THUIR at TREC 2005 Terabyte Track[1]

Le Zhao, Rongwei Ceng, Min Zhang, Yijiang, Jin, Shaoping Ma
State Key Lab of Intelligent Tech. & Sys., CST Dept, Tsinghua University, Beijing 100084, China
zhaole@tsinghua.org.cn

Abstract: IR group of Tsinghua University this year has used its TMiner text retrieval system for indexing and retrieval of the Terabyte track ad hoc and named-page subtasks. In doing the two tasks, we used the in-link anchor texts (the anchor of the URLs that point to the current page in the collection) together with the content texts of the web pages for building the indices. When retrieving, the word-pair method [1] was used and proved effective on 2004 and 2005 Terabyte ad hoc task topics and the 2005 named-page task. We analyze the performance of word-pair method in comparison with the Markov random field term dependence model of [2] and a generative phrase model we proposed, which is natural on the language modeling framework [3].

1. TMiner at Terabyte 2005

On a PC of 2GB memory, with one CPU and IDE hard disks, TMiner could index 50GB text (about 200GB HTML files) with tolerable time. But since the terabyte collection contains about 100GB pure text (110GB including anchor texts), building one single index for such a large collection would cost TMiner too much time. We built 27 indices for the 27 parts of the collection in our experiments. When retrieving, we summed the DF values of the query terms from each index, and assigned the BM2500 RSV to documents in the collection according to the DF sum. This distributed index system returns exact RSV as if only one single index is constructed for the whole collection (at the expense of additional query processing time).

In the ad hoc and named-page tasks, the index of in-link anchor combined with page content was used. This is the most effective way of combining anchor text for retrieval in our observation and we didn't build indices that contain no in-link anchor for comparison.

In addition to the use of anchor text, since the indices we built contains full position information for the index terms, the word-pair method [1] was used in both tasks.

2. Query length and the word-pair method

In this section we provide performance of the word-pair method on the Terabyte 2004 ad hoc topics. We varied the query length, from keyword queries (title only) to verbose queries (title + description + narrative).

Table 1. MAP/BPREF measures for the word-pair(wp) method with different parameters

2004 TB Ad hoc	wp 0.0 (baseline)	wp 0.1	wp 0.2	%**Max increase**
Keyword query	0.2611/0.3259	**0.2676**/**0.3332**	0.2532/0.3315	2.49%/2.24%
Verbose query	0.3015/0.3841	0.3198/0.4008	**0.3280**/**0.4084**	8.79%/6.33%

[1] This work is supported by the Chinese National Key Foundation Research & Development Plan (2004CB318108), Natural Science Foundation (60223004, 60321002, 60303005, 60503064) and the Key Project of Chinese Ministry of Education (No. 104236)

Increase for a limited range of word-pair parameter (0.1-0.2) could always be observed over the baseline. The word-pair improvement for keyword queries is lower than that of the verbose queries. This can be justified through how users construct keyword queries. Since most keywords are key terms from longer sentences or phrases, keyword queries do not tend to be exact phrases which can be used by the word-pair method. However, for the case of verbose queries, more phrases will occur and will be adopted for improvement of the retrieval performance by the word-pair method. Not surprisingly, the word-pair method performs better on precise queries than on the more noisy queries, as experiments of Chinese text retrieval revealed [7].

3. Comparison between term dependency models

$$W = (1-\lambda) \sum_{t \in query\ terms} W^{(1)} \frac{(k_1+1)\, tf_t}{(K+tf_t)} \frac{(k_3+1)\, qtf_t}{(k_3+qtf_t)} + \lambda \sum_{wp \in query\ wordpairs} W_2$$

$$where \quad W^{(1)} = \log \frac{N - df_t + 0.5}{df_t + 0.5},\ W_2 = W_2^{(1)} \frac{(k_1+1)\, tf_{wp}}{(K+tf_{wp})} \frac{(k_3+1)\, qtf_{wp}}{(k_3+qtf_{wp})},\ W_2^{(1)} = \log \frac{N - df_{wp} + 0.5}{df_{wp} + 0.5}$$

(1.1)

The above formula is the word-pair method based on the BM2500 retrieval formula. In this method, the RSV from the word-pairs is interpolated linearly with that of the unigram terms. This method is actually a discriminative model [6] which incorporates unigram and bigram features of the text.

