ICTNET at Microblog Track TREC 2011[†]

Peng Cao[1,2], Jinhua Gao[1,2], Yubao Yu[1,2], Shenghua Liu[1], Yue Liu[1], Xueqi Cheng[1]
[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190
[2]Graduate School of Chinese Academy of Sciences, Beijing, 100190

1. Introduction

The ICTNET group has participated in the Microblog track of TREC 2011. The main task is to search the messy tweets for those about a topic that is represented by a query. There are 50 queries, i.e. topics given for the track totally. Besides the topic description, a query time is also given for each query, indicating the exact time of the query issuance. The search is supposed to be conducted onsite, and those tweets later than the query time should not be returned. Furthermore, the issuers wishes to see the most recent but relevant information to the query. Hence, our system should answer a query by providing a list of relevant tweets ordered from newest to oldest, starting from the time the query was issued.

The existing related work about microblog is mainly focused on users [1-2], information flow [3-4], and tweets' content [5-6]. Our work is to query the tweets' content to find relevant, interesting, and fresh tweets. With exploring the features of tweets' content, hashtags, urls, post time etc., we employ SVM ranking model to rank our query results. The model is trained on pair-wise labeled data. Query extension both within tweets and external Wikipedia articles and Google search results are conducted by pseudo relevance feedback method and keywords extracting. In our experiment, 4 running results of 50 queries are collected on more than 5.6 million English tweets. There are 64.6% relevant tweets retrieved in less than 1000 returned results, and 449 relevant tweets are retrieved in top 30 according our ranking scores.

The rest of the report is organized as follows. Section 2 introduces the data preprocessing. Section 3 describes our main method to rank the search results namely the learning to rank. Section 4 shows the experiment results, and section 5 concludes the paper.

2. Data preprocess

Only English tweets are considered in our work. With estimating the post time of the retweeted and removing those tweets in other languages, we then do the word stemming and stop word removing. Besides, consecutively repeating one letter of a word becomes popular in tweets to express some emotions, such as "goooood", "tooooold", and etc. al. In such case, we simply reduce the times of the repeatness into 2 to map such expressions, and get "good", "toold" as their stems. Afterwards, urls and hashtags of a tweet are extracted out as another two features for the tweet. The "@" links in a tweet are not considered in our model, so "@" links are removed from the tweet content.

One of the most distinguishing features of tweets is that there are quite a bit of slangs and sloppiness words in them, which results in vocabulary mismatch problem. It is obvious that the mismatch problem may affect the query result, since a misspelled word cannot be hit by a correct query word. To solve this problem, we introduce a misspelling list which consists of the mappings from correct words to their commonly misspelled forms. We then extend each query with such a list, so tweets containing misspelled words can be hit in the search result. A Bayes-based model is used for building the misspelling list.

[†]This work is sponsored by NSF of China Grants No. 60903139 and No. 60873243, and by 863 Program of China Grants No. 2010AA012502 and No. 2010AA012503.

3. Learning to Rank

The SVM rank is employed to train our ranking model. A query for tweets is described as a topic, which is more than a list of query words. So it becomes more important to mine the additional words around the topic, *i.e.* query extension. Afterwards, we proposed 6 features, including enhanced BM25, the times of retweetness, freshness, tweet length, the number of urls, and hashtags. With the 6 features and query extension, we train a SVM ranking model based on the labeled data.

3.1 Query extension

There are two ways for the query extension, internal extension and external extension. The internal extension indicates that we only use the corpus itself to extend our query, while the external extension can use the other web resources. If we query with the original query words, the top tweets that have the most ranking scores are more relevant to the topic described by the query. Therefore, the top-K tweets from the original query are used for extracting extension words. When viewing the top-K tweets as a document, the TF-IDF weight is employed to measure the extension words. For each query, we use Wikipedia and search engine such as Google to extent the query words. For each original description of query, we extract the keywords of the search results with TFIDF weight as the query extension. It is worth to mention that most of the words extended from the Wikipedia articles are the same to that from Google search. Besides, the misspelling list are also applied to the query extension.

3.2 Features

We investigate 6 features to model the relevance, interestingness, and freshness. The elapsed time from posting to querying intuitively reflects the freshness. The length of a tweet is generally defined as the number of words after content cleansing and decomposition. Furthermore, the length of tweets must reflect the increment, so that the length of the left part after removing the original tweet is calculated for the retweeted ones. The other 4 features are described in the following, and the distributions of the features are studied as well.

