Different Retrieval Models and Hybrid Term Indexing

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

Abstract

Retrieval effectiveness depends on both the retrieval model and how terms are extracted and indexed. For Chinese, Japanese and Korea text, there are no spaces to delimit words. Indexing using hybrid terms (i.e. words and bigrams) was not very effective in NTCIR-II open evaluation. In this evaluation, we found that using the 2-Poisson model with hybrid term indexing can be effective in retrieval. With our pseudo-relevance feedback, the performance can be enhanced to a level that is comparable to the best performance in the formal runs. Therefore, we found that hybrid term indexing is promising when the 2-Poisson model is used.

Keywords: *Chinese information retrieval, indexing, 2-Poisson model, and evaluation.*

1 Introduction

Hybrid term indexing [1] was developed with the aim to enhance the retrieval effectiveness of Chinese information retrieval by using words and bigrams in a complimentary manner. Although earlier work [2] demonstrated the robustness and effectiveness of this indexing strategy, it was evaluated with a small test collection. When it was used for the NTCIR-II open Chinese information retrieval evaluation, the results were not good [3]. By comparison, other retrieval systems were performing better in terms of retrieval effectiveness in the NTCIR-II. It was not known whether the problems of hybrid term indexing were due to the indexing strategy per se, or due to the lack of sophistication in our retrieval system, which is called `IR`, or due to both.

A study was carried out to increase the level of sophistication of our retrieval system by comparing our implementation of the current state-of-the-art retrieval models implemented in our `IR` system, including our implementation of the vector-space [5], 2-Poisson [6], logistic-regression [7] and Pircs retrieval models [8] with those in the literature, using conventional indexing strategies (i.e. character, word, short word, bigram and Pircs indexing strategies), evaluated using TREC-5, TREC-6, TREC-9 and NTCIR-II (Chinese) test collections. An attempt to compare the retrieval effectiveness for different retrieval models and indexing strategies based on title and long queries were made separately, without pseudo-relevance feedback (PRF). The study [4] showed that our retrieval system achieved comparable results with those in the literature although direct comparisons are difficult to make, due to varying conditions and incomplete data. Also, the study [4] showed that robust and good retrieval effectiveness was achieved by our 2-Poisson retrieval model using our bigram indexing.

Initially, we were hoping to target our effort in cross-lingual information retrieval, as well as Chinese information retrieval tasks, for NTCIR-III participation. However, limited by resources and without the knowledge of whether hybrid term indexing strategy was good, we focused our effort to evaluate hybrid term indexing for NTCIR-III. It is also interesting to evaluate Chinese information retrieval tasks in its own right since the test collection of NTCIR-III has about triple the number of documents compared with that of the other open test collections (i.e., TREC5-9 and NTCIR-II).

The rest of the paper is organized as follows. Section 2 reviewed hybrid term indexing. 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 Hybrid Term Indexing

From previous work [9,10,11], 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 [12,13] but there are concerns about the recall performance of automatic extractions or the concerns about the scope of word formation rules [14]. 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.

However, bigrams have high storage cost. To reduce this effect, bigrams and words are not exhaustively indexed in the document. Instead, bigrams are extracted at parts of the documents where the out-of-vocabulary problem is likely to occur. One method is to extract bigrams only at regions where the Chinese phrases or sentences are segmented into individual character sequences. In this way, the number of extracted unique bigrams are reduced and therefore the storage cost is kept low. This idea of extracting information from single-character sequences was already applied in word extraction [15] but it was not applied in indexing for information retrieval.

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.

Algorithm A summarizes the discussion of using 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.

3 Retrieval Models

In NTCIR-II, we modified the vector space model to compute a (length) weight that depends on the length of the term and that computes the minimum weight of the set of terms extracted from a concept term. Here, the length weighting scheme is used but other weighting mechanism is not used because it is replaced by those specified by the specific retrieval model.

In the previous study [4], it was found that the best combination of our retrieval model and our indexing strategy was the 2-Poisson model and bigram indexing. However, this result was obtained using TREC-5, TREC-6, NTCIR-II and TREC-9 test collections, which have about a third of the number of documents in the collections compared with NTCIR-III. It would be interesting to know if this result still holds for NTCIR-III. Therefore, we examined our implementation of the vector space model, the 2-Poisson model, the logistic regression model and the Pircs retrieval model.

