Improving Retrieval Effectiveness by Using Key Terms in Top Retrieved Documents

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

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

Abstract. In this paper, we propose a method to improve the precision of top retrieved documents in Chinese information retrieval where the query is a short description by re-ordering retrieved documents in the initial retrieval. To re-order the documents, we firstly find out terms in query and their importance scales by making use of the information derived from top *N* (*N*<=30) retrieved documents in the initial retrieval; secondly, we re-order retrieved *K* (*N*<<*K*) documents by what kinds of terms of query they contain. That is, we first automatically extract key terms from top *N* retrieved documents, then we collect key terms that occur in query and their document frequencies in the *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%-26.9% 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 (please see appendix for some examples), many retrieval models, indexing strategies, query expansion strategies and document re-ordering methods have been proposed. Different from most of the western languages, Chinese sentence is a contiguous Chinese character sequence without white space between Chinese words. 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 [9]. The comparison between the three kinds of indexing units (single Characters, bi-grams and short-words) is given in [7]. 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. [11] 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)

D.E. Losada and J.M. Fernández-Luna (Eds.): ECIR 2005, LNCS 3408, pp. 169 – 184, 2005.

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.

Regarding retrieval models, two models are most widely used in Chinese Information Retrieval, i.e., Vector Space Model [15] and Probabilistic Retrieval Model [3].

For query expansion, most strategies make use of global analysis or local analysis [2, 10, 13, 17]. For global analysis, the expansion terms are acquired by analyzing the whole document collection. For local analysis, the top *N* retrieved documents in initial retrieval will be used. 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 [8]. 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. [6] proposes 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. [12] 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 [18,19] 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. We first find out terms in query and their importance scales by making use of the information derived from top N (N<=30) retrieved documents in the initial retrieval; secondly, we re-order retrieved K (N<<K) documents by what kinds of terms of query they contain.

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-rank 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 Key Term Extraction

Key term extraction concerns the problem of what is a key term. Intuitively, key terms in a documents are some conceptual terms that are prominent in document and play main roles in discriminating itself from other documents. In other words, key terms in a document can represent the main content of the document. Generally, in the viewpoint of conventional linguistic studies, key terms maybe are some NPs, NP-phrases or some kind of VPs, adjectives that can represent some specific concepts in document content representation.

We use a seeding-and-expansion mechanism to extract key terms from documents [4, 5]. The procedure of key term extraction consists of two phases, seed positioning and term determination. Intuitively, a seed for a candidate term is an individual 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 Chinese Characters to reflect their significance in the text in some way. To do so, we make use of a very large corpus r (2GB data from NTCIR3 dataset, LDC's Mandarin Chinese News Text and news articles from www.sina.com.cn) as a *reference*. Suppose s is a document, w is an individual Chinese Character in the text, let $P_r(w)$ and $P_s(w)$ be the probability of w occurring in r and s respectively, we adopt *relative probability* or *salience* of w in s with respect to r [16], as the criteria for evaluation of seed words.

$$P_s(w) \,/\, P_r(w) \qquad (1)$$

We call w a *seed* if $P_s(w) \,/\, P_r(w) \geq \delta$ (δ>=1). That is, its probability in document must be equal or great than its average probability in large corpus.

Although it is difficult to give out the definition of key term, we try to give some assumptions about a key term. We have the following assumptions about a key term in a document.

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

The above assumptions tell us a key term is an independent maximal string which must occur at least 2 times in a document and contain a seed. For example, given document *d*, suppose Chinese Character 博 (bo3) is a seed in *d*, 故宫博物院 (National Palace Museum) occurs 3 times in *d*, 博物院 (Museum) occurs 5 times in *d*, if we set the parameter *L* in ii) as 2, then both string 故宫博物院 (National Palace Museum) and 博物院 (Museum) are terms in *d*; but if we set the parameter *L* in ii) as 3, then 故宫博物院 (National Palace Museum) is term in *d*, but 博物院 (Museum) is not a term in *d* because its independent occurrence is 2 (excluding 3 occurrences as substring in 故宫博物院 (National Palace Museum)).

