Hybrid Term Indexing: an Evaluation

Robert W.P. Luk

Dept. Computing
Hong Kong Polytechnic University
Email: csrluk@comp.polyu.edu.hk

K.F. Wong

Dept. Sys. Eng. & Eng. Man.
Chinese University of Hong Kong
Email: kfwong@se.cuhk.edu.hk

K.L. Kwok

Dept. Computer Science
Queen's College
CUNY
Email: kwok@ir.cs.qc.edu

ABSTRACT

Retrieval effectiveness depends on how terms are extracted and indexed. For Chinese text (and others like Japanese and Korean), there are no space to delimit words. Indexing using hybrid terms (i.e. words and bigrams) were able to achieve the best precision amongst homogenous terms at a lower storage cost than indexing with bigrams. However, this was tested with conjunctive queries, using a small test data set. Here, we extended the vector space model using the cosine measure, for processing hybrid terms. We also introduced weighting based on the length of the term. Our evaluation shows that the averaged precision of hybrid term indexing is about the same as the best precision, achieved using bigram indexing, but incurring less storage (about 61% of the storage for bigram indexing). The precision performance of hybrid term indexing is consistently better than that of word indexing, even though their storage cost is about the same. Ranking based on the length of the query terms slightly improves the retrieval effectiveness of hybrid term indexing but degrades the retrieval effectiveness of word indexing. Even though our best performance is the worst compared with that of the other participants, this may be due to some common factors across different indexing strategies (e.g. stop words, term weights and query term processing), and may not be due to the indexing strategies that we are evaluating.

Keywords: Chinese information retrieval, indexing, IR models, and evaluation.

1. Introduction

Chinese documents are becoming widely available in the Internet. Chinese newspapers in different parts of the world are now accessible on-line, for example Ming Bao in Hong Kong, Lianhe Zaopao in Singapore, Renmin Raobao in mainland China, China Times in Taiwan and CANews in USA. There has been rapid development of Internet in China, Hong Kong, Taiwan and Singapore. Yahoo! has set up its Chinese portal in Hong Kong to capture this growing market.

With the increasing large amount of information on the Internet, an apparent problem is to find the relevant information via the Internet. Chinese information retrieval is becoming more important in the advent of this development. Indexing techniques using inverted file, model-based signature [1], superimposed coding signature [2], variable bit-block compression signature [2] and pat-tree [3,4] were modified to index Chinese (Japanese) documents, as well as mixed Chinese-English documents.

In general, these indexing techniques only affect the storage and speed performance and occasionally there is trade-off between this performance with retrieval effectiveness (e.g. recall and precision). On the other hand, defining what terms to index in the document directly affect retrieval effectiveness, with the exception of PAT-trees [3,4], which incurs a significant storage overhead.

Recently, research work [5,6,7] compared the retrieval effectiveness using different types of terms (i.e. characters, bigrams and words). In general, retrieval based on characters has the best recall where as retrieval based on words or based on bigrams has the best precision. Unlike words, bigrams do not have the out-of-vocabulary problem but they incur significant storage overhead. To overcome the shortcoming of one type of terms over the others, research workers have merged the retrieval lists from different indexed terms. Leong and Zhou [8] have found little improvement in merging retrieval lists but Kwok [5] have found significant improvement when merging retrieval

lists of words and bigrams. One disadvantage of merging retrieval lists is the high overhead to store two indices and to process 2 lists of results. Recently, Tsang *et al.* [9] proposed to merge the index, instead of the retrieval lists. Effectively, the index contains different types of terms and it is called a *hybrid index*. Instead of exhaustive indexing, bigrams are extracted only at locations where the out-of-vocabulary problems are likely to occur. In this way, the index size and bigram dictionary size are kept low, and the retrieval performance can still be improved (around 10% in terms of precision). However, the evaluation was carried out for conjunctive queries.