3.1 Our Vector Space Model

The vector space model (VSM) ranks the retrieved documents according to the similarity, *sim(.,.)*, between the query vector, *q*, and the retrieved document vector, *d*. Lee [16] showed that the cosine measure achieved better performance compared with the Jaccard coefficient. The cosine measure is defined as:

$$\cos(q,d) \equiv \frac{q \bullet d}{\| q \|_2 \times \| d \|_2}$$

where $q \cdot d$ is the inner product of *q* and *d* and $\|.\|_2$ is the Euclidean distance of the argument. In fact, the cosine measure can be considered as a normalized value of the inner product similarity.

Acronym	Equation
TW	$w_{i,j} = \frac{t_{i,j}}{n_j}$
Ltw1	$w_{i,j} = \log(t_{i,j}+1) \times \log(\frac{N}{n_j}+1)$

Table 1: TF-IDF Weights

Both query and document vectors are vectors over a set of term weights. A number of term weights, known as the *tf-idf* weights, were proposed in the past for document vectors. These term weights are defined based on the frequency of the term in a particular document and the number of documents that contain that term. Based on results in our previous experiments [4], the following two term

weights were found better and used in this evaluation, where Ltw1 is a common variant of the *tf-idf* weight.

One problem with using the term weights is that the document length is a distance function of the term weights, which are dependent on the document frequency of a term and its term frequency. This implies that the document lengths need to be re-computed whenever new documents are added, and this is not desirable for incremental indexing. In this evaluation, we try to find the type of term weights that can work best with the document length defined as the Euclidean distance of term frequencies, i.e., $\| d_i \|_2 = \sqrt{\sum_j t_{i,j}^2}$. In this way, the cosine measure can be used for dynamic collections and the type of term weights can be found. In summary, the inner product is computed based on the *tf-idf* weights and the denominator is simply $||d_i||_2$ since $||q||_2$ has no effect on the ranking, i.e.,

$$m\cos(q, d_i) \equiv \frac{c(q) \bullet c(d_i)}{\| d_i \|_2}$$

where $c(q)$ is a vector with weights computed by $c(q_j)$ ° q_j for all j, and $c(d_i)$ is a (document) vector with weights computed by $c(d_{i,j}) \equiv w_{i,j}$ (Table 1) for all j, and $c(q) \cdot c(d_i)$ ° S_j $c(q_j)$ ´ $c(d_{ij})$.

3.2 Our 2-Poisson Model

A well-known probabilistic model of information retrieval is the 2-Poisson model by Robertson and Sparck-Jones [17]. There are many variations of this model [18] and the most common one computes the well-known okapi weight (BM11 in [6]) of a document using the following simplified summation of the BM11 weights:

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

where N is the total number of documents, n_j is the number of documents that have term j, q_j is the weight for the query term j, $t_{i,j}$ is the occurrence frequency of term j in document i, len_i is the length of document i, and len is the average document length. The above BM11' weights were simplified using the best retrieval results in [19], where the three parameters of the okapi weights (k_1, k_2 and k_3) are set to $k_1 = 1.0$, $k_2 = 0.0$ and k_3 = ¥ for the general BM11 definition. Based on our previous study [4], the document lengths were measured based on the Euclidean distances. If length weighting is applied, there is no need to re-normalize the document length, since it is independent of the query weight.

3.3 Our Logistic Regression Model

The logistic-regression (LR) model [20] is another well-known probabilistic information retrieval model. It tries to estimate the parameters (or weights) of the traditional Bayesian probabilistic model [17] in a principled manner by using regression to model the dependencies among the data, instead of assuming binary independence or validating the estimation of parameters using the 2-Poisson model. By examining the training data for English documents, up to four interactions are modeled. This results in a (relatively) more complex expression for the document weights, $LR(q, d_i)$, that depends on four mathematical terms, $X_1 \ldots X_4$,:

$$LR(q,d_i) \equiv \frac{1}{1+e^{-\log O(r|d_i,q)}}$$

$$\log O(r \mid d_i, q) = -3.51 + 37.4X_1 + 0.33X_2 - 0.1937X_3 + 0.929X_4$$

$$X_1 = \frac{1}{\sqrt{N}+1}\sum_j \frac{q_j}{\|q\|_1 + 35}$$

$$X_2 = \frac{1}{\sqrt{N}+1}\sum_j \log\frac{t_{i,j}}{\|d_i\|_1 + 80}$$

$$X_3 = \frac{1}{\sqrt{N}+1}\sum_j \frac{\sum_i t_{i,j}}{\sum_{m,n} t_{m,n}}$$

$$X_4 = N$$

where $\|q\|_1$ is the City-Block length of the query q, and $||d||_1$ is the City-Block length of the document d. We used the above formulae to compute our logistic regression similarity scores for each document in the `IR` system.