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

```
Given threshold $\delta$ ($\delta$>=1) and L ($L$>1);
 Let $F_d(t)$ represents the frequency of term $t$ in document $d$;
$T = \{\}$;
Collect every Seed $w$ in $d$ into $E$ by $P_d(w) / P_r(w) \geq \delta$
For all $c \in E$ {
   Let $Q = \{t: t$ contains $c$ and $F_d(t) \geq L\}$;
   While $Q \neq NIL$ {
      $max\text{-}t \leftarrow$ the longest string in $Q$;
      $T \leftarrow T + \{ max\text{-}t \}$;
      Remove $max\text{-}t$ from $Q$;
      For all other $t$ in $Q$ {
         If $t$ is a substring of $max\text{-}t$ {
            $F_d(t) \leftarrow F_d(t) - F_d(max\text{-}t)$;
            If $F_d(t) < L$ {
               Remove $t$ from $Q$; }
         }
      }
   }
}
Return $T$ as key terms in document $d$;
```

Fig. 1. Term Extraction from Document *d*

3 Document Re-ordering

For Chinese information retrieval where query is a short description by natural language, we argue that different terms in query may play different roles. While many terms in query are descriptive or functional, some terms in query are important and may represent the main point of query. The difficulty is that we cannot find out if a term is important or not directly from the query itself. One alternative is we may make use of some information derived from top retrieved documents in initial retrieval. Firstly, we assume, like most pseudo feedback methods, that the top N (N<=30) retrieved documents are relevant with query q; secondly, we extract separately key terms from these documents by using our term extraction algorithm introduced at section 2; thirdly, for each key term t, we collect its document frequency DF_t in top N retrieved documents, that is, we collect how many documents of top N retrieved

Given q is a query, N is the number of top pseudo relevant documents, and K is the number of returned documents to be re-ordered in initial retrieval.

Step 1: Find out terms in q and their weight by information in top N documents;

Step 1.1 Extract key terms from each document d in top N retrieved documents by using term extraction algorithm in Fig. 1;

Step 1.2 For each key term t, collect its document frequency, that is, how many documents of top N retrieved document it occurs in;

Step 1.3 Collect key terms that occur at query q and their document frequencies;

Let $T=\{T_1, T_2, ..., T_n\}$ is the set of collected key terms;

Let $D=\{DF_1, DF_2, ..., DF_n\}$ is the set of document frequencies of terms in T;

Step 1.4 Assign each term T_i in T a weight W_i by:

$$W(T_i) = \mathrm{sqrt}(|T_i|)\ \mathrm{X}\ \mathrm{sqrt}(DF_i) \qquad (2)$$

where $|T_i|$ is the length of term T_i, i.e., the number of Chinese characters in term T_i. The weight reflects the scale of importance of T_i in query q.

Step 2: Re-order top K retrieved documents by terms in q and their weight;

Step 2.1 For each document d_i in top K retrieved documents, calculate its re-ordered similarity value S_i by its initial similarity value R_i in the initial retrieval;

$$w = \sum_{t_j \in q, d_i} W(t_j) \qquad (3)$$

$$S_i = \begin{cases} w \times R_i & (w>0) \\ R_i & (w=0) \end{cases} \qquad (4)$$

Step 2.2: Re-order top K retrieved documents by their new re-ordered similarity values $\mathrm{S}=\{\mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_\mathrm{i}, \ldots, \mathrm{S}_\mathrm{K}\}$.

Fig. 2. The Procedure of Document Re-ordering

documents term *t* occurs in; fourthly, we pick up these key terms which occur in query topic as terms of query *q* and regard their document frequencies (in practise, we use square root of document frequencies to smooth them) as their weight in query; these weight reflect their importance in query, that is, more important term has more document frequency, descriptive or functional term has less document frequency; furthermore, for each term of query *q*, we also use their length (number of Chinese characters in term) as weight to reflect an observation that long term may contain more information.

After having valued each term in query *q*, we can use the information to re-order retrieved documents. Firstly, for each document *d* in returned documents, we find out what query terms occur in it; secondly, we sum the weight of these query terms in *q* and use the accumulated value to re-calculate the similarity between document *d* and query *q*; finally, we use the new similarity value (it is not a real similarity value but a value which is used to rank documents) to order retrieved documents.

Figure 2 gives out the pseudo code of the procedure of document re-ordering for query *q* and top *K* retrieved documents.

