Pseudo-Relevance Feedback and Title Re-Ranking for Chinese Information Retrieval

Robert W.P. Luk
Department of Computing
The Hong Kong Polytechnic University
Email: csrluk@comp.polyu.edu.hk

K.F. Wong
Dept. Systems Engineering & Engineering Management
Chinese University of Hong Kong
Email: kfwong@se.cuhk.edu.hk

Abstract

In our formal runs, we have experimented with the retrieval based on character-based indexing and hybrid term indexing because these are more distinct types of indexing for better pooling. We confirmed that character-based indexing did not produce relatively good retrieval effectiveness. We have also experimented with three new pseudo-relevance feedback (PRF) methods. These new methods were able to achieve slightly better performance compared with our original PRF method for short queries at the same time using only 60 expanded terms instead of 140 expanded terms, thereby reducing the retrieval time by about 30 percentage points. We have experimented with a novel re-ranking strategy, called title re-ranking. This strategy rewards documents which title terms match with the terms in the title query. Title re-ranking is able to improve the performance but it hurts the performance of long queries when PRF is used with it. For title queries, our best MAP achieved was 24.9% using both title re-ranking and PRF based on rigid relevance judgment. *For long queries using our original PRF, our best MAP was 26.0%.*

Keywords: *Chinese information retrieval, indexing, 2-Poisson model, relevance feedback, re-ranking and evaluation.*

1 Introduction

In pooling, it is assumed that different search strategies are used so that different relevant documents of the same query can be identified by different search engines, thereby enabling the identified number of pooled relevant documents to be close to the true number of relevant documents of the query. However, as open evaluation workshops mature, many search engines in the workshops use similar retrieval models and indexing strategies, which resulted in producing highly correlated retrieval lists. This meant that many relevant documents in different retrieval lists are the same and many relevant documents are missing from the pooled relevant set. Trying to offset this effect, some of our formal runs are based on character-based indexing and the other runs are based on the hybrid-term indexing, which are relatively uncommon indexing strategies.

While character-based indexing was one of earlier common indexing strategies, it has become less popular because of its lower retrieval effectiveness. There is almost no participant who used character-based indexing in the NTCIR-3 workshop. Here, we also want to examine whether character-based indexing is achieving consistently lower retrieval effectiveness in this test collection, where the document collection is the same as that of the NTCIR-3 workshop.

Pseudo-relevance feedback (PRF) is one of the most well known [1] and widely applied technique in the open IR evaluation workshops in order to improve the retrieval effectiveness. Here, we examine four PRF methods to see which one produces better results. On the one hand, we wish to obtain similar results with less expanded terms so that the retrieval efficiency can be increased. On the other hand, we are hoping that better retrieval effectiveness performance can be obtained.

The advantage of PRF as a re-ranking strategy is that it is able to improve performance quite robustly. However, one disadvantage of PRF as a re-ranking strategy is that it incurs significant overhead in terms of processing time during retrieval. The major portion of the time is spent on reformulating the queries and the second retrieval. If other re-ranking strategies can capitalize the advantage of PRF without incurring significant processing time overhead during retrieval, then re-ranking strategies can be applied in many practical situations. To avoid incurring significant processing time overhead during retrieval, these re-ranking strategies should not need to analyze the individual documents and re-ranking is done only for the top 1000 documents, instead of processing the query again. However, we should not be too concerned if there is a significant amount of time spent on indexing so long as this does not affect retrieval efficiency. Do such re-ranking strategies exist? Here, we examined a simple re-ranking strategy based on matching title query terms and the title terms in the documents.

The rest of the paper is organized as follows. Section 2 discussed our formal runs. Section 3 reviewed various retrieval models used. Section 4 has a set of evaluations, including a comparison between indexing strategies, retrieval models and using PRF. Finally, Section 5 summarizes our findings.

2 Standard Runs

Here, we report our formal runs and the runs using our previous PRF method.

2.1 Formal Runs

We used the 2-Poisson model with the okapi BM11' weighting function as follows:

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

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 collection, n_j is the document frequency of the j-th term, $t_{i,j}$ is the j-th term frequency in the i-th document and len_i is the the Euclidean document length for the i-th document and len is the average Eucliden document length.

