On the Use of Words and N-grams for Chinese Information Retrieval

Jian-Yun Nie [1]

Département d'informatique et de recherche opérationnelle
Université de Montréal,
Email: nie@iro.umontreal.ca

Jiangfeng Gao, Jian Zhang, Ming Zhou

Microsoft Research
Email: {jfgao, mingzhou}@microsoft.com

Abstract: In the processing of Chinese documents and queries in information retrieval (IR), one has to identify the units that are used as indexes. Words and n-grams have been used as indexes in several previous studies, which showed that both kinds of indexes lead to comparable IR performances. In this study, we carry out more experiments on different ways to segment documents and queries, and to combine words with n-grams. Our experiments show that a combination of the longest-matching algorithm with single characters is the best choice.

Keywords: Information retrieval, Chinese, word, n-gram.

1. Introduction

It is now well known that the major difference between Chinese information retrieval (IR) and IR in European languages lies in the absence of word boundaries in sentences. Words have been the basic units of indexing in traditional IR. As Chinese sentences are written as continuous character strings, a pre-processing has to be done to segment sentences into shorter units that may be used as indexes. Units may be of two kinds: words or n-grams. In the previous studies, several experiments have been carried out using these two kinds of indexing units [4, 8, 11]. It turns out that they only lead to a marginal difference in IR performance when they are used separately.

However, several questions are still not answered satisfactorily: Does the accuracy of word segmentation have a significant impact on IR performance? Is it worthwhile to combine words with n-grams in Chinese IR? How should this be done? These are the questions we will examine in this study. The purpose of the study is to complete the previous results concerning the relationship between word segmentation, the use of n-grams and the performance of Chinese IR. A series of tests will be conducted. This is a step forward to find a good way to index Chinese texts.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies and not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and / or a fee.
Proceedings of the 5th International Workshop Information Retrieval with Asian Languages

Copyright ACM 1-58113-300-6/00/009 … $5.00

2. Chinese segmentation

There are two methods for segmenting a continuous character string into shorter units: using n-grams and using words. The advantage of using n-grams is that it does not require any linguistic knowledge. This is the main reason for using n-grams in Chinese and other Asian languages [7, 12]. A string is simply cut down into units of fixed length. Usually, one uses uni-grams (or characters) and/or bi-grams. For example, a string ABCD (where each letter represents a Chinese character) can be segmented into bi-grams AB BC CD, or uni-grams A B C D.

It is always possible to use longer n-grams. However, the cost for indexing in IR would be much higher as there will be a lot more possible units to be considered, thereby increasing the number of indexes. This additional cost does not seem to be necessary for Chinese IR because most meaningful Chinese words are composed of one or two characters (our statistics shows that the average length of

[1] This work was done while the author was visiting Microsoft Research China.

words in usage is 1.59). Therefore, bi-grams can successfully cover most of the words.

The segmentation of Chinese sentences into words requires linguistic knowledge. Several types of knowledge may be used: manually constructed dictionary which stores a set of known words, heuristic rules on word formation, or some statistical measures based on co-occurrences of characters. These three types of knowledge may be combined in different ways. For example, an approach based on a dictionary often uses also a set of heuristic rules. A statistical approach may also incorporate a set of heuristic rules.

Various experiments have been carried out on different segmentation approaches in the past 15 years. There is no one single approach shown to be clearly superior to the others. Most elaborated approaches can achieve a segmentation accuracy of over 90%. This performance has been believed to be sufficient for IR. Most people think that a few segmentation errors would not have a significant impact on IR performance. This is true if the segmentation errors do not concern critical words that are the most significant for a document or a query. In [10] it has been shown that segmentation accuracy does have an impact on IR performance. However, the experiment was done on a small document collection. In the present study, we would carry out the comparison again with a larger test collection – TREC corpus [3].

In this study, we will not use a statistical approach for word segmentation because it would require a large set of segmented texts as training data, which is not available in our case. Instead, we have two dictionaries for a dictionary-based segmentation. Therefore, we will only test dictionary-based segmentation methods.

A dictionary-based segmentation [2, 9, 14, 15] tries to identify all the occurrences of the dictionary words in a sentence. If there are ambiguities, the longest-matching algorithm is usually used to select the best choice.