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 top 10 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 separately use vector space model and probabilistic retrieval model as our retrieval models. The initial retrieval result is used as 1^{st} baseline to evaluate our proposed method.

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.

4.1 Vector Space Model

In our first group experiments, we use vector space model to represent documents and queries. We also use Yang L.P et.al. [18]'s result on NTCIR3 CLIR dataset as 2^{nd} baseline. 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* weighting scheme:

$$w(t, d)=\log(\mathrm{T}(t, d)+1) * \log(N/\mathrm{D}(t)+1) \quad (5)$$

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

The weight of bi-gram *t* in query *q*, $w(t, q)$, is given by the following weight scheme:

$$w(t, q) = \mathrm{T}(t, q) \quad (6)$$

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 comparison of precisions at different parameters setting is given at table 1-6. 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%)**

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

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%)	0.133 (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 7 and table 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.

The comparison of the precisions of 42 query topics before and after document re-ordering at parameter setting (δ=1, N=25, L=4) is given at Fig. 3-4. From Fig. 3-4, 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.

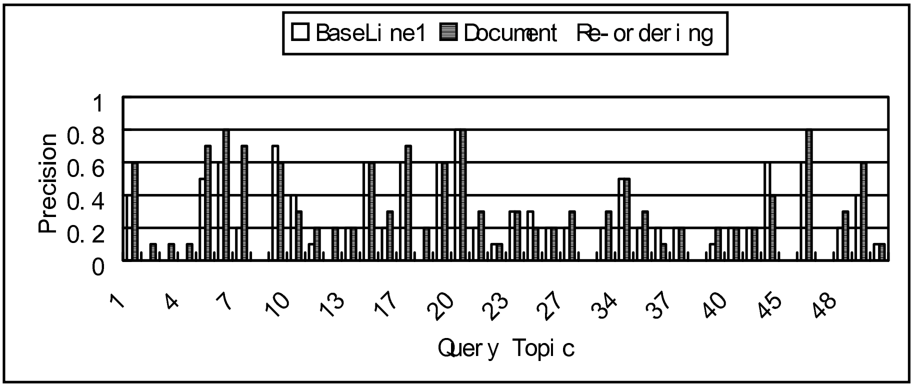

Fig. 3. PreAt10 at rigid relevance judgment (δ=1, N=25, L=4)

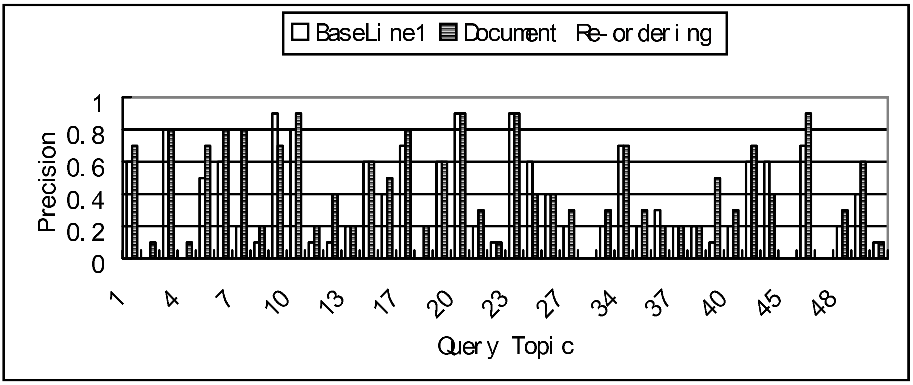

Fig. 4. PreAt10 at relax relevance judgment (δ=1, N=25, L=4)

4.2 Probabilistic Retrieval Model

In our second group experiments, we use the famous OKAPI BM11 [14] model as retrieval model. The other parameter settings are the same as that in our first group experiments except no [BaseLine2] is used.

OKAPI BM11 is a kind of probabilistic retrieval model based on 2-Possion model. We use the following BM11 weighting function:

$$BM11(q, d_i) = \sum_j q_j \log\left(\frac{N - n_j + 0.5}{n_j + 0.5}\right)\left(\frac{t_{i,j}}{t_{i,j} + \frac{len_i}{len}}\right)$$

where q is the query, d_i is the i-th document, q_j is the j-th query term weight, *N* is the number of documents in the document collection, n_j is the number of documents which contain the j-th term, $t_{i,j}$ is the number of occurrence of j-th term in i-th document, *len*$_i$ is the Euclidean document length of the i-th document and *len* is the average Euclidean document length.

