QCRI at TREC 2013 Microblog Track

Tarek El-Ganainy
Qatar Computing Research Institute
Doha, Qatar
telganainy@qf.org.qa

Zhongyu Wei[*]
The Chinese University of Hong Kong
Hong Kong, China
zywei@se.cuhk.edu.hk

Walid Magdy
Qatar Computing Research Institute
Doha, Qatar
wmagdy@qf.org.qa

Wei Gao
Qatar Computing Research Institute
Doha, Qatar
wgao@qf.org.qa

ABSTRACT

We report our work in the real-time ad hoc search task of TREC-2013 Microblog track. Our system focuses on improving retrieval effectiveness of Microblog search through query expansion and reranking of search results. We apply web-based query expansion algorithm for enriching the microblog queries with additional terms from concurrent webpages related to the search topic. Later we apply results reranking through utilizing state-of-the-art learning to rank algorithms to train 12 different ranking models using relevance judgment of Tweets2011-12 queries, for which we conduct feature engineering, validation dataset selection, and the ensemble of these models. Our approach differs from salient approaches in the previous Microblog tracks that are based on document expansion utilizing embedded URLs and that leverage some single ranking model for tweets re-ranking. Our pipeline constructed using the hybrid of these two components showed promising retrieval results on Tweets2013 benchmark dataset.

1. INTRODUCTION

This year comes the third edition of TREC Microblog track[1] consisting of a single task – real-time ad hoc search, while the real-time filtering task introduced last year is eliminated this time. Although the basic concept of the search task is the same as previous years, there are two new genres:

- Firstly, a new version of tweets collection is provided for evaluation with much larger size than before which includes approximately 240 million tweets spread over a two-month period: from February 1, 2013 to March 31, 2013 (inclusive).
- Secondly, instead of maintaining a local copy of the data independently per group, the new tweets collection is stored on a remote server shared by all participants and a set of search APIs[2] are provided to users for interaction with the data. The server will return up to 10,000 results for a specific query each time utilizing a state-of-the-art baseline retrieval model. Since in previous years the local copy can be different among groups, it is now fairer to share an identical corpus remotely by all users so that participants can focus on core techniques and compare with the same baseline run.

We, the group of participants from Qatar Computing Research Institute (QCRI), submitted four runs for the ad hoc search task, which were configured differently by using query expansion for retrieval [13] and learning-to-rank [9] models to re-rank the search results. The four runs are named as QCRI1, QCRI2, QCRI3 and QCRI4 whose configurations are described as follows:

- QCRI1 uses a single ranking model for re-ranking based on the results obtained from the expanded queries using standard pseudo relevance feedback (PRF) [17].
- QCRI2 uses a single ranking model for re-ranking with a selected validation set for model selection based on the results obtained from the expanded queries using PRF.
- QCRI3 combines multiple ranking models for re-ranking based on the results obtained from the expanded queries using PRF.
- QCRI4 combines multiple ranking models for re-ranking based on the results obtained from the expanded queries that combine the expansion terms from the standard PRF and those extracted from the corresponding Google search results of the same query.

2. SYSTEM OVERVIEW

The architecture of our real-time Microblog search and re-ranking system is described in Figure 1.

We fetched 10,000 retrieval results for each topic with the expanded queries via search API. Two query expansion schemes were adopted in our system, one utilizing the standard PRF technique and the other mining query expansion terms from Google search results based on the same set of Microblog search queries and time-bounded by the query timestamp. The run QCRI4 was obtained by retrieving the tweets using the combination of two sets of expansion terms which resulted from the corresponding query expansion schemes, while the other three runs were conducted using the expanded queries which resulted from PRF only and did not use any external information. Then we processed the initially retrieved tweets by removing non-English tweets and filtering retweets. After that, we extracted four categories of ranking features including content-based features, Twitter-specific features, account authority features and temporal features from this tweets set, for re-ranking

[*] The work was performed during the author's internship at Qatar Computing Research Institute

