DUTIR at TREC 2011 Microblog Track

Cunhui Shi, Kejiang Ren , Hongfei Lin, Shaowu Zhang
{smart, renkj}@mail.dlut.edu.cn   hflin@dlut.edu.cn
School of Computer Science and Technology, Dalian University of Technology, Dalian 116024

Abstract

In TREC 2011 Microblog Track, we explore the use of pseudo relevance feedback to expand original query terms in topics. Hyperlink is used to enhance the performance of the retrieval results. And we set a threshold of entropy to filter retrieval results. Microblog is a Realtime Adhoc Task, so we make use of average querytweettime that comes from pseudo relevance feedback to change retrieval score. We combine two models to improve retrieval results. The results show that our model is effective at realtime relevance retrieval.

Keywords

Microblog retrieval, Language Model, VSM, Feedback

1. Introduction

In this work, we investigated a strategy of pseudo relevance feedback to expand original query terms in topics, as well as the combining two models of LM and VSM to improve the quality of a search result list. Our experiments mainly focus on the following three aspects: (1) how does the traditional retrieval models perform in microblogging environments? (2) How does the traditional retrieval models combining with pseudo relevance feedback and other microblog feature, such as entropy, hyperlink, perform in microblog information retrieval. (3) Due to the short text feature of microblog, can combining two retrieval models improve the quality of a search result list?

The Microblog track examines search tasks and evaluation methodologies for information seeking behaviors in microblogging environments. The goal of the task is to return a ranking of the documents in the collection in the order of querytweettime of relevance. The results are assessed mainly by P@30[1].

The rest of this paper is organized as follows: In Section 2, we describe the traditional retrieval models and relevance feedback model used in this paper. In section 3, we present the experimental for Microblog Track task. In section 4, we present our official results in microblog track. In section 5, we conclude the paper and future work.

2. Retrieval Models

2.1 Vector Space model (VSM)

In VSM, documents and queries are assumed to be part of a t-dimensional vector space, where t is the number of index terms. The typical form of document term weighting in the VSM[2] are:

$$w_{ij} = \frac{(\lg tf_{ij} + 1.0) \times idf_j}{\sum_{j=1}^{t} [(\lg tf_{ij} + 1.0) \times idf_j]^2}, \quad idf_j = \log \frac{N}{n_j}, \quad tf_{ij} = \frac{f_{ij}}{\sum_{k=1}^{t} f_{ik}}$$

Where idf_j is the inverse document frequency weight for term j and tf_{ij} is the term frequency weight of term j in document I, w_{ij} is the weight of a term j in document i. N is the total number of documents in the collection, and n_j is the number of documents in which term k occurs. In our experiments, we take advantage of Lucene toolkit[3] to implement VSM.

2.2 Language model (LM)

A language model, which first used in information retrieval by Ponte and Croft[4], representation of a document can be used to generate the query by sampling terms according to the probability distribution. The probability of producing the query for a given document model as follows:

$$\hat{p}(Q \mid M_d) = \prod_{t \in Q} \hat{p}(t \mid M_d) \times \prod_{t \notin Q} 1.0 - \hat{p}(t \mid M_d)$$

Where t is a term in query Q, the function $\hat{p}(t \mid M_d)$ is as follows:

$$\hat{p}(t \mid M_d) = \begin{cases} p_{ml}(t,d)^{(1.0-\hat{R}_{t,d})} \times p_{avg}(t)^{\hat{R}_{t,d}} & if \ \ tf_{(t.d)} > 0 \\ \frac{cf_t}{cs} & otherwise \end{cases}$$

Where cf_t is the count of term t in the collection and cs is the collection size or the total number of tokens in the collection, $tf_{(t,d)}$ is the raw term frequency of term t in document d. The other functions calculate as follows:

$$\hat{R}_{t,d} = (\frac{1.0}{1.0+\overline{f}_t}) \times (\frac{\overline{f}_t}{1.0+\overline{f}})^{tf_{(t,d)}}, \quad \hat{p}_{avg}(t) = \frac{(\sum_{d_{(t \in d)}} p_{ml}(t \mid M_d))}{df_t}, \quad \hat{P}_{ml}(t \mid M_d) = \frac{tf_{(t,d)}}{dl_d}$$

Where $\overline{f}$ is the mean term frequency of term t in documents where t occurs, df_t is the document frequency of t, dl_d is the total number of tokens in document d. θ_Q for query and the other is LM θ_D for document.