Notice that the RSV from the BM2500 formula is actually the logarithm of the probability that a document is relevant with respect to the query. Thus, the linear combination seems very artificial; the coefficients no longer distribute probabilistic mass, in stead, the log(probability) is distributed. A better model based on probabilistic theory should use λ to model the prior probability that the underlying model is a bigram phrase model (or word-pair model) in stead of a unigram model. This prior probability should be allowed to vary as the query terms change and as the documents change, since different words have different probabilities to form phrases (word-pairs) and authors of different articles should be allowed to have different preferences of using certain contiguous words as phrases. [4] also expressed such an intention.

A more rigorous model should be:

$$\mathrm{P}(Q \mid D) = \sum_{\theta_i} \{\mathrm{P}(Q \mid \theta_i, D) \mathrm{P}(\theta_i \mid D)\} \tag{1.2}$$

This model is a generative phrase model based on the language modeling framework for IR. Our λ and $1-\lambda$ is now $\mathrm{P}(\theta_i \mid D)$, which models the probabilities of the underlying phrase models (whether query words q_i and q_{i+1} are treated as a phrase, thus as a single term in retrieval) which the author of document D assumed when composing D. We know that the sum $\sum_i \mathrm{P}(\theta_i \mid D) = 1$, just as $\lambda + (1-\lambda) = 1$ in the word-pair method.

Based on the language model assumption, the generation probability of a query *Q* by a model θ_i is surely independent of the actual document *D* which is generated by θ_i. The above model

could be simplified as

$$\mathrm{P}(Q \mid D) = \sum_{\theta_i} \{\mathrm{P}(Q \mid \theta_i) \mathrm{P}(\theta_i \mid D)\} \tag{1.3}$$

Of course query could also have underlying generative models. But let us first discuss the above less general model which is already problematic.

The model of (1.3) suffers from the sparse data problem already, the traditional smoothing methods like Dirichlet smoothing [5] fails on bigram phrases, because bigrams are even sparser than non-common terms. We have not devised an appropriate way for smoothing the bigrams and this method performed badly. This failure is actually the failure of linearly combining the probabilities directly. Interestingly, linear combination of the log(probabilities) based on language model was observed to be quite effective [2] (though the combining coefficients were not individualized for each document in the cited work). Questions arise when we compare the above three dependency models – linear combination of log-probability (1.1), linear probability (1.3) and the linear log probability (Markov random field) model in [2]. Should we combine the probabilities only after we take logarithm for retrieval? Why does direct linear combination of probabilities from different models like phrase model and unigram model fail since it follows directly from the Bayes rule? Previous research machine learning and recent works in IR [6] showed that possibly because of the sparsity of the data, generative methods could fail from the intermediate step of estimating the class-conditional probability and discriminative models would be favored more.

4. Results submitted

Table 2. TREC 2005 Terabyte ad hoc/named-page tasks THUIR submitted runs

ad hoc Run Tag	Description	MAP	R-Precision	BPREF
THUtb05SQWP1	Keyword query. wp 0.1	0.3032	0.3650	0.3202
THUtb05LQWP1	Verbose query. wp 0.1	**0.3366**	**0.3835**	**0.3484**
THUtb05VQWP2	Verbose query. wp 0.2	0.3351	0.3796	0.3464
named-page Tag	Description	MRR	success in top10	failure in top1000
THUtb05npB	baseline BM2500	0.426	138/252	48/252
THUtb05npW15	word-pair 0.15	**0.463**	**155/252**	**45/252**
THUtb05npWP2	word-pair 0.2	0.455	150/252	**45/252**

The results are consistent on 2005 tasks as on the 2004 topics. Word-pair method is stable.

Reference

[1] M. Zhang, C. Lin, Y. Liu, L. Zhao, S. Ma, THUIR at TREC 2003: Novelty, Robust and Web. In the proceedings of the Twelfth Text REtrieval Conference (TREC 2003), page 556 Gaithersburg, Maryland, 2003.

[2] D. Metzler and W.B. Croft, A Markov random field model for term dependencies. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 472--479, 2005.

[3] J. Ponte and W.B. Croft, A language modeling approach to information retrieval. In

Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, 275--281, 1998.

[4] F. Song and W.B. Croft, A general language model for information retrieval. In Proceedings of Eighth International Conference on Information and Knowledge Management (CIKM'99), 1999.

[5] C. Zhai and J. Lafferty, A study of smoothing methods for language models applied to ad hoc information retrieval. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001.

[6] R. Nallapati. Discriminative models for information retrieval. In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 2004.

[7] L. Zhao, R. Cen, C. Wang, W. Qi, Y. Jin, M. Zhang and S. Ma. 2005 THUIR Report for 863 Information Retrieval Evaluation. To appear in Journal of Chinese Information. 2005.