[1] `https://github.com/lintool/twitter-tools/wiki/TREC-2013-Track-Guidelines`

[2] `https://github.com/lintool/twitter-tools/wiki/TREC-2013-API-Specifications`

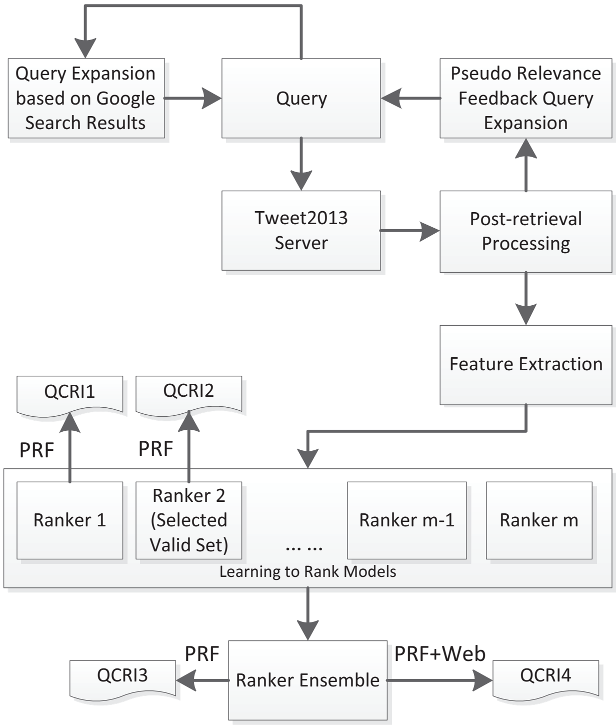

Figure 1: Real-time Microblog search and re-ranking system

the search results by different learning to rank models. For the run QCRI1 and QCRI2, a single ranking model MART [6] was applied for re-ranking, and an ensemble of 12 trained rankers was used for both QCRI3 and QCRI4. The difference between QCRI1 and QCRI2 was that the latter utilized an automatically selected validation set.

Overall, we designed our pipeline to combine query expansion and result re-ranking. For query expansion, besides the commonly used PRF, we also made use of the search result from Google for query expansion. The details will be presented in Section 4. For result re-ranking, our system resorted to learning to rank, the application of which although successful in previous Microblog tracks, was still shallow for the task. Therefore, we performed extensive engineering work for re-ranking including feature engineering, validation set selection and the ensemble of various ranking models. In addition to previously used content-based features, Twitter-specific features and account authority features, we also incorporated temporal features into our system for the sake of real-time fashion of Microblog search. Also, we tried to select validation set using the query similarity measure described in [3] for selecting the single best performing ranking model. Finally, we explored different methods to combine multiple ranking models. The details about ranking model learning will be given in Section 5.

3. CORPUS STATISTICS & SEARCH API

A new tweets collection was provided this year for evaluation that contains 243,271,538 tweets and is an order of magnitude larger than the Tweets2011 collection. A brief comparison between the new collection and Tweet2011 corpus is displayed in Table 1. Other than the total size, the average length of tweets in Tweet2011 is 2.4 tokens shorter than that of Tweet2013.

Table 1: The general comparison to previous tweets collection

Collection	# of tweets	# of terms	Tweet length
Tweet2013	243,271,538	2,928,041,436	12.04
Tweet2011	16,141,809	155,562,660	9.64

The collection was stored and indexed on a remote server processed using some basic operations such as tokenization, normalization and stemming. And a set of search APIs were released for accessing to the corpus where three services were provided including baseline retrieval, text and API-supplied metadata from retrieved tweets, and access to corpus-level statistics. The baseline retrieval was based on negative KL-divergence language modeling approach with Dirichlet prior smoothing parameter μ set to 2,500. For each tweet returned, there were 13 fields of information provided, including tweet id, tweet content, score given by language model, time stamp, language identity, and so on. For corpus-level statistics, information about term frequency and document frequency for terms appearing more than 10 times were provided. The open-source search engine Lucene[3] was employed for indexing and providing the baseline retrieval result.

