Hybrid Term Indexing for Different IR Models

Ken C.W. Chow[1], Robert W.P. Luk, K.F. Wong[2], K.L. Kwok[3]

[1]Hong Kong Polytechnic University
Dept. Computing,
Kowloon, Hong Kong
Email: csrluk@comp.polyu.edu.hk

[2]Chinese University of Hong Kong
Dept. Systems Eng. and Eng. Management
Shatin, Hong Kong
Email: kfwong@se.cuhk.edu.hk

[3]Queens College, CUNY
Dept. Computer Science
New York, USA
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. Here, we extended the weighted Boolean models using fuzzy and *p*-norm measures, as well as the vector space model using the cosine measure, for processing hybrid terms. Our evaluation shows that all IR models using hybrid terms achieve better average precision over those using words. Across different recall values, the weighted Boolean model using fuzzy measures with hybrid terms achieve consistently about 8% higher than those using words. The vector space model using the cosine measures with hybrid terms achieved the best improvement in the average recall and precision.

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

Proceedings of the 5th International Workshop Information Retrieval with Asian Languages

Copyright ACM 1-58113-300-6 00 009 ... $5.00

In general, these indexing techniques only affect the storage and speed performance and occasionally there are trade-off between these 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 $\{ w \} \cup \{ b \}$
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** $w \in D$ **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 $w \in D$
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 weighted Boolean models based on Fuzzy measures and the one based on p-norm measures, as well as 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, we use the scale weight assignment scheme in Table 1.

Type	Length	Scale Weight
Words	> 2 characters	1
Words	= 2 characters	0.6
Bigram	= 2 characters	0.4
Words	= 1 character	0.2

Table 1: Scale weight assignment scheme.

3.2 Boolean Model Extension

The extension of Boolean models for hybrid term indexing occurs when the query term q is not an index term. In this case, word segmentation is applied to the query term. If any part of the query term is segmented into single character sequences (of length larger than 1), then bigrams are extracted. Thus, a single query term can be expanded into a set of index terms, which can be words or bigrams. For a single query term to be considered to have occurred in the document, all the related index terms must also appear. Therefore, the single query term is expressed as the conjunction of all the related index terms.

Formally, a single query term can be expressed in Equation 3, as:

$$q = \prod_{x \in WS(q)} x \qquad (3)$$

where Π represents the conjunction and $WS(.)$ returns a set of index terms after applying algorithm A to the argument.

For different Boolean models, the computation of the weights for the query term is exactly the same as computing a conjunctive query of all the related index terms. For the fuzzy model, the similarity $F(.,.)$ between the query term q and the document D is defined according to Equation 4, as follows:

$$F(q, D_j) = \min_{x \in WS(q)} \{S(x, D_j)\} \qquad (4)$$

where as the similarity $P(.,.)$ for conjunction in the p-norm model is defined in Equation 5, as follows:

$$P(q, D_j) = \frac{1}{\left[\dfrac{\sum_{x \in WS(q)} w(x)^p \left(1 - R(x, D_j)\right)^p}{\sum_{x \in WS(q)} w(x)^p} \right]^{\frac{1}{p}}} \qquad (5)$$

where $R(.,.)$ is the normalized weight of $W(.,.)$ in p-norm and $w(x)$ is the user weight of the query term x.

For simplicity, typically, the query term weight $w(x)$ is equals to a constant. In this case, the modified similarity $P'(.,.)$ of the conjunction in p-norm is simplified to Equation (6), as follows:

$$P'(q, D_j) = \frac{1}{\left[\dfrac{\sum_{x \in WS(q)} \left(1 - R(x, D_j)\right)^p}{card(WS(q))} \right]^{\frac{1}{p}}} \qquad (6)$$

where $card(.)$ return the cardinality value of the set.

3.3 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 7:

$$C(Q, D_j) = \frac{\sum_{q \in Q} \min_{x \in WS(q)} \{w(x) \times W(x, D_j)\}}{len(Q, D_j) \times |D_j|} \qquad (7)$$

where $|D|$ is the vector length of the document D and the vector length of Q is now modified to form Equation 8:

$$len(Q, D_j) = \sum_{q \in Q} \left(w(x)^2 \mid x = \arg\min_{y \in q} \{w(y) \times W(y, D_j)\} \right) \qquad (8)$$

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 9:

$$C'(Q, D_j) = \frac{\sum_{q \in Q} \min_{x \in WS(q)} \{W(x, D_j)\}}{|D_j|} \qquad (9)$$

4 Evaluation

We used the PH corpus [15] for evaluation. In total, there are 3,480 documents derived from the Xinhua News Agency, occupying about 7Mbytes. A subject is asked to formulate 11 (Boolean type) queries (Table 2). To generate the queries for the vector space models, we simply discarded the Boolean operators in the Boolean queries. To measure recall performance, the subject has to read all the 3,480 documents, which is labour intensive. Instead, after query formulation, we randomly sampled 200 documents from the 3,480 documents and the subject read all the 200 documents before (s)he decides whether they are relevant or not. Although this sampling method it not as rigorous as those using pooling or Monte Carlo [16] methods, we are interested in the relative retrieval performance instead of the absolute retrieval performance.