The comparison of precisions at different parameters setting is given at table 9-14.

Table 9. Statistics on (δ=1, L=2)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3333	0.2452	0.1529	0.1026
N=20	0.3595 (7.9%)	0.2571 (4.9%)	0.1681 (9.9%)	0.1117 (8.9%)
N=25	**0.3667 (10%)**	**0.2595 (5.8%)**	**0.1688 (10.4%)**	**0.1129 (10%)**
N=30	0.3595 (7.7%)	0.2576 (5.1%)	0.1671 (9.3%)	0.1114 (8.6%)

Table 10. Statistics on (δ=1, L=3)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3333	0.2452	0.1529	0.1026
N=20	**0.3762 (12.9%)**	**0.269 (9.7%)**	0.1721 (12.6%)	0.116 (13.1%)
N=25	0.369 (10.7%)	0.2667 (8.8%)	**0.1729 (13.1%)**	**0.1162 (13.3%)**
N=30	0.3571 (7.1%)	0.2571 (4.9%)	0.1719 (12.4%)	0.1145 (11.6%)

Table 11. Statistics on (δ=1, L=4)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3333	0.2452	0.1529	0.1026
N=20	**0.3738 (12.2%)**	0.2713 (10.6%)	**0.1798 (17.6%)**	**0.1212 (18.1%)**
N=25	**0.3738 (12.2%)**	**0.2738 (11.7%)**	0.1771 (15.8%)	0.1198 (16.8%)
N=30	0.369 (10.7%)	0.2667 (8.8%)	0.1733 (13.3%)	0.1164 (13.5%)

Table 12. Statistics on (δ=10, L=2)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3333	0.2452	0.1529	0.1026
N=20	0.4 (20%)	0.2881 (17.5%)	0.1836 (20%)	0.1248 (21.6%)
N=25	**0.4048 (21.5%)**	**0.2952 (20.4%)**	**0.185 (21%)**	**0.1264 (23.2%)**
N=30	0.3929 (17.9%)	0.2857 (16.5%)	0.1807 (18.2%)	0.1229 (19.8%)

Table 13. Statistics on (δ=10, L=3)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3333	0.2452	0.1529	0.1026
N=20	0.3952 (18.6%)	**0.2929 (19.5%)**	0.1883 (23.2%)	**0.1293 (26%)**
N=25	0.3952 (18.6%)	0.2881 (17.5%)	**0.189 (23.6%)**	0.1288 (25.5%)
N=30	**0.3976 (19.3%)**	0.2905 (18.5%)	0.1881 (23%)	0.1283 (25%)

Table 14. Statistics on (δ=10, L=4)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3333	0.2452	0.1529	0.1026
N=20	0.3786 (13.6%)	0.2762 (12.6%)	**0.1912 (25%)**	**0.1314 (28.1%)**
N=25	**0.3952 (18.6%)**	**0.2857 (16.5%)**	**0.1905 (25%)**	0.1302 (26.9%)
N=30	**0.3952 (18.6%)**	**0.2857 (16.5%)**	0.1893 (23.8%)	0.1295 (26.2%)

From table 9-14, our proposed method gets better result than [BaseLine1] in every parameter setting. If only considering PreAt100, it seems we may get better result by using terms in top 20 or 25 retrieved documents; but if only considering PreAt10, it seems we may get better result by using terms in top 25 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. Table 15-16 gives the comparison of precisions on different term extraction parameter settings using terms in top 25 retrieved documents.

Table 15. Statistics on (δ=1, N=25)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3333	0.2452	0.1529	0.1026
L=2	0.3667 (10%)	0.2595 (5.8%)	0.1688 (10.4%)	0.1129 (10%)
L=3	0.369 (10.7%)	0.2667 (8.8%)	0.1729 (13.1%)	0.1162 (13.3%)
L=4	**0.3738 (12.2%)**	**0.2738 (11.7%)**	**0.1771 (15.8%)**	**0.1198 (16.8%)**

Table 16. Statistics on (δ=10, N=25)

