Document Re-ordering Based on Key Terms in Top Retrieved Documents

Yang Lingpeng, Ji Donghong, Nie Yu, and Zhou Guodong

Institute for Infocomm Research,
21, Heng Mui Keng Terrace,
Singapore 119613
{lpyang, dhji, ynie, zhougd}@i2r.a-star.edu.sg

Abstract. In this paper, we propose a method to improve the precision of top retrieved documents by re-ordering the retrieved documents in the initial retrieval. To re-order the documents, we first automatically extract key terms from top N (N<=30) retrieved documents, then we collect key terms that occur in query and their document frequencies in top N retrieved documents, finally we use these collected terms to re-order the initially retrieved documents. Each collected term is assigned a weight by its length and its document frequency in top N retrieved documents. Each document is re-ranked by the sum of weights of collected terms it contains. In our experiments on 42 query topics in NTCIR3 Cross Lingual Information Retrieval (CLIR) dataset, an average 17.8%-27.5% improvement can be made for top 10 documents and an average 6.6%-12% improvement can be made for top 100 documents at relax/rigid relevance judgment and different parameter setting.

1 Introduction

For Chinese Information Retrieval where query is a short description by natural language, many retrieval models, indexing strategies, query expansion strategies and document re-ordering methods have been proposed. Chinese Character, bi-gram, n-gram (n>2) and word are the most widely used indexing units. The effectiveness of single Chinese Characters as indexing units has been reported in [7]. The comparison between the three kinds of indexing units (single Characters, bi-grams and short-words) is given in [5]. It shows that single character indexing is good but not sufficiently competitive, while bi-gram indexing works surprisingly well and it's as good as short-word indexing in precision. [9] suggests that word indexing and bi-gram indexing can achieve comparable performance but if we consider the time and space factors, it is preferable to use words (and characters) as indexes. It also suggests that a combination of the longest-matching algorithm with single characters is a good method for Chinese IR and if there is a module for unknown word detection, the performance can be further improved. Some other researches give similar conclusions. Bi-gram and word are considered as the top two indexing units in Chinese IR and they are also used in many reported Chinese IR systems.

A. Gelbukh (Ed.): CICLing 2005, LNCS 3406, pp. 572 – 583, 2005.

Regarding retrieval models, two models are most widely used in Chinese Information Retrieval, i.e., Vector Space Model [12] and Probabilistic Retrieval Model [2].

For query expansion, most strategies make use of the top *N* retrieved documents in initial retrieval [11]. Generally, it selects *M* indexing units from the top *N* documents according to some criteria and adds these *M* indexing units to original query to form a new query. In such a process of query expansion, it's supposed that the top *N* documents are related with original query. However in practice, such an assumption is not always true. Although many literatures report that query expansion can improve the recall in many situation, they also suggest that the actual relevance quality of top retrieved documents affects the effectiveness of query expansion.

While query expansion tries to improve the recall of top retrieved documents, document re-ordering is used to improve the precision of top retrieved documents.

Lee, K. et.al. propose a document re-ranking method which uses document clusters [6]. Firstly, they build a hierarchical cluster structure for the whole document set; secondly, they divide top retrieved documents into some clusters, that is, they find sub-trees in hierarchical cluster structure which contain some retrieved documents by some criteria; finally, they calculate similarity between each cluster and each query topic, and use the similarity to adjust the similarity between query and each document in this document cluster. It's reported their method achieves significant improvements on their experiments on Korean corpus. One difficulty of this method is it needs to build hierarchical cluster structure for document set.

Kamps, J. [4] propose a method to re-order retrieved documents by making use of manually assigned controlled vocabularies in documents. By building a controlled vocabulary - controlled vocabulary matrix on co-occurrences, each document can be represented as a vector by controlled vocabularies which occur in and each query can be represented as a vector by the vectors of top N retrieved documents. Finally, each document is re-ordered by the distances between the document vector and query vector. It's reported this re-ranking strategy significantly improves retrieved effectiveness on their experiments on German GIRT and French Amaryllis collections. This method depends on the controlled vocabularies assigned to document, but in most case, no controlled vocabulary is assigned to documents.