Query No.	Query String
1	香港特別行政區
2	恆指上升有限
3	中英 or 聲明
4	一國兩制
5	電影 or 導演
6	主權回歸
7	投資者
8	單議席 and 單票制
9	單議席 or 單票制
10	大學 or The Open University of Hong Kong
11	中國 and not 香港

Table 2: (Boolean) queries for evaluation.

For this evaluation, we implemented the simplified p-norm similarity $P'(.,.)$ for conjunction only and we also implemented the simplified cosine measure $C'(.,.)$. Further, we have set $p = 2$ for the p-norm model. The evaluation compares the performance of word-based indexing with the hybrid term indexing since word-based indexing is amongst the best in precision and has good retrieval efficiency [17] (i.e. storage overhead and retrieval speed).

4.1 Recall and Precision Performance

Table 3 shows the 10-point average recall and precision performances of word-based indexing and hybrid term indexing, over the 11 queries. For all the different IR models, the average precision performances of hybrid terms are better (about 2%-9%) than those of word-based indexing. However, this is at the expense of the recall performance. The weighted Boolean based on p-norm measure appears to have little differences in performance between word-based and hybrid term indexing but the vector space model achieved a larger improvement in precision (8%) than in the degradation of recall (2%). The weighted Boolean based on Fuzzy measure achieved the best precision performance (93%). One reason why the precision performance is high is due to the small sample of documents since there are not many irrelevant documents from a small sample.

Figure 1 shows the recall and precision curves for the different IR models using different word or hybrid term indexing schemes. For the vector space model using word-based indexing, the precision degrades as the recall increase, which is typical of the precision-recall trade-off. However, the vector space model using hybrid term indexing has a fairly constant precision even for large recall values. This shows that the conjunction affects most with documents of lower rank. For the higher rank documents, the precision of the hybrid term indexing of the vector space model is lower (about 8%) than that of the word-based indexing.

IR Model	Indexing	Average Recall	Average Precision
Fuzzy Set	Hybrid	42%	93%
	Word	50%	84%
	Difference	-8%	9%
p-norm	Hybrid	45%	88%
	Word	46%	86%
	Difference	-1%	2%
Vector space (cosine)	Hybrid	55%	77%
	Word	58%	69%
	Difference	-3%	8%

Table 3: Average Recall and Precision Performance for Word and Bigram Indexing. The absolute performance is not representative since the document collection is small. However, the difference in performance is of interest. Hence, the Difference row is the performance of the Hybrid indexing method minus the corresponding performance of the word indexing method.

For the weighted Boolean model using Fuzzy measure, the precision of the hybrid term indexing is consistently higher than that of the word-based indexing by about 8% for all recall values. Similarly, the weighted Boolean model using p-norm measure achieved a better precision with hybrid

term indexing than that with word-based indexing, except the precision at 10% recall.

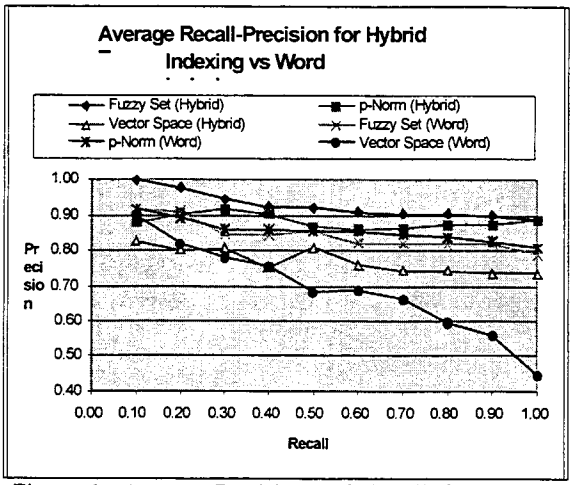

Figure 1: Average Precision and Recall for Word and Hybrid Term Indexing

4.2 Dictionary Size Variation

Two advantages of the hybrid approach are in reducing the storage demand of the bigram and in solving the out-of-vocabulary problems. To examine the effect of the different dictionary coverage, the hybrid approach is evaluated with different dictionary sizes, which are generated by randomly sampling words from the full-size dictionary.

Figure 2 shows the total storage demand using different dictionary sizes for both hybrid term and word-based indexing. Unlike [9], we showed the storage demand for the 200 indexed documents, instead of the 3,480 documents and the hybrid term indexing has almost no difference compared with the use of the full dictionary. Similar to [9], as the dictionary size decrease the number of storage demand increase.