4. QUERY EXPANSION BASED ON WEB SEARCH RESULTS

The main challenge in finding relevant tweets to a given topic is word mismatch between search query and tweets text. Some TREC reports in Microblog track [8, 11, 13, 16] showed that query expansion helped in improving the Microblog retrieval effectiveness since it could enrich the query with additional terms that led to better matching with more relevant tweets. Our work last year [13] tested expanding the microblog query with title of the top concurrent web result from searching Google with the microblog query. Results showed significant improvement in retrieval effectiveness compared to standard pseudo relevance feedback (PRF) from the tweets collection [13]. In this year, we further investigate the web-based query expansion approach through testing advanced settings and configurations rather than just simply appending the title of the top web result to the query.

Teevan et al. [14] showed that Microblog search queries were typically time-related since users usually searched social platform for updates about events happening at the time of search. Thus, our query expansion approach is to leverage the web search results obtained by using the Microblog search query aiming to find the concurrent relevant webpages, such as news articles or websites discussing the topic of search. Then we extract expansion terms from the Web search result pages to enrich the initial expansion terms produced by the standard PRF. The process is described as follows:

- The original query Q_0 is used to search in the tweets collection in an initial step. The most frequent n_t terms (excluding stop words) appearing in the top retrieved n_D tweets are extracted with standard PRF [17]. Extracted expansion terms are denoted as Q_{PRF}.
- Q_0 is used to search the Web via search engine in the same time frame of the query for the concurrent results, in which we extract two types of information: (1) The title of the topmost search result is extracted and pruned by removing stop words and website name (similar to [13]). The title part usually contains delimiters like '-' and '|' that separate the

[3] http://lucene.apache.org/

real title content and the domain name of the webpage, e.g., "... | CNN.com", "... - Wikipedia, the free encyclopedia". Only the real title is used for expansion, referred to as Q_{title}. (2) Both titles and snippets of the top-10 ranked results are collected. Then all terms appearing more than n_w times are extracted and used for expansion, referred to as Q_{web}.

- All expansion terms are combined and appended with a given weight to the original query as follows:

$$Q_{exp} = (1 - \alpha) \cdot Q_0 + \alpha \cdot (Q_{PRF} \cup Q_{title} \cup Q_{web})$$

where Q_{exp} is the final expanded query used for searching tweets at the second time and α is the weight assigned to the expansion terms.

The final formulated query Q_{exp} is expected to be richer in information about the topic than the original query, and potentially leads to better search results.

5. ENSEMBLE OF RANKING MODELS FOR TWEETS RE-RANKING

In our participation, we employed multiple ranking models and their ensemble for re-ranking the retrieved tweets. Our models were learned using Tweets2011-12 qrels and tested with Tweets2013 queries.

Our approach aims to improving re-ranking effectiveness by using validation set selection and the combination of different ranking models. We employed six state-of-the-art ranking algorithms in our system: RankNet [2], RankBoost [5], Coordinate Ascent [12], MART [6], LambdaMART [16], and RandomForests [1]. All these algorithms have been implemented in the RankLib package[4].

5.1 Feature Description

We defined a set of 21 features belonging to 4 different categories. The brief description of all the features can be found in Table 2. Some of these features are detailed in this section.

5.1.1 Content-based Features

Content-based features aim to capture textual similarity between query and the target tweet. This kind of feature is widely used in Web search and full-text retrieval, and has been proved indicative.

- **BM25 & BM25_Exp: BM25** measures the content relevancy between original query Q_0 and tweet T by BM25 weighting function. The standard BM25 is formulated as:

$$\sum_{q_i \in Q_0} \frac{IDF(q_i) * TF(q_i, T) * (k_1 + 1)}{TF(q_i, T) + k_1 * (1 - b + b * \frac{Length(T)}{Avg_{length}})}$$

where $Length(T)$ denotes the length of T, $TF(q_i, T)$ is the frequency of term q_i in tweet T, Avg_{length} stands for average length of tweets in the tweet collection, and $IDF(q_i)$ is inverse document frequency. Both average length and IDF are provided by the collection server as corpus-level statistics. For **BM25_Exp**, the similarity is computed between the expanded query and the target tweet.

- **LM & LM_Exp: LM** measures the content relevancy between Q and T by KL-divergence. For **LM_Exp**, the content relevance is computed between the expanded query and the target tweet. Since the scoring function used by the search API is KL-divergence, we utilize the relevance score of the target tweet as feature value here.