Qu, Y. L. [10] uses manually built thesaurus to re-rank retrieved documents. Each term in query topic is expanded with a group of terms in thesaurus. It's a hard job to manually build a large thesaurus for unexpected query topics.

Bear J. el al. [1] use manually constructed or automatically learned small grammars for topics to re-order documents by matching grammar rules in some segment in articles. But grammar construction itself is a difficult problem in Chinese language.

Yang, L.P., et. al [14,15] use extracted long terms in query and document to re-order retrieved documents in Chinese IR. Firstly, they cluster the whole document set into some clusters; secondly, they automatically extract global key terms from these clusters; thirdly, they make use of these global terms and their frequencies to find local terms in a query or a document; finally, they use long local terms to re-calculate the similarity between query and document, and use the new similarity value to re-order retrieved documents. Their experiments show that long terms play an important role in document re-ordering, since they tend to be more significant for the retrieval precision than short terms. It's reported their experiments based on NTCIR3 CLIR dataset can achieve an average 10%-11% improvement for top 10 documents and an average 2%-5% improvement for top 100 documents. One difficulty of this method is how to identify local key terms in query and document because there are a few parameters needed to set.

In this paper, we propose an approach to re-order retrieved documents. Firstly, we automatically extract key terms from each document in document set; secondly, we use key terms in top N retrieved documents and their document frequencies to re-order top retrieved K (N<K) documents.

The rest of this paper is organized as following. In section 2, we describe how to automatically extract key terms from document. In section 3, we describe how to re-order retrieved documents. In section 4, we evaluate the performance of our proposed method on NTCIR3 CLIR dataset and give out some result analysis. In section 5, we present the conclusion and some future work.

2 Term Extraction

We use a seeding-and-expansion mechanism to extract terms from documents. The procedure of term extraction consists of two phases, seed positioning and term determination. Intuitively, a seed for a candidate term is an individual word (or Chinese character) within the term, seed positioning is to locate the rough position of a term in the text, while term determination is to figure out which string covering the seed in the position forms a term.

To determine a seed needs to weigh the individual words to reflect their significance in the text in some way. To do so, we make use of a very large corpus r as a *reference*. Suppose s is the text of the collected summaries, w is an individual word in the text, let $P_r(w)$ and $P_s(w)$ be the probability of w occurring in r and s respectively, we adopt 1), *relative probability* or *salience* of w in s with respect to r [13], as the criteria for evaluation of seed words.

1) $P_s(w) / P_r(w)$

We call w a *seed* if $P_s(w) / P_r(w) \geq \delta$ (δ>0).

We have the following assumptions about a term.

i) a term contains at least a seed.
ii) a term occurs at least L $(L>1)$ times in the text.
iii) a *maximal word string* meeting i) and ii) is a term.
iv) for a term, a *real maximal substring* meeting i) and ii) without considering their occurrence in all those terms containing it is also a term.

Here a *maximal word string* meeting i) and ii) refers to a word string meeting i) and ii) while no other longer word strings containing it meet i) and ii). A *real maximal substring* meeting i) and ii) refer to a real substring meeting i) and ii) while no other longer real substrings containing it meet i) and ii).

Figure 1 describes the procedure to extract key terms from a document *d*.

```
let F_d(t) represents the frequency of t in d;
let O is a given threshold (O>1);
T = { };
collect Seeds in d into S;
for all c∈S
        let Q = {t: t contains c and F_d(t)≥O};
        while Q ≠ NIL
                max-t ← the longest string in Q;
                T ← T + { max-t };
                Remove max-t from Q;
                for all other t in Q
                        if t is a substring of max-t
                                F_d(t)← F_d(t)- F_d(max-t);
                                if F_d(t)<O
                                        removing t from Q;
return T as key terms in document d;
```

Fig. 1. Key Term Extraction from Document *d*

3 Document Re-ordering

We make use of the information of key terms and their document frequencies in top N (N<=30) retrieved documents to re-order top K (N<K) retrieved documents. Firstly, we automatically extract key terms from each document; secondly, we collect key terms which are sub-string of query topic and their document frequencies in top N retrieved documents; thirdly, we assign collected key terms weight by their length and document frequency, that is, more weight is given to more long key term and more weight is given to more document frequent key term; finally, we re-rank retrieved documents by the sum of weight of key terms they contain.

