HIT at TREC 2012 Microblog Track

Zhongyuan Han[1,2], Xuwei Li[1], Muyun Yang[1], Haoliang Qi[2], Sheng Li[1], Tiejun Zhao[1]

1. School of Computer Science and Technology, Harbin Institute of Technology
Harbin, Heilongjiang, China, 150001
2. School of Computer Science and Technology, Heilongjiang Institute of Technology
Harbin, Heilongjiang, China, 150050
{zyhan, xwli, ymy}@mtlab.hit.edu.cn, haoliang.qi@gmail.com, {lisheng, tjzhao}@hit.edu.cn

ABSTRACT

This paper describes our approaches to the TREC 2012 Microblog Track. We explore the query expansion and document expansion techniques to address the retrieval of short tweet texts. Further, we examine the webpages linked by the URL in a tweet as an external source to improve the performance. Then learning to rank technique is adopted to combine all features for better performance. Finally, we accomplish the microblog filtering via comparing the new tweet against top m relevant tweet retrieved in the history.

1. INTRODUCTION

Microblog, most notably Twitter, is becoming increasingly popular around the world. For users, the real-time search in microblog is an effective way to have a grasp of latest development or other's thoughts of a topic. However, the microblog retrieval challenges the classical retrieval technologies with its extremely short text, e.g. Twitter with no more than 140 characters. In TREC 2011 Microblog Track, the learning to rank technique are generally adopted to combine text-based features with non-text features such as URL, hashtag, time and so on[1-3]. In our view, the the current research is mainly focused on the application of non-text features, leaving the difficulty of modeling caused by microblog's short text less touched. In particular, the short document modeling is not well addressed. Therefore, we try to examine the document expansion and query expansion under the classical language model framework to accomplish the microblog retrieval in this year. Meanwhile, we also examine the webpage contents indicated by the URL in a tweet for its contribution to retrieval performance. In addition, we make use of learning to rank technique to combine other non-text features for better performance. At last, we accomplish the microblog filtering via comparing the new tweet against top m relevant tweet retrieved in the history.

TREC 2012 microblog track intends to fulfill two defined tasks: Real-time Adhoc Task and Real-time Filtering Pilot Task. The Remainder of this paper is arranged as follows: Section 2 introduces our approaches and results in Real-time Adhoc Task. Section 3 presents the way we accomplish the Real-time Filtering Pilot Task, and Section 4 concludes the paper.

2 REAL-TIME ADHOC TASK

We investigate the query expansion, the document expansion for tweet search, and apply learning to rank technique to adopt to combine all features for better performance. In other way, We incorporate the scores of the tweet texts and the scores of the URL in them to improve retrieval performance.

2.1 KL-divergence Retrieval Model

The KL-divergence model is a classical language modeling approach for retrieval. It can support feedback more naturally. In this approach, a query and a document are assumed to be generated from a unigram query language model θ_Q and a unigram document language model θ_D, respectively. Given a query Q and a document D, we would compute an estimate of the corresponding query model ($\hat{\theta}_Q$) and document model ($\hat{\theta}_D$), and then KL-divergence of the two models is defined as :

$$KL(\hat{\theta}_Q \| \hat{\theta}_D) = \sum_{w \in V} p(w \mid \hat{\theta}_Q) * \log \frac{p(w \mid \hat{\theta}_Q)}{p(w \mid \hat{\theta}_D)}$$

where V is the set of all the words in vocabulary.
The estimation of the query model $\hat{\theta}_Q$ is described in sub-section 2.2 and the estimation of the document model is often done through smoothing with the global collection language model θ_C , we used the Dirichlet smoothing with μ =100:

$$P(w \mid \hat{\theta}_D) = \frac{c(w, D) + \mu P(w \mid \theta_C)}{|D| + \mu}$$

Because of short text of tweet, document expansion is used for better estimation of the document language model as is described in sub-section 2.3.

