Applying Machine Learning to Text Segmentation for Information Retrieval

Xiangji Huang[†], **Fuchun Peng**[†], **Dale Schuurmans**[†], **Nick Cercone**[†]

[†] School of Computer Science, University of Waterloo

Waterloo, Ontario, Canada, N2L 3G1

{jhuang, f3peng, dale, ncercone}@ai.uwaterloo.ca

Stephen E. Robertson[‡]

[‡]Microsoft Research Ltd., Cambridge, U.K. and City University, London, U.K.

ser@microsoft.com

August 22, 2002

Abstract

We propose a self-supervised word segmentation technique for text segmentation in Chinese information retrieval. This method combines the advantages of traditional dictionary based, character based and mutual information based approaches, while overcoming many of their shortcomings. Experiments on TREC data show this method is promising. Our method is completely language independent and unsupervised, which provides a promising avenue for constructing accurate multi-lingual or cross-lingual information retrieval systems that are flexible and adaptive. We find that although the segmentation accuracy of self-supervised segmentation is not as high as some other segmentation methods, it is enough to give comparable (in some cases even better) retrieval performance. It is commonly believed that word segmentation accuracy is monotonically related to retrieval performance in Chinese information retrieval. However, for Chinese, we find that the relationship between segmentation and retrieval performance is in fact *nonmonotonic*; that is, at around 70% word segmentation accuracy an over-segmentation phenomenon begins to occur which leads to a reduction in information retrieval performance. We demonstrate this effect by presenting an empirical investigation of information retrieval on Chinese TREC data, using a wide variety of word segmentation algorithms with word segmentation accuracies ranging from 44% to 95%, including 70% word segmentation accuracy from our self-supervised word-segmentation approach. It appears that the main reason for the drop in retrieval performance is that correct compounds and collocations are preserved by accurate segmenters, while they are broken up by less accurate (but reasonable) segmenters, to a surprising advantage. This suggests that words themselves might be too broad a notion to conveniently capture the general semantic meaning of Chinese text. Our research suggests machine learning techniques can play an important role in building adaptable information retrieval systems and different evaluation standards for word segmentation should be given to different applications.

Keywords: Machine learning, Word segmentation, EM algorithm, Chinese information retrieval.

1 Introduction

The increasing interest in cross-lingual and multilingual information retrieval has created the challenge of designing accurate information retrieval systems for Asian languages such as Chinese, Thai and Japanese. For multilingual information retrieval it is important to have an adaptable system which can be easily ported to new domains and languages. However, in designing information retrieval systems for these languages one faces the challenge of addressing the word segmentation problem as part of the retrieval process. That is, unlike English, in Asian languages, words are not explicitly delimited by whitespace. This creates significant problems both in interpreting queries and in indexing the text corpus.

The word segmentation problem in Asian languages has been heavily researched in the past decade [2, 5, 13, 19, 26, 27, 32]. In the retrieval task for these languages, the first step to indexing is to tokenize the collection. Traditionally there have been three approaches taken to tokenization: the dictionary based approach, the character based approach and the mutual information based statistical approach [6, 18, 23, 24]. In the dictionary based approach, one pre-defines a lexicon containing a large number of Chinese words and then uses heuristic methods such as maximum matching to segment the Chinese sentences. In the character based approach, sentences are tokenized simply by taking each character to be a basic unit. In mutual information based statistical approach, one uses the lexical statistics of the Chinese characters in corpora to mark the word boundaries. The lexical statistics include the occurrence frequency of a character in text corpora, and the co-occurrence frequency of two characters in text corpora. All these three approaches have advantages and disadvantages. The dictionary based approach is the earliest and most widely used method in Chinese IR. It has the advantage of requiring smaller inverted index file, achieving faster retrieval speed, and flexibly allowing additional linguistic information to be incorporated into the retrieval system (e.g. synonyms). The most prominent disadvantage of the dictionary based approach is that it requires a large pre-defined lexicon, which normally must be constructed by hand with significant amount of labor and time. Moreover, the lexicon constructed in one language/domain is not portable to another languge/domain, and it is virtually impossible to list all the Chinese words in a dictionary since the set of words is open-ended [6]. An additional shortcoming of the traditional maximum matching method used in the dictionary approach is that a character sequence is always segmented the same way regardless of context, which clearly violates the true nature of Chinese text. In the character based approach, the most prominent advantage is that it does not require pre-defined lexicon. Each character is considered as a basic unit. However, the disadvantages include huge index file, much slower retrieval speed, and the fact that it is difficult to incorporate linguistic information of any kind.

Both the dictionary based and the character based approaches have been successfully applied to Chinese information retrieval in recent work [3, 6, 17]. Overall, the character based approach has tended to yield better retrieval precision [4, 18, 35], and therefore there is an argument about whether word segmentation is necessary at all for Chinese IR. However, some researchers [23, 24] have argued that there exits some inherent difficulties in the character based approach. For example, a modern Chinese IR system should be able to take into account more than just character information, but should also be able to exploit sophisticated techniques such as latent semantic indexing [16]. Kowk [20] proposed using overlapping bi-grams for word segmentation which has large index file due to overlapping. Chen et al [6] proposed using mutual information based approaches for word segmentation which is found to yield better retrieval performance than the bi-gram segmentation. However, the mutual information based statistical approach limits words to be at most two characters. Although most Chinese words are of one or two characters, there are still plenty of words beyond two characters. We want to know if using machine learning for Chinese information retrieval without dictionaries can achieve comparable performance as the mutual information based statistical approach, the character based approach and those that do word segmentation using dictionaries.

In this paper, we propose an EM-based method for Chinese information retrieval which has many of the advantages of all the character based approach, the dictionary based approach and the mutual information based approach, while overcoming many of their shortcomings. We call our approach *self-supervised segmentation*. In *self-supervised segmentation*, no predefined lexicon is required. Instead, all that is needed is a large unsegmented training corpus—which is almost always easy to obtain. We automatically learn a lexicon and lexical distribution from the training corpus by using the EM algorithm [10], and then segment the collections using the Viterbi algorithm [28]. Unlike previous EM word segmentation methods [13], where one lexicon is learnt, we learn two lexicons (for reasons outlined below). Since our segmentation approach is completely unsupervised and language independent, it can be easily adapted to other languages.

We implemented the mutual information based [1], the character based and the dictionary based methods to study and compare the retrieval effectiveness at different word segmentation accuracy using TREC Chinese track data. The best segmentation accuracy of the self-supervised segmentation is around 77%. This is not high enough compared to many supervised learning segmenters [15, 33]. Is this enough for Chinese information retrieval? The relationship between Chinese word segmentation accuracy and information retrieval performance has recently been investigated in the literature. Foo and Li [11] have conducted a series of experiments which suggests that the word segmentation approach does indeed have effect on IR performance. Specifically, they observe that the recognition of words of length two or more can produce better retrieval performance, and the existence of ambiguous words resulting from the word segmentation process can decrease retrieval performance. Similarly, Palmer and Burger [25] observe that accurate segmentation tends to improve retrieval performance. All of this previous research has indicated that there is indeed some sort of correlation between word segmentation accuracy and retrieval performance. However, the nature of this correlation is not well understood.

One reason why the relationship between segmentation and retrieval performance has not been well understood is that previous investigators have not considered using a variety of Chinese word segmenters which exhibit a wide range of segmentation accuracies, from low to high. In order to solve this problem, we employ three families of Chinese word segmentation algorithms from the recent literature. The first technique we employed was the standard maximum matching dictionary based approach. The remaining two algorithms were selected because they can both be altered by

[1] We did not implement the overlapping bi-gram approach because Chen et al [6] found non-overlapping mutual information based approach performed better than the overlapping bi-gram approach.

simple parameter settings to obtain different word segmentation accuracies. Specifically, the second Chinese word segmenter we investigated was the minimum description length algorithm of Teahan et al. [33], and the third was the EM based technique of Peng and Schuurmans [26]. Overall, these segmenters demonstrate word identification accuracies ranging from 44% to 95% on the PH corpus [2, 15, 33].

The rest of the paper is organized as follows. We first introduce the self-supervised word segmentation algorithm in Section 2, and then briefly describe the weighting methods we use in the experiments in Section 3. Following that, we present the experiments we have conducted on the TREC data set in Section 4, 5 and 6. Finally, discussions and conclusions are given in Section 7 and Section 8.

2 Self-supervised Segmentation

For the convenience of non-Chinese speaking reader, we illustrate the Chinese segmentation problems by employing an artificial English segmentation problem where the whitespace has been removed between words and the task is to recover the proper word segmentation from the conjoined text. To understand how one can begin to make progress on this problem, note that in a general word segmentation task where there are no identifying markers between words, one could effectively exploit *known* words to guide the segmentation of unknown words [9]. For example, if the word "*computer*" is already known then upon seeing the text "*computerscience*" it is natural to segment "*science*" as a possible new word. To exploit this observation, we develop an EM based word discovery method that is a variant of standard EM training, but avoids getting trapped in local maxima by keeping two lexicons: a *core* lexicon which contains words that are judged to be trustworthy, and a *candidate* lexicon which contains all other candidate words that are not in the core lexicon. The remainder of this section describes our unsupervised word segmentation algorithm in detail.