Given query *q*, following is the procedure to re-order the *K* initial retrieved documents by making use of top *N* (*N*<*K*) retrieved documents:

Step 0: Let $\{d_1, d_2, \ldots, d_i, \ldots, d_K\}$ denote the top *K* initial retrieved documents;

Let $R=\{R_1, R_2, \ldots, R_i, \ldots, R_K\}$ denote the similarity values between q and d_i in initial retrieval;

Let $S=\{S_1, S_2, \ldots, S_i, \ldots, S_K\}$ denote the similarity values between q and d_i after document re-ordering;

Step 1: Extract key terms from top *N* retrieved documents;

Step 2: Collect key terms which occur at query *d*;

Let these collected key terms form term set $T=\{T_1, T_2, \ldots, T_n\}$; their document frequencies in top *N* retrieved documents form set $D=\{DF_1, DF_2, \ldots, DF_n\}$;

Step 3: Assign each term T_i in T a weight W_i by following formula:

$$W(T_i) = \text{sqrt}(|T_i|) \times \text{sqrt}(DF_i).$$

where $|T_i|$ is the length of term T_i, i.e., the number of Chinese characters in term T_i and sqrt(x) is the square root of x.

Step 4: Calculate new similarity value S_i between query q and each document d_i in K initial retrieved documents;

Step 4.1: Calculate re-ranking weight w of d_i by key terms in d_i;

$w = 0$;
$T' = T$;
do while (T' is not empty)
  Find the longest key term t in T';
  If t occurs in document d_i, Then
    $w = w + W(t)$
    Remove all occurrence of t in d_i
    Discard t from T';

Step 4.2: Calculate new similarity value S_i between q and d_i

if $w > 0$ then
  $S_i = w \times R_i$
else
  $S_i = R_i$

Step 5: Re-order top K retrieved documents by their new similarities values $S=\{S_1, S_2, \ldots, S_\text{i}, \ldots, S_\text{K}\}$.

4 Experiments and Evaluation

We use NTCIR3 CLIR dataset as our test dataset. The dataset contains Chinese document set CIRB011 (132,173 documents from China Times, China Times Express, Commercial Times, China Daily News and Central, Daily News) and CIRB20 (249,508 documents from United Daily News). We also use the Chinese-Chinese D-run query topics in NTCIR3 CLIR as query topics. There are 50 query topics released in NTCIR3, but only 42 topics are finally used to evaluate. Each query is a simple description of a topic by Chinese language. (Appendix lists the 42 query topics. You may also find more information about NTCIR3 CLIR task from http://research.nii.ac.jp/ntcir-ws3/work-en.html).

For initial retrieval, we use bi-gram as index unit and we use vector space model to represent documents and queries. Each document or query is represented as a vector in vector space where each dimension of vector is a bi-gram. The weight of bi-gram t in document d is given by the following TF/IDF weight scheme:

$$w(t, d)=\log(\text{T}(t, d)+1) \times \log(N/\text{D}(t)+1)$$

where $w(t, d)$ is the weigh given to t in d, $\text{T}(t, d)$ is the frequency of t in d, N is the number of documents in document set, $\text{D}(t)$ is the number of documents in document set which contain t.

The weight of bigram t in query q, $w(t, q)$, is given by the following weight scheme:

$$w(t, q) = \mathrm{T}(t, q)$$

where $\mathrm{T}(t, q)$ is the frequency of t in q.