2.2 Query Expansion

We applied the relevance feedback model[4] for query expansion. The relevance model deems that a query term is generated by a relevance model p(w|θ_R), which is derived by top-ranked feedback documents by assuming them to be samples from the relevance model as follows:

$$P(w \mid \theta_R) \propto \sum_{d \in F} P(w \mid d) P(d \mid \theta_R)$$

where F denotes the feedback documents, usually chosen as the top-ranked retrieval documents for the query (set as 20 in our experiment); p(w|d) is the probability that the term w appearing in the document d, and the relevance θ_R is approximated by the original query, thus we can obtain:

$$P(w \mid \theta_R) \propto \sum_{d \in F} P(w \mid \theta_d) P(\theta_d) \prod_{i=1}^{m} P(q_i \mid \theta_d)$$

The above relevance model is used to enhance the original query model by the following interpolation:

$$P(w \mid \theta_q^{'}) = (1-\alpha)P(w \mid \theta_q) + \alpha P(w \mid \theta_R)$$

where α is the interpolation weight (set as 0.8 in our experiments).

2.3 Document Expansion

The use of a limited and very small number of characters by twitter's very definition (i.e. 140 characters) drastically reduces the length of the tweet text. In very short microblog messages, most terms occur only once, making statistical estimation less reliable. Meanwhile, the term mismatch in microblog search is also non-trivial issue. In this sense, how to best estimate the document language model P(w|D) is crucial.

Since |D| is small (about 12 terms in length in our experiments after tweet message preprocessing), the word count n(w,D) is frequently observed as 1. To enhance the classical the maximum likelihood estimation method, we applied DELM (Document Expansion Language Model)[5] to improve the representation of short tweet in microblog search. That is, the model for document D is smoothed with contents obtained from its k nearest neighbors $D_1,\ldots,D_k$, with each document's influence in the smoothed model being proportional to its cosine similarity with D.

The precise definition of neighborhood document (i.e. tweet) is measured by the cosine similarity between any two document models X and Y. Then it assigns a confidence value $r_d(b)$ to every document b in the collection to indicate the confidence we believe b is sampled from d's hidden model. The cosine similarity and confidence value are defined respectively as below:

$$\mathrm{sim}(X,\ Y) = \frac{\sum_i x_i \times y_i}{\sqrt{\sum_i x_i^2 \times y_i^2}}$$

$$\mathrm{rd}(d,\ b) = \frac{\mathrm{sim}(d,\ b)}{\sum_{b' \in C-\{d\}} \mathrm{sim}(d,\ b')}$$

In fact the confidence value $r_d(b)$ is set by normalizing the cosine similarity scores. Then a pseudo document d' is obtained with the following pseudo term count:

$$c(w,d') = \beta c(w,d) + (1-\beta) \sum_{b \in C-\{d\}} (r_d(b) \times c(w,b))$$

Here it uses a parameter β to control the degree of relying on neighborhood document. This technique is proved to be valid in improving search results in TREC texts by [5]. In our experiments, β=0.8 and the number of neighborhood documents is set 100.

2.4 URL

The research of TREC 2011 Microblog Track shows that whether containing a URL is an important feature for a tweet[1]. A tweet which contains a URL is more likely to contain substantial contents for a topic. We take advantage of a simple linear model to combine the score of tweet and the score of URL. Given a query Q and a tweet D, the relevance score(Q,D) is computed according to the following equation:

$$\text{Score}(Q, D) = (1 - \lambda) * \text{score\_D}(Q, D) + \lambda * \text{Score\_Url}(Q, D) * \delta$$

in which Score_D(Q,D) is the relevance score for Q and D estimated by Kullback-Leibler (KL) divergence; Score_URL(Q,D) is the relevance score of Q and Webpage of URL in D, which is also estimated by KL divergence.

The λ is the parameter to control the effect of URL content to the retrieval process. We empirically set it as a constant of 0.8. The δ is a zoom ratio to enable the Score_URL(Q,D) to be comparable to Score_D(Q,D), which is decided as the ratio of the average of Score_D to the average of Score_URL.

2.5 Learning to Rank

Previous works proved that microblog's non-text features are positive to retrieval performance. The learning to rank technique, which was successfully used by the several teams in TREC 2011 Microblog Track, is also adopted in this paper.