3.4 Our Pircs Retrieval Model

Pircs [21] is another well-known probabilistic indexing and retrieval system, which has consistently performed well in the Chinese information retrieval tasks of various open evaluations (TREC-5, TREC-6, TREC-9 and NTCIR). The document weight $P(q, d_i)$ for the i-th document is a linear weighted sum of the activation of query terms and documents in a conceptual network of query terms, index terms and documents:

$$P(q,d_i) \equiv a\sum_j w_{i,j} \times S\left(\frac{q_j}{\|q\|_1}\right) + (1-a)\sum_j w_{i,j} S\left(\frac{t_{i,j}}{\|d_i\|_1}\right)$$

$$w_{i,j} \equiv \log\left[\frac{t_{i,j}}{\|d_i\|_1 - t_{i,j}} \times \frac{\sum_{m,n} t_{m,n} - \|d_i\|_1 - \sum_m t_{m,j} + t_{i,j}}{\sum_m t_{m,j} - t_{i,j}}\right]$$

$$w_{i,j} \equiv \log\left[\frac{q_j}{\|q\|_1 - q_j} \times \frac{\sum_{m,n} t_{m,n} - \|d_i\|_1 - \sum_m t_{m,j}}{\sum_m t_{m,j}}\right]$$

This mixture model in which weighting depends on the direction of matching is an important extension of the Bayesian probabilistic models discussed previously. The mixture parameter a determines the contribution of the weights applied to the query terms and the index terms. According to our pilot study, a has little impact on the retrieval performance. For subsequent evaluation, a is set to 0.5.

Although there are probabilistic weights, like $w_{i,j}$ and $w_{i,j}$, for query terms and documents, respectively, these weights are scaled by some normalized document and query term weights, after a signal transformation function, $S(.)$. Based on our previous study [4], the ramp function is used and our implementation of Pircs only computed the ranking score for the entire document, instead of based on sub-documents. Length weighting is achieved by scaling the query term frequencies. Since the activation involves the computation of query lengths, the query lengths need to be adjusted using the length weighted query term frequencies for valid results.

One important difference between our Pircs retrieval model and the one in the literature is the Pircs retrieval model in the literature typically uses sub-documents for retrieval whereas our Pircs retrieval model uses the entire document for ranking. It is not known whether this would have an impact on the retrieval effectiveness.

4 Evaluation

Based on the NTCIR-III test data, we examined performance of various types of query and indexing strategies (i.e. bigram, Pircs, hybrid and short hybrid). The test data occupies about 747M bytes, including mark up and directory structures and the evaluation was carried out using 42 queries. The evaluation was carried out using the `IR` system configured for different retrieval models and different indexing strategies. It is emphasised that the retrieval models were implemented in our system and they may not correspond to the retrieval systems of other participants.

4.1 Space efficiency

Table 1 shows the storage cost of the inverted index and the dictionary in megabytes. It is well known that bigram indexing has the largest storage cost. Hybrid term indexing incurs more index storage than that of Pircs indexing by about 10%. Since the dictionary storage cost of our Pircs indexing and hybrid term indexing are similar, the difference in storage between Pircs and hybrid term indexing is due to the index (i.e. posting).

Indexing Strategy	Index (Mbytes)	Dictionary (Mbytes)	Total (Mbytes)
Bigram	1,329	84	1,413
Pircs	674	54	728
Hybrid	728	53	781
Short Hybrid	751	55	806

Table 2: Storage cost (in megabytes) of the inverted index and the dictionary.

4.2 Formal Runs

Three formal runs were submitted based on the 2-Poisson model using short hybrid term indexing, which uses only two character words for word segmentation for the concept queries (C), the concept and title queries (TC), and the question, title and narrative queries (TDN). The performances are summarised in Table 3.