From previous indexing work [2, 3, 4], it is clear that words are the preferred index terms if there is no out-of-vocabulary problem. To solve the out-of-vocabulary problem, words can be extracted automatically [5, 6] but there are concerns about the recall performance of automatic extractions or the concerns about the scope of word formation rules [7]. Instead, we propose to use bigrams to solve the out-of-vocabulary problem. Bigrams have the advantage that it is a completely data-driven technique, without any rule maintenance problem. Bigrams can be extracted on the fly for each document. There are no requirements to define a somewhat arbitrary threshold (or support) and there is no need to extract and test any templates for word extraction.

Algorithm A combined both word-based indexing and bigram-based indexing. Note that Algorithm A does not index single-character words unless the single-character segmented substring is a single character and it is not a stop word. To secure better recall instead of precision, Algorithm A can be changed to index all single-character words that are not stop words. In this case, step 5 of Algorithm A is modified to:

if w is not a stop word **then**,

and steps 13, 14 and 15 can be deleted. In this evaluation, instead of using words, we used just two character words and our indexing strategy is called short hybrid term indexing.

Input: Document d and the word dictionary D
Output: Index terms { w } È { b }
Method: Hybrid Term Indexing
Step 1 Segment text into sequences s_k
Step 2 **For each** sequence s_k of Chinese characters in the document d **do**
Step 3 Segment s_k using the word dictionary D
Step 4 **For each** word w Î D matched in s_k **do**
Step 5 **if** $|w| > 1$ character **and** w is not a stop word **then**
Step 6 Index w
Step 7 **end**
Step 8 **For each** single-character segmented substring $s_{k,m}$ in s_k **do**
Step 9 **if** $|s_{k,m}| > 1$ character **then**
Step 10 **For each** bigram b in $s_{k,m}$ **do**
Step 11 Index b
Step 12 **end**
Step 13 **else**
Step 14 **if** $s_{k,m}$ is not a stop word **then**
Step 15 Index $s_{k,m}$ as a word w Î D
Step 16 **end**
Step 17 **end**

Algorithm A. Hybrid term indexing.

Table 1 shows the retrieval effectiveness of our system, which is rather poor even compared with the median and average for both title and description queries. However, our system is using the basic retrieval performance without any PRF and without filtering duplicate documents. As expected, the retrieval effectiveness of character-based indexing is consistently lower than the hybrid-term indexing for both title and description queries.

Query Type	Idx Unit	Rigid (%)		Relax (%)	
		MAP	P@10	MAP	P@10
T	C	15.8	18.5	19.1	26.3
	H	17.7	22.0	22.0	30.5
D	C	12.5	14.4	15.5	21.2
	H	15.3	19.8	20.2	29.0

Table 1: Retrieval effectiveness of submitted formal runs. Key: T for title queries, D for description queries, C for character-based indexing and H for hybrid-term indexing.

Figure 1 shows the relative precision of hybrid-term indexing for title queries. The relative precision is the precision of our system for a particular title query minus the average precision of that title query. Even though our system is performing lower than average, the retrieval effectiveness of some queries was performing surprisingly better than average.

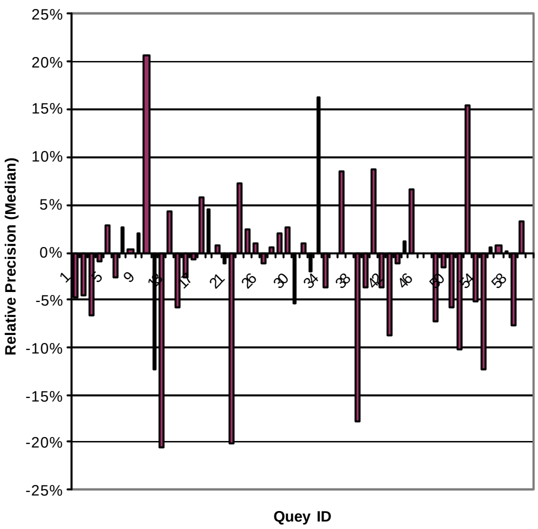

Figure 1: The precision of different T queries relative to the precision of the corresponding queries averaged across all formal runs.