In this paper, we will explore the use of hybrid term indexing for 2 general types of IR models: the extended Boolean model and the vector space model. In the next section, we will give a brief review of hybrid indexing. In section 3, we will describe how the 2 general types of IR models are extended for hybrid term indexing. In section 4, the evaluation of using hybrid term indexing, for the 2 types of IR models are reported. Finally, we conclude.

2 Hybrid Term Indexing

From previous work, 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 [10,11] but there are concerns about the recall performance of automatic extractions or the concerns about the scope of word formation rules [12]. 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 [13] 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: Word and Bigram 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. Word+bigram 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.

3 IR Model Extension

Two common IR models, weighted Boolean and the vector space model, can rank documents according to their similarities with the query. We will examine the vector space models based on the cosine measures.

3.1 Weights

To compute the similarity *S(q,D)* between the query *q* and the document *D*, both models rely on assigning weights to the index terms and the query terms. Typically, the index terms are weighted [14]

by the term occurrence frequency and by the inverse document frequency as in Equation 1:

$$w(t_i, D_j) = t_{i,j} \times d_j \qquad (1)$$

where $t_{i,j}$ is the occurrence frequency of term t_i in document D_j and d_i is the inverse document frequency of the term t_i.

In the hybrid term indexing, different types of term have different importance if they are matched. For instance, an index term, which is a long word, is a reliable indicator of relevance because it is seldom to match any long sequences and this type of term is likely to be technical terms or proper nouns. In addition, since the index term is a word in the system dictionary, it was applied in word segmentation, instead of exhaustively extracted using a sliding window. Thus, it is more difficult to find a match and hence it is more reliable. On the other hand, bigrams were extracted exhaustively at specific regions of the text. To reflect their relative importance, we assign a scale weight $z(t_i)$ in addition to the weight $w(t_i,D_j)$ so that the total weight $W(t_i,D_j)$ becomes Equation 2.

$$W(t_i, D_j) = z(t_i) \times w(t_i, D_j) \qquad (2)$$

Smaller scale weights are assigned to bigrams compared with the weights of 2 character words. Since bigrams are more discriminating than single character words, we assign a larger scale weight to bigrams. For evaluation, if the index term is a bigram, it is assigned with a weight equals to 1.5. Otherwise, the index term is assigned a weight equals to its length.

3.2 Vector Space Model Extension

In the vector space model, extension is needed when the query term is not an index term. Similar to the Boolean model, word segmentation is applied to that query term and the bigrams are extracted from the single character sequences. Since the set of related index term extracted from the query term must all occur, we consider the index terms are conjoined together. For simplicity, the conjunction is evaluated using the Fuzzy model (i.e. taking the minimum of the weights of all the related index terms).

Formally, the cosine similarity $C(.,.)$ is extended and is defined as in Equation 3:

$$C(Q, D_j) = \frac{\sum_{q \in Q} \min_{x \in WS(q)} \{w(x) \times W(x, D_j)\}}{len(Q, D_j) \times |D_j|} \qquad (3)$$

where $|D|$ is the vector length of the document D and the vector length of Q is now modified to form Equation 4:

$$len(Q, D_j) = \sum_{q \in Q} \left(w(x)^2 \mid x = \arg\min_{y \in q} \{w(y) \times W(y, D_j)\} \right) \qquad (4)$$

Note that $len(Q,D_j)$ depends on the document D_j since the identification of the index term x depends on the particular document D_j.

For simplicity and speed of computation, typically, $w(x)$ is set to a constant, which is equals to $w(q)$. Since the ranking is not affected by any monotonic scaling, $len(Q,D_j)$ and the weights $w(x)$ can be discarded. In this case, the new cosine similarity $C'(.,.)$ can be simplified to Equation 5:

$$C'(Q, D_j) = \frac{\sum_{q \in Q} \min_{x \in WS(q)} \{W(x, D_j)\}}{|D_j|} \qquad (5)$$

4 Evaluation

Based on the NTCIR Workshop 2 test data, we examined performance of various types of query (i.e. title queries and very short queries) and indexing strategies (i.e. word, bigram and hybrid). The test data occupies about 490M bytes and evaluation was carried out using 50 queries.