Query	Rigid (%)			Relax (%)		
Type	MAP	RP	R	MAP	RP	R
C	24	27	74	29	33	72
TC	27	30	78	33	36	75
TDN	28	32	80	35	38	79

Table 3: Performance of our formal runs in percentages. Key: MAP for mean average precision, RP for R-precision and R is the recall.

In general, the more information in the queries, the better the precision and recall. The precision performances evaluated using the relax judgement were consistently higher than those that were judged rigidly. By contrast, the recall performance was better for rigidly judged relevant documents than relaxedly judged relevant documents. This phenomenon may be due to the fact that there are more documents judged as relevant by relaxed judgements. Since only the top 1,000 documents are examined whether evaluated using rigidly or relaxed judged documents, the likelihood of observing a relevant document judged in a relaxed manner is higher than that for rigidly judged relevant documents.

Figure 1 shows the precision of individual TDN queries relative to (or minus) the precision of the corresponding query averaged across different formal runs. The precisions of 8 queries out of 42 were below average. The precisions of two queries (i.e., 35 and 37) were substantially lower than average. One query is about "war crime prosecutions" and the other query is about "cloning human bodies". We are not entirely sure why the precisions were substantially worst than average.

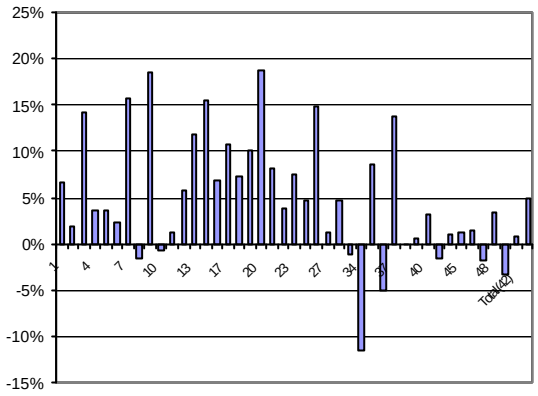

Figure 1: The precision of different TDN queries relative to the precision of the corresponding queries averaged across all formal runs.

Figure 2 shows an interesting relationship between precision and recall of individual queries, for 2-Poisson model using hybrid term indexing. In this figure, queries are ranked by their precision in descending order. The general trend is that the lower the precision, the lower the recall of a particular query, with a correlation value of 66%. It suggested that recall and precision of individual queries were related rather than completely independent measures of performance. In the rest of this paper, the mean average precision will be reported and the recall performance will be only reported when there are sufficient space.

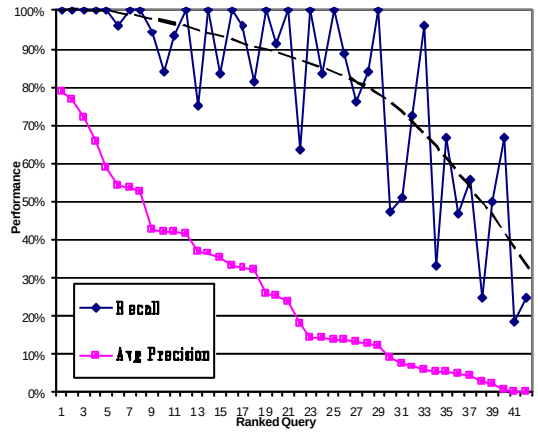

Figure 2: Retrieval effectiveness performance (i.e. recall and precision) against individual queries ranked by the precision performance in descending order.

Direct comparison of our formal runs with other formal runs is difficult because of the different parameter settings and different levels of sophistication. If we simply examine the overall MAP values, then other groups like APL, Berkeley, PIRCS, and CRL have obtained higher MAP values than ours. If we compare on the basis of retrieval without pseudo relevance feedback (PRF), then our MAP values are amongst the best for each of the submitted query types. Hence, we believed that our retrieval model and indexing strategy are comparable to the best performing groups.

4.3 Comparison with Other Indexing Methods

In our previous study [4], it was found that the 2-Poisson model using bigram indexing achieved robust and good retrieval performance. Since Pircs indexing has consistently achieved good results, we examined bigram and Pircs indexing with the 2-Poisson retrieval model for comparison purposes.