To examine whether duplicate documents have an impact on retrieval effectiveness, we have re-run our system with title and description queries, filtering the known duplicate documents. Table 2 shows the effectiveness with and without filtering the duplicate documents. The difference in performance is not substantial, typically within just 1% difference. Note that filtering duplicate documents can gain some mean average precision (MAP) but this is at the expense of loosing the precision at the top-10 documents for both rigid and relax judgment. In the subsequent runs, we filtered the duplicate documents.

Query	Dup	Rigid (%)		Relax (%)	
Type	Filter	MAP	P@10	MAP	P@10
T	No	17.7	22.0	22.0	30.5
	Yes	17.9	21.0	22.0	29.7
D	No	15.3	19.8	20.2	29.0
	Yes	15.7	19.5	20.3	28.6

Table 2: Retrieval effectiveness runs for title and description queries with and without filtering duplicate documents. Key: T for title queries, D for description queries, C for character-based indexing and H for hybrid-term indexing.

2.2 PRF Runs

Here, we used our previous PRF settings, i.e. examine the top 6 documents in the first retrieval, identifying the top 140 terms that have the highest product of term frequency and document frequency in the top 6 documents. Since our performance is not as good as others, we have also examined whether bigram indexing will bring any better performance. Since character indexing was not performing very well in the formal runs, we have stopped examining whether character-based indexing further.

Table 3 shows the retrieval effectiveness using PRF with hybrid-term indexing and bigram indexing. After using PRF, the retrieval effectiveness appeared to be comparable to those in other formal runs (i.e., at least the results are better than the average). The best performance in the formal runs for title queries was very high (i.e., about 31% MAP for the rigid judgment) and we suspect that this is the performance of manual runs rather than automatic runs because the best MAP of the other runs is only 25%. The difference in performance between the best formal run for title queries and ours is about 6% MAP, which is statistically significant according to [8].

Bigram indexing appeared to perform better than hybrid term indexing for all the query types except description queries. Therefore, our subsequent runs only used bigram indexing. Due to some errors in our script, we have omitted the runs for narrative queries.

Query	Idx	Rigid (%)		Relax (%)	
Type	Unit	MAP	P@10	MAP	P@10
T	H	22.0	21.2	26.8	30.2
	B	23.3	24.2	28.0	32.5
D	H	20.2	22.5	26.5	32.9
	B	19.9	22.7	25.6	32.9
C	H	21.6	21.5	26.9	30.7
	B	23.1	26.3	28.4	35.6
TC	H	23.6	27.1	29.1	37.0
	B	24.2	27.0	30.0	37.1
TCDN	H	24.4	27.5	29.8	37.5
	B	26.0	28.1	31.5	38.6

Table 3: Retrieval effectiveness of runs with PRF. Key: T for title queries, D for description queries, C for concept queries, N for narrative queries, TC for combining title and concept queries, TCDN for long queries, B for bigram indexing and H for hybrid-term indexing.

3 Term Selection for PRF

In NTCIR-3, we used the product S_0 of the total term frequency and the number of top n documents containing the term as a term ranking function. Although good results were obtained, the performance improvement using PRF based on this term selection is not very substantial. We wonder whether there are better term selection methods that are unexplored. Since our original PRF is using 140 terms to obtain the better performance, our system using PRF is very slow. We want to find other PRF that can achieve at least similar performance but that need much less expanded terms to obtain good performance so that the retrieval speed can be enhanced, as the speed time is proportional to the number of unique query terms.

Here, three additional term selection methods were experimented, namely S_1, S_2 and S_3. The first term selection uses a term ranking function that is the product S_1 of S_0 and the inverse document frequency idf_j of the j-th term, i.e. $S_1(j) = S_0(j) \times idf_j$.

The second term selection method uses S_2 which is S_0 if the term has a document frequency less than k_1; Otherwise, S_2 is $2S_0$, i.e.:

$$S_2(j) = \begin{cases} S_0(j) & for \quad n_j \geq k_1 \\ 2S_0(j) & for \quad 1 < n_j < k_1 \\ 0 \quad for \quad n_j = 1 \end{cases}$$

where n_j is the j-th document frequency and k_1 is a parameter. Basically, S_2 rewards specific terms by scaling their S_0 scores by two. How specific is the term is set by the parameter k_1. Terms that occurred once are not included because those terms will not bring any additional relevant documents since they were found from the top n documents in the first retrieval.