3.2.1 Enhanced BM25

As for measuring the content relevance, we proposed a scoring method for tweets based on BM25. Let Q be a query that consists of query words q_1, q_2, q_3, … . In the BM25 model, the socre $B(T_i, Q)$ for tweet T_i is calculated in the following way.

$$B(T_i,Q)=\sum_{j=1}^{|Q|} IDF(q_j)\bullet\frac{f(q_j,T_i)\bullet(k_1+1)}{f(q_j,T_i)+k_1\bullet(1-b+b\bullet\frac{|T_i|}{avgtl})}$$

where q_j is the jth query word, |Q| is the total number of query words, *avgtl* is the average tweet length, and $|T_i|$ is the length of tweet T_i. The function $f(q_j, T_i)$ indicates the frequency of the query word q_j in tweet T_i. The model parameters k_1 and b are 2.0 and 0.75 separately. $IDF(q_j)$ is defined as follows.

$$IDF(q_j)=\log\frac{N-n(q_j)+0.5}{n(q_j)+0.5}$$

where $n(q_j)$ is the number of tweets that contain query word q_j. It is seen that the score of a tweet matching two different words is the same to that of another matching the same word twice, if the two query words has the same *IDF* value. Since tweets are extremely short, the word frequency actually does not count much, and matching different query words makes much more sense. Therefore, we need a way to boost such a case. The enhanced BM25 finally defines as follows with boosting.

$$B(T_i,Q)=\left(\sum_{j=1}^{|Q|} w_j\bullet IDF(q_j)\bullet\frac{f(q_j,T_i)\bullet(k_1+1)}{f(q_j,T_i)+k_1\bullet(1-b+b\bullet\frac{|T_i|}{avgtl})}\right)\bullet\sum_{j=1}^{|Q|} h_{i,j}$$

where w_j is the weight of query word q_j, and $h_{j,i}$ is a 0-1 binary that indicates whether query word q_j

hits tweet i. w_j is used to measure the confidence that the extension query word q_j belongs to the query topic. As for the misspelling extension, the edit distance divides the weight of the correct one. Simply, we give those original query words weight to be 5, and extension query word 2.

3.2.2 The times of retweetness

In Twitter, the times of retweetness reflect the popularity of a tweet. We include those tweets that are exactly the same to the tweet, and contain “RT” followed by the tweet as a substring for the estimation of retweetness. It shows that the ratio of the tweets that have never been retweeted is about 94.96% in the corpus. Thus we only consider the tweets that are retweeted in the distribution. There are a large number of tweets that are retweeted once, while only a few have been retweeted more than 20 times. The distribution for those retweeted follows the power law. And the retweetness feature is calculated as the following formula.

$$\ln (r_i + 1)$$

where r_i is the times of retweetness of tweet i. r_i equals zero if it is not retweeted.

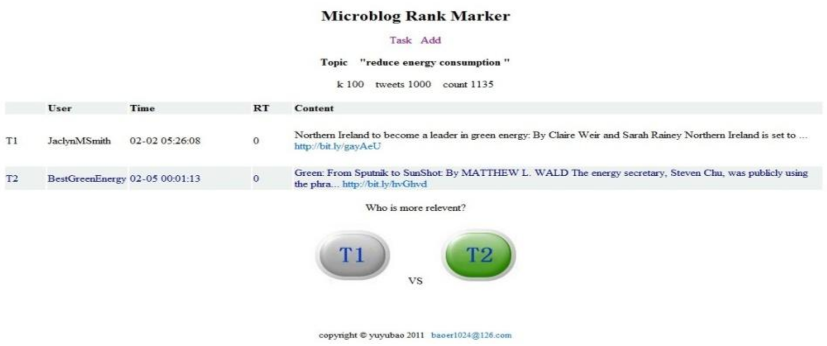
3.2.3 Urls and hashtags

With the data preprocess, we got urls and hashtags extracted from the original tweet content. Since crawling the content of urls is not available, and may bring in the spam link content, we use the number of urls in each tweet as a feature. A tweet with urls usually contains more information than others. Thus the number of urls reflects the value of a tweet in our ranking model.

Hashtags are designed to reflect the topic of a tweet, so they are given much weight in the ranking model. Since a hashtag maybe a combination of several words, we simply use substring match, instead of whole word match as a query hits a hashtag. Similarly as the way we boost content match, the score is also boosted by multiplying the number of different query words got matched. Thus the feature score H_i of hashtags for tweet i is calculated as follows.

$$H_i = \left(\sum_{j=1}^{|Q|} IDF(q_j) \bullet h_{j,i}\right) \bullet \sum_{j=1}^{|Q|} h_{j,i}$$

where $h_{j,i}$ is a 0-1 binary that indicates whether query word q_j hits the hashtages in tweet i.