Segmentation ambiguities may be of two kinds: combinatory ambiguity and overlapping ambiguity. The first kind refers to the case where the string AB (where A and B may be single characters or strings of characters) may be considered as a single word, and it can also be separated into A and B. In other words, the words A and B may be combined to form a longer word. The second case refers to the case where ABC may be segmented either as AB C or A BC, i.e. the words AB and BC overlap. Using the longest matching, in the first case, the longer word AB is preferred. In the second case, either solution may be selected according to the direction in which the longest matching algorithm is applied. If we start the segmentation from the beginning (forward application of longest matching), the first solution will be chosen. If we start from the end (backward application), we will choose the second solution. However, there is no clear difference in segmentation accuracy between these two directions.

The longest matching algorithm has proven to be effective. In fact, in the first case, if two words may be combined into a longer word, and this longer word is stored in the dictionary, it is generally the case that the longer word is well accepted and denotes a specific meaning. This is the case, in particular, if a word is composed of single-character words. In many such cases, single-character words usually have quite different meanings, or archaic meanings, in comparison with the meaning of the compound in modern Chinese. For example, the word 系统 (system) may be decomposed into 系 (department, attach, etc.) and 统 (unite, sum, all, etc.). However, the meanings of the characters are very different from the compound 系统. In the case where a compound word is composed of shorter compound words, the difference between the meanings of the compound word and the component words is much less. For example, the word 操作系统 (operating system) can be separated into 操作 (operating) and 系统 (system). The meaning of the long compound is similar to those of the component word. Some argue that such a long compound is not a word, but a phrase. Others may argue that 操作系统 (operating system) corresponds to a specific concept in computer science. Therefore, it is better to consider it as a word. The key issue behind this debate is that there is no clear definition of the notion of word in Chinese. Without entering into this debate, we will adopt a loose definition of word for our IR purpose: We will consider every entry in our dictionary as a word; no matter it is a short word or a long phrase.

In practice, many words such as date expressions (e.g. 一九三四年 – year 1934), suffix structures (e.g. 使用者 - user), etc. can be more efficiently recognized using heuristic rules. Therefore, dictionary-based segmentation is often complemented by a set of heuristic rules to identify such words. However, it is also possible to store all these words in the dictionary in order to gain a higher speed.

3. Possible impacts of segmentation on IR

One may tend to use the same longest-matching approach as described above to segment Chinese documents and queries for IR. The advantage of doing this is that long words usually describe more precise meanings than short words. It may be expected that the retrieval precision (the proportion of relevant documents among those retrieved) may be high. However, as we can notice, if a long word contains several short words, then only the long word will be identified as an index. The short words included are ignored. For example, if 操作系统 (operating system) is identified as a word, 操作 (operating) and 系统 (system) will not. In practice, very often, we can also refer to an "operating system" by just "system". Although the word "system" is included in "operating system", it will be considered as a completely independent index from "operating system" by IR systems. The effect of this is the loss in recall, or the phenomenon of silence. That is, some relevant documents will not be retrieved.

There may be several ways to solve this problem:

1. Instead of only extract the longest words from a sentence, we can also extract those that are included in the long words, i.e. we first apply the longest matching strategy, and then extract the short words involved within long words. In this way, the short words involved in long words will also be used as indexes. In most cases, single characters are also words (although their meaning may become archaic in modern Chinese). So this decomposition can go forth till single characters. In fact, we can even not apply the longest-matching strategy but use a simple linear look-up as follows:

   Consider a sentence as a string; extract every word that appears at the beginning of the string no matter how long it is; remove the first character at the beginning of the string and repeat the same process until the string is completely removed.

   This simple process extracts all the words in a sentence. It is much faster than the longest matching strategy, because no special processing is required to deal with ambiguities, even in the case of overlapping ambiguities. We will call this method *full segmentation*. This approach turned out to be quite effective in [11].

2. The longest words may be combined with characters. In fact, single characters may ensure a certain level of recall. Therefore, the combination of longest words with characters may be a reasonable compromise between precision and recall. This approach has been used in [5] with success.

Instead of using words, n-grams may also be used as indexes. One may use only bi-grams. However, our previous study [11] showed that a combination of bi-grams with uni-grams (characters) is a better solution. In fact, some single characters are completely meaningful alone (e.g. 造 – build). If only bi-grams are used, such meaningful characters are forced to combine with another character. There is a high chance that the character is combined with different characters in a document and a query, thus preventing the document from matching the query on the basis of this character. This is a possible explanation why bi-grams and characters together lead to a better performance.