Zhai et al.[9] proposed a more universal language model which known as the Kullback-Leibler divergence retrieval model. Assume that a query q is obtained as a sample from a generative model θ_Q and a document d is generated by a model θ_D. The distance value of d with respect to q can be measured by KL divergence:

$$D(\theta_Q \mid \theta_D) = -\sum_w P(w \mid \theta_Q) \log P(w \mid \theta_D) + cons(q)$$

The cons(q) is dropped because of no affecting ranking of document. Therefore the final ranking is based on the cross entropy of the query language model with respect to the document language model. The language model is based on the Indri Retrieval Toolkit[5].

2.3 Rocchio Query expansion

Rocchio's[6] approach which used the VSM to rank document provides a general framework for implementing relevance feedback. Rocchio feedback model can also be used for pseudo relevance based on query expansion. The terms weight of expanded query is calculated as follows:

$$Q' = \alpha Q + \beta \sum_{rel} R_i - \gamma \sum_{non-rel} S_i$$

Where Q' and Q represent the expanded and original query vectors, R_i and S_i are individual terms of the relevant set of R and non-relevant set of S. The weights α , β and γ are weights.

In our experiments, the parameters α and γ are set constant value 1 and 0, the parameter β is a variable value changing with the document score which calculate from VSM or LM in the initial retrieval. The first ten documents which initial retrieval are considered relevance. The terms which are in the first ten retrieval document leaved for the count greater than two. The terms weight of the set R is calculated by TF-IDF, in which the collection size is set ten of the pseudo relevance documents.

3. Experiments

This section introduces our experiments in detail. The structure of the retrieval system is shown in Figure 1.

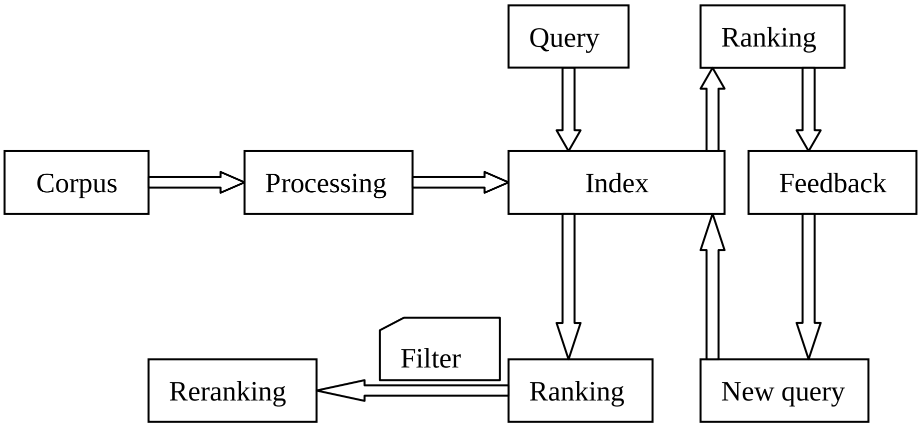

Figure 1. The framework of the microblog retrieval system.

The preprocessing phase is designed to extract the English tweets from the raw corpus for the reason that the non-English tweets are judged non-relevant. The future evidence does not used in the system in order to meet the real-time retrieval. Consequently before the first retrieval, we built index for each query respectively in which the tweet querytweettime is less than query querytweettime. Also the external evidence did not used in the system.

Owing to the final retrieval results are ranked by the mean of tweets' querytweettime, the parameter t is used to decrease the score of the result which the querytweettim is greater but non-relevance. It is calculated as follows:

$$t = \begin{cases} -e^{-\frac{t_i - t_{avg}}{t_{\max} - t_{avg}}} & if \quad t_i - t_{avg} > 0 \\ e^{-\frac{t_i - t_{avg}}{t_{\max} - t_{avg}}} & others \end{cases}$$

Where t$_i$ is the querytweettime of tweet in result, t$_{avg}$ and t$_{max}$ are the average querytweettime and the maximum querytweettime of pseudo relevance feedback result list of ten respectively.

Since microblog is a short text containing no more than 140 characters, whether microblog has a hyperlink is important feature that imply the microblog has more information. It is used to enhance the final score of the retrieval results. The final score SC$^{'}$(q,d) of ranking is calculated as follows:

$$SC^{'}(q,d) = SC(q,d) \times (1 + a * t + b * L)$$

Where SC(q,d) is the score of query q and document d calculated by VSM or LM combing with pseudo relevance feedback, if the tweet contains a link, L is assigned a value of positive one, otherwise L is zero, the weights a and b are used to adjust the weight of t and L. In VSM, we set the weights a and b are both 0.5. In LM, they are set 0.5 and 0.1 respectively.