2.1 EM segmentation and training

Assume we have a sequence of characters $C = c_1c_2...c_T$ that we wish to segment into chunks $S = s_1s_2...s_M$, where T is the number of characters in the sequence and M is the number of words in the segmentation. Here chunks s_i will be chosen from the core lexicon $V_1 = \{s_i, i = 1, ..., |V_1|\}$ or the candidate lexicon $V_2 = \{s_j, j = 1, ..., |V_2|\}$. If we already have the probability distributions $\theta = \{\theta_i|\theta_i = p(s_i), i = 1, ..., |V_1|\}$ defined over the core lexicon and $\phi = \{\phi_j|\phi_j = p(s_j), j = 1, ..., |V_2|\}$ over the candidate lexicon, then we can recover the most likely segmentation of the sequence $C = c_1c_2...c_T$ into chunks $S = s_1s_2...s_M$ as follows. First, for any given segmentation S of C, we can calculate the joint likelihood of S and C by

$$prob(S, C|\theta, \phi) = \prod_{i=1}^{M_1} \frac{p(s_i)}{2} \prod_{j=1}^{M_2} \frac{p(s_j)}{2} = \frac{1}{2^M} \prod_{k=1}^{M} p(s_k)$$

where M_1 is the number of chunks occurring in the core lexicon, M_2 is the number of chunks occurring in the candidate lexicon, and s_k can come from either lexicon. (Note that each chunk s_k must come from exactly one of the core or candidate lexicons.) Our task is to find the segmentation S^* that achieves the maximum likelihood:

$$\begin{aligned} S^* &= \underset{S}{argmax}\{prob(S|C; \theta, \phi)\} \\ &= \underset{S}{argmax}\{prob(S, C|\theta, \phi)\} \end{aligned} \tag{1}$$

Given a probability distribution defined by θ and ϕ over the lexicon, the Viterbi algorithm [28] can be used to efficiently compute the best segmentation S of character string C. However, *learning* which probabilities to use given a training corpus is the job of the EM algorithm. Following [10], the parameter re-estimation formulas are as follows.

$$\theta_i^{k+1} = \frac{\sum_S \#(s_i, S) \times prob(S, C|\theta^k, \phi^k)}{\sum_{s_i} \sum_S \#(s_i, S) \times prob(S, C|\theta^k, \phi^k)} \tag{2}$$

$$\phi_j^{k+1} = \frac{\sum_S \#(s_j, S) \times prob(S, C|\theta^k, \phi^k)}{\sum_{s_j} \sum_S \#(s_j, S) \times prob(S, C|\theta^k, \phi^k)} \tag{3}$$

where $\#(s_i, S)$ is the number of times s_i occurring the segmentation S. In both cases the denominator is a weighted sum of the number of words in all possible segmentations, the numerator is a normalization constant, and (2) and (3) therefore are weighted frequency counts. Thus, the updates can be efficiently calculated using the forward and backward algorithm.

2.2 Self-supervised lexicon growing

The above algorithm requires two lexicons. Here we describe how they can be constructed automatically. Let us define C_1, C_2 as the training corpus and the validation corpus respectively, and let V_1 and $V2$ be the core candidate lexicons respectively. Initially, V_1 is set to be empty and V_2 is initialized to contain all candidate "words" that are generated from the training corpus by enumerating contiguous character strings of length 1 to some predefined maximum length L. In a first pass, starting from the uniform distribution, EM is used to increase the likelihood of the training corpus C_1. When the training process stabilizes, the M words with highest probability are selected from V_2 and moved to V_1, after which all the probabilities are re-scaled so that V_1 and V_2 each contain half the total probability mass. EM is then run again. The rationale for shifting half of the probability mass to V_1 is that this increases the influence of core words in determining segmentations and allows them to act as more effective guides in processing the training sequence. We call this procedure of successively moving the top M words to V_1 *forward selection*.

Forward selection is repeated until the segmentation accuracy of Viterbi on the validation corpus C_2 leads to a decrease in F-measure (which means we must have included some erroneous words in the core lexicon). After forward selection terminates, M is decremented and we carry out a process of *backward deletion*, where the M words with the lowest probability in V_1 are moved back to V_2, and EM training is successively repeated until F-measure again decreases on the validation corpus C_2 (which means we must have deleted some correct core words). The two procedures of forward selection and backward deletion are alternated, decrementing M at each alternation, until $M \leq 0$; as shown in Fig. 1. As with EM, the outcome of this self-supervised training procedure is the probability distributions over the lexicons that can be used by Viterbi to segment test sequences.

0. Input
Completely unsegmented training corpus C_1
Validation corpus C_2

1. Initialize
V_1 = empty;
V_2 contains all potential words;
$OldFmeasure$ = infinite small;
$bForwardSelection$=true;
set M to a fixed number;

2. Iterate
```
while (M > 0){
  Perform EM based on current V1 and V2 until convergence;

  Calculate NewFmeasure on validation corpus C2;

  if(NewFmeasure < OldFmeasure){
    // change selection direction
    bForwardSelection = ¬bForwardSelection;
    M = M-5;
  }

  OldFmeasure = NewFmeasure;

  //SelectCoreWords(true) do forward selectoin
  //SelectCoreWords(false) do backward selectoin
  SelectCoreWords(bForwardSelection);
}
```

Figure 1: Self-supervised Learning

2.3 Mutual information lexicon pruning

Note that the likelihood is defined by a product of individual chunk probabilities (making the standard assumption that segments are independent), which means that the more chunks a segmentation has, the smaller its likelihood will tend to be. For example, in English, given a character sequence *sizeofthecity* and a uniform distribution over multi-grams, the segmentation *sizeof|thecity* will have higher likelihood than segmentation *size|of|the|city*. Therefore, maximum likelihood will tend to prefer fewer chunks in its segmentation and consequently put large probability on long non-word sequences like *sizeof* and *thecity*. These words are called erroneous agglomerations. Another difficulty is that EM is known to have trouble escaping local maxima in similar sequence models [28]. To eliminate erroneous agglomerations and to pull EM out of the local maxima, we employ a mutual information based criterion to prune the lexicon.

Recall that the mutual information between two random variables is given by

$$MI(X,Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x) \times p(y)} \tag{4}$$

where a large value indicates a strong dependence between X and Y, and zero indicates independence. In our case, we want to test the dependence between two chunks s_1 and s_2. Given a long word, say $s=$*"abcdefghijk"*, we consider splitting it into its most likely two-chunk segmentation, say $s_1=$*"abcd"* and $s_2=$*"efghijk"*. The *pointwise mutual information* [22] between s_1 and s_2 is

$$MI(s_1,s_2) = \log \frac{p(s)}{p(s_1) \times p(s_2)}. \tag{5}$$

To apply this measure to pruning, we set two thresholds $\gamma_1 > \gamma_2$. If the mutual information is higher than the threshold γ_1, we say s_1 and s_2 are strongly correlated, and do not split s. (That is, we do not remove s from the lexicon.) If mutual information is lower than the lower threshold γ_2, we say s_1 and s_2 are independent, and remove s from the lexicon and redistribute its probability to s_1 and s_2. If mutual information is between the two thresholds, we say s_1 and s_2 are weakly correlated, and therefore shift some of the probability from s to s_1 and s_2, by keeping a portion of s's probability for itself (1/3 in our experiments) and distributing the rest of its probability to the smaller chunks, proportional to their probabilities. The idea is to shift the weight of the probability distribution toward shorter words. This splitting process is carried out recursively for s_1 and s_2.

2.4 Summary

We have described our approach to automatically learning a lexicon given a corpus of Chinese text. Note that once the lexicon is established in this manner it can be used to segment the Chinese corpus for use in future IR tasks. That is, our automatic segmentation technique facilitates a dictionary based approach to IR without having to construct the dictionary by hand, making the ease and flexibility with which a retrieval system can be constructed comparable to that of a character based approach. Below we evaluate how well a dictionary based approach built on an automatically constructed lexicon compares to using a hand built dictionary and how it compares to other approaches. And also we investigate whether the self-supervised word segmentation is enough for Chinese information retrieval.

3 Information Retrieval Environment

We conducted our information retrieval experiments using the OKAPI system [18, 29]. To construct a dictionary based IR system we considered two different single unit weighting functions. They are both extended versions of ICF,[2] which include document length and within-document and within-query frequencies as providing further evidence. Adding this evidence makes the term-weighting dependent on the document, which has been shown to be highly beneficial in English text retrieval [29].