5.1.2 Twitter-specific Features

Twitter provides many special characteristics and we identify some of them as features for the ranking models.

- **Has_URL & URL#:** An informative tweet always contains URL for information extension. However, a tweet embedding too many URLs might be a spam. We use two features to capture the URL related information of the target tweet. **Has_URL** is a binary feature which is assigned 1 if the tweet contains at least one URL, and 0 otherwise. **URL#** indicates the number of URLs included in the tweet.

- **Has_Hashtag & Hashtag#:** Users always use hashtag within a tweet to highlight a topic. These two features are used to capture the hashtag usage in the target tweet. **Has_Hashtag** is a binary feature which is assigned 1 if the tweet contains at least one hashtag, and 0 otherwise. **Hashtag#** indicates the number of hashtags included in the tweet.

- **RT# & RT#_Level (RTL):** Generally, if a tweet is more informative, it is more likely to be reposted by other users. We use two features to indicate the popularity of the target tweet. **RT#** is the number of times the tweet is reposted, and **RTL** is an integer between 0 and 4 (inclusive) indicating the level of the retweet count. The value of **RTL** can be computed as follows corresponding to the border points of **RT#** such as 0, 1, 10, 100 and 1000:

$$RTL = \begin{cases} 0, & if \;\; RT\# = 0 \\ 1, & if \;\; RT\# \in [1, 9] \\ 2, & if \;\; RT\# \in [10, 99] \\ 3, & if \;\; RT\# \in [100, 999] \\ 4, & if \;\; RT\# \in [1000, 9999] \end{cases}$$

5.1.3 Account-related Features

A tweet becomes more informative if it is posted by authoritative users. Therefore, we also utilize some straightforward count related to this intuition as features.

- **Follower# & Follower#_Level(FL): Follower#** is the number of followers the author who publishes the target tweet owns, and **FL** is an integer between 0 and 5 (inclusive) indicating the level of follower count. The value of **FL** is computed based on **Follower#** with separating points at 10, 100, 1000, 10000 and 100000.

- **Status# & Status#_Level(SL): Status#** is the number of tweets the author publishes, and **SL** is an integer between 0 and 5 (inclusive) indicating the level of **Status#**. The value of **SL** is computed based on **Status#** with separating points at 10, 100, 1000, 10000 and 100000.

5.1.4 Temporal Features

According to our previous work on Microblog search [15, 7], temporal features appeared effective for both result re-ranking and query expansion. Therefore, we propose two temporal features.

- **Recency_Degree (RD): RD** indicates whether the tweet is published recently according to the query time. Time difference between tweet and query is used here to measure the recency degree:

$$RD = Time_{query} - Time_{tweet}$$

where $Time_{query}$ stands for the time stamp (in millisecond) the query is issued and $Time_{tweet}$ denotes the time stamp (in millisecond) the target is posted.

[4] `http://sourceforge.net/p/lemur/wiki/RankLib/`

Table 2: Feature description (t: a tweet; Q_o: original query; Q_{exp}: expanded query)

Feature category	Feature name	Feature description
Content-based	BM25	BM25 similarity between Q_o and t
	LM	Language model similarity between Q_o and t
	Length	The number of tokens in t
	Unique_TF	The number of unique terms in t that match terms in Q_o
	TF	The frequency of terms in t that match terms in Q_o
	BM25_Exp	BM25 similarity between Q_{exp} and t
	LM_Exp	Language model similarity between Q_{exp} and t
	Unique_TF_Exp	The number of unique terms in t that match terms in Q_{exp}
	TF_Exp	The frequency of terms in t that match terms in Q_{exp}
Twitter-specific	Has_URL	Whether t contains at least one URL
	Has_HashTag	Whether t contains at least one hashtag
	URL#	The number of URLs in t
	HashTag#	The number of hashtags in t
	RT#	The counts that t has been reposted
	RT#_Level	The level of RT#
Account authority	Status#	The number of tweets the user publishes
	Follower#	The number of followers the user owns
	Status#_Level	The level of status count the user publishes
	Follower#_Level	The level of follower count the user owns
Temporal	Recency_Degree	The gap between query time and tweet time
	Is_Peak	Whether the tweet is published in the peak date of the query

- **Is_Peak (IP): IP** is a binary feature indicating whether the target tweet is posted in the peak time of queried topic. Peak-finding algorithm [10] is used to identify the peak time for the query. Following the strategy used in our real-time tweet search system last year [7], we apply peak-finding for the top-n search results and treat the first k largest peaks as the real peak of the query.