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. Surprisingly, hybrid term indexing incurs less index storage than that of words indexing. Since there are bigrams in hybrid term indexing, the storage cost of the dictionary for hybrid term indexing is much larger than that of word indexing. The overall storage cost of hybrid term indexing (i.e. index plus dictionary storage cost) is about the same as that of word indexing (c.f. 223 versus 228). Hybrid term indexing is only 61% of the total storage of bigram indexing.

Index Strategy	Index Size (Mbytes)	Dictionary Size (Mbytes)
Word	224	4
Bigram	364	56
Hybrid	202	21

Table 1: Storage cost (in megabytes) of the inverted index and the dictionary.

4.2 Retrieval Effectiveness

4.2.1 General Results

Figure 1 shows the recall-precision curve for title queries. Since there are a number of indexing strategies, the results appeared cluttered. We summarise the performance in terms of the 11-point averaged precision values.

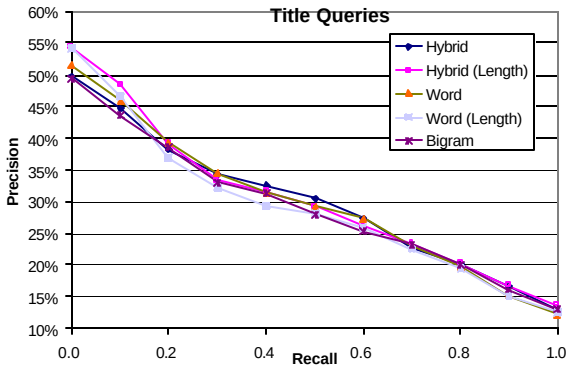

Figure 1: Precision-recall curve for title queries.

Table 2 shows the 11-point averaged precision of various indexing techniques for title queries. The best-averaged interpolative precision is 30.5%, achieved using hybrid term indexing, with length weighting, for both rigid and relaxed judgement. However, the best and the near best performance were not substantially different (within 1%). For title queries, we can say that hybrid term indexing is as good as bigram indexing up to this point. Apart from interpolative precision, there is also the averaged top *N* document precision values for comparison. Typically, the 11-averaged interpolative and 11-point averaged the top N document precision values follow similar trends but the former usually has a higher value than the latter. Bigram indexing does not have any length weighting since each bigram has identical length (i.e. 2 characters) and length weighting would have no effect on the ranking of documents.

Table 3 shows the 11-point averaged precision of various indexing techniques for very short queries. In this case, bigram indexing achieved the best 11-point averaged precision of 45% and 55% for rigid and relaxed judgement, respectively. The second best performance is within 2% lower than the best, which is achieved by hybrid term indexing, with length weighting.

Indexing		Hybrid		Word		Bigram
Length Weighting		N	Y	N	Y	N/A
Inter.	Rigid	29.9%	**30.5%**	29.9%	29.3%	29.2%
	Relaxed	**41.2%**	**41.5%**	40.1%	39.0%	41.0%
Doc	Rigid	15.3%	15.7%	16.1%	15.4%	15.1%
	Relaxed	28.3%	28.6%	29.0%	28.0%	28.7%

Table 2: 11 point averaged precision for title queries. Key: N/A for not applicable, Inter means interpolative precision, Doc means top N document precision, Rigid refers to rigid judgement results and Relax refers to relaxed judgement results.

Indexing		Hybrid		Word		Bigram
Length Weighting		N	Y	N	Y	N/A
Inter.	Rigid	42.0%	43.2%	40.7%	37.8%	**45.3%**
	Relaxed	52.4%	54.1%	50.4%	47.2%	**55.5%**
Doc	Rigid	21.9%	22.9%	21.9%	19.6%	23.8%
	Relaxed	35.5%	36.9%	35.3%	33.1%	37.7%

Table 3: 11-point averaged precision for very short queries.