3.1 BM25 weighting function

The first function, called BM25 [1], is given as

$$w = \frac{(k_1+1)*tf}{K+tf} * \log\frac{N-n+0.5}{n+0.5} * \frac{(k_3+1)*qtf}{k_3+qtf} \quad \oplus \quad k_2 * nq * \frac{(avdl-dl)}{(avdl+dl)} \tag{6}$$

[2]ICF is defined as $w = \log\frac{N-n+0.5}{n+0.5}$, where N is the number of indexed documents in the collection and n is the number of documents containing a specific term [30].

where N is the number of indexed documents in the collection, n is the number of documents containing a specific term, tf is within-document term frequency, qtf is within-query term frequency, dl is the length of the document, $avdl$ is the average document length, nq is the number of query terms, the k_is are tuning constants (which depend on the database and possibly on the nature of the queries and are empirically determined), K equals to $k_1 * ((1 - b) + b * dl/avdl)$, and $\oplus$ indicates that its following component is added only once per document, rather than for each term. The component:

$$k_2 * nq * \frac{(avdl - dl)}{(avdl + dl)}$$

is called *correction factor* which was designed to take into account the length of a document. The value of the correction factor decreases with dl, from a maximum as $dl \to 0$, at which $dl = avdl$, and to a minimum as $dl \to \infty$, as shown in Figure 2.

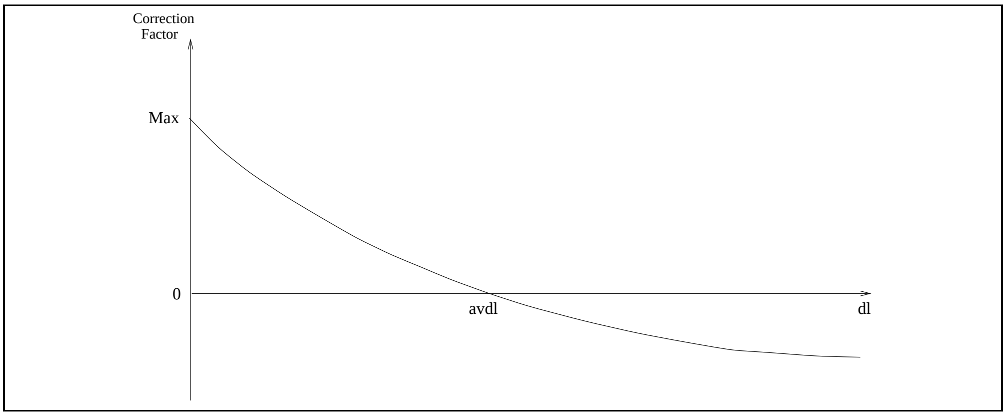

Figure 2: Curve for BM25's correction factor

This design of the correction factor assumes that, the shorter the document is, the more value the correction factor should have, i.e., the more possible the document is relevant.

In our experiments, the values of k_1, k_2, k_3 and b in the BM25 function are set to be 2.0, 0, 5.0 and 0.75 respectively. Note that we set k_2 to be 0, which means that the correction factor is not considered. The setting of these numbers was obtained from previous extensive experiments for English text retrieval and from initial experiments for Chinese text retrieval. For example, we found that the system produces better results if we set k_2 to be 0.

3.2 BM26 weighting function

The fact that better performance of BM25 is achieved when k_2 is set to be zero (i.e., the correction factor is ignored) indicates that the correction factor in BM25 is not designed properly. To tackle this problem, we propose an enhanced version of BM25, referred to as BM26, which is based on the following two assumptions: (1) overly short documents are not relevant; (2) the function curve for the correction factor should be consistent with the distribution of relevant documents in the standard text collection provided by the TREC conferences. More details can be found in [18]. BM26 is defined as follows:

$$w = \frac{(k_1+1)*tf}{K+tf} * \log\frac{N-n+0.5}{n+0.5} * \frac{(k_3+1)*qtf}{k_3+qtf} \quad \oplus \quad k_d * y \tag{7}$$

where all the parameters have the same meaning as in BM25 except k_d is a tuning constant and

$$y = \begin{cases} \ln(\frac{dl}{avdl}) + \ln(x_1) & \text{if } 0 < dl <= rel\_avdl; \\ (\ln(\frac{rel\_avdl}{avdl}) + \ln(x_1))(1 - \frac{dl - rel\_avdl}{x_2 * avdl - rel\_avdl}) & \text{if } rel\_avdl < dl < \infty. \end{cases} \tag{8}$$

in which dl is the length of the document, $avdl$ is the average document length, rel_avdl is the average relevant document length calculated from previous queries based on the same collection of documents, x_1 and x_2 are two parameters to be set. The parameter k_d in BM26 is set to have different values in our experiments. When k_d is 0, BM26 becomes BM25 since we set the parameter k_2 in BM25 to be 0 in our experiments.

The difference between BM26 and BM25 is in the y bit of the correction factor. In BM26, y will reach a maximum as $dl \to rel\_avdl$, and a minimum as $dl \to 0$ (or $dl \to \infty$). This relation is shown in Figure 3. In our experiments, x_1 and x_2 were set to 3 and 26 respectively.

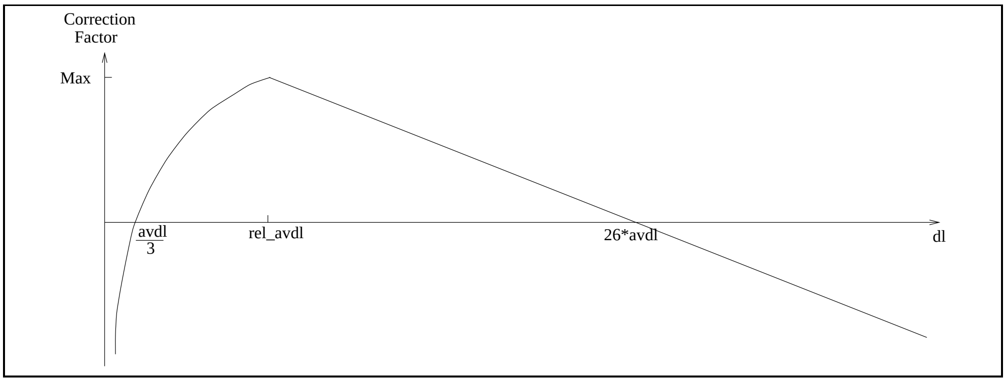

Figure 3: Curve for the new correction factor

4 Experiments Setup

4.1 Data sets

The test collection we use is from TREC-5 and TREC-6 (Text REtrieval Conferences) [34]. It contains 164,768 documents and consists of 139,801 articles selected from the *People's Daily* newspaper and 24,988 articles selected from the *Xinhua newswire*, with 0 bytes as the minimum file size, 294,056 bytes as the maximum size and 891 bytes as the average file size. 54 Chinese topics (28 for TREC-5 and 26 for TREC-6) were used in our experiments. The document collection used in TREC-6 Chinese track was identical to the one used in TREC-5. All the original articles were tagged using SGML. Chinese characters inside these articles were encoded using the GB (GuoBiao) coding scheme.

Our segmenter is trained on the training set C_1 with validation set C_2, where C_1 is 10M data which contains a subset of one year of *People's Daily* news service stories (www.snweb.com) and C_2 is a randomly selected 2000 sentence subset of the Chinese Treebank from LDC[3] which is has been segmented by hand. The parameter k_d in Table 1 and Table 2 is a tuning constant in BM26 (see equation 7). When k_d is 0, BM26 becomes BM25 in our experiments.

[3]http://www.ldc.upenn.edu/ctb

4.2 Measuring segmentation accuracy

We evaluated segmentation accuracy on the Mandarin Chinese corpus, PH, due to Guo Jin [2, 15, 33]. This corpus contains one million words of *segmented* Chinese text from newspaper stories of the *Xinhua news agency of the People's Republic of China* published between January 1990 and March 1991.

To make the definitions precise, first define the original *segmented* test corpus to be S. We then collapse all the whitespace between words to make a second unsegmented corpus U, and then use the segmenter to recover an estimate $\hat{S}$ of the original segmented corpus. We measure the segmentation accuracy by precision, recall, and F-measure on detecting correct words. Here, a word is considered to be correctly recovered if and only if [25]

1. a boundary is correctly placed in front of the first character of the word
2. a boundary is correctly placed at the end of the last character of the word
3. and there is no boundary between the first and last character of the word.

Let N_1 denote the number of words in S, let N_2 denote the number of words in the estimated segmentation $\hat{S}$, and let N_3 denote the number of words correctly recovered. Then the precision, recall and F measures are defined

$$\text{precision: } p = \frac{N_3}{N_2}$$
$$\text{recall: } r = \frac{N_3}{N_1}$$
$$\text{F-measure: } F = \frac{2 \times p \times r}{p+r}$$

In this paper, we only report the accuracy in F-measure, which is a comprehensive measure that combines precision and the recall.

4.3 Measuring retrieval performance

In our experiments, the TREC relevance judgments for each topic came from the human assessors of the National Institute of Standards and Technology (NIST). Statistical evaluation was done by means of the TREC evaluation program. Several measures are used to evaluate the retrieval result which is an ordered set of retrieved documents. The measures include Average Precision: average precision over all 11 recall points (0.0, 0.1, 0.2,..., 1.0); R Precision: precision after the number of documents retrieved is equal to the number of known relevant documents for a query; and Precision at 100 docs: precision after 100 documents have been retrieved. Detailed descriptions of these measures can be found in [34].