The third method uses the ranking function S_3 that combines the ranking format of S_2 and the previous ranking function S_1 as follows:

$$S_3(j) = \begin{cases} S_1(j) & for \quad n_j \geq k_1 \\ 2S_1(j) & for \quad 1 < n_j < k_1 \\ 0 \quad for \quad n_j = 1 \end{cases}$$

The justification for the format is the same as S_2 except that the rankings about based on S_1, i.e. the product of the term frequency in the top n documents, the document frequency in the top n documents and the inverse document frequency (of the collection).

To tune the term selection methods, we compared the retrieval with PRF for title queries only because they can be processed faster and because usually the performance can be scaled up. Using one term selection method, in particular S_1, we determine the near best parameter settings, i.e. the top n documents, the mixture parameter α that provides the relative weighting to the selected terms and the query terms.

# Terms	Rigid (%) MAP	Rigid (%) P@10	Relax (%) MAP	Relax (%) P@10
30	22.5	25.1	26.8	33.9
50	23.3	26.4	27.3	35.4
60	**23.4**	26.4	27.3	35.3
70	23.3	25.3	**27.4**	33.9
90	23.1	**27.3**	26.9	**35.6**
140	22.0	26.1	25.3	33.9

Table 4: Retrieval effectiveness of bigram indexing with PRF using S_1 term ranking function and α = 0.3, for title queries.

Table 4 shows the retrieval effectiveness of bigram indexing with PRF where α = 0.3, using top 6 documents in the first retrieval and S_1 to select terms. The best MAP performance for rigid judgment was achieved using just 60 terms, instead of 140 terms. Since the MAP for relax judgment of the PRF using 60 terms is already the near best (c.f. 27.3% and 27.4% for 70 terms), our subsequent runs used 60 selected terms for the second retrieval of our PRF. Note that the retrieval effectiveness using just 60 terms is already performing similar to that of our earlier PRF using 140 terms.

α	Rigid (%) MAP	Rigid (%) P@10	Relax (%) MAP	Relax (%) P@10
0.1	**23.4**	**26.4**	**27.3**	**35.3**
0.3	**23.4**	**26.4**	**27.3**	**35.3**
0.5	23.4	26.3	27.3	35.1
0.7	23.4	26.3	27.3	35.3
0.9	23.4	26.3	27.3	35.3

Table 5: Retrieval effectiveness of bigram indexing with PRF using S_1 term ranking function and 60 selected terms, for title queries.

Table 5 shows the retrieval effectiveness of bigram indexing with PRF using 60 selected terms from the top 6 documents, ranked by S_1. The mixture parameter α is incremented by 0.2 from 0.1. The MAP for both rigid and relax judgment appeared to be independent of the mixture parameter. This suggested that our new term selection mechanism is fairly robust. Using the precision for the top 10 documents to differentiate performance, we have chosen to set a = 0.1 for subsequent runs.

Table 6 shows the retrieval effectiveness of bigram indexing with PRF. The best performance of the MAP and the precision of the top 10 documents for rigid and relax judgment are achieved for different top n documents. This suggested that the performance is more sensitive the top n documents than the mixture parameter. In the subsequent runs, we will use the top 7 documents.

Top n documents	Rigid (%)		Relax (%)	
	MAP	P@10	MAP	P@10
5	23.5	25.8	27.1	34.1
6	23.4	26.4	**27.3**	35.3
7	**23.6**	26.1	27.2	35.0
8	22.9	26.1	27.0	34.8
9	22.0	**27.1**	26.6	**35.9**
10	23.0	26.5	25.3	34.8

Table 6: Retrieval effectiveness of bigram indexing with PRF using S_1 term ranking function and 60 selected terms, for title queries.

Term Select	Zero Filter	Rigid (%)		Relax (%)	
		MAP	P@10	MAP	P@10
S_0	No	22.2	25.9	25.3	33.7
	Yes	22.2	25.9	25.3	33.7
S_1	No	23.7	26.1	27.3	34.6
	Yes	23.7	26.1	27.3	34.6
S_2	No	23.4	25.9	27.1	34.2
	Yes	23.6	26.6	27.3	**35.1**
S_3	No	**23.9**	**26.6**	**27.5**	34.9
	Yes	**23.9**	**26.6**	**27.5**	34.9

Table 7: Retrieval effectiveness of bigram indexing with PRF using a = 0.1, the top 7 documents and 60 selected terms, for title queries. The zero filter will discard expanded terms with a zero weight instead of using them.