4.2.2 Is Hybrid Term Indexing Better?

Table 4 shows the 11-point averaged precision of hybrid term indexing minus that of other indexing strategies for title queries. On average, there was an insignificant better precision of 0.4%. The best case is the performance of hybrid term indexing with length weighting against word indexing with length weighting. In this case, there were consistent better performance across interpolative and top *N* document precisions, as well as between rigid and relaxed judgement results.

For very short queries, hybrid term indexing appeared to have a slightly better precision against word indexing, across interpolative and top *N* document precisions, as well as rigid judgment results and relaxed judgment results. However, hybrid term indexing is performing worst than bigram indexing, consistently across interpolative and top *N* document precisions, as well as rigid judgment results and relaxed judgment results. Even though the overall difference in 11-point averaged precision between hybrid term indexing and other indexing strategies is positive (i.e.1.1%), the overall average is biased towards the performance difference between hybrid term indexing and word indexing.

		Hybrid - Word		Hybrid - Bigram	
Length Weighting		N	Y	N/A	Average
Inter.	Rigid	0.1%	1.2%	0.7%	0.7%
	Relaxed	1.2%	2.5%	0.3%	1.3%
Doc	Rigid	-0.7%	0.3%	0.2%	-0.1%
	Relaxed	-0.8%	0.6%	-0.5%	-0.2%
Average		-0.1%	1.1%	0.2%	0.4%

Table 4: Difference in 11-point averaged precision between hybrid term indexing and other indexing strategies, for title queries.

Indexing		Hybrid - Word		Hybrid - Bigram	Average
Length Weight		N	Y	N/A	
Inter.	Rigid	1.3%	6.1%	-3.3%	1.4%
	Relaxed	2.1%	6.9%	-3.0%	2.0%
Doc	Rigid	0.0%	3.4%	-1.8%	0.5%
	Relaxed	0.2%	3.8%	-2.1%	0.6%
	Average	0.9%	5.1%	-2.6%	1.1%

Table 5: Difference in 11-point averaged precision between hybrid term indexing and other indexing strategies, for very short queries.

4.2.3 Is length weighting effective?

Table 6 shows the difference in performance between ranking with length weighting and without length weighting, for the same indexing strategy. Interestingly, length weighting improves the precision of hybrid term indexing but degrades the performance of word indexing, consistently across interpolative and top *N* document precisions, as well as rigid and relaxed judgment results. Similar pattern can be observed for ranking with and without length weighting for very short queries (Table 7). In summary, length weighting slightly improves the retrieval effectiveness of hybrid term indexing but slight degrades the retrieval effectiveness of word indexing.

Indexing		Hybrid	Word	Average
Inter.	Rigid	0.59%	-0.54%	0.03%
	Relaxed	0.21%	-1.05%	-0.42%
Doc	Rigid	0.40%	-0.64%	-0.12%
	Relaxed	0.29%	-1.04%	-0.38%
Average		0.37%	-0.82%	-0.22%

Table 6: Difference in 11-point averaged precision between ranking with and without length weighting, for title queries.

Indexing		Hybrid	Word	Average
Inter.	Rigid	1.9%	-2.9%	-0.5%
	Relaxed	1.6%	-3.2%	-0.8%
Doc	Rigid	1.0%	-2.4%	-0.7%
	Relaxed	1.3%	-2.2%	-0.5%
	Average	1.5%	-2.7%	-0.6%

Table 7: Difference in 11-point averaged precision between ranking with and without length weighting, for very short queries.