5 Effects of Self-supervised Segmentation on Chinese IR

In this section, we first investigate the influence of maximum word length L, and then use the optimal length $L = 3$ for further comparison with other term extraction methods.

5.1 Influence of maximum word length L

According to *Frequency dictionary of modern Chinese,1980*, (see [12]), among the top 9000 most frequent words: 26.7% are uni-grams, 69.8% are bi-grams, 2.7% are tri-grams, 0.007% are 4-grams, and 0.002% are 5-grams. So most Chinese words are within 4 characters long. In our training algorithm, we set a maximum word length constraint L. To evaluate the effect of L, we experimented with L set to 2, 3 or 4. Table 1 and Table 2 shows the *average precision/R-precision*— results on TREC-5 and TREC-6 data sets.

k_d	$L = 2$	$L = 3$	$L = 4$
0	0.3264/0.3639	0.3422/0.3857	0.3246/0.3550
2	0.3326/0.3707	0.3504/0.3901	0.3319/0.3585
6	0.3430/0.3819	0.3613/0.3981	0.3416/0.3692
8	0.3453/0.3849	0.3641/0.3996	0.3419/0.3692
10	0.3450/0.3832	0.3661/0.4027	0.3422/0.3733
15	0.3403/0.3773	0.3601/0.3923	0.3412/0.3747
20	0.3320/0.3744	0.3536/0.3836	0.3368/0.3737
50	0.2756/0.3271	0.2982/0.3444	0.2865/0.3316

Table 1: Influence of L on TREC-5 using different weighting methods

k_d	$L = 2$	$L = 3$	$L = 4$
0	0.4363/0.4572	0.4660/0.4849	0.4531/0.4652
2	0.4459/0.4579	0.4754/0.4897	0.4632/0.4718
6	0.4595/0.4693	0.4906/0.4949	0.4781/0.4792
8	0.4635/0.4687	0.4950/0.4939	0.4822/0.4822
10	0.4661/0.4667	0.4968/0.4973	0.4841/0.4839
15	0.4659/0.4685	0.4970/0.5001	0.4820/0.4799
20	0.4603/0.4624	0.4928/0.4976	0.4758/0.4798
50	0.3990/0.4244	0.4186/0.4566	0.4101/0.4459

Table 2: Influence of L on TREC-6 using different weighting methods

Here one can see that the best results were achieved when $L = 3$. An explanation of this observation is that although more than 96% of Chinese words are within 2 characters long (which is the reason why the bi-gram indexing works well [6, 20, 21]), there are still many words that are longer than 2 characters, and breaking them down reduces the precision. Obviously, 3-grams and 4-grams will have less ambiguity than bi-grams. However, 4-grams will include many more combinations than 3-grams, which reduces the reliability with which their occurrence probabilities can be estimated relative to 3-grams. Therefore, statistical over-fitting explains why $L = 4$ yields worse performance than $L = 3$.

One can also see that the different weighting methods have a large effect on performance. Using the BM26 weighting function makes a significant positive contribution to the quality of retrieval compared to using BM25. By setting k_d to 10 for TREC-5 and 15 for TREC-6, the best retrieval performance in terms of average precision can be obtained for all the EM-based word segmentation methods. Here we find that the performance on TREC-6 data set is much better than on TREC-5.

5.2 Comparisons with other term extraction methods

The first extraction method uses a hand built dictionary of words, compound words and phrases to index the texts. We refer to this method as the dictionary-based approach. The dictionary we use in the experiments contains 69,353 Chinese words and phrases [18]. The second extraction method we compare to is a standard character based approach, in which documents are indexed by the single Chinese characters that appear in the text. However, we would like to emphasize that using single characters for indexing does not imply that we use single characters as keywords in search. For the character based approach we used in our experiments, a search could be conducted for any multi-character word or phrase identified at search time, whether or not this word or phrase appeared in the dictionary. Therefore the experimental results we report for the character based approach use the character based method for indexing and use a dictionary based method for topic processing. The third extraction method we compare to is a mutual information based statistical approach [6], in which one first collects occurrence frequencies for uni-grams and bi-grams from the corpus, and second uses a mutual information based criterion to segment Chinese text [32] and then uses these segmented terms to index the documents.

The motivation for proposing the self-supervised segmentation method is to incorporate the advantages of all the character based, the dictionary based, and mutual information based approaches, while overcoming their shortcomings. For the TREC-5 comparison, we choose one best run in terms of Average Precision from the EM based method, one best run from mutual information based method, one best run using only single words as retrieval keywords from the dictionary based method, and one run from the character based method where only BM25 was used. For the TREC-6 comparison, we also compare one best run in terms of Average Precision from the EM based method, one best run from mutual information based method, one best run using only single word from the dictionary based method, and one run of the character based method. No phrase weighting has been used in the above six selected runs.

The topic processing method we used in the experiments is automatic and simple. First we rank the words extracted from topics by the values of their weights multiplied by the within-query frequencies, and then choose the top 19 [4] ranking as retrieval keywords. The detailed description can be found in [18]. The segmentation method used in topic processing is consistent with that used in document processing except for the character based approach. That is, when EM method, dictionary based method or mutual information based method is used for document processing, topic

[4] We chose the number of 19 because it gave the best result among the three numbers we tried in our experiments. There may be a better way to do it. But it does not affect our results.

processing also uses EM method, dictionary based method or mutual information based method respectively; However, the *character based method* we use in the paper is mixture of a pure character based method and dictionary based method, i.e., it uses the character based method for indexing but uses a dictionary based method for topic processing. This hybrid system yields far superior results to the pure character based method, which we did not present in this paper.

The size of the character based index built for Chinese TREC collection is about one gigabyte, which is about twice the size of the document collection. The size is doubled since because it stores positional information about each character's occurrence. The index for the EM-based methods are at the same level as the dictionary based method. The sizes of the index files for the dictionary based, EM based and mutual information (MI) based methods are given in Table 3. In terms of retrieval time, the EM based methods are at the same level as the dictionary based methods. MI based method is a little bit slower than the EM based and the dictionary based. All of these three methods are about three times faster than the character based approach.

	character	dictionary	EM2	EM3	EM4	MI
index	139,734	1,691,575	2,087,509	1,431,104	1,868,539	11,723,406
invert	1,077,393,536	678,257,616	648,472,816	667,634,000	677,900,752	730,013,936

Table 3: Size of Index Files (unit is byte)

We show the experimental results of the three methods on TREC-5 data in Tables 4 and 5, and the results on TREC-6 data sets in Tables 6 and 7. Here the *relevant retrieved* is the number of relevant documents retrieved out of the 2182 or 2958 documents in the collection for TREC-5 or TREC-6 respectively. We set the dictionary based method as the baseline.

On TREC-5 data, we find the EM based segmentation gives a 5.57% improvement in average accuracy over the dictionary based method, but it does a little worse than the character based method. In terms of R-precision, the EM based method yields better performance than both the character based and dictionary based methods. On TREC-6 data, the EM based method yields slightly worse results than both the dictionary based and the character based methods. On both TREC-5 and TREC-6 datasets, the EM based method produces better results than mutual

Recall	character	dictionary	EM-based	MI-based
0.00	0.7764	0.7681	0.7358	0.7473
0.10	0.6243	0.6261	0.6249	0.6263
0.20	0.5507	0.5075	0.5429	0.5528
0.30	0.4987	0.4531	0.4998	0.4976
0.40	0.4458	0.4034	0.4406	0.4289
0.50	0.4247	0.3558	0.3954	0.3935
0.60	0.3591	0.3180	0.3343	0.3415
0.70	0.2711	0.2463	0.2803	0.2569
0.80	0.2236	0.1760	0.1859	0.1818
0.90	0.1408	0.1154	0.1082	0.0993
1.00	0.0266	0.0082	0.0157	0.0275
average precision	0.3795	0.3468	0.3661	0.3627
improvement	9.43%	baseline	5.57%	4.58%
relevant retrieved	1986	1883	1939	1893

Table 4: TREC-5: comparing precision at specified recall rate

information based method.

Previous research [3, 18] has suggested that exact segmentation may not be necessary in the IR task. The results that we have obtained here also support this point: although the segmentation accuracy of the EM based method is not necessarily very high, it achieves comparable performance to a hand built dictionary approach.

6 Relationship between Segmentation Accuracy and Retrieval Performance

The advantages of using EM for word segmentation has in fact been considered in previous research [13, 26]. However, due to the low segmentation accuracies [5] these methods obtain, they still do not tend to be regarded as good methods for Chinese IR. Nevertheless, the results presented so far suggest that this need not be the case. In fact, we have shown that state of the art Chinese

[5]The segmentation accuracies of EM based and mutual information based methods whose retrieval performances shown in Tables 4, 5, 6 and 7 are 77% and 61% respectively.