5.2 Validation Set Selection

In machine learning, validation dataset is commonly used to select the model with better performance. Since a final ranker is supposed to achieve the best performance on validation set, the more similar the validation set is to the test set, the higher performance the ranker may obtain on test set. In our system, we tried to select validation set by choosing the training queries that bear high similarity to test queries. We adopted two different query selection methods by following the strategy described in [3], namely, the document feature aggregation and the query comparison methods. According to our experiments, the document feature aggregation method demonstrated higher performance which therefore was used in our system.

5.3 Ensemble of Rankers

To improve the ranking effectiveness, some retrieval systems tried to combine the ranked lists from different ranking models. The result of Yahoo Learning to Rank Challenge [4] also revealed that the ensemble of ranking models are powerful in Web search. In our system, we learned a number of ranking models separately and used an ensemble approach to incorporate them. We trained the following 12 models for the ensemble:

1. A Rankboost model is trained without validation set;
2. A RandomForest model is trained without validation set;
3. Two MART models are learned by selecting 20% training queries into validation set and optimizing on P@30, where the validation set of one model is selected using the query selection method described in [3] and the validations set of the other is selected just randomly;
4. Two RankNet models are learned with validation set in the same way as above;
5. Two Coordinate Ascent models are learned with the validation set in the same way as above, but one of them optimizes on MAP instead of P@30;
6. Four LambdaMART models are learned with the validation sets as above, where two of them are validated using the selected validation set and the other two are validated using the random validation set, and in each group there is one model optimizing on P@30 and the other optimizing on MAP.

The detail of configuration regarding the training of these separate rankers can be found in Table 3. For the ensemble of rankers, we compared two different approaches that combined these 12 resulted rankers: (1) linear combination by LambdaRank; (2) simply averaging the ranking scores, and the result showed that the latter – simply adding up the normalized model scores – produced higher performance. Therefore, it was eventually adopted for the ensemble in our submissions.

6. EVALUATION

6.1 Setup

In TREC ad-hoc Microblog track, two different tweets collections and three sets of queries have been released so far. The tasks of the first two years shared the same collection Tweets2011 which contains around 16 million tweets. The much larger collection Tweets2013 containing 243,271,538 tweets was newly constructed. There are 3 different query sets, one for each year, which are denoted as QS2011, QS2012 and QS2013 containing 50, 60 and 60 queries, respectively. The basic statistics about the relevance judgment of all query sets of the three years are given in Table 4.

Table 3: The 12 ranking models learned for the ensemble

Algorithm	Validation%	Validation set	Optimized metric
RankBoost [5]	–	–	–
RandomForest [1]	–	–	–
RankNet [2]	20%	selected	P@30
	20%	random	P@30
MART [6]	20%	selected	P@30
	20%	random	P@30
Coordinate Ascent [12]	20%	random	P@30
	20%	random	MAP
LambdaMART [16]	20%	selected	P@30
	20%	selected	MAP
	20%	random	P@30
	20%	random	MAP

Table 4: The statistics of relevance judgement

	QS2011	QS2012	QS2013
# of queries	50	60	60
# of annotated tweets	40,855	73,073	71,279
# of highly relevant	558	2,572	3,155
# of all relevant	2,864	6,286	9,011

All the parameters of query expansion and result re-ranking models we used this year were optimized using the qrels of QS2011 and QS2012 as the training sets. Following the track benchmark, we report P@30 as the major evaluation metric, and we will also report mean average precision (MAP) for reference. The assessment was based on two levels of strictness regarding the relevance judgement: (1) All: All relevant and highly relevant tweets were considered as relevant; (2) High: Only highly relevant ones were treated as relevant.