	PreAt10(relax)	PreAt10(rigid)	PreAt100(relax)	PreAt100(rigid)
BaseLine1	0.3333	0.2452	0.1529	0.1026
L=2	**0.4048 (21.5%)**	**0.2952 (20.4%)**	0.185 (21%)	0.1264 (23.2%)
L=3	0.3952 (18.6%)	0.2881 (17.5%)	0.189 (23.6%)	0.1288 (25.5%)
L=4	0.3952 (18.6%)	0.2857 (16.5%)	**0.1905 (25%)**	**0.1302 (26.9%)**

From table 15 and table 16, our proposed method can improve precision at every parameter setting. We also see that the respectively results in table 16 is better than these in table 15. Since the only difference between table 15 (δ=1, where almost all terms are considered equally) and table 16 (δ=10, where more prominent terms are considered) is the setting of parameter δ, we may come to a conclusion: important key terms (more prominent terms) in topic play key roles and it can be used to improve precision.

From table 16, our proposed method can improve PreAt10 by 18.6%-21.5% from 0.3333 to 0.3952-0.4048 in relax relevance judgment and improve PreAt10 by 16.5%-20.4% from 0.2452 to 0.2881-0.2952 in rigid relevance judgment. In PreAt100 level, our method can improve 21%-25% and 23.2%-26.9% in relax relevance judgment and rigid relevance judgment.

The comparison of the precisions of 42 query topics before and after document re-ordering at parameter setting (δ=1, N=25, L=4) is given at Fig 5-6. From Fig. 5-6, for 42 topics in NTCIR3, there are only 3 query topics (topic 21, 24 and 38) whose precisions are slightly decreased after document re-ordering, the other 39 topics are all improved after document re-ordering.

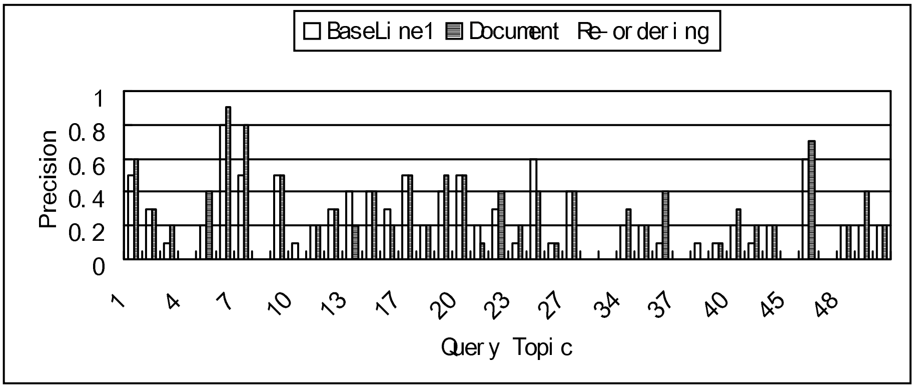

Fig. 5. PreAt10 at rigid relevance judgment (δ=1, N=25, L=4)

From our two group experiments based on bi-gram as index unit and vector space model and probabilistic retrieval model as retrieval models, our proposed method can improve precision at every parameter setting. From table 6 and table 16(δ =10, N=25), our method can improve 18.6%-23.7% and 16.5%-27.5% in relax relevance judgment and rigid relevance judgment; in PreAt100 level, our method can improve 8.1%-25% and 6.6.2%-26.9% in relax relevance judgment and rigid relevance judgment.

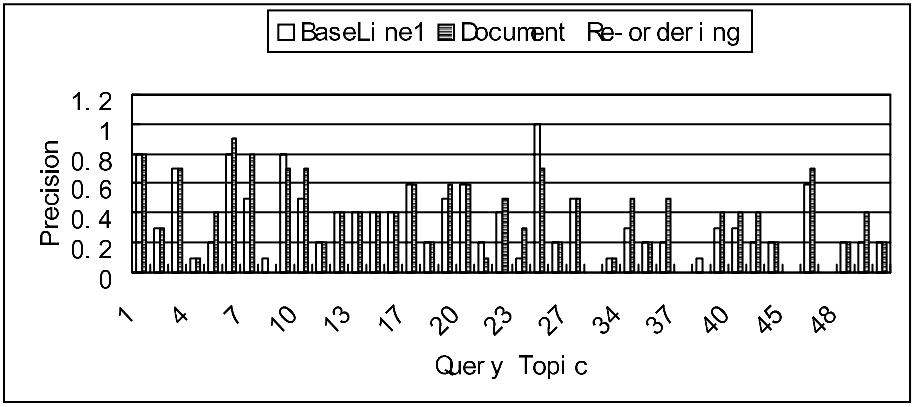

Fig. 6. PreAt10 at relax relevance judgment (δ=1, N=25, L=4)