The similarity (distance) between a document d and a query q is calculated by the cosine of the document vector and the query vector.

The initial retrieval result is used as 1st baseline to evaluate our proposed method; we also use Yang L.P et.al. [14]'s result on NTCIR3 CLIR dataset as 2nd baseline.

Our experiments re-rank the top 1000 initial retrieved documents and evaluate the effectiveness by precisions at different document levels. We use NTCIR3's relax relevance judgment and rigid relevance judgment to measure the precision of retrieved documents. Relax Relevance Judgment considers highly relevant documents, relevant documents and partially relevant documents, while Rigid Relevance Judgment only considers highly relevant documents and relevant documents. We use PreAt10 and PreAt100 to separately represent the precision of top 10 retrieved documents and top 100 retrieved documents.

Our experiments focus on two parts: Which kind of key terms in documents will be used to re-order retrieved documents? How many top retrieved documents should we use to extract key terms from? For the first part, we extract different key terms by using different parameters in our term extraction method. There are two parameters in our term extraction method. One parameter is δ - the minimum saliency of seed in term, the other parameter is L - the minimum occurrence of term in document. For the second part, we only test parameter N - the number of top retrieved documents that are used to extract terms from. Following is the parameter setting in our experiments:

δ=1, 10: We consider terms which contain at least a seed whose salience is 1 or 10;

L=2, 3, 4: We consider terms which occur at least 2 times, 3 times or 4 times in document;

N=20, 25, 30: We consider top 20, 25 or 30 retrieved documents as related documents and extract key terms from them to re-order retrieved documents.

Table 1-6 gives the comparison of precisions at different parameters setting. In table 1-6, column [PreAt10(relax)] represents the average precision of 42 topics on PreAt10 relax relevance judgment; Column [PreAt10(rigid)] represents the average precision of 42 topics on PreAt10 rigid relevance judgment; Column [PreAt100(relax)] represents the average precision of 42 topics on PreAt100 relax relevance judgment; Column [PreAt100(rigid)] represents the average precision of 42 topics on PreAt100 rigid relevance judgment. Row [BaseLine1] represents the initial retrieved result; Row [BaseLine2] represents experiment result reported on Yang et. al [14]; Row [N=20] represents the re-ordered result which make use of key terms in top 20 retrieved documents; Row [N=25] represents the re-ordered result which make use of key terms in top 25 retrieved documents; Row [N=30] represents the re-ordered result which make use of key terms in top 30 retrieved documents. Each item in table represents the precision and its improvement over [BaseLine1] at the conditions expressed by Column and Row.

Table 1. Statistics on (δ=1, L=2)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3619	0.2595	0.1886	0.1279
BaseLine2	0.4052 (12%)	0.2871 (10.6%)	0.1926 (2.1%)	0.133 (4%)
N=20	0.4143 (14.5%)	0.3024 (16.5%)	**0.2055 (9%)**	**0.1376 (7.6%)**
N=25	**0.4262 (17.8%)**	**0.3143 (21.1%)**	0.2052 (8.8%)	0.1371 (7.2%)
N=30	0.4167 (15.1%)	0.3119 (20.2%)	0.2048 (8.6%)	0.1369 (7%

Table 2. Statistics on (δ=1, L=3)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3619	0.2595	0.1886	0.1279
BaseLine2	0.4052 (12%)	0.2871 (10.6%)	0.1926 (2.1%)	0.133 (4%)
N=20	0.4119 (13.8%)	0.3001 (15.6%)	0.205 (8.7%)	0.1376 (7.6%)
N=25	**0.4333 (19.7%)**	**0.3167 (22%)**	0.2079 (10.2%)	0.1381 (8%)
N=30	**0.4333 (19.7%)**	**0.3167 (22%)**	**0.2083 (10.4%)**	**0.1388 (8.5%)**