Microblog Rank Marker

Task Add

Topic "reduce energy consumption "

k 100 tweets 1000 count 1135

	User	Time	RT	Content
T1	JaclynMSmith	02-02 05:26:08	0	Northern Ireland to become a leader in green energy: By Claire Weir and Sarah Rainey Northern Ireland is set to ... http://bit.ly/gayAeU
T2	BestGreenEnergy	02-05 00:01:13	0	Green: From Sputnik to SunShot: By MATTHEW L. WALD The energy secretary, Steven Chu, was publicly using the phra... http://bit.ly/hvGhvd

Who is more relevent?

copyright © yuyubao 2011 baoer1024@126.com

Figure 1 Labeling System

3.3 Labeling

To train our ranking model, we need some acknowledged rank list for some queries which reflects the target of the Ad Hoc search. Unfortunately, the TREC committee does not provide such criteria. We then organized 8 well-trained graduate students in our lab to label the ranking list of several queries that are randomly picked from the 50 ones. As shown in Fig. 1, we build a labeling system to let people compare in pairs, and use the method of quick sort to form the final ranking list. So if we want to label a ranking list of length n, it costs O(logn) comparisons on average.

The labeling system looks like Fig 1. For each query we ask volunteers to decide the 2 tweets namely T1 and T2, which one is better than anther. And we accumulate the result and rank the tweets to train a model.

4. Experiments and Evaluations

There are totally 16,141,812 tweets crawled according to the official seeds of TREC2011, and 5,650,490 English tweets are selected as the benchmark. 49 topics are tested. We submitted 4 different runs. In the 1st run, ICTNET11MBR1, we only use internal query extension and misspelling list. The 2nd run, ICTNET11MBR2, use takes logarithm of enhanced BM25 and different query weight, compare to the first one. The 3rd run extends the queries with the keywords from related wiki articles and google search results, based on the first one. And the last run, ICTNET11MBR4, uses both future tweets and articles for query extension.

Table I. Experimental Results

Total Relevant retrieve @1000 – the total number of retrieved relevant tweets in the 1000 returned results.
Total Relevant retrieve @30 – the total number of retrieved relevant tweets in the top 30 returned results

	Total Relevant retrieve @1000			Total Relevant retrieve @30			MAP			R-precision		
	Official	submit	submit@30	best	submit	submit@30	best	submit	submit@30	best	submit	submit@30
Run1	2864	1839	--	899	67	440	0.51	0.10	0.15	0.61	0.10	0.20
Run2	2864	1593	--	899	63	348	0.51	0.09	0.12	0.61	0.08	0.17
Run3	2864	1845	--	899	67	449	0.51	0.10	0.14	0.61	0.09	0.20
Run4	2864	1850	--	899	59	416	0.51	0.10	0.14	0.61	0.08	0.19

Table I illustrate the experimental results of our submission. The column "Official" gives the total number of the labeled relevant tweets in the corpus. The column "best" gives the total number of the best retrieved tweets among all the team. And the column "submit" is our submission. It is need to mention that we misunderstood the real meaning returned results, and thought that it evaluates according to score sequence, instead of post time sequence. It is quite different in our ranking model. Thus we also evaluate the top 30 ranked tweets in our submission by ourselves, shown in column "submit@30". Nevertheless, the total number of retrieved tweets is 1850 while there is 2864 totally. In the "submit@30", the total number of retrieved tweets in top 30 is 449, while the total best is 899.

5. Conclusion

We propose 6 features to build the SVM Rank model, and by training on user labeled data, 4 runs are submitted with different query extensions and enhanced BM25 models.

Reference

[1] What is Twitter, a Social Network or a News Media? Haewoon Kwak, Changhyun Lee, Hosung Park, Sue Moon WWW2010
[2] TwitterRank: Finding Topic-sensitive Influential Twitters Jianshu Weng, Ee-Peng Lim, Jing Jiang, Qi He WSDM2010
[3] Differences in the Mechanics of Information Diffusion Across Topics: Idioms, Political Hashtags, and Complex Contagion on Twitter Daniel M. Romero, Brendan Meeder, Jon Kleinberg WWW2011
[4] Who Says What to Whom on Twitter Shaomei Wu,Jake M. Hofman, Winter A. Maso, Duncan J. Watts WWW2011
[5] Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors Takeshi Sakaki,Makoto Okazaki,Yutaka Matsuo WWW2010
[6] Predicting the Future with Social Media[J]. Arxiv preprint arXiv:1003.5699. 2010. S. Asur, B. A. Huberman.
[7] T. Joachims, Training Linear SVMs in Linear Time, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), 2006
[8] T. Joachims, *Optimizing Search Engines Using Clickthrough Data*, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002