R	character	dictionary	EM-based	MI-based
5	0.5571	0.5429	0.5500	0.5286
10	0.5429	0.5143	0.5107	0.5214
15	0.4881	0.4810	0.4881	0.4952
20	0.4732	0.4732	0.4857	0.4946
30	0.4369	0.4321	0.4595	0.4571
100	0.3189	0.3150	0.3243	0.3139
200	0.2380	0.2302	0.2418	0.2329
500	0.1272	0.1216	0.1269	0.1208
1000	0.0709	0.0672	0.0693	0.0676
R-Precision	0.3963	0.3863	0.4027	0.3988
improvement	2.59%	baseline	4.25%	3.24%

Table 5: TREC-5: comparing R-precision

word segmentation, limited as it is in terms of accuracy, still facilitates retrieval performance that it comparable with the best of the current Chinese IR systems. In the following, we investigate the relationship between word segmentation accuracy and retrieval performance in Chinese IR.

6.1 Word Segmentation Algorithms

Beside the self-supervised word segmentation method and mutual information based statistical method we describe above, we take another two methods for segmentation. One is the dictionary based maximum matching segmentation and another is a compression based methods. The segmentation accuracy of the self-supervised word segmentation can be controlled by varying the number of training iterations and by applying lexicon pruning techniques. For the compression based segmentation method, we can control the segmentation accuracy by varying its parameters. For the sake of completeness we briefly describe the dictionary based word segmentation and the compression based methods here.

6.1.1 Dictionary based word segmentation

The dictionary based approach is the most popular Chinese word segmentation method. The idea is to use a hand built dictionary of words, compound words, and phrases to index the text. In our

Recall	character	dictionary	EM-based	MI-based
0.00	0.9604	0.9558	0.9110	0.9120
0.10	0.8144	0.8217	0.8021	0.8105
0.20	0.7396	0.7351	0.7482	0.7371
0.30	0.6957	0.6586	0.6580	0.6564
0.40	0.6662	0.5958	0.5935	0.5921
0.50	0.5937	0.5507	0.5267	0.5402
0.60	0.5195	0.4708	0.4608	0.4659
0.70	0.4284	0.3844	0.3779	0.3683
0.80	0.3224	0.2892	0.2738	0.2755
0.90	0.1966	0.1485	0.1846	0.1751
1.00	0.0239	0.0023	0.0297	0.0050
average precision	0.5348	0.5044	0.4970	0.4966
improvement	6.03%	baseline	-1.47%	-1.55%
relevant retrieved	2569	2536	2540	2532

Table 6: TREC-6: comparing precision at specified recall rate

experiments we used the longest forward match method in which text is scanned sequentially and the longest matching word from the dictionary is taken at each successive location. The longest matched strings are then taken as indexing tokens and shorter tokens within the longest matched strings are discarded. In our experiments we used two different dictionaries. The first is the Chinese dictionary used by Gey et al. [14], which includes 137,659 entries. The second is the Chinese dictionary used by Beaulieu et al. [1], which contains 69,353 words and phrases.

6.1.2 Compression based word segmentation

The PPM word segmentation algorithm of Teahan et al. [33] is based on the text compression method of Cleary and Witten [8]. PPM learns an n-gram language model by supervised training on a given set of hand segmented Chinese text. To segment a new sentence, PPM seeks the segmentation which gives the best compression using the learned model. This has been proven to be a highly accurate segmenter [33]. Its quality is affected both by the amount of training data and by the order of the n-gram model. By controlling the amount of training data and the order of language model we can control the resulting word segmentation accuracy.

R	character	dictionary	EM-based	MI-based
5	0.7615	0.8077	0.7538	0.7846
10	0.7731	0.7885	0.7846	0.7808
15	0.7615	0.7667	0.7564	0.7641
20	0.7385	0.7404	0.7346	0.7423
30	0.6910	0.6936	0.6833	0.6949
100	0.5035	0.4923	0.4831	0.4758
200	0.3615	0.3521	0.3406	0.3354
500	0.1832	0.1808	0.1798	0.1765
1000	0.0988	0.0975	0.0977	0.0974
R-Precision	0.5404	0.5055	0.5001	0.4955
improvement	6.90%	baseline	-1.07%	-1.99%

Table 7: TREC-6: comparing R-precision

6.2 Segmentation accuracy control

By using the forward maximum matching segmentation strategy with the two dictionaries [6, 18], Berkeley and City, we obtain the segmentation accuracy of 71% and 85% respectively. For the PPM algorithm, by controlling the order of the n-gram language model used (specifically, order 2 and order 3) we obtain segmenters that achieve 90% and 95% word recognition accuracy respectively. The segmentation accuracy for the mutual information based approach is 61%. Finally, for the self-supervised learning technique, by controlling the number of EM iterations and altering the lexicon pruning strategy we obtain word segmentation accuracies of 44%, 49%, 53%, 56%, 59%, 61%, 70%, 75%, and 77%. Thus, overall we obtain 13 different segmenters that achieve segmentation accuracies of 44%, 49%, 53%, 56%, 59%, 61%, 70%, 71%, 75%, 77%, 85%, 90%, and 95%.

6.3 Experimental results

Now, given the 13 different segmenters, we conducted extensive experiments on the TREC data sets using different information retrieval methods (achieved by tuning the k_d constant in the term weighting function described in Section 3).

Table 8 shows the *average precision* and *R-precision* results obtained on the TREC-5 and TREC-

6 queries when basing retrieval on word segmentations at 12 different accuracies, for a single retrieval method, $k_d = 10$. To illustrate the results graphically, we re-plot this data in Figure 4, in which the x-axis is the segmentation performance and the y-axis is the retrieval performance.

seg. accuracy	TREC-5	TREC-6
44%(EM)	0.2231/0.2843	0.3424/0.3930
49%(EM)	0.2647/0.3259	0.3848/0.4201
53%(EM)	0.2999/0.3376	0.4492/0.4801
56%(EM)	0.3056/0.3462	0.4473/0.4727
59%(EM)	0.3097/0.3533	0.4740/0.4960
61%(MI)	0.3627/0.3988	0.4953/0.4942
70%(EM)	0.3721/0.3988	0.5044/0.5072
71%(Berkeley)	0.3656/0.4088	0.5133/0.5116
75%(EM)	0.3652/0.4000	0.4987/0.5097
77%(EM)	0.3661/0.4027	0.4968/0.4973
85%(City)	0.3468/0.3863	0.5044/0.5055
90%(PPM)	0.3213/0.3663	0.4983/0.5008
95%(PPM)	0.3189/0.3669	0.4867/0.4933

Table 8: Average precision and R-precision results on TREC queries when $k_d = 10$.

Clearly these curves demonstrate a nonmonotonic relationship between retrieval performance (on the both P-precision and the R-precision) and segmentation accuracy. In fact, the curves show a clear uni-modal shape, where for segmentation accuracies 44% to 70% the retrieval performance increases steadily, but then plateaus for segmentation accuracies between 70% and 77%, and finally decreases slightly when the segmentation accuracy increase to 85%,90% and 95%. This phenomenon is robustly observed as we alter the retrieval method by setting $k_d = 0, 6, 8, 15, 20, 50$, as shown in Figures 4 to 10 respectively.

The highest segmentation accuracy that the self-supervised segmentation method can achieve is 77%. However, the best retrieval performance for this method can be obtained by setting the segmentation accuracy to 70%.

To give a more detailed picture of the results, Figures 11 and 12 we illustrate the full *precision-recall* curves for $k_d = 10$ at each of the 12 segmentation accuracies, for TREC-5 and TREC-6 queries respectively. In these figures, the 44%, 49% segmentations are marked with stars, the 53%, 56%,

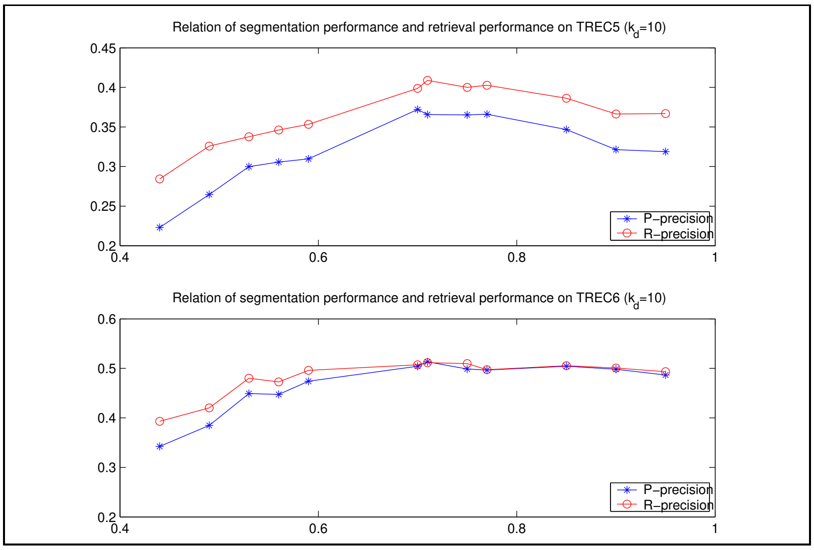

Figure 4: Retrieval performance (y-axis) versus segmentation accuracy (x-axis) for $k_d = 10$.