The advantage of bi-grams in comparison with words lies in its robustness to unknown words. For example, proper nouns are not all stored in the dictionary, such as 大亚湾 (a place in southern China). The word segmentation will segment the proper noun into three characters: 大, 亚, 湾. Using bi-grams, we can still use part of the proper nouns as indexes: 大亚, 亚湾. If both bi-grams occur in the same document, there is a higher chance that the document concerns 大亚湾, than if the three single characters occur in it. Political terms or abbreviations (e.g. 三乱 – three turmoils), and foreign names (e.g. 皮纳图博火山 - Mount Minatubo) are similar cases that can be dealt with effectively by bi-grams. Therefore, bi-grams can consider unknown words and abbreviations in a better way than words do.

Words and bi-grams represent two different ways to represent a text – one relies on linguistic knowledge and the other on statistical information only. It is a common practice to combine different evidence to judge document relevance. So it is also reasonable to combine n-grams with words.

The approaches suggested above for Chinese IR are very similar to some approaches in IR for European languages. In fact, the indexing process is a step to create a representation of a Chinese text or query. As IR for European languages, there may be different ways to build such a representation: by means of keyword, compound terms, or a certain combination of them. The indexing problem in Chinese is similar. If we use the three kinds of indexes described above, we can create three possible representations for a document and a query as shown in the following figure:

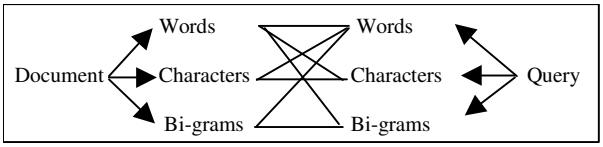

Fig. 1. Possible representations in Chinese IR

In this figure, we see that some correspondences may be created across representations if different representations are integrated. If we consider each representation form as a source of evidence, then the correspondence between a document and a query shown in Fig. 1 combines multiple evidence sources, an approach commonly used in traditional IR [6].

A closer comparison between Chinese IR and IR in European languages is possible. Roughly speaking, we can consider Chinese characters as lexemes in European languages, and Chinese words as words or phrases. It is a common practice in traditional IR to apply a stemming process in order to find a reduced form of word. This operation is similar to a decomposition of compound Chinese words into characters.

The use of bi-grams is very similar to the attempts to create word couples as indexes in European languages, except we do not apply any linguistic knowledge to filter bi-grams. The full segmentation approach is similar to some approaches in English IR in which compound phrases are combined with single words. [13] is one of them. It was shown that when noun phrases are combined with words, a slight increase in IR performance is observed. We hope that the similar combination of long and short words, as well as characters, would also lead to some improvement in IR performance.

So, from an IR point of view, the approaches described above are not imagined without foundation. They are based on the previous experience in IR. We will see through our experiments that these same approaches apply equally well to Chinese, once one establishes equivalence between the roles of units in Chinese and European languages.

4. Experimental settings

The tests are conducted on the TREC Chinese corpus [3]. The documents in the collection are articles published in the People's Daily from 1991 to 1993, and a part of the news released by the Xinhua News Agency in 1994 and 1995. A set of 54 queries has been set up and evaluated by people in the NIST (National Institute of Standards and Technology). This is the only large Chinese corpus available for IR tests. Some characteristics of the test corpus are given in the following tables.

NB of doc.	Total Size (megabyte)	Average length
164 789	167.4	507 characters

NB of queries	Average length
54	119 characters

Table 1. Characteristics of the test collection

Once Chinese sentences have been segmented in separate items, traditional IR systems may be used to index them. These separate items are called "terms" in IR. For our experiments, we used a modified version (the modifications are made in order to deal with Chinese) of the SMART system [1].

The indexing result for a document is a vector of weights:

$$D_i \longrightarrow (d_{i1}, d_{i2}, ..., d_{im}).$$

where d_{ik} ($1 \le k \le m$) is the weight of the term t_k in the document D_i, and m is the size of the vector space, which is determined by the number of different terms found in the collection. In our case, terms are Chinese words and/or n-grams. The weight d_{ik} of a term in a document is calculated according to its occurrence frequency in the document (*tf* - term frequency), as well as its distribution in the entire collection (*idf* - inverse document frequency). More precisely, we used the following formula in Smart (the *ltc* weighting scheme):

$$d_{ik} = \frac{[\log(f_{ik}) + 1.0] * \log(N/n_k)}{\sqrt{\sum_j [(\log(f_{jk}) + 1.0) * \log(N/n_k)]^2}}$$

where f_{ik} is the occurrence frequency of the term t_k in the document D_i, N is the total number of documents in the collection; n_k is the number of documents that contain the term t_k. This is one of the most used *tf*idf* weighting schemes in IR.