4.3 System Comparison

If we compare the best precision of our system with other participants', then our best performance is the worst amongst others best. We believe that

this is due to the common factors across different indexing strategies, instead of differences between individual indexing strategies. First, our stop word list is different from other systems. Our stop word list is just a list of single characters but it is known that certain stop words should have multiple characters (e.g. conjunction). Second, many other systems used the okapi score [15] where as we used the simple TF-IDF score for ranking, as pointed out by Prof. Gey. Initially, we considered the use of a modified version [16] of the 2-Poisson model. Unfortunately, there might be some errors in our implementation and the results were abandoned. Otherwise, we can assess the impact of different weighting schemes on the retrieval effectiveness. Third, we did not use any pseudo-relevance feedback or incorporating any concepts into the queries to boost the performance. Finally, our query pre-processing is very primitive. At present, it simply segments the query into a sequence of terms. Some form of query term weighting scheme should be developed to observe which query term may be more important than others, similar to [17].

Unfortunately, very few systems report on the storage cost and retrieval cost. So that it is hard to visualize the trade off between additional average precision gained and other factors. Vines and Zobel [18] have shown that although bigram indexing has good retrieval effectiveness, its retrieval efficiency is not as good as other indexing strategy. Here, we consider that hybrid term indexing is promising since it has similar retrieval effectiveness as the best indexing strategy, and it has substantial reduction in storage cost compared with the most retrieval-effective indexing strategy.

5 Conclusion and Future Work

We evaluated hybrid term indexing for the NTCIR Workshop 2, Chinese information retrieval task. According to our evaluation, hybrid term indexing achieved the best or near best retrieval effectiveness compared with word and bigram indexing, incurring only 61% of the storage needed by bigram indexing. The introduction of ranking using length weights for the query terms improves the retrieval effectiveness of hybrid term indexing but degrades the retrieval effectiveness of word indexing. Even though our best retrieval effectiveness is the worst amongst other participants' best, we believe that as more sophisticated techniques are employed, as in the other systems, hybrid term indexing remains a promising approach if storage cost and retrieval speed are considered significant.

In the future, the retrieval system should be enhanced with okapi weighting method or regression weighting method. In addition, some pseudo-relevance feedback should be explored. Hopefully, this can bring the retrieval effectiveness up by another 10% or 20%, comparable with the other participants' best. In addition, more variation of hybrid term indexing can be experimented.

Acknowledgement

This work was carried out when Robert Luk was on leave at University of Massachusetts (UMASS). We thank the Center for Intelligent Information Retrieval, UMASS, for providing the computing facility to carry out the evaluation. Note that this information retrieval system is developed independently and separately from the Inquiry system. We are grateful to ROCLING for providing the dictionary. We would like to thank Prof. Gey for his helpful comments. This work is supported in part by Departmental Earnings Account Project H-ZJ88.

References

[1] Chien, L-F. A Model-Based Signature File Approach for Full-text Retrieval of Chinese Document Databases, *Computer Processing of Chinese and Oriental Languages*, 1995.

[2] Chan, S.K., Y.C. Wong and R.W.P. Luk. Variable bit-block compression signature for English-Chinese information retrieval, *Proceedings of IRAL 98*, KRDL, National University of Singapore, 61-66, 1998.

[3] Chien, L-F. PAT-Tree-Based Keyword Extraction for Chinese Information Retrieval, *ACM SIGIR 97 Conference*, Philadelphia, USA, 50-58, 1997.

[4] Shishibori, M., M. Fiketa, K. Ando and J-I. Aoe, A Construction Method for the Index Represented by a Pointerless Patricia Trie, *Proceedings of IRAL 97*, Japan, 1997.

[5] Kwok, K.L. Comparing Representations in Chinese Information Retrieval, *Proc. of 20th Ann.Intl. ACM SIGIR Conf. on R&D in IR*, July 27-31, 1997. pp. 34-41.

[6] Lam, W., 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.

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

[8] Leong, M-K. 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, 19-21, 1997.

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

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

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

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

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

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

[15] Robertson, S.E. and Walker, S. Okapi/keenbow at TREC-8, *Proc. TREC-8*, 1999.

[16] Robertson, S.E. and Walker, S. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. *Proc. ACM SIGIR 92*, 232-241, 1992.

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

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