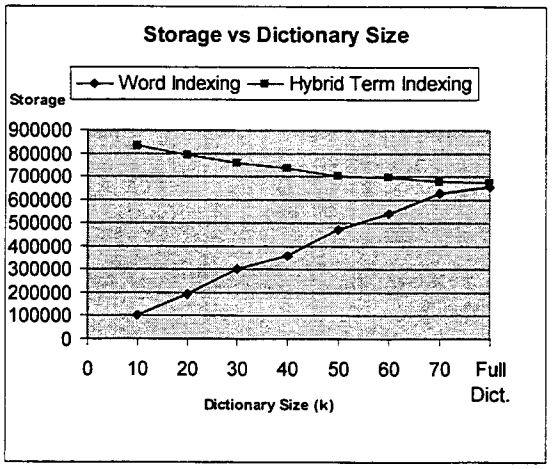

Figure 2: Storage demand of word and hybrid term indexing for different dictionary sizes.

Figure 3 shows the average recall variations with different dictionary sizes, for the different IR models. The average recall of various IR models using word-based indexing increases as the dictionary size increases while the recall decreases sharply initially and degrade slightly later, using hybrid term indexing.

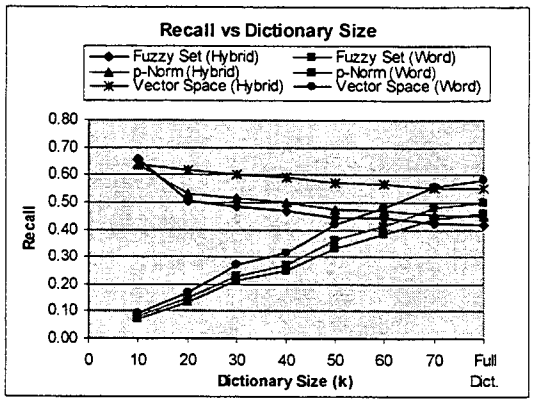

Figure 3: Average Recall for different dictionary sizes.

In Figure 4, the average precision of various IR models using word-based indexing decreases as the dictionary size increases but the precision increases for IR models using hybrid term indexing. The change in recall and precision becomes steadier when the dictionary size is over 70,000.

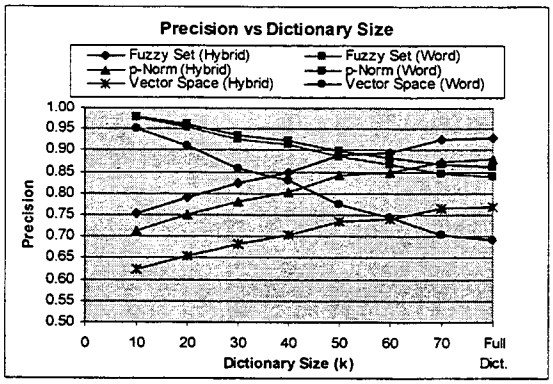

Figure 4: Average Precision for different dictionary sizes.

5 Conclusion and Future Work

We proposed a hybrid Chinese term indexing strategy: the word+bigram approach, for the weight Boolean model and the vector space model. This approach combines both the use of words in a dictionary and bigrams extracted from the documents. The number of bigrams was kept low by restricting the bigram extraction to areas in the text where word segmentation is likely to have the out-of-vocabulary problem.

In our evaluation, the precision performances of all IR models using hybrid term indexing are better than those

using word-based indexing. In particular, the weighted Boolean model using Fuzzy measure achieved about 8% consistently higher than that using word-based indexing and this Boolean model achieved the best precision-recall performance. However, the use of hybrid-term indexing incurs a slightly higher storage overhead than word-based indexing. In addition, the recall performance is lower for hybrid term indexing than word-based indexing. Since Internet search engines typically returns thousands of web pages for a single query, a better precision might be more desirable for those applications.

Our future work is to evaluate the use of hybrid term indexing with a much larger sample (e.g. the TREC Chinese corpus), in order to demonstrate its wider applicability. Also, we have not examined how different scale weights would affect the retrieval performance. In the future, it would be interesting to determine the best or optimal (defined in some sense) weights, for instance using genetic algorithms.

Acknowledgement

We are grateful to ROCLING for providing the dictionary and to Guo and Liu for providing the PH corpus. This work is supported by PolyU Central Research Grant 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, 1998. pp. 61-66.

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

[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*, 1997, **35**, pp.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, 1997, pp. 19-21.

[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, 1999, pp. 112-117.

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

[11] Guo, J. Critical tokenization and its properties, *Computational Linguistics*, 1997, **23**(4), 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*, 1995, **46**(2), 83-96.

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

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

[15] Guo, J. and H.C. Liu, "PH – a Chinese corpus for pinyin-hanzi transcription", *ISS Technical Report, TR93-112-0*, Institute of Systems Science, National University of Singapore, 1992.

[16] Burgin, R., The Monte Carlo method and the evaluation of retrieval system performance, *Journal of the American Society for Information Science*, 1999, **50**(2), 181-191.

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