A query is indexed in a similar way, and a vector is also obtained for a query:

$$Q_j \longrightarrow (q_{i1}, q_{i2}, ..., q_{im}).$$

Similarity between D_i and Q_j is calculated as the inner product of their vectors, that is:

$$Sim(D_i, Q_j) = \sum_k (d_{ik} * q_{jk}).$$

5. Experiments

We will conduct the following tests in order to find out the best units to be used as indexes for Chinese IR:

1. using the longest matching with a small dictionary and with a large dictionary
2. combining the first method with characters
3. using full segmentation with or without adding characters
4. using bi-grams and characters
5. combining words with bi-grams and characters
6. adding an unknown word detection

5.1. Impact of dictionary in word segmentation

We first tested the use of longest matching as a segmentation means. Two different dictionaries are used in order to examine the impact of the completeness of the dictionary on IR performance. The small dictionary contains 65 502 entries. The large dictionary contains 220K entries. Note, however, that a certain number of the entries in the large dictionary are expressions (e.g. of time) that can also be recognized with heuristic rules. The inclusion of these expressions in the dictionary only allows for a higher speed of the segmentation process. We can assume that the second dictionary is quite complete. In both cases, we use the same forward longest-matching strategy.

Using the first dictionary, we obtained an average precision[2] of 0.3797. Using the second dictionary, the average precision is increased to 0.3907.

Through these two experiments, we can see that a better dictionary can increase IR effectiveness to some extent. This result is coherent with that in [10] that a better segmentation leads to a better IR result. However, the increase is very limited in comparison with the number of additional entries. From this, it may be concluded that

[2] The average precision is measured as the average of the precision ratios at 11 recall points : 0.0, 0.1, …, 1.0.

increasing the size of the dictionary is not a very effective way to increase IR performance. Other means should be used in addition.

5.2. Combining single characters with longest words

As we stated in the last section, one of the problems with the longest-matching strategy is that the short words included in long words are ignored. This may affect the recall ratio (thus the average precision). The first solution we suggested is to complement the longest words by single characters. Using this approach, we obtained some improvements: In the case of the small dictionary, the average precision becomes 0.4058 (an improvement of 6.9%). In the case of large dictionary, it becomes 0.4290 (9.8% improvement). These increases in performance are consistent with the results obtained by [5].

In comparison with the last series of experiments in section 5.1, we can see that simply adding single characters is a more effective way to increase IR performance than trying to increase the size to the dictionary.

5.3. Using full segmentation

Another way to increase recall is to extract also the short words implied in long words (full segmentation). We only report the experiment with the large dictionary here.

Using full segmentation, we obtained an average precision of 0.4090. In comparison with the longest-matching algorithm in section 5.1, this performance is higher. This confirms our intuition that full segmentation may gain much in recall (although there is a certain loss in precision).

However, this performance is lower than the previous one (section 5.2). One of the main differences between them is the cross-word segmentation phenomenon, i.e. some words are extracted that are composed of parts of two different words. For example, from the string 开发油田 (exploit a oilfield), we not only extract the correct words 开发 (exploit) and 油田 (oilfield), but also 发油 (hair oil). In the same way, from 意外事故 (accident), we will extract the wrong word 外事 (foreign affaires). Obviously these wrong words will have a great impact on the retrieval results.

As the combination of characters with the longest-matching algorithm has been beneficial, it is also intuitive to combine the full segmentation with characters. This combination may seem strange because single characters are already extracted from texts. The difference created by adding single characters once again lies in the weights we attribute to single characters. In the full segmentation, compound words are implicitly attributed higher weights, because they are represented several times: as compound words and as single characters. However, a single-character word is only represented once. Therefore, if a query contains several compound words and some other single-character words, the former may match documents several times (as compound and as single characters), whereas the latter only once. So the addition of single characters may be seen as a means to better balance the weight between compound and single-character words.