Table 3. Statistics on (δ=1, L=4)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3619	0.2595	0.1886	0.1279
BaseLine2	0.4052 (12%)	0.2871 (10.6%)	0.1926 (2.1%)	0.133 (4%)
N=20	0.4262 (17.8%)	0.3143 (21.1%)	**0.2117 (12.2%)**	**0.14 (9.5%)**
N=25	**0.4357 (20.4%)**	0.319 (22.9%)	0.2098 (11.2%)	0.1393 (8.9%)
N=30	0.4333 (19.7%)	**0.3214 (23.9%)**	0.2105 (11.6%)	0.1395 (9.1%)

Table 4. Statistics on (δ=10, L=2)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3619	0.2595	0.1886	0.1279
BaseLine2	0.4052 (12%)	0.2871 (10.6%)	0.1926 (2.1%)	0.133 (4%)
N=20	0.4262 (17.8%)	0.3119 (20.2%)	**0.2043 (8.3%)**	**0.1369 (7%)**
N=25	**0.4381(21.1%)**	**0.3214 (23.9%)**	0.2038 (8.1%)	0.1364 (6.6%)
N=30	0.4357(20.4%)	**0.3214 (23.9%)**	0.2038 (8.1%)	0.1362 (6.5%)

Table 5. Statistics on (δ=10, L=3)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3619	0.2595	0.1886	0.1279
BaseLine2	0.4052 (12%)	0.2871 (10.6%)	0.1926 (2.1%)	0.133 (4%)
N=20	0.4286 (18.4%)	0.3119 (20.2%)	0.2076 (10.1%)	0.1379 (7.8%)
N=25	**0.4476 (23.7%)**	**0.331(27.5%)**	0.2064 (9.4%)	0.1383 (8.1%)
N=30	0.4405 (21.7%)	0.319 (22.9%)	**0.2086 (10.6%)**	**0.14 (9.5%)**

Table 6. Statistics on (δ=10, L=4)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3619	0.2595	0.1886	0.1279
BaseLine2	0.4052 (12%)	0.2871 (10.6%)	0.1926 (2.1%)	0.133 (4%)
N=20	**0.4405 (21.7%)**	**0.3262 (25.7%)**	**0.2129 (12.9%)**	**0.141 (10.2%)**
N=25	**0.4405 (21.7%)**	0.3238 (24.8%)	0.2112 (12%)	0.1402 (9.6%)
N=30	0.4381(21.1%)	0.3238 (24.8%)	0.21 (11.3%)	0.139 (8.7%)

Table 7. Statistics on (δ=1, N=25)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3619	0.2595	0.1886	0.1279
BaseLine2	0.4052 (12%)	0.2871 (10.6%)	0.1926 (2.1%)	0.133 (4%)
L=2	0.4262 (17.8%)	0.3143 (21.1%)	0.2052 (8.8%)	0.1371 (7.2%)
L=3	0.4333 (19.7%)	0.3167 (22%)	0.2079 (10.2%)	0.1381 (8%)
L=4	**0.4357 (20.4%)**	**0.319 (22.9%)**	**0.2098 (11.2%)**	**0.1393 (8.9%)**

Table 8. Statistics on (δ=10, N=25)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3619	0.2595	0.1886	0.1279
BaseLine2	0.4052 (12%)	0.2871 (10.6%)	0.1926 (2.1%)	0.1330 (4%)
L=2	0.4381(21.1%)	0.3214 (23.9%)	0.2038 (8.1%)	0.1364 (6.6%)
L=3	**0.4476 (23.7%)**	**0.331(27.5%)**	0.2064 (9.4%)	0.1383 (8.1%)
L=4	0.4405 (21.7%)	0.3238 (24.8%)	**0.2112 (12%)**	**0.1402 (9.6%)**

From table 1-6, our proposed method gets better result than [BaseLine1] and [BaseLine2] in every parameter setting. If only considering PreAt100, it seems we may get better result by using terms in top 20 retrieved documents; but if only considering PreAt10, it seems we may get better result by using terms in top 25 or top 30 retrieved documents. If considering PreAt10 and PreAt100 together, we regard that we may get better and stable result by using terms in top 25 retrieved documents.