59% segmentations are marked with circles, the 70%, 71%, 75%, 77% segmentations are marked with diamonds, the 85% segmentation is marked with hexagrams, and the 90% and 95% segmentations are marked with triangles. We can see that the curves with the diamonds are above the others, while the curves with stars are at the lowest positions.

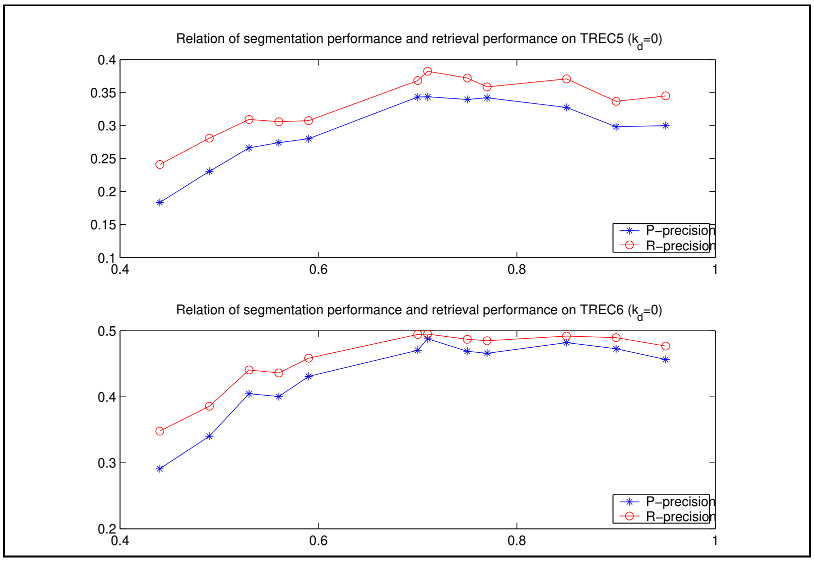

Figure 5: Retrieval performance (y-axis) versus segmentation accuracy (x-axis) for $k_d = 0$.

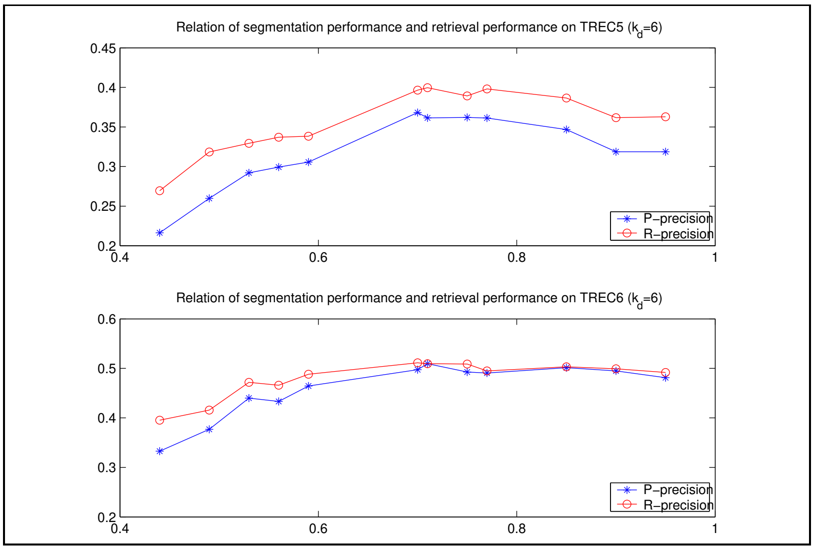

Figure 6: Retrieval performance (y-axis) versus segmentation accuracy (x-axis) for $k_d = 6$.

7 Related Work and Discussions

In this section, we relate our self-supervised word segmentation to previous research and analyze the experimental results reported in Section 5 and 6.

7.1 Self-supervised word segmentation

Our self-supervised word segmentation is a variant of EM based approach. EM as an iterative optimization approach has been extensively used many applications. In Chinese word segmentation, Ge, et al. [13] uses a soft counting version of EM to learn how to segment Chinese. To augment the influence of important words, Ge et al. [13] shifts probability mass to likely words by soft counting. In our model, we shift half of the probability space to the core words by dividing the lexicon to two parts. Sproat and Shih [32] uses a more complicated Hidden Markov Model (HMM) model that includes special recognizers for Chinese names and a component for morphologically derived words. As pointed out in [13], standard EM segmentation can be thought of as a zero order HMM.

In our self-supervised learning, we used mutual information for lexicon pruning. In the literature of Chinese word segmentation, other researchers have considered using mutual information to build a lexicon. For example, Sproat and Shi [32] uses mutual information to build a lexicon, but only deals

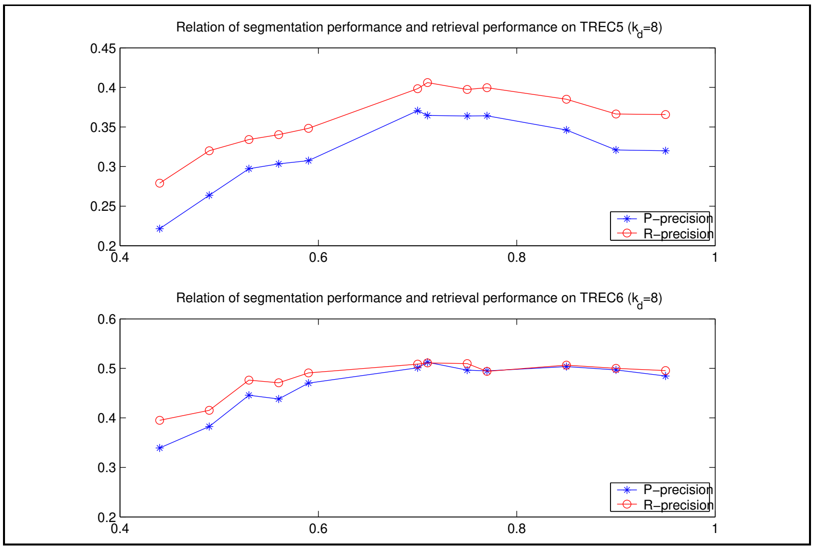

Figure 7: Retrieval performance (y-axis) versus segmentation accuracy (x-axis) for $k_d = 8$.

with words of up to 2 characters. Chang and Su [5], Ponte and Croft [27] use mutual information and context information to build a lexicon based on the statistics directly obtained from the training corpus. By contrast, we are using mutual information to prune a given lexicon. That is, instead of building a lexicon from scratch, we first add all possible words and then use mutual information to prune away illegal words after training with EM. Hence the statistics we use for calculating mutual information are more reliable than those directly obtained from corpus by frequency counting. Another difference is that we are using a probabilistic splitting scheme that sometimes just shifts probability between words, instead of completely discarding words.

When considering applying word segmentation to Chinese IR, our work is most related to the work of Chen et al. [6]. There too it was proposed that Chinese IR could be conducted without using a dictionary. In their method, Chen collects occurrence frequencies from the corpus and limits the word length to at most 2 characters in order to use mutual information for segmenting Chinese text. Similarly, we also use the frequencies from the corpus and also use mutual information during the process. However, our work differs from theirs in many respects. First we do not limit the word length to 2 characters. The maximum word length could be set arbitrarily to suit the application. In fact, our best results are achieved when $L = 3$. Second, the statistics we used were optimized by an iterative EM process, which is guaranteed to achieve at least a local optimum. This

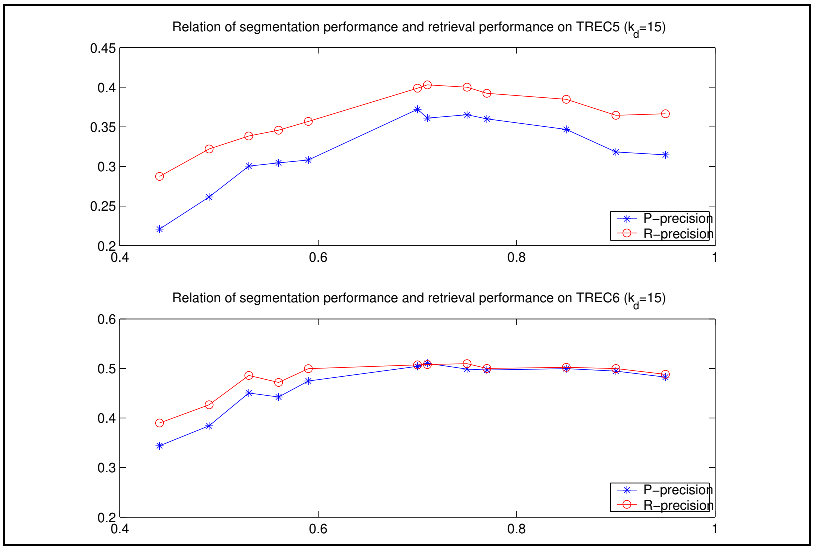

Figure 8: Retrieval performance (y-axis) versus segmentation accuracy (x-axis) for $k_d = 15$.

approach should be more reliable than the statistics direct from the corpus. This is confirmed by the experimental results shown in Tables 4, 5, 6 and 7. Our work is also strongly related to Kwok [20, 21] where he proposed to use a mixture of uni-gram and bi-gram, or mixture of uni-gram and short words to represent text. The mixture was done with some predefined rules. In our work, this mixture is incorporated implicitly because our EM-based algorithm can segment text into words ranging from 1 character to L characters, where L is the predefined maximum word length.