Table 4 shows the mean averaged precision (MAP) and the recall performance for bigram, Pircs and short hybrid term indexing strategies for four common types of queries: title (T), concept (C), title and concept (TC) and merging all query types together (TDNC), without PRF. Clearly, our Pircs indexing strategy without any PRF was not performing as good as bigram indexing. Although bigram indexing and our short hybrid term indexing were performing similarly for different types of queries using the 2-Poisson model, the recall performance of hybrid term indexing was slightly better than that of bigram indexing.

Index	Query Type	Rigid		Relax	
		MAP	R	MAP	R
Bigram	T	21%	65%	26%	62%
	D	19%	64%	24%	61%
	C	25%	72%	29%	68%
	TC	27%	78%	32%	74%
	TDCN	29%	82%	35%	77%
Our PIRCS	T	16%	54%	19%	49%
	D	16%	53%	19%	50%
	C	19%	63%	22%	58%
	TC	21%	68%	24%	63%
	TDCN	21%	70%	25%	64%
Short Hybrid	T	21%	66%	27%	63%
	D	20%	66%	27%	63%
	C	24%	74%	29%	72%
	TC	27%	78%	33%	75%
	TDCN	29%	82%	36%	78%

Table 4: The precision and recall performance of the 2-Poisson model using bigram, Pircs and short hybrid indexing for five common types of queries. Note that the rows with grey background are results based on the formal runs. Key: T for title queries, D for description queries, C for concept queries, TC for title and concept queries and TDCN for combining all types of queries together.

Apart from indexing storage to differentiate the performance of bigram and hybrid term indexing, retrieval time is also another important discriminating performance measure. Figure 3 shows the retrieval time per query against the number of terms per query, for bigram and short hybrid term indexing, similar to [22]. For both bigram indexing and short hybrid term indexing, the retrieval time varies linearly with the number of terms in a query, with over 90% correlation. However, it appeared that short hybrid term indexing takes longer to retrieve compared with bigram indexing if the number of terms in a query is identical. This indicates that the posting lists of hybrid term indexing are longer than for bigram indexing.

However, the number of query terms indexed using bigrams is larger than that using short hybrid term indexing for the same query. Since the retrieval time is a linear function of the number of terms in the query, the total retrieval time may be longer for bigram indexing than short hybrid term indexing and it may be sufficient to compare the average retrieval time per query for bigram indexing and short hybrid term indexing. Using TC and TDCN queries together, bigram indexing takes about 37s per query, compared with 24s per query. Therefore, the retrieval speed of hybrid term indexing is about 1.5 times faster than that using bigram indexing. In our past study [4], the retrieval speed of word indexing was the fastest and it was only 1.15 times faster than bigram indexing. Hence, it seemed the retrieval speed for hybrid term indexing would be even faster than word indexing although further evaluation is necessary.

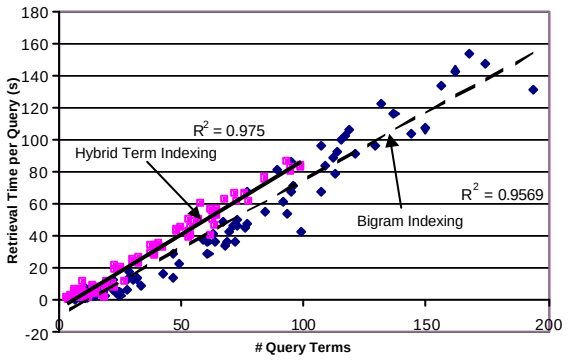

Figure 3: Retrieval time per query for bigram indexing and short hybrid term indexing for TC and TDCN queries.

In summary, hybrid term indexing achieved similar retrieval effectiveness performance as bigram indexing using the 2-Poisson model and it is more efficient in terms of index storage and retrieval speed than bigram indexing.

4.5 Comparison with Other Retrieval Models

To verify the findings of the previous study [4], we examined whether the retrieval effectiveness of 2-Poisson model combined with bigram indexing was the best, compared with the vector-space model, logistic-regression model and Pircs model. For the vector-space model, Ltw1 weighting scheme (Table 1) is used.