Tables 7 and 8 gives the comparison of precisions on different term extraction parameter settings using terms in top 25 retrieved documents.

From Table 7 and 8, our proposed method can improve PreAt10 by 17.8%-23.7% from 0.3619 to 0.4262-0.4476 in relax relevance judgment and improve PreAt10 by 21.1%-27.5% from 0.2595 to 0.3143-0.331 in rigid relevance judgment. In PreAt100 level, our method can improve 8.1%-12% and 6.6%-9.6% in relax relevance judgment and rigid relevance judgment. Even in worst case, our proposed method get better result than [BaseLine2] with 18.8%, 21.1%, 8.1% and 6.6% improvement at PreAt10(relax), PreAt10(rigid), PreAt100(relax) and PreAt100(rigid) level compared with 12%, 10.6%, 2.1% and 4% improvement in [BaseLine2].

From table 7 and table 8, we may conclude that using key terms that occur at least 3 times or 4 times in documents may get better results.

The above experiments on NTCIR3 dataset show our method can achieve significant improvements on PreAt10 and PreAt100 results.

Fig. 2-5 gives the comparison of the precisions of 42 query topics before and after document re-ordering at parameter setting (δ=1, N=25, L=4).

From Fig. 2–5, for 42 topics in NTCIR3, there are only 2 query topics (topic 9 and 43) whose precisions are slightly decreased after document re-ordering, the other 40 topics are all improved after document re-ordering.

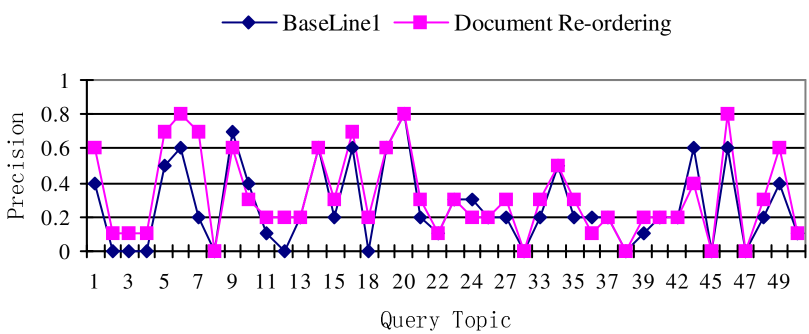

Fig. 2. PreAt10 at rigid relevance judgment (δ=1, N=25, L=4)

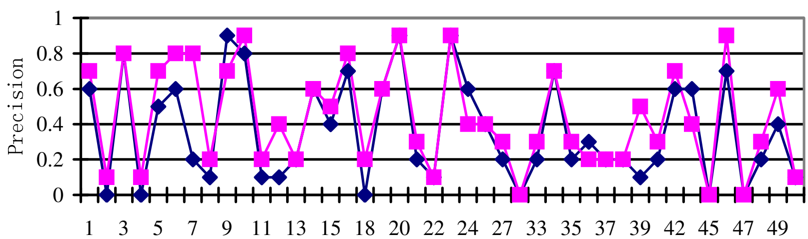

Fig. 3. PreAt10 at relax relevance judgment (δ=1, N=25, L=4)

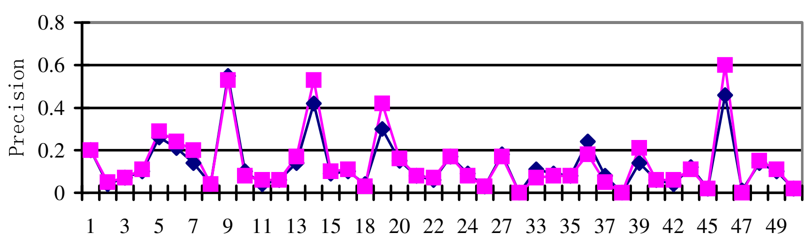

Fig. 4. PreAt100 at rigid relevance judgment (δ=1, N=25, L=4)