6.2 Parameter Tuning for Query Expansion

We examined the effectiveness of our different query expansion strategies and tried to find reasonable configuration for each. Then we tested the combined expansion method with the appropriate configuration. The level of **All**, the less strict assessment, was used here. For the ease of comparison with the best systems available in the track, we evaluated our system on the query sets of QS2011 and QS2012. Later, we applied the best configuration of expansion parameters to QS2013.

We initially investigated the best configurations to PRF using the retrieved tweets only (referred to as PRF thereafter) and to Web-search-based query expansion, each separately. Later, we combined expansion terms coming from both methods and added to the original query Q_0 to formulate the final search microblog query.

Regarding the PRF, the best achieved improvement over the baseline, measured by P@30, when tested on the TREC2011 collection using QS2011 and QS2012 was:

- The optimal number of tweets n_D to be used in the feedback process was 50.
- The optimal number of feedback terms n_t was 12.
- The optimal weight for the expansion queries α was 0.2.

For our Web-search-based query expansion, the timestamp provided with the topics was utilized to simulate the live query expansion from the web described in Section 4. The query of each topic was used to search Google for relevant webpages appeared at the time of the topic. In our experiments, the time was specified between the earliest tweet time in the collection and the time of the topic issued. This assured that the returned results were temporally aligned with the time of the query issued on Twitter [13]. We examined the effectiveness of expansion with different feedback information (i.e., Q_{title} and Q_{web}). It was found that query expansion using Q_{title} lead to slight improvement in the retrieval effectiveness over the baseline. Also, we found that the optimal n_w for Q_{web} was equal to 3, which led to some improvement over the baseline as well. However, t-test indicated that these improvements were statistically insignificant. Therefore, we simply combined all the expansion terms from Q_{PRF}, Q_{title} and $Q_{web}(n_w = 3)$ to add them into the final expanded query (referred to as PRF+Web thereafter). By combining the expansion terms it led to higher retrieval effectiveness. We found that PRF+Web got the peak scores when α=0.2 and α=0.3.

Therefore, our final expansion configuration were set as:

$$Q_{exp} = 0.8 \cdot Q_0 + 0.2 \cdot (Q_{PRF} \cup Q_{title} \cup Q_{web})$$

where Q_{PRF} is configured with n_D=50, n_t=12, and Q_{web} is configured with n_w=3.

Table 5 demonstrates some example queries with the expansion terms produced from the three expansion strategies, where we can see that expansion terms can alleviate word mismatches between queries and tweets.

6.3 Results with Re-ranking

For tweets re-ranking, there are three parts of parameters which need to be tuned. Firstly, the parameters n (top-n search results) and k (the k largest peaks) of peak-finding for the **Is_Peak** feature were validated based on Tweet2011 corpus using QS2011 and QS2012, which were fixed as $n = 100$ and $k = 2$.

Secondly, for the validation set selection method, we trained our rankers on QS2011 and validated them on QS2012, and our result showed that the document feature aggregation method [3] performed better. Thus, we choose it as the query similarity measure for validation set selection.

Thirdly, for the choice of ranker ensemble approach, we compared the linear combination of the rank scores by LambdaRank and simply averaging the rank scores from the 12 ranking models, for which we trained the rankers on QS2011 and validated them on QS2012. The result showed that the simple average is better. Thus, we just summed over the normalized model scores as the final rank score of each tweet.

For the training of the individual ranking models, we used the default parameters given by RankLib.

Table 6 shows the overall performance of our four submitted runs. We also listed the performance of two baseline runs for comparison: LM(PRF) – language-model-based retrieval with PRF; LM(PRF+Web) – language-model-based retrieval plus PRF with Web-search-based query expansion. Besides, the official median performance of each query among the 65 automatic runs were added up and then averaged to provide an official reference.

As shown, there is no much difference between QCRI1 run and the baseline LM(PRF), which indicates that the shallow application of learning to rank for tweets re-ranking may not be helpful especially when the baseline itself can already perform fairly well. Therefore, some deeper techniques are needed.