The disadvantage of the EM-based approach is that the training procedure is time-consuming. In our experiments, it takes 2 days to learn the lexicon on a PC (750M HZ Pentium IV CPU, 2G memory). However, this should not be a problem because this procedure is only done one time. Once the lexicon is constructed, Viterbi algorithm can be used to segment a new sentence in linear time to the length of the sentence [28].

7.2 Relationship between segmentation accuracy and IR performance

The observations of the relationship were surprising to us at first, although they suggest that there is an interesting phenomenon at work. To attempt to identify the underlying cause, we break the explanation into two parts: one for the first part of the curves where retrieval performance increases with increasing segmentation accuracy, and a second effect for the region where retrieval

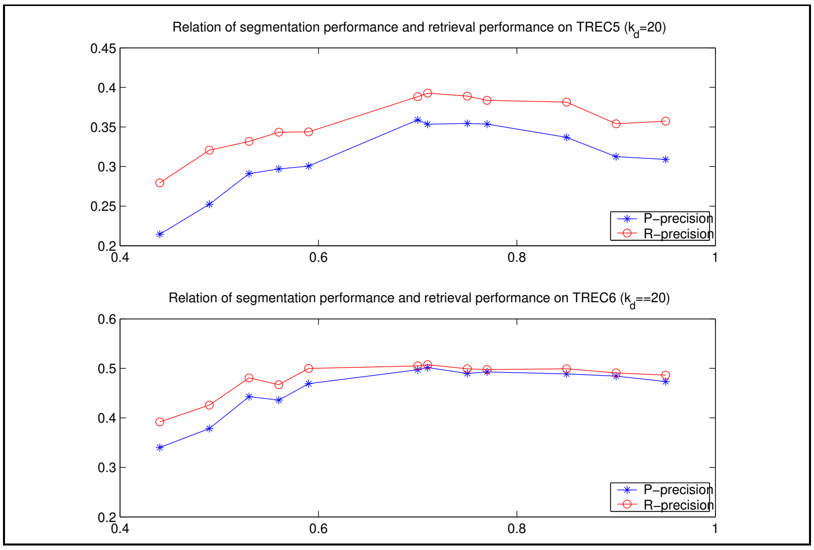

Figure 9: Retrieval performance (y-axis) versus segmentation accuracy (x-axis) for $k_d = 20$.

performance plateaus and eventually decreases with increasing segmentation accuracy.

The first part of these performance curves seems easy to explain. At low segmentation accuracies the segmented tokens do not correspond to meaningful linguistic terms, such as words, which hampers retrieval performance because the term weighting procedure is comparing arbitrary tokens to the query. However, as segmentation accuracy improves, the tokens behave more like true words and the retrieval engine begins to behave more conventionally.

However, after a point, when the second regime is reached, retrieval performance no longer increases with improved segmentation accuracy, and eventually begins to decrease. One possible explanation for this which we have found is that a weak word segmenter accidentally *breaks* compound words into smaller constituents, and this, surprisingly yields a beneficial effect for Chinese information retrieval.

For example, one of the test queries, Topic 34, is about the impact of droughts in various parts of China. Retrieval based on the EM-70% segmenter retrieved 84 of 95 relevant documents in the collection, whereas retrieval based on the PPM-95% segmenter retrieved only 52 relevant documents. In fact, only 2 relevant documents were missed by EM-70% but retrieved by PPM-95%, whereas 34 documents retrieved by EM-70% and not by PPM-95%. In investigating this phenomenon, one finds that the performance drop appears to be due to the inherent nature of written Chinese. That

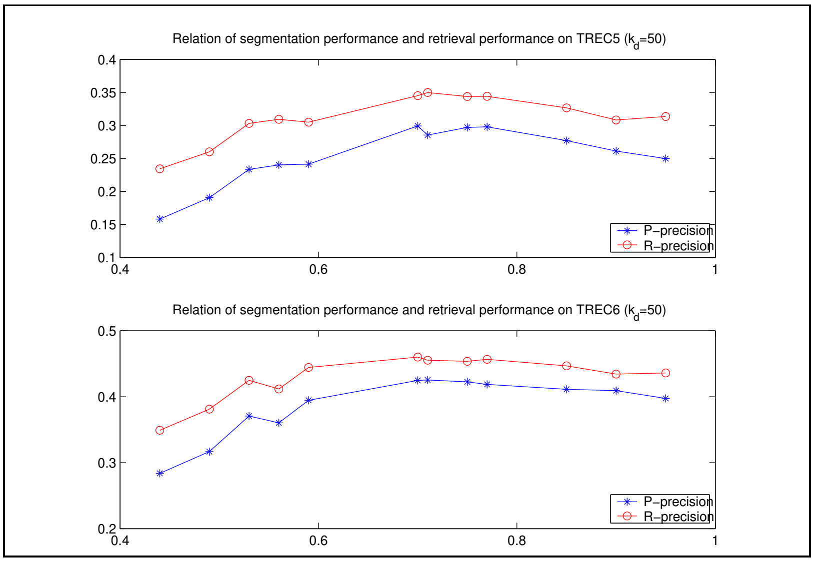

Figure 10: Retrieval performance (y-axis) versus segmentation accuracy (x-axis) for $k_d = 50$.

is, in written Chinese many words can often legally be represented their subparts. For example, 农作物(agriculture plants) is sometimes represented as 作物(plants). So for example in Topic 34, the PPM-95% segmenter correctly segments 旱灾 as 旱灾(drought disaster) and 农作物 correctly as 农作物 (agriculture plants), whereas the EM-70% segmenter incorrectly segments 旱灾 as 旱(drought) and 灾(disaster), and incorrectly segments 农作物 as 农(agriculture) and 作物(plants). However, by inspecting the relevant documents for Topic 34, we find that there are many Chinese character strings in these documents that are closely related to the correctly segmented word 旱灾(drought disaster). These alternative words are 春旱，旱魔，受旱，干旱，抗旱，旱区 etc. For example, in the relevant document "pd9105-832", which is ranked 60th by EM-70% and 823rd by PPM-95%, the correctly segmented word 旱灾 does not appear at all. Consequently, the correct segmentation for 旱灾 by PPM-95% leads to a much weaker match than the incorrect segmentation of EM-70%. Here EM-70% segments 旱灾 into 旱 and 灾 , which is not regarded as a correct segmentation. However, there are many matches between the topic and relevant documents which contain only 旱. This same phenomenon happens with the query word 农作物 since many documents only contain the fragment 作物 instead of 农作物, and these documents are all missed by PPM-95% but captured by EM-70%.

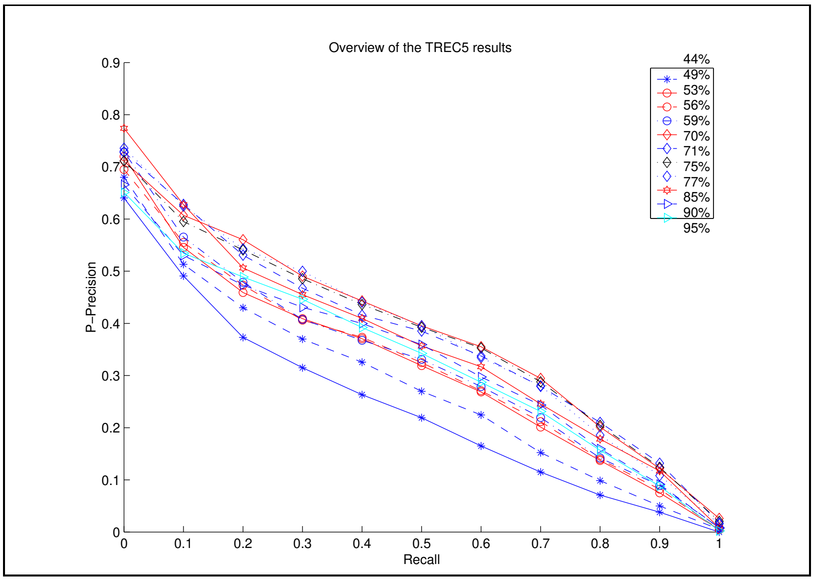

Figure 11: TREC5 precision-recall comprehensive view at $k_d = 10$

8 Conclusions and Future work

We have proposed a novel EM based method for text segmentation for the purposes of Chinese information retrieval, and presented experimental results on recent TREC data. Our method has the advantages of all the character based, dictionary based and mutual information based methods, while overcoming many of their shortcomings. Although our EM based segmentation method does not yield completely accurate segmentations by itself, it nevertheless performs well as a basis for Chinese IR. We achieve retrieval performance that is comparable (and sometimes even better) than the manual dictionary based, character based and mutual information based statistical methods. Our results demonstrate the machine learning techniques can be successfully applied to text segmentation and information retrieval to build adaptable systems.