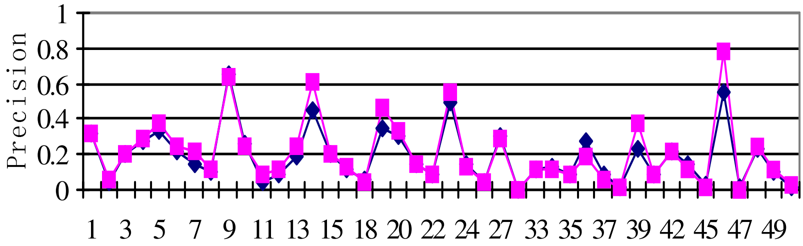

Fig. 5. PreAt100 at relax relevance judgment (δ=1, N=25, L=4)

5 Conclusion and Future Work

Document re-ordering is very important for improving the precision. In this paper, we introduce our approach to re-order retrieved documents in Chinese IR. For each query topic, firstly, we automatically extract key terms from top N (N<=30) retrieved documents; secondly, we collect these terms that occur at query topic and their document frequencies in top N retrieved documents; thirdly, we re-order top K (N<K) retrieved documents by collected key terms each document contains. Each collected key term is given a weight by its length and its document frequency in top N retrieved documents. Weight is given to reflect an observation: long key term may contain more precise information and it'll be given more weight; key term occurred in more top retrieved documents tends to play more important role in distinguishing query topic and it'll be given more weight.

With bi-gram as indexing units and vector space model as retrieval model, our experiments in the Chinese tasks of CLIR in NTCIR3 show that our method using key terms can improve the average performance of Chinese IR on 42 query topics by 17.8%-27.5% at top 10 documents and 6.6%-12% at top 100 documents at all kinds of parameter settings and relax relevance judgment or rigid relevance judgment.

In future, we'll apply our method on Chinese IR which uses Probabilistic Retrieval Model as retrieval model or uses word as indexing unit. We also want to do more experiments on Chinese IR which uses long description as query topic.

References

1. Bear J., Israel, D., Petit J., Martin D.: Using Information Extraction to Improve Document Retrieval. Proceedings of the Sixth Text Retrieval Conference, 1997.
2. Fuhr. N.: Probabilistic Models in Information Retrieval. The Computer Journal. 35(3):243-254, 1992.
3. Ji D.H., Yang L.P., Nie Y.: Chinese Language IR based on Term Extraction. In The Third NTCIR Workshop, 2002.
4. Kamps, J.: Improving Retrieval Effectiveness by Reranking Documents Based on Controlled Vocabulary. The 21th European Conference on Information Retrieval, 2004.
5. Kwok K.L.: Comparing Representation in Chinese Information Retrieval. In Proceedings of the ACM SIGIR-97, pp. 34-41.1997
6. Lee K., Park Y., Choi, K.S.: Document Re-ranking model using clusters. Information Processing & Management V, 2001.

7. Li. P. Research on Improvement of Single Chinese Character Indexing Method, Journal of the China Society for Scientific and Technical Information, Vol. 18 No. 5. 1999.
8. M. Mitra., A. Singhal. and C. Buckley. Improving Automatic Query Expansion. In Proc. ACM SIGIR'98, Aug. 1998.
9. Nie J.Y., Gao J., Zhang J., Zhou M.: On the Use of Words and N-grams for Chinese Information Retrieval. In Proceedings of the Fifth International Workshop on Information Retrieval with Asian Languages, IRAL-2000, pp. 141-148, 2000
10. Qu, Y.L., Xu, G.W.,Wang J.: Rerank Method Based on Individual Thesaurus. Proceedings of NTCIR2 Workshop. 2000.
11. Robertson S.E., Walker, S.: Microsoft Cambridge at TREC-9: Filtering track: In NIST Special Pub. 500-264: The Eight Text Retrieval Conference (TREC-8), pages 151-161, Gaithersburg, MD, 2001.
12. SALTON, G., MCGILL, M.: Introduction to Modern Information Retrieval, McGraw-Hill.1983
13. Schutze, H.: The hypertext concordance: a better back-of-the-book index. Proceedings of First Workshop on Computational Terminology. 101-104, 1998.
14. Yang L.P., Ji D.H., Tang L.: Document Re-ranking Based on Automatically Acquired Key Terms in Chinese Information Retrieval. Proceedings of 20th International Conference on Computational Linguistics (COLING'2004). 2004.
15. Yang L.P., Ji D.H., Tang L.: Chinese Information Retrieval based on Terms and Ontology. Proceedings of NTCIR4 Workshop. 2004.