The effectiveness of QCRI2 is higher than QCRI1 which indicates that the validation dataset selection method is helpful, but their difference is not statistically significant. The performance is improved further when we combined multiple ranking models as it could be shown that QCRI3 outperformed QCRI2. This suggests that different models can perform differently on each individual query and the simple summation over the normalized ranking

Table 5: Examples of expansion terms extracted by each of the expansion method

Q_0	Q_{PRF}	Q_{title}	Q_{web} ($n_w = 3$)
2022 FIFA soccer	11 playing world cup qatar winter president sports blatter sepp 2010	qatar unveils new green stadium designs for 2022 fifa	stadiums qatar cup 2018 world
Moscow airport bombing	bomb suicide killed 35 domodedovo blast people 31 dead kills afp russias	moscow bombing carnage at russias domodedovo airport	blast dead busiest terrorist moscows killed 100 suicide 35 russias attack russian people domodedovo
Bieber and Stewart trading places	jon justin kristen video daily new trade sex stewarts bodies good teens	justin bieber trades places with jon stewart raps for sean kingston	jon daily justin night bodies
US behind Chaevez cancer	survival cancer-treatment firms special rosy claims report breast reuters reason @cancerfollowers rally	us rejects venezuelas conspiracy claims over hugo chavezs cancer	venezuelan embassy death venezuela maduro chavez united hugo chavezs officials states
Tony Mendez	argo cia real #argo spy lives ben hero affleck hostage latino aka	argo what really happened in tehran cia agent remembers	oscars film talks cia real oscar argo agent

scores can generally improve the overall re-ranking.

Basically, leveraging external information could result in performance gain in large margin, evidenced as the obvious improvement made by LM(PRF+Web) (even without re-ranking) over QCRI3, where the differences in terms of P@30 and MAP were statistically significant according to the ALL level assessment. Further plus re-ranking based on LM(PRF+Web) resulted in QCRI4 that leads to higher performance, and all the improvements over QCRI3 became statistically significant.

Figure 2 shows the performance of P@30 on individual test queries in QS2013. The P@30 values varied widely across all the 60 queries. Our system obtained very high P@30 values on 6 queries, namely MB126 (“Pitbull rapper”), MB127 (“Hagel nomination filibustered”), MB129 (“Angry Birds cartoon”), MB141 (“Mila Kunis in Oz movie”), MB157 (“Kardashian maternity style”) and MB169 (“Honey Boo Boo Girl Scout cookies”). These topics are easy queries due to their relatively large number of relevant tweets in qrels (all of them have more than 50 relevant tweets and 4 of which even have over 300 relevant tweets). The sufficiency of the relevant tweets could also boost the performance of query expansion. On the other hand, the worst 4 queries on which our system achieved P@30 lower than 0.1 include MB134 (“The Middle TV show”), MB150 (“UK wine industry”), MB160 (“social media as educational tool”) and MB165 (“ACPT Crossword Tournament”). Further examination revealed that MB150 and MB165 had only 5 and 7 relevant tweets in qrels, respectively, and each relevant tweet contained just 2 or even less original query terms which made them difficult search queries. Furthermore, fewer relevant information also harmed the effectiveness of query expansion. Although the rest of two queries MB134 and MB160 were not lack of relevant tweets, our system still did not generate good performance. For the topic MB134, we found that most of the retrieved tweets were about another TV show “Malcolm in the Middle” which although containing the term ‘Middle’, was actually retrieved due to the term ‘Malcolm’ introduced as an expansion term by our query expansion that led to a query topic shift. For the topic MB160, our expanded query gave a larger weight to ‘tool’ than ‘education’, which lost the real focus of the original query, and as a consequence, we found most of the retrieved tweets were related to “using social media as tool”.

7. CONCLUSIONS

In this work, we reported an efficient and effective method for real-time ad hoc Microblog search task based on query expansion and ensemble of rankers for tweets re-ranking. Our techniques are innovative in a sense that (1) query expansion based on PRF was enriched by using expansion terms extracted from the concurrent Web search results; (2) instead of using a single ranking model, we exploited an ensemble of multiple ranking models trained by using different state-of-the-art learning to rank algorithms for better tweets re-ranking. Our method is very efficient since it does not require the time-consuming processing of external information such as the webpages given by the URLs embedded in tweets. Our pipeline constructed by integrating the two components demonstrate promising retrieval effectiveness on TREC 2013 Microblog track datasets.