We also observe that the relationship between the word segmentation and the retrieval performance is non-monotonic. The retrieval performance first increases as the word segmentation accuracy increases, but it saturates after some point and even decrease when the segmentation accuracy is too high. Our self-supervised word segmentation method is enough to produce comparable retrieval performance, although it can produce the best result on TREC-5 data by setting the parameters that lead to the segmentation accuracy of 70%.

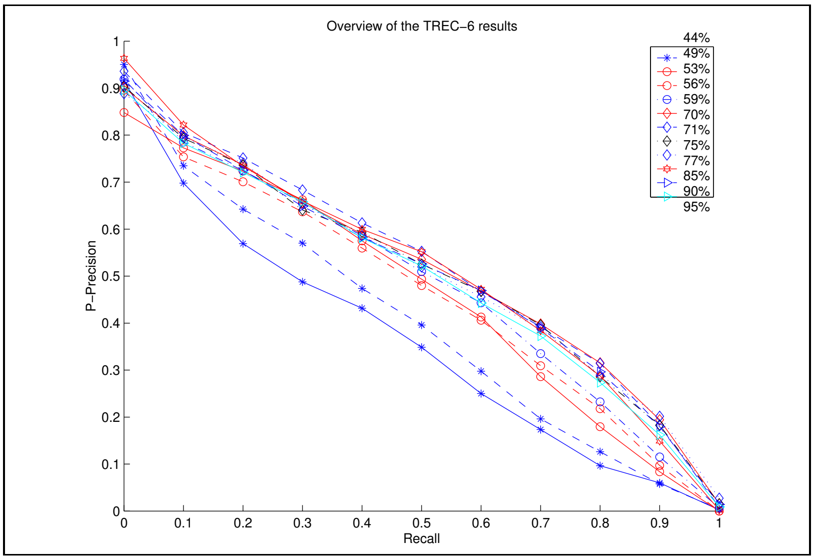

Figure 12: TREC6 precision-recall comprehensive view at $k_d = 10$

Although straightforward, these observations suggest a different trajectory for future research on Chinese information retrieval. Instead of focusing on achieving accurate word segmentation, we should pay more attention to issues such as keyword weighting [18] and query keyword extraction [7]. Our current keyword extraction method is very rough, and we are investigating more sophisticated extraction methods such as those used in [6, 7]. Also, we find that the weak unsupervised segmentation method based yields better Chinese retrieval performance than the other approaches, which suggests a promising new avenue to apply machine learning techniques to IR [31]. Of course, despite these results we expect highly accurate word segmentation to still play an important role in other Chinese information processing tasks such as information extraction and machine translation. This suggests that some different evaluation standards for Chinese word segmentation should be given to different NLP applications.

Finally, a successful Chinese IR system should employ many of the same techniques that are effective for English IR, such as latent semantic indexing [16] and query expansion. Combining our unsupervised Chinese word segmentation technique with latent semantic indexing has the potential to make an adaptable and high performance Chinese IR system. This is ongoing work.

9 Acknowledgments

Research supported by Bell University Labs, MITACS and NSERC. We sincerely thank Dr. William Teahan for supplying us the PPM segmenters.

References

[1] M. Beaulieu, M. Gatford, X. Huang, S. Robertson, S. Walker, and P. Williams. Okapi at TREC-5. In *D.K.Harman (ed): Proceedings of TREC-5*, pages 143–166, 1997.

[2] M. Brent and X. Tao. Chinese Text Segmentation With MBDP-1: Making the Most of Training Corpora. In *Proceedings of ACL2001*, France, 2001.

[3] C. Buckley, A. Singhal, and M. Mitra. Using Query Zoning and Correlation within SMART: TREC-5. In *Proceedings of TREC-5*, pages 105–118, 1997.

[4] C. Buckley, J. Walz, M. Mitra, and C. Cardie. Using Clustering and SuperConcepts Within SMART: TREC-6. In *Proceedings of TREC-6*, pages 107–124, 1998.

[5] J.-S. Chang and K.-Y. Su. An Unsupervised Iterative Method for Chinese New Lexicon Extraction. *International Journal of Computational Linguistics & Chinese Language Processing*, 1997.

[6] A. Chen, J. He, L. Xu, F. C. Gey, and J. Meggs. Chinese Text Retrieval Without Using a Dictionary. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49. ACM, 1997.

[7] L.-F. Chien, T.-I. Huang, and M-C. Chien. Pat-tree-based Keyword Extraction for Chinese Information Retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–58. ACM, 1997.

[8] J. Cleary and I. Witten. Data compression using adaptive coding and partial string matching. *IEEE Trans on Communications*, 32(4):396–402, 1984.

[9] D. Dahan and M. Brent. On the discovery of novel word-like units from utterances: An artificial-language study with implications for native-language acquisition. *Journal of Experimental Psychology: General*, 128:165–185, 1999.

[10] A. Dempster, N. Laird, and D. Rubin. Maximum-likelihood from Incomplete Data via the EM algorithm. *J. Royal Statist. Soc. Ser.*, B(39), 1977.

[11] S. Foo and H. Li. Chinese Word Segmentation Accuracy and Its Effects on Information Retrieval. *TEXT Technology*, 2001.

[12] P. Fung. Extracting key terms from Chinese and Japnese text. *International Journal on Computer Processing of Oriental Language,Special Issue on Information Retrieval on Oriental Languages*, 99-121, 1998.

[13] X. Ge, W. Pratt, and P. Smyth. Discovering Chinese Words from Unsegmented Text. In *Proceedings of the 22th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 271–272, 1999.

[14] F. Gey, A. Chen, J. He, L. Xu, and J. Meggs. Term importance, boolean conjunct training, negative terms, and foreign language retrieval: probabilistic algorithms at TREC-5. In *Proceedings of the Fifth Text REtrieval Conference*, pages 181–190. NIST special publication, 1996.

[15] J. Hockenmaier and Brew C. Error driven segmentation of Chinese. *Communications of COLIPS*, 8(1):69–84, 1998.

[16] T. Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 1999.

[17] X. Huang and S. Roberton. Okapi Chinese Text Retrieval Experiments at TREC-6. In *Proceedings of TREC-6*, pages 137–142, 1998.

[18] X. Huang and S. Robertson. A probabilistic approach to Chinese information retrieval: theory and experiments. In *Proceedings of the BCS-IRSG 2000: the 22nd Annual Colloquium on Information Retrieval Research*, Cambridge, England, 2000.

[19] W. Jin. Chinese Segmentation and its Disambiguation. In *MCCS-92-227*, Computing Research Laboratory, New Mexico State University, Las Cruces, New Mexico, 1992.

[20] K. L. Kwok. Comparing Representations in Chinese Information Retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 34–41. ACM, 1997.

[21] K. L. Kwok. Employing Multiple Representations for Chinese Information Retrieval. *Journal of the American Society for Information Science (JASIS)*, 50(8):709–723, 1999.

[22] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, Massachusetts, 1999.

[23] J.Y. Nie and F. Ren. Chinese information retrieval: using characters or words? *Information Processing and Management*, 35:443–462, 1999.

[24] J.Y. Nie, X. Ren, and M. Brisebois. On Chinese text retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 225–233. ACM, 1996.

[25] D. Palmer and J. Burger. Chinese Word Segmentation and Information Retrieval. In *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval, Electronic Working Notes*, 1997.

[26] F. Peng and D. Schuurmans. Self-supervised Chinese Word Segmentation. In *F. Hoffman et al. (Eds.): Advances in Intelligent Data Analysis, Proceedings of the Fourth International Conference (IDA-01), LNCS 2189*, pages 238–247, Cascais, Portugal, 2001. Springer-Verlag Berlin Heidelberg.

[27] J. Ponte and W. Croft. Useg: A retargetable word segmentation procedure for information retrieval. In *Symposium on Document Analysis and Information Retrival 96 (SDAIR)*, 1996.

[28] L. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of IEEE*, 77(2), 1989.

[29] S. E. Robertson and S. Walker. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241, 1994.

[30] K. Sparck-Jones. Search Relevance Weighting Given Little Relevance Information. *Journal of Documentation*, 35(1), 1979.

[31] K. Sparck-Jones. The Role of Artificial Intelligence in Information Retrieval. *Journal of the American Society for Information Science*, 42(8):558–565, 1991.

[32] R. Sproat and C. Shih. A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Language*, 4:336–351, 1990.

[33] W. J. Teahan, Y. Wen, R. McNab, and Witten I. H. A Compression-based Algorithm for Chinese Word Segmentation. *Computational Linguistics*, 26(3):375–393, 2001.

[34] E. Voorhees and D. Harman. Overview of the Sixth Text REtrieval Conference (TREC-6). In *Proceedings of the sixth Text REtrieval Conference (TREC-6)*. NIST Special Publication, 1998.

[35] R. Wilkinson. Chinese Document Retrieval at TREC-6. In *Proceedings of TREC-6*, 1998.