Appendix: 22 Query Topics in NTCIR3 (First Part of 42 Topics)

001: 查询故宫博物院所举办之千禧汉代文物大展相关内容
(Find information of the exhibition "Art and Culture of the Han Dynasty" in the National Palace Museum)

002: 查询台湾加入WTO後各产业可能面对的问题
(Find possible problems that industries will meet after Taiwan's joining WTO.)

003: 查询大学学术追求卓越计划的相关内容
(Find the content of Program for Promoting Academic Excellence of Universities.)

004: 查询何谓电子商务及电子商务之内容
(Find what E-Commerce is and its contents)

005: 查询朱熔基担任中国总理後所提出的经济改革计划。
(Find Zhu Rong ji's economic reform after his serving as the premier)

006: 查询有关一九九八年诺贝尔物理学奖的相关报导
(Retrieve reports relating to 1998 Nobel Prizes in Physics)

007: 查询有关华航於桃园中正机场失事的相关报导
(Retrieve reports about China Airlines' crash while trying to land at Taoyan international airport.)

008: 查询一九九八年电影「铁达尼号」获得奥斯卡奖之相关报导
(Retrieve reports of Oscar winners, Titanic, in 1998)

009: 查询有中新一号卫星相关报导及评论
(Find reports and comments related to satellite ST1)

010: 查询何谓反圣婴现象及其与圣婴现象的比较与影响
(Find what the anti-El Nino is and the comparison with El Nino)

011: 查询阿里山蒸气火车的历史及其与观光业、森林业的关系
(Find the history of steam locomotive in Mount Ali and its relationship with forestry and sightseeing)
012: 查询曼谷亚洲运动会的相关报导
(Find news of Asian Games in Bangkok)
013: 查询精省的法规内容有哪些，以及台湾省废省後前省长宋楚瑜的态度
(Find the content of Province-refining enactment and Mr. James Soong's attitudes after the Province-refining)
014: 有关於感染电脑病毒引起的问题之文章
(Articles about problems caused by computer virus infection.)
015: 与使用被称为体细胞核移植的技术创造复制牛相关的文章
(Articles relating to the birth of cloned calves using the technique called somatic cell nuclear transfer.)
017: 有关於北野武导演的电影之文章
(Articles relating to Director Takeshi Kitano's films.)
018: 有关最後审判日或世界末日的宗教思想的关连事件。
(Incidents relating to religious thought about doomsday, or the end of the world.)
019: 有关於欧洲货币组织的经济影响之文章
(Articles relating to economic influence of European monetary union.)
020: 有关於日产与雷诺汽车公司资本结合的文章
(Articles relating to a capital tie-up of Nissan Motor Company of Japan and Renault of France.)
021: 有关於1999年西土耳其大地震造成的破坏与救援行动及灾害情形与难民的文章
(Articles relating to the damage, the rescue operations, and the damage situation and victims of a big earthquake in Western Turkey in 1999.)
022: 描述有关柬埔寨的前首相Pol Pat 战争罪行的文章
(Articles describing the war crimes of former Prime Minister Pol Pot of Cambodia.)
023: 有关金大中总统对亚洲的政策之文章
(Articles relating to President Kim Dae-Jung's policy toward Asia)