Type	RM	Rigid (%)			Relax (%)		
		b	p	h	b	p	h
T	VSM	17	8	15	22	11	21
	LR	20	18	20	26	20	**27**
	2-P	**21**	16	**21**	26	19	**27**
	PP	19	16	18	25	20	23
D	VSM	15	8	16	20	11	20
	LR	18	17	19	23	20	25
	2-P	19	16	**20**	24	19	**27**
	PP	16	15	13	21	19	18
C	VSM	18	8	17	22	10	21
	LR	23	19	23	28	22	28
	2-P	**25**	19	24	**29**	22	**29**
	PP	23	19	19	27	21	23
TC	VSM	21	10	20	26	13	25
	LR	25	21	25	29	25	31
	2-P	**27**	21	**27**	32	24	**33**
	PP	25	20	20	29	24	24
TDCN	VSM	23	13	21	28	18	27
	LR	28	19	26	33	24	33
	2-P	**29**	21	**29**	35	25	**36**
	PP	23	16	16	28	20	18

Table 5: The mean averaged precision based on the rigid and relax judgment of relevance for the different retrieval models (RMs): vector space model (VSM), logistic-regression model (LR), the 2-Poisson model (2-P) and our Pircs retrieval model (PP), for five types of queries, comparing bigram (b), our Pircs (p) and short hybrid (h) indexing strategies.

Table 5 shows the mean average precision based on the rigid judgment of relevance for different retrieval models using bigram, Pircs and short hybrid term indexing for five types of queries (i.e., T, D, C, TC and TDCN). Clearly, the 2-Poisson model achieved the highest precision using bigram indexing for the four types of queries based on the rigid relevance judgment. This confirms our previous study [4]. However, the highest precision achieved using he 2-Poisson model and bigram indexing is the same as that corresponding to the 2-Poisson model using short hybrid term indexing, differing by no more than 1% in just one type of queries (i.e., concept queries). The results in Table 5 also shows that the highest precisions were obtained if short hybrid term indexing is used for the 2-Possion model. In particular, for relax judgment, the (equal)

best performances were consistently obtained by the 2-Poisson model using short hybrid term indexing.

4.6 Pseudo-Relevance Feedback

All of the evaluation carried out did not use pseudo-relevance feedback (PRF) because this facilitates direct comparison of different retrieval models using different indexing strategies. However, if we compare the best performance (Table 6) without regard to the sophistication of the retrieval systems, our performance is still about 5% lower than the best formal runs. An immediate question is whether the performance of the 2-Poisson model using short hybrid term indexing can be comparable by increasing the sophistication of our retrieval system further. Therefore, we explored the use of PRF.

Query Type	# formal runs	Rigid		Relax	
		MAP	RP	MAP	RP
T	1	19%	22%	25%	28%
D	14	29%	30%	36%	38%
C	4	24%	27%	29%	33%
TC	4	30%	31%	38%	39%
TDCN	8	34%	35%	42%	43%

Table 6: Best performance of all the formal runs for T, D, C, TC and TDCN queries. Key: MAP for mean average precision and RP for R-precision.

Our PRF scheme collects the top six documents and extracts the best, 140 terms by our ranking score, which are simply the product of the total term frequency and the number of documents that the term has appeared in the top six documents.

Table 7 shows the mean averaged precision, R-precision and recall of the 2-Poisson model using hybrid term indexing, with our PRF. Although our performance with PRF for title queries were much better than the best in the formal runs (Table 6), there was only one formal run submitted for title queries. Hence, it is difficult to conclude whether the 2-Poisson model using short hybrid term indexing was performing better than others for title queries. Again for C queries and TC queries, there were only four formal runs. For TDCN queries, there were eight formal runs. In this case, the MAP performance of the 2-Poisson model using short hybrid term indexing with PRF was comparable to the best results in the formal runs. Since the D queries were used for comparison in the workshop, the number of participating systems was the largest. In this case, our 2-Poisson model with short hybrid term indexing achieved similar performance as the best results in the formal runs for D queries. However, it is unclear whether better results would be obtained in the informal runs of other participating retrieval systems. For future comparisons, the average precisions for the top 10 (retrieved) documents using the 2-Poisson model with short hybrid term indexing are reported in Table 7.

Query Type	Rigid (%)				Relax (%)			
	P	RP	P_{10}	R	P	RP	P_{10}	R
T	28	28	32	74	34	35	47	71
D	32	33	40	81	39	40	54	77
C	32	32	41	81	38	39	55	76
TC	33	33	42	81	39	39	55	76
TDCN	35	35	43	86	41	42	59	80

Table 7: Performance of the 2-Poisson model using short hybrid term indexing, with PRF. Key: P for mean average precision, RP for R-precision, P_{10} is the average precision for the top 10 (retrieved) documents and R is the recall.