Using the combination, the average precision is increased to 0.4117. This shows that the combination indeed creates a better balance between compound words and single characters among indexes. However, this performance is still lower than that in section 5.2.

5.4. Chinese IR using n-grams

In the previous studies [4, 11] it is found that bi-grams may result in a performance comparable to words. In [11] it is further shown that if bi-grams are combined with uni-grams (characters), the performance is better. We repeat this experiment here, and obtained an average precision of 0.4254. This performance is comparable to the best performance we obtained using words. It may seem surprising because many bi-grams are meaningless, especially bi-grams containing functional characters (e.g. 的 – of). Notice, however, that the IR indexing process also includes a weighting scheme. If a bi-gram occurs very often in a document, it is important in that document (the *tf* factor). However, if it appears in many documents, then its importance will be diminished (the *idf* factor). For most bi-grams with functional characters, it is very likely that they appear in many documents. Therefore, their *idf* weight (and the total weight) will be reduced. In addition, many bi-grams will have little incidence on the global IR results because they do not appear in the queries (of our test corpus). This is why the global IR effectiveness does not seem to suffer despite so many meaningless bi-grams.

As we stated in Section 3, the combination of n-grams with words may gain in robustness. In the TREC queries, a number of proper nouns and political abbreviations are used. These words are not stored in the dictionary. By using bi-grams, they may be better taken into account.

The disadvantage of bi-grams with respect to words is the much larger number of indexes produced. As a consequence, the indexing time is more than doubled (from about 2 hours to more than 5 hours). The disc space requirement and the retrieval time are increased at about the same rate. This could raise problems for dealing with larger document collections.

5.5. Combinations of words and n-grams

As bi-grams and words have their own advantages, is it possible to combine them to benefit from both of them? Theoretically, such a combination would yield a better precision (due to words) and an increased robustness for unknown words (due to n-grams).

Words and n-grams may be combined in different ways: 1) During the segmentation process, one can extract words, bi-grams and characters at the same time. All the extracted

elements will be used as indexes. 2) Another combination is to segment documents and queries in two ways: by words, and by bi-grams and characters. The queries will be evaluated separately. Then the retrieved results may be combined by simply sum up the similarities produced by the two separate retrievals (or multiplied by a relative importance). The idea behind the second combination is that, if a document is retrieved by both methods, then it is likely that the document is relevant. This is the same idea as the combination of different evidence for document ranking commonly used in IR [6].

The disadvantage of such a combination is the sharp increase of processing time and disc space to store the indexing results. If we segment texts into words, bi-grams and characters, the total size of the document corpus is increased to 700 MB from originally 167 MB. As a matter of fact, we failed to index the whole corpus using Smart because of the limit of our resources and that of the Smart system.

We only succeeded in the second kind of combination. Two cases of combination are tested:

1) combining 5.2 (longest-matching + characters) and 5.4 (bi-grams + characters),

2) combining 5.3 (full segmentation) and 5.4 (bi-grams + characters).

We obtained respectively an average precision of 0.4260 and 0.4400. They represent slight changes from the uncombined cases. This result is consistent with those obtained in TREC-5 and 6 Chinese track (e.g. [8]). However, the changes are marginal. Whereas the space and the time are roughly the sum of those required by the separate runs.

5.6. The impact of unknown word detection

After word segmentation, we noticed that some important proper nouns and noun phrases have not been recognized as words, but segmented into single characters. For example, 皮纳图博火山 (Mount Minatubo) has been segmented as 皮 纳 图 博 火山. The word 蜂窝式 (cellular) is also segmented into 蜂窝 (bee's nest) and 式 (type). This is because the words are not stored in the dictionary. This phenomenon is common in Chinese because no dictionary can store all the words, and new words are created constantly. Therefore, it is important to have a mechanism to automatically recognize such words in texts.

We used a NLP analyzer developed in Microsoft – NLPWin – to recognize such unknown words. NLPWin first tags texts using a Chart-parser (with a dictionary). For unknown words, a category is guessed according to its context. Special rules have also been integrated to recognize proper nouns. As a consequence, most Chinese or non-Chinese proper nouns can be tagged and recognized correctly. Some political terms and abbreviations (e.g. 中越 - Sino-Vietnam) can also be recognized. Using NLPWin, we created another set of words that is added to our original dictionary. From the 54 queries, 80 new words have been recognized. Most of them are proper nouns or noun phrases. Table 2 contains some examples of queries for which the addition of new words has positive impacts.

