Introducing the Sequence Model for Text Retrieval

Yih-Kuen Tsay and Yu-Fang Chen

Department of Information Management, National Taiwan University

Abstract. We propose and explore a novel approach, called the sequence model, to text retrieval. The model differs from classical ones in the extent of how positional information of term occurrences is used for relevance judgment. In the sequence model, documents and queries are viewed as sequences of term-position pairs and the relevance of a document to a query is judged by the similarity between their respective representative sequences. We suggest three primitive measures of sequence similarity, each capturing a distinct aspect of resemblance between two sequences. These similarity measures can be combined in various ways to suit different information needs. We have developed a prototype system with the sequence model as its core. Experimental results show that our sequence-based approach is often more effective than appearance-based approaches.

1 Introduction

All of the three classic information retrieval (IR) models—Boolean, vector space, and probablistic—and their variations view documents and quries as represented essentially by a set of index terms or keywords, each possibly associated with a constant or varying weight [1, 14]. In these IR models, occurrences of index terms are the primary basis of relevance judgement, but it is the *number* of occurrences, not their *order* or *proximity*, that counts. The importance of positional information has not been completely ignored, though. Researchers have shown its usefulness in matching phrases, specifying context, or simply pinpointing relevant parts of a document. Practical text retrieval systems, mostly based on classical models, have started to incorporate these usages of positional information. Still, the relative positions of term occurrences play a secondary role in judging relevance.

Research and practice of text retrieval in Chinese (and other similar Asian languages) have followed the same path. However, as a sentence in Chinese is just a string of characters without clear boundaries of words, to fit into the conventional word-based models, Chinese text processing faces the additional problem of word segmentation. Dictionary and statistics-based techniques exist for tackling the problem, but each has its own difficulty.

It is possible to avoid the word segmentation problem. The character-based approach views a Chinese sentence of m characters simply as a *set* of m independent terms. Typically, positional relations among characters, such as order and

distance, are ignored. These relations, nonetheless, are useful hints for capturing the meanings of Chinese words in a sentence. For instance, the Chinese word "電腦" (computer) is formed by the combination of two characters "電" (electric) and "腦" (brain) with particular order and distance.

Relevance judgment in text retrieval ultimately has to resort to syntactical similarity between the document and the query both of which are just strings. Such similarity is not much different from what has been studied in the context of string similarity problems. This suggests that traditional string similarity algorithms can be used to solve part of the relevance judgement problem. However, for large text collections, these string algorithms are mostly too slow. They are slow largely due to the very fine similarity measures for which they are designed for. For text retrieval, coarser similarity measures may suffice.

We observe that the *order* and *proximity* of characters/terms are almost always involved in similarity measures for strings and should be essential for relevance judgement in text retrieval. Furthermore, in the context of text retrieval, when we evaluate the similarity between a document string and a query string, only terms appearing in both strings are relevant. Other terms can be ignored without affecting the evaluation result. This leads us to consider (not necessarily consecutive) sequence of terms. Sequence similarity measures then can play the role of string similarity measures for text retrieval.

The sequence model was conceived along the above line of thinking. In the model, documents and queries are viewed as sequences of term-position, or token-position pairs and the relevance of a query to a document is judged by the similarity between their representative sequences, which may be obtained by certain heuristics. A "token" in this model can be a character, word, or a phrase. The granularity of texts to be modelled is decided by its language features. We suggest three primitive measures of sequence similarity, namely token appearance, token ordering and token consecutiveness, that capture distinct aspects of resemblance between two sequences. These similarity measures can be combined in various ways to suit different information needs. We have developed a prototype system with the sequence model as its core. Experimental results show that our sequence-based approach is often more effective than appearance-based approaches.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 describes our sequence model. Section 4 presents the implementation and experiments to evaluate our method. Section 5 briefly compares our method with word-based and bigram methods. Finally, Section 6 concludes the paper.

2 Related Work

In English and other Indo-European languages, texts are composed of words separated by spaces and punctuation marks. There are explicit word boundaries in the texts, so words are naturally basic indexing and retrieval units. In the case of Chinese and many other Asian languages such as Japanese and Korean, texts are strings consisting of ideographs, without spaces to specify word boundaries.

Many text segmentation methods are proposed to solve this problem, and they can be classified as follows:

- Statistical methods [5], [3]. This kind of methods uses statistical measures such as character frequencies to find out word boundaries.
- Dictionary-based methods [3], [11]. These approaches are based on the comparison with a dictionary to separate texts into words. When there is ambiguity (several segmentation choices), grammar rules are often used to resolve it.
- Hybrid methods [12], [8]. The former two classes of method are combined. Other kinds of extra knowledge such as semantic rules or syntax analysis may be added as well.

Segmentation is not a ideal solution for Chinese text retrival. First, segmentation methods based on corpus analysis, which require computation, could be time consuming if the corpus is large. On the other hand, segmentation methods based on extra knowledge, such as a dictionary, syntax and domain knowledge, may not be able to segment all the texts. Moreover, queries and documents may be segmented differently, and this will degrade the effectiveness of retrieval results. Another significant shortcoming of segmentation is that once done, segmentation results are likely to remain for a long duration, especially when the document collection is large. This will make it infeasible to re-segment the collection, owing to the improvement of the segmentation method. Incorrect segmentation of the texts is also another problem. Although many segmentation approaches claim to reach a high accuracy of above 90%, the incorrectly segmented words will still degrade the performance.

On the other hand, approaches that do not require segmentation — N-gram-based, character-based and hybrid approaches — are developed. N-gram-based approaches exhaustively tokenize all overlapping substrings of length n during document and query processing; therefore, word boundaries identification are not required.

In general, one may consider that words are more meaningful than n-grams in European languages, so choose words as the tokens in an IR system may be the most plausible. However, this is less true for Chinese IR [11], [8], [12]. Nie *et al.* provided the following reasons:

- Chinese characters (ideographs) alone are more meaningful than characters (alphabets) in European languages. Single Chinese characters can serve as words.
- Words lengths in Chinese are more static than in