5 Summary

In this participation, we have demonstrated that our 2-Poisson model using hybrid term indexing was an effective and efficient combination of retrieval model and indexing strategy. Further, the effectiveness of this combination of retrieval model and indexing strategy can enhance the mean averaged precision performance to be comparable to the best formal run results using PRF. Therefore, we conclude that short hybrid term indexing strategy is a promising indexing method if used with the 2-Poisson model.

We have also demonstrated that the sophistication of our `IR` system is comparable to other participating systems in terms of retrieval effectiveness. It should be emphasised that some participating groups are interested in understanding specific information retrieval issues, which may not relate to achieving comparative good retrieval effectiveness. Also, it should be emphasised that the results obtained are based on our implementation of different retrieval models, in which there may be unknown differences with those published in the literature.

Acknowledgement

We would like to thank 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. Also, Robert is thankful to Prof. Noriko Kando for pointing out the importance of term weighting before the NTCIR participation (although the okapi weighting scheme were not ready for our system to participate in NTCIR-II) and Prof. Gey for raising the issue of the sophistication of our retrieval

system, in the last NTCIR-II open evaluation. Robert is also in debt to Prof. K.L. Kwok for commenting our implementation of Pircs, which differs from his system and to Prof. K.F. Wong for making available the TREC data for experimentation. Robert is also grateful to ROCLING for providing the dictionary. This work is supported in part by Departmental Earnings Account Project H-ZJ88.

References

[1] T.F. Tsang, R.W.P. Luk and K.F. Wong. Hybrid term indexing using words and bigrams. *Proceedings of IRAL 1999*, Academia Sinica, Taiwan, 1999, pp. 112-117.

[2] K.C.W. Chow, R.W.P. Luk, K.F. Wong and K.L. Kwok. Hybrid term indexing for weighted Boolean and vector space models. *International Journal of Computer Processing of Oriental Languages*, **14(2)**: 1-19, 2001.

[3] R.W.P. Luk, K.F. Wong and K.L. Kwok. Hybrid term indexing: an evaluation. *Proceedings of 2nd NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization*, National Institute of Informatics, Tokyo, Japan, 2001, pp. 130-136.

[4] R.W.P. Luk and K.L. Kwok. An evaluation of Chinese document indexing strategies and retrieval models. Submitted to *ACM Trans. Asian Language Information Processing*.

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

[6] S.E. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. *Proceedings of ACM SIGIR '92 R&D in IR*, 1992, pp. 232-241.

[7] A. Chen, J. He, L. Xu, F.C. Gey and J. Meggs. Chinese text retrieval without using a dictionary. *Proceedings of ACM SIGIR '97 R&D in IR*, 1997, pp. 42-49.

[8] K.L. Kwok. Improving Chinese and English ad-hoc retrieval: a Tipster text phase 3 project report, *Information Retrieval*, **1(3)**: 217-250, 1999.

[9] 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.

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

[11] 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.

[12] 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.

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

[14] 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.

[15] R.W.P. Luk. Chinese-word segmentation based on maximal-matching and bigram techniques, *Proceedings of ROCLING VII*, 1994, pp. 273-282.

[16] D.L. Lee, H. Chuang and K. Seamons. Document ranking and the vector-space model. *IEEE Software*, **14(2)**: 67-75, 1997.

[17] S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*. **27**: 129-146, 1976.

[18] S.E. Roberston, S. Walker, S. Jones, M. Hancock-Beaulieu and M. Gatford. Okapi at TREC-2, *Proceedings of TREC-2*, 1993, pp. 21-25.

[19] S.E. Roberston, S. Walker, S. Jones, M. Hancock-Beaulieu and M. Gatford. Okapi at TREC-3, *Proceedings of TREC-3*, 1994, pp. 109-128.

[20] W.S. Cooper, A. Chen and F.C. Gey. Full text retrieval based on probabilistic equations with coefficients fitted by logistic regression. *Proceedings of TREC-2*, 1994, pp. 57-66.

[21] K.L. Kwok. A network approach to probabilistic information retrieval. *ACM Trans. on Information Systems*. **12**:325-353, 1996.

[22] P. Vines and J. Zobel. Efficient building and querying of Asian language document databases, *Proceedings of IRAL 1999*, Academia Sinica, Taiwan, 1999, pp. 118-125.