Among the 54 queries, the addition of unknown words had positive impact for 10 of them. For 4 queries, negative impacts have been observed. In particular, for query 10, the negative impact has been important (from 0.3086 to 0.1359) because 中国新疆 (China Xinjiang) was recognized as a word. In the documents, however, they are often separated into two words 中国 (China) and 新疆 (Xinjiang); and more often, only 新疆 appears in texts. For the other 40 queries, no significant impact has been observed. Globally, the recognition of unknown words has a positive impact on the IR performance. The average precision for the 54 queries is changed from 0.4290 to 0.4342.

Q	original av.prec.	New av.prec.	Impr.	New words added
9	0.3648	0.4173	14.4%	毒品买卖 (drug sale)
23	0.3940	0.5154	30.8%	联合国安理会 (Security committee of UN), 和平建议 (peace proposal)
28	0.4824	0.5034	4.4%	蜂窝式 (cellular), 交换网 (interchange network)
46	0.3483	0.4192	20.4%	中越 (Sino-Vietnam)
47	0.5369	0.5847	8.9%	皮纳图博火山 (Mount Minatubo), 臭氧层 (ozone layer)
54	0.6778	0.7005	3.3%	F-16, 八．一七 (August 17)

Table 2. Impact of unknown word recognition on some queries.

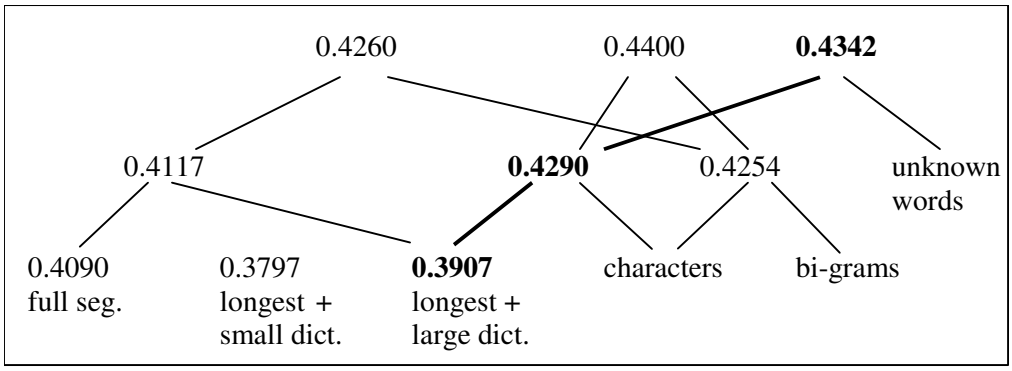

Fig. 2. Summary of experiments

5.7. Summary

This series of experiments may be summarized in the figure 2. We can clearly see that as long as different kinds of indexes are combined, the IR performance increases. The question now is whether the combination is worth the cost. Some combinations do not increase much the cost in time and space. This is the case for the combinations of words with characters. There are only about 6 000 Chinese characters in the GB codes. The addition of these characters does not increase much the vector space, and the cost of indexing and retrieval.

On the other hand, any indexing scheme that involves bi-grams is very costly in both time and space. Virtually, there are 6 000 * 6 000 possible bi-grams in Chinese. Although many of these bi-grams actually do not appear, the number is still much higher than the possible words and characters in Chinese. This will result in a very large vector space, leading to excessive indexing time and space. Compared with the combination of words and characters, there is no advantage for bi-grams, except that it does not require a dictionary. However, it is no longer a problem to acquire a high quality Chinese dictionary nowadays. So the use of bi-grams is not justified.

If we place the above experiments in the context of cross-language IR (CLIR), the use of bi-grams is even less justified. In fact, there may be simple methods to translate queries from or into Chinese words using either a bilingual dictionary, or using a set of parallel texts as training data. However, there is no dictionary for bi-grams. The use of parallel texts is also constrained by the huge amount of space and time required for training a translation model with bi-grams. So the use of bi-grams in CLIR is impracticable.

On the other hand, the recognition of unknown words may be a feasible way to improve IR performance. It is possible to select those unknown words above some frequency threshold. In this way, the indexing space and time will not be increased drastically.