Specifically, we designed a ranking logistic regression model to learn a pair-wise ranking from twitter retrieval. The ranking logistic regression algorithm will be reported in the future. For a training sample consisting of a relevant tweet and an irrelevant tweet, the following features are applied to the modeling process:

- Text-based features

KL(Q,D): KL divergence of the original query and the original tweet;
KL(EQ,D): KL divergence of the expanded query and the original tweet;
KL(Q,ED): KL divergence of the original query and the expanded tweet;
KL(EQ,ED): KL divergence of the expanded query and the expanded tweet;

- Non-text features

Has_URL: whether the tweet contain a URL (binary valued)
Has_hashtag: whether the tweet contain a hashtag (binary valued)
Retweet_count: frequency of the tweet re-posted

- User features

Followers_count: how many people are following this author
Friends_count: how many people this author is following
Listed_count: how many groups is the user in

Note that to avoid using external source, the content of URL here is not used as in sub-section 2.4 in the ranking modeling process.

2.6 Experimental Results

1) Data setting

We download 10,397,336 tweets by twitter crawler provided by track organizers. After preprocessing, 3,754,077 tweets are indexed. The main steps are as follows:

- The null tweets were removed.
- The retweets without "RT" would be judged as non-relevant and thus were removed. The retweets with "RT" are removed if there is nothing contents in front of "RT". But

once there exists description in the beginning of their tweet text, we only keep the words before "RT".

- We filtered out all the non-English tweets using language identifier tool provided by Nutch[1].
- Porter stemmer is used for stemming and stop words are filtered.

The statistics of dataset obtained by our group are shown in Table 1. Moreover, we extracted URL links in tweets and downloaded them as an external resource. The number of successfully downloaded webpage is 814,817.

At last, we completed the real-time index. Actually, we indexed tweets which only posted before the timestamp for every query.

Table1. Statistics of tweets in dataset

#Total Tweets	# of Null Tweets	# of Retweets	# of Non-English tweets	# of Indexed Tweets
10,397,336	0	342,652	6,300,607	3,754,077

2) TREC Results

We used TREC 2011 microblog task data to train the parameters as mentioned before and submitted four official runs as follows:

- hitQryFBrun4: a baseline run uses KL divergence with query expansion only;
- hitDELMrun2: both query expansion and document expansion are applied;
- hitURLrun3: using external source: a linear combination of score in Run2 and the URL derived score as described in sub-section 2.4;
- hitLRrun1: result from learning to rank technique as described in sub-section 2.5;

The formal results provided by TREC are listed in Table 2 and Table 3. Comparing hitDELMrun2 with hitQryFBrun4, it indicates that the straight forward integration of the document expansion into the query expansion does not significantly improve the performance. The corresponding analysis is still under going now. Unsurprisingly, the learning to rank technique in hitLRrun1 brings about a better retrieval performance.

The result deserves further elaboration is that the hitURLrun3 achieves best performance. This fact indicates that the webpage of the URL in a tweet is a valid help to decide if the tweet is relevant to a topic. This is somewhat natural since a webpage content is more informative than 140 characters.

Table 2. Results for Highly Relevant Tweets

	P@30	R-Precision	MAP
hitQryFBrun4	0.2345	0.2471	0.2302
hitDELMrun2	0.2350	0.2522	0.2257
hitURLrun3	0.2701	0.2872	0.2642
hitLRrun1	0.2446	0.2628	0.2411

[1] http://nutch. apache. org/

Table 3. Results for All Relevant Tweets

	P@30	R-Precision	MAP
hitQryFBrun4	0.4424	0.3655	0.3186
hitDELMrun2	0.4345	0.3636	0.3197
hitURLrun3	0.4695	0.3751	0.3469
hitLRrun1	0.4379	0.3777	0.3355

3. REAL-TIME FILTERING PILOT TASK

The main measure of this sub-task is T11SU which is biased for precision. In TREC 2011 Microblog Track, we can observe many queries have a retrieval results over 0.4 at P@30[6]. This fact motivates us to use information retrieval results for a good T11SU score. And the retrieval model adopted for this task is what we described at sub-section 2.5. The tweets prior to querytweettime are used as background training data for query expansion, document expansion and θ_C in Dirichlet smoothing method.