Table 7 shows the retrieval effectiveness of bigram indexing with PRF using different term ranking functions but the same setting, i.e. α = 0.1, 60 selected terms and assuming that the top 7 documents are relevant. The zero filter is added here to discard any selected terms with a zero weight, which surprisingly only affected S_2 slightly. The best performance was obtained using S_3, except for the precision at the top 10 documents based on relax judgment. Based on this observation, subsequent runs are based on the S_3 term ranking function. Note that the performance using S_3 is better than that using our original PRF with 140 terms, except for the MAP based on the relax judgment.

Table 8 shows the retrieval effectiveness of bigram indexing using PRF with the following settings: α = 0.1, top 7 documents are assumed to be relevant, 60 terms are selected using S_3. While the performance for the title and description queries was good using S_3 compared with our original PRF, the long queries were substantially lower than that of our original PRF. This is rather disappointing since we are hoping to achieve better results with the new PRF.

Query Type	Rigid (%)		Relax (%)	
	MAP	P@10	MAP	P@10
T	**23.9**	**26.6**	**27.5**	34.9
D	20.3	23.7	25.6	33.1
C	22.5	23.7	26.9	32.9
N	21.1	26.4	26.2	**36.8**
TC	22.5	24.1	26.8	33.2
TCDN	21.2	25.8	26.0	35.1

Table 8: Retrieval effectiveness of bigram indexing with PRF using a = 0.1, S_3 term ranking function, the top 7 documents and 60 selected terms, for different query types.

The retrieval time of long queries for the PRF using 60 expanded terms is about 69.8% of the retrieval time of long queries for our original PRF using 140 expanded terms. This retrieval time saving is not the same as the ratio of the number of query terms (i.e. 60/140) because there is the overhead time to examine the top n documents.

3 Title Re-ranking

Title re-ranking tries to re-rank the documents based on the matching score between the title query and the title of the documents. The re-ranking function $sim'(.)$ is:

$$sim'(q, d_i) = (sim(q, d_i) - m) \times M(q_t, t(d_i)) + m$$

where $sim(q,d_i)$ is the original similarity score, m is the minimum original similarity score in the top n documents, $t(d_i)$ is the title of the i-th document, q_t is the corresponding title query of q, and $M(.)$ is the number matched specific terms between the title query and the title of the document. This re-ranking function guarantees the top n documents will remain in the top n ranks of the re-ranked list because $sim'(q,d_i)$ ³ m for all top n documents.

Table 9 shows the retrieval effectiveness of bigram indexing with title re-ranking for re-ranking different top n documents. The best performance was obtained by re-ranking the top 500 to 1000 documents, except for the precision at the top 10 documents based on rigid judgment. Clearly, title re-ranking improved the retrieval effectiveness for whatever top n documents that was re-ranked.

Top n	Rigid (%)		Relax (%)	
documents	MAP	P@10	MAP	P@10
0	18.9	23.2	22.5	29.8
5	20.1	23.2	23.3	29.8
10	20.4	**24.8**	23.6	31.4
20	20.0	24.4	23.8	32.4
30	20.0	24.1	23.9	33.1
100	20.3	23.9	24.2	33.1
200	20.4	23.7	24.4	33.1
500	**20.5**	23.7	**24.6**	**32.9**
1000	**20.5**	23.7	**24.6**	**32.9**

Table 9: Retrieval effectiveness of bigram indexing with title re-ranking for title queries.

Query Type	Top n	Rigid (%)		Relax (%)	
		MAP	P@10	MAP	P@10
T	0	18.9	23.2	22.5	29.8
	10	20.4	**24.8**	23.6	31.4
	1000	**20.5**	23.7	**24.6**	**32.9**
D	0	15.3	18.5	20.1	27.8
	10	16.2	19.5	21.6	29.5
	1000	17.8	20.3	23.3	30.1
C	0	19.6	22.7	24.1	30.7
	10	19.7	23.4	24.2	32.4
	1000	21.1	25.9	25.6	35.8
N	0	22.1	24.8	27.2	34.4
	10	20.7	25.3	25.0	35.1
	1000	22.7	28.0	27.8	39.2
TC	0	21.1	23.2	25.0	32.0
	10	20.5	23.7	24.9	33.1
	1000	21.6	25.9	26.4	36.4
TCDN	0	23.3	25.1	28.1	35.3
	10	21.0	25.3	25.4	34.4
	1000	22.9	28.0	27.8	38.3

Table 10: Retrieval effectiveness of bigram indexing with title re-ranking for different query types.