Entropy is another important characteristic of information in microblog that the more the number of words differently the greater the entropy. The calculation function is as follows:

$$E = -\sum_{i=1}^{n} p_i \log_2 p_i$$

Where p_i is the frequency within the document, and n is the total independent term's size of the document. We set a threshold of entropy to filter retrieval results. The result which entropy is more than 2 is withheld. Relevance score is also used to filter the final results. The retrieval results which score less than 0.15 are discarded. And then the results are re-ranking using the querytweettime.

Different relevance ranking model in information retrieval are likely to retrieval new documents which are not shared between them. We combine the LM and VSM in a linear:

$$MixSC(q,d) = \lambda SC(q,d)_{LM} + (1-\lambda) SC(q,d)_{VSM}$$

Where MixSC(q,d) is the score after combination, $SC(q,d)_{LM}$ and $SC(q,d)_{VSM}$ are score of LM and VSM respectively. In our experiments λ is set 0.7.

In all the considered approaches, we did a topic-based normalization before combination on the score of different methods, in order to normalize the positive values into values in the range [0, 1]. The normalized value of the score $SC(q, d_i)$ between query q and document d_i is calculated as follows:

$$SC_i(q,d_i) = \frac{SC_i(q,d_i) - \min\{SC(q,d_i)\}}{\max\{SC(q,d_i)\} - \min\{SC(q,d_i)\}}$$

Where the $\max\{SC(q, d_i)\}$ and $\min\{SC(q, d_i)\}$ are the maximum and minimum value of $SC(q, d_i)$ respectively for all the documents in the collection.

4. Results

There are four runs submitted. Table1shows our official runs considering all relevant and highly relevant tweets as relevant being over 49 topics for Microblog Track. The runs dutirTfidfFb and dutirLmFb take account of VSM and LM respectively combining with pseudo relevance feedback. The other two runs dutirMixSp and dutirMixFb are combination of two retrieval models LM and VSM, but the latter run used the pseudo relevance feedback.

Table 1: MAP and P@30 of 4 runs for all relevance

Run id	MAP	R-prec	P@30
dutirTfidfFb	0.2219	0.3100	0.2939
dutirLmFb	.0.2851	0.3483	0.3224
dutirMixSp	0.2674	0.3426	0.3129
dutirMixFb	0.2936	0.3633	0.3408

Table2 shows our official runs considering only highly relevant tweets being over 33 topics.

Table 2: MAP and P@30 of 4 runs for highly relevance

Run id	MAP	R-prec	P@30
dutirTfidfFb	0.1684	0.1980	0.0949
dutirLmFb	0.2375	0.2262	0.1071
dutirMixSp	0.2177	0.2137	0.1020
dutirMixFb	0.2352	0.2374	0.1162

From Table1 and Table2, the performance of runs using the pseudo relevance feedback is significantly better than runs do not use feedback. And different relevance ranking model in information retrieval are retrieval new documents which are not shared between them. Since we filter the results taking advantage of entropy and final score, the result which has little terms of difference and low relevant score are discarded. And also enhance score of document having hyperlink, in general it usually include more information. In realtime task, the retrieval results which the querytweettime is huge have greater impact on the final ranking, so we decrease the score of those documents and increase the score of results of small querytweettime using the average querytweettime calculating from pseudo relevance feedback results. The effectiveness is obvious using querttweettime feature to improve results of ranking.

5．Conclusion

In this paper, we present our methods and experiments in Microblog track 2011. We evaluate traditional IR models VSM and LM in our experiments, combining with pseudo relevance feedback and other microblog feature, such as entropy and hyperlink. Taking account of the microblog track is a Realtime Adhoc Task, we use entropy to filter the tweets containing fewer information. And also we set a threshold discarding the results which the relevance score is small.

References

[1] 2011 Track Guidelines. http://sites.google.com/site/microblogtrack/

[2] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. Information Processing and management, 24 (5):513-523, 1988

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

[4] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. SIGIR’93, Melbourne, Australia.

[5] http://www.lemurproject.org/indri/.

[6] J. Rocchio. Relevance Feedback in Information Retrieval, pages 313-323. 1971.

[7] D. A Grossman and O. Frieder. Information retrieval: algorithms and heuristics, second Edition. Pp.11-18, 96-97, 2009.

[8] Zhai C and Lafferty J. Model-based feedback in the language modeling approach to information retrieval.CIKM'01: Proceedings of the tenth international conference on Information and knowledge management, 2001: 403-410.