3.1 Retrieval Score Based Filtering

The chief difference between the retrieval and filtering is that we have the whole document collection for retrieval, yet we have only partial documents preceding a give time *T* in filtering. To apply the retrieval model, we treat the documents ahead of *T* as the collection, and then retrieve top *m* tweets as the relevant set. To tell whether a newly arrival tweet *X* is relevant to the topic, we simply compute its retrieval score and compare it with the *m-th* tweet. If *X* has smaller score, it is filtered as irrelevant otherwise it is updated into the relevant set.

We examine three strategies to determine the *m* as follows:

1) Fixed top-m method: the baseline method which always keeps most relevant *m* tweets in the relevant set.
2) Dynamic top-m method: to take advantage the observation that the relevant tweets tends to occurred at a close period, we extend *k* tweets in the relevance set; and an irrelevant tweet will result a decrease of |k/1000| in the number of relevant set.
3) Combined Dynamic top-m with tweet's content and Fixed top-m with URL: this method is designed specifically for tweets with URL: only the tweets both in dynamic top-m with tweets' content and fixed top-m with webpage of the URL are judged relevant, otherwise are judged irrelevant; and tweets without URL are all treated as irrelevant.

3.2 Experimental Results

We perform the same data processing as described in sub-section 2.6. At the starting point for a given topic, we used the entire corpus prior to querytweettime as background training data, and the query and querytweet as a positive training example.

Similarly, we submitted four official runs as follows:

- window2run: results from fixed top-m method with m=2;
- hitRSW: results from dynamic top-m method with m=2 and k=8;
- hitUWT: results from a combined method of dynamic top-m with content and fixed top-m with URL , again m=2 and k=8 in dynamic method and m=1000 in fixed method.
- URLAllFB: all tweets are judged as positive to examine the recall of our data collection;

TREC released results of the four runs are provided in Table 4. From these summary results, the fixed top-m method(windows2run)gets a better T11SU but the recall is too low. The dynamic top-m method (hitRSW) improves the recall rate, but more irrelevant tweets are judged positive, which causes a better F_0.5 and a lower T11SU. The dynamic top-m with URL (hitUWT) achieves best T11SU and F_0.5 by introducing the webpage indicated by the URL as external resources. Also, it seems that we lost almost 10% of the relevant tweets in our data collections, which deserves further investigation.

Table 4. Results for Real-time Filtering Pilot Task

	T11SU	F_0. 5	Precision	Recall
window2run	0.3321	0.2055	0.3860	0.0987
hitRSW	0.2942	0.2699	0.2838	0.3440
hitUWT	0.4117	0.3338	0.6219	0.1740
URLAllFB	0	0.0001	0	0.9146

4. CONCLUSION

In TREC 2012 microblog track, we explore the query expansion and document expansion approaches to tweet retrieval. It seems that current document expansion approach is still far from a perfect solution to tweet document modeling. Instead, we find that the webpage of the URL in a tweet can benefit the retrieval process significantly.

In the sub-task of Real-time Filtering Pilot Task, we developed a approach using information retrieval model to filter out the relevant tweet. The results show the approach works well.

5. ACKNOWLEDGEMENTS

The work of this paper is supported by the key project of National High Technology Research and Development Program of China (863 Program, No. 2011AA01A207), the project of National Natural Science Foundation of China (Grant No. 61105072 & 61272384) and the Fundamental Research Funds for the Central Universities.

6. REFERENCES

[1] D. Metzler, C. Cai. USC/ISI at TREC 2011: Microblog Track. In *Proceedings of TREC 2011*, Gaithersburg, USA, 2011.

[2] T. Miyanishi, N. Okamura, X. Liu, K. Seki, K. Uehara. TREC 2011 Microblog Track Experiments at Kobe University. In *Proceedings of TREC 2011*, Gaithersburg, USA, 2011.

[3] Y. Duan, L. Jiang, T. Qin, M. Zhou and H.-Y. Shum. An Empirical Study on Learning to Rank of Tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 295–303, 2010.

[4] V. Lavrenko and W. B. Croft. Relevance-based Language Models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 120-127, 2001.

[5] T. Tao, X. Wang, Q. Mei and C. Zhai. Language Model Information Retrieval with Document Expansion. In *Proceedings of the main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pp. 407-414, 2006.

[6] I. Ounis, J. Lin and I. Soboroff. Overview of the TREC-2011 Microblog Track. In *Proceedings of TREC 2011*, Gaithersburg, USA, 2011.