Table 10 shows the retrieval effectiveness of bigram indexing with title re-ranking for different types of queries using top 10 and top 1000 documents for re-ranking. For relatively short queries, title re-ranking can enhance the retrieval effectiveness. For relatively long queries, title re-ranking actually hurts the retrieval effectiveness instead of enhancing it.

4 PRF plus Title Re-ranking

We are curious whether combining both title re-ranking and PRF together can yield even better performance. Table 11 shows the retrieval effectiveness of bigram indexing using title re-ranking followed by PRF using different term ranking functions. The PRF setting is the following: $\alpha = 0.1$, assuming the top 7 documents are relevant, select 60 terms based on the corresponding term ranking function. Again, the best term ranking function S_3 working with title re-ranking. For the best performance based on rigid relevance judgment, re-ranking the top 10 documents is the most effective. For the best performance based on the relax judgment, re-ranking the top 1000 documents is the most effective.

Term Select	Top n docs	Rigid (%)		Relax (%)	
		MAP	P@10	MAP	P@10
S_0	10	22.8	27.3	26.0	36.3
	1000	22.0	25.1	26.6	33.9
S_1	10	23.9	26.4	27.3	34.9
	1000	23.2	27.5	28.0	**36.6**
S_2	10	23.5	27.0	27.2	35.9
	1000	21.7	25.9	26.6	34.4
S_3	10	**24.6**	**28.0**	27.8	35.8
	1000	23.8	27.8	**28.1**	**36.6**

Table 11: Retrieval effectiveness of bigram indexing with title re-ranking followed by PRF for title queries.

Query Type	Rigid (%)		Relax (%)	
	MAP	P@10	MAP	P@10
T	24.6	28.0	27.8	35.8
D	21.3	25.9	27.6	36.1
C	23.0	24.8	28.2	35.3
N	22.0	27.3	27.2	38.6
TC	23.7	27.1	28.4	37.1
TCDN	22.4	28.1	27.8	29.4

Table 12: Retrieval effectiveness of bigram indexing with title re-ranking (for the top 10 documents) followed by PRF using S_3 term ranking function, for different query types.

Table 12 shows the retrieval effectiveness of bigram indexing with title re-ranking for the top 10 documents, followed by PRF using S_3 term ranking function to select expansion terms. Unfortunately, PRF did not substantially improve the retrieval effectiveness of long queries to a level that is competitive to our original PRF, even without title re-ranking. Therefore, we experimented with another combination, i.e. perform PRF first, followed by title re-ranking.

Query Type	Rigid (%)		Relax (%)	
	MAP	P@10	MAP	P@10
T	**24.9**	27.1	**28.3**	35.8
D	21.1	24.2	26.1	33.9
C	22.8	24.2	27.1	33.6
N	22.1	27.3	27.1	37.1
TC	22.9	25.3	27.2	34.8
TCDN	22.3	26.6	27.2	36.4

Table 13: Retrieval effectiveness of bigram indexing with PRF using S_3 term ranking function, followed by title re-ranking (for the top 10 documents), for different query types.

Table 13 shows the retrieval effectiveness of bigram indexing with PRF using S3 term ranking function, followed by title re-ranking for different query types. The MAP performance of the title queries based on the rigid judgment is the highest (i.e., 24.9%), which is similar to the best performance of the formal runs of other query types (i.e., 25.1%). Unfortunately, the retrieval effectiveness of the long queries is still lower than that of the original PRF for long queries (c.f. 22.3% and 26.0%).