In the future, we plan to combine query expansion and document expansion within an efficient framework to further improve retrieval effectiveness. For result re-ranking, we will study the time-related features more deeply and broadly for integrating more of them into our framework, and we will also try to improve the re-ranking of individual queries by using some query sensitive approaches.

8. REFERENCES

[1] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 89–96, 2005.

[3] P. Cai, W. Gao, A. Zhou, and K. Wong. Query weighting for ranking model adaptation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 112–122, 2011.

Table 6: The overall performance of all our runs based on ALL relevance and HIGH relevance. ** and * indicate significance at 99% and 95% confidence level, respectively, compared to our second best run QCRI3

	ALL		HIGH	
	P@30	MAP	P@30	MAP
Median (65 auto)	0.4217	0.2126	-	-
LM(PRF)	0.4849	0.3030	0.2578	0.2156
QCRI1	0.4678	0.2993	0.2706	0.2212
QCRI2	0.4733	0.3001	0.2706	0.2246
QCRI3	0.4817	0.3068	0.2728	0.2268
LM(PRF+Web)	0.5356**	0.3444**	0.2828	0.2464
QCRI4	**0.5372****	**0.3494****	**0.2983***	**0.2656****

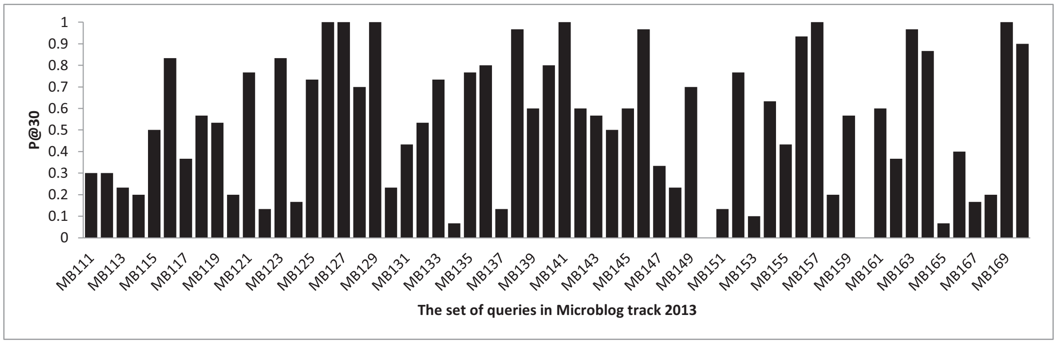

Figure 2: The performance of P@30 on all the queries in QS2013 by our run of QCRI4

[4] O. Chapelle and Y. Chang. Yahoo! learning to rank challenge overview. *Journal of Machine Learning Research–Proceedings Track*, 14:1–24, 2011.

[5] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[6] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[7] W. Gao, Z. Wei, and K.-F. Wong. Microblog search and filtering with time sensitive feedback and thresholding based on bm25. In *TREC*, 2012.

[8] Z. Han, X. Li, M. Yang, H. Qi, S. Li, and T. Zhao. Hit at trec 2012 microblog track. In *TREC*, 2012.

[9] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[10] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 227–236, 2011.

[11] D. Metzler and C. Cai. Usc/isi at trec 2011: Microblog track. In *TREC*, 2011.

[12] D. Metzler and W. B. Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[13] A. Saad El Din and W. Magdy. Web-based pseudo relevance feedback for microblog retrieval. In *TREC*, 2012.

[14] J. Teevan, D. Ramage, and M. R. Morris. # twittersearch: a comparison of microblog search and web search. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pages 35–44, 2011.

[15] Z. Wei, W. Gao, L. Zhou, B. Li, and K.-F. Wong. Exploring tweets normalization and query time sensitivity for twitter search. In *TREC*, 2011.

[16] Q. Wu, C. J. Burges, K. M. Svore, and J. Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010.

[17] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, pages 403–410, 2001.