5 Conclusion and Future Work

Document re-ordering is very important for improving the precision of retrieved documents. In this paper, we introduce our approach to re-order retrieved documents in Chinese IR. For each query topic *q*, firstly, we try to find out its terms and give each term a weight by using information of key terms automatically extracted from top *N* (*N*<=30) retrieved documents and their document frequencies; secondly, we re-calculate the similarity between query *q* and document *d* in top retrieved *K* (*N*<<*K*) documents by what kinds of query terms it contains; finally, we re-order retrieved *K* documents by their re-calculated similarity value.

Our experiments on 42 query topics in NTCIR3 CLIR task, with bi-gram as indexing units, shows our proposed approach produced significant improvement in retrieval precision by 17.8%-27.5% average improvement at top 10 documents level and 6.6%-26.9% average improvement at top 100 documents level at all kinds of parameter settings and relax relevance judgment or rigid relevance judgment.

The experimental results show some idea under our approach may be useful for Chinese information retrieval, that is, we may use key terms in top *N* retrieved documents to determine terms of query *q*, and we also can use the number of documents in top *N* retrieved documents which contain these query terms to reflect the importance of query terms in query *q*; moreover, long query term may contain more precise information and can be used to improve precision.

Our experiments are all based on Chinese information retrieval. In the future, we'll do some experiments on other languages. We also want to try other term extraction approaches to analyse what kind of role each part plays in our approach.

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] Carpineto, C., Romano, G., Giannini, V.: Improving Retrieval Feedback with Multiple Term-Ranking Function Combination. In ACM Transactions on Information Systems, Vol. 20, n. 3, pp. 259-290. 2002.

[3] Fuhr. N.: Probabilistic Models in Information Retrieval. The Computer Journal. 35(3):243-254, 1992.

[4] Ji D.H., Yang L.P., Nie Y.: Chinese Language IR Based on Term Extraction. In The Third NTCIR Workshop, 2002.

[5] Ji D.H., Yang L.P., Nie Y., Tang L.: Online Discovery of Relevant Terms from Internet. IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLPKE2003), Beijing, China, Oct, 2003.

[6] Kamps, J.: Improving Retrieval Effectiveness by Reranking Documents Based on Controlled Vocabulary. The 21th European Conference on Information Retrieval, 2004.

[7] Kwok K.L.: Comparing Representation in Chinese Information Retrieval. In Proceedings of the ACM SIGIR-97, pp. 34-41.1997

[8] Lee K., Park Y., Choi, K.S.: Document Re-ranking Model Using Clusters. Information Processing and Management. V. 37 n.1, p1-14, 2001.

[9] 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.

[10] M. Mitra., A. Singhal. and C. Buckley. Improving Automatic Query Expansion. In Proc. ACM SIGIR’98, Aug. 1998.

[11] 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

[12] Qu, Y.L., Xu, G.W.,Wang J.: Rerank Method Based on Individual Thesaurus. Proceedings of NTCIR2 Workshop. 2000.

[13] 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.

[14] Robertson S.E., Walker, S., Jones S.: Okapi at TREC-2. In The Second Text Retrieval Conference (TREC-2). 1994.

[15] SALTON, G., MCGILL, M.: Introduction to Modern Information Retrieval, McGraw-Hill.1983

[16] Schutze, H.: The Hypertext Concordance: A Better Back-of-the-Book Index. Proceedings of First Workshop on Computational Terminology. 101-104, 1998.

[17] Vechtomova O., Robertson S.E., Jones S. Query Expansion With Long-Span Collocates. Information Retrieval, 6(2), 2003, pp. 251-273.

[18] 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.

[19] Yang L.P., Ji D.H., Tang L.: Chinese Information Retrieval Based on Terms and Ontology. Proceedings of NTCIR4 Workshop. 2004.

Appendix: 10 Query Topics in NTCIR3 (Part of 42 Query 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)