Query Type	Rigid (%)		Relax (%)	
	MAP	P@10	MAP	P@10
T	22.5	27.6	25.6	36.4
D	19.3	23.9	25.0	34.2
C	21.6	24.4	26.8	35.9
N	20.8	26.4	26.1	38.3
TC	22.0	25.8	26.7	36.6
TCDN	21.4	27.0	26.3	38.8

Table 14: Retrieval effectiveness of bigram indexing with title re-ranking (for the top 10 documents), followed by the original PRF using 140 expanded terms for different query types.

Table 14 shows the retrieval effectiveness of bigram indexing with title re-ranking for the top 10 documents, followed by our original PRF using 140 expansion terms. The surprising fact is that title re-ranking actually hurts the retrieval performance for almost all the query terms. A plausible reason is that our original PRF is not very robust. After the title re-ranking, the (near) optimal operating point has changed by the re-ranking. On the other hand, the new PRF using S_3 term ranking function is more robust and therefore it is able to leverage the benefit of title re-ranking. In view of this, we are experimenting with performing the original PRF first, followed by title re-ranking and the corresponding results are shown in Table 15. These performances were worse than the corresponding performances in Table 14.

Query Type	Rigid (%)		Relax (%)	
	MAP	P@10	MAP	P@10
T	22.8	25.8	25.7	34.6
D	19.1	22.0	23.6	31.9
C	20.3	22.4	24.4	30.7
N	19.4	23.4	24.3	33.4
TC	21.8	23.9	25.3	32.2
TCDN	20.4	23.9	24.9	33.7

Table 15: Retrieval effectiveness of bigram indexing with the original PRF using 140 expanded terms for different query types followed by title re-ranking (for the top 10 documents).

5 Summary

In this participation, we have shown that character indexing was not very effective, as expected. Hybrid term indexing was performing reasonably and bigram indexing was the more robust indexing strategies.

We have experimented with three newer term selection methods. The best and more robust new selection method relied on only 60 terms to enhance retrieval performance that is similar to our original PRF with 140 selected terms for short queries. For long queries, our original PRF still performed significantly well compared with the new PRF. For the TCDN queries, the MAP of our original PRF is comparable to the best performance of the formal runs of other query types.

We have experimented with the title re-ranking strategy. This strategy enlarges the similarity score of documents which have title terms that match with terms in the title query. Title re-ranking was found to improve the retrieval effectiveness. Title re-ranking can work together with PRF to yield our best MAP for title queries, which is as high as 24.9%, comparable to the MAP of long queries with PRF. We are still experimenting with our original PRF with title re-ranking but the results were disappointing.

Acknowledg ment

We would like to thank ROCLING for providing the word list and the Center for Intelligent Information Retrieval, University of Massachusetts (UMASS), for facilitating Robert Luk to develop in part the IR system, when he was on leave at UMASS. This work is supported by the CERG Project # PolyU 5183/03E.

References

[1] C. Buckley, G. Salton, J. Allan. Automatic retrieval with locality information using Smart. *Proceedings of TREC 1*, 1992, pp. 59-72.

[2] W. Lam, C-Y Wong and K.F. Wong. Performance Evaluation of Character-, Word- and N-Gram-Based Indexing for Chinese Text Retrieval, *Proceedings of IRAL 97*, Japan, 1997.

[3] J-Y. Nie and F. Ren. Chinese information retrieval: using characters or words, *Information Processing and Management*, **35**:443-462, 1997.

[4] M-K. Leong and H. Zhou. Preliminary qualitative analysis of segmented vs bigram indexing in Chinese, *Proceedings of the Sixth Text Retrieval Conference (TREC-6)*, Gaithersburg, Maryland, November, 1997, pp. 19-21.

[5] P. Fung and D. Wu. Statistical Augmentation of a Chinese Machine-readable dictionary, *Proceedings of Workshop on Very Large Corpora*, Kyoto, August, 1994, pp. 69-85.

[6] J. Guo. Critical tokenization and its properties, *Computational Linguistics*, **23:4:** 569-596, 1997.

[7] Z. Wu and G. Tseng. ACTS: An Automatic Chinese Text Segmentation System for Full Text Retrieval, *Journal of the American Society of Information Science*, **46(2)**: 83-96, 1995.

[8] E.M. Voorhee and C. Buckley. The effect of topic set size on retrieval experiment scores, *Proceedings of ACM SIGIR 2002 Research and Development in Information Retrieval*, 2002, pp. 316-323.