In conclusion, the best way for Chinese IR and CLIR with Chinese is to use a combination of words and characters. This corresponds to the bold lines in figure 2.

6. Conclusions

Many experiments have been done on Chinese IR. The main concern was on the segmentation of Chinese texts into smaller units. Two approaches have been proposed: using words and using n-grams. However, there was still no conclusive result about the ideal segmentation method to be used for Chinese IR. In this study, we made a series of experiments to examine the impact of different segmentation methods on IR performance. Our experiments show that words and n-grams can achieve comparable performances. However, if we consider the time and space factors, then it is preferable to use words (and characters) as indexes.

The previous experiments [4, 11] have tested several indexing methods that turn out to be reasonable for Chinese IR. In this paper, we tested several additional approaches. It turns out that a combination of the longest-matching algorithm with single characters is a good method for Chinese IR. In addition, if there is an unknown word detection, the performance can be further improved. The size of vector space produced is bounded by the number of known and unknown words, and that of characters. The indexing and retrieval speed is much faster than that with bi-grams.

This series of tests is only the first step of our ongoing research program. In a later stage, Chinese IR will be used as a step in English-Chinese cross-language IR. For this task, it is even more difficult to use bi-grams as indexes, because there is no effective means to translate English words to Chinese bi-grams. This is another reason why we privilege words and characters as indexes for Chinese texts.

References

1. Buckley, C. *Implementation of the SMART information retrieval system*, Technical report, #85-686, Cornell University, 1985.

2. Chen, K.-J. and Kiu, S.-H. Word identification for Mandarin Chinese sentences. *5th International Conference on Computational Linguistics*, 1992. pp. 101-107.

3. Harman, D. K. and Voorhees, E. M., Eds. *Information Technology: The Fifth Text REtrieval Conference (TREC-5)*, NIST SP 500-238. Gaithersburg, National Institute of Standards and Technology, 1996.

4. Kwok, K. L. Comparing representations in Chinese information retrieval. *Conference on Research and Development in Information Retrieval, ACM-SIGIR*, 1997, pp. 34-41.

5. Kwok, K.L. and Grunfeld, L. TREC-5 English and Chinese retrieval experiments using PIRCS, *The Fifth Text Retrieval Conference (TREC-5),* NIST special publication 500-238, 1997, pp. 133-142.

6. Lee, J. H. Combining multiple evidence from different properties of weighting schemes. *Conference on Research and Development in Information Retrieval, ACM-SIGIR*, Seattle, 1995, pp. 180-188.

7. Lee, J. H. and Ahn, J. S. Using n-grams for Korean text retrieval. *Conference on Research and Development in Information Retrieval, ACM-SIGIR*, Zurich, (996, pp. 216-224.

8. Leong, M.-K. and Zhou, H. Preliminary qualitative analysis of segmented vs. bigram indexing in Chinese, *The Sixth Text Retrieval Conference (TREC-6),* NIST special publication 500-240, 1998, pp. 551-557.

9. Li, B.-Y., Lien, S., Sun, C.-F. and Sun, M.-S. A maximal matching automatic Chinese word segmentation algorithm using corpus tagging for ambiguity resolution. *R.O.C. Computational Linguistics Conference (ROCLING-IV)*, Taiwan, 1991, pp. 135-146.

10. Nie, J.-Y., Brisebois, M. and Ren, X. On Chinese text retrieval. *Conference on Research and Development in Information Retrieval, ACM-SIGIR*, Zurich, 1996, pp. 225-233.

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

12. Ogawa, Y. A new character-based indexing organization using frequency data for Japanese documents. *Conference on Research and Development in Information Retrieval, ACM-SIGIR*, Seattle, 1995, pp. 121-129.

13. Strzalkowski, T., Lin, F. and Perez-Carballo, J. Natural language information retrieval TREC-6 report, *The Sixth Text Retrieval Conference (TREC-6),* NIST special publication 500-240, 1998, pp. 347-366.

14. Yao, T.-S., Zhang, G.-P. and Wu, Y.-M. A rule-based Chinese automatic segmentation system. *Journal of Chinese Information Processing*, 1990, 4(1): 37-43.

15. Yeh, C.-L. and Lee, H.-J. Rule-based word identification for Mandarin Chinese sentences - A unification approach. *Computer processing of Chinese and Oriental Languages*, 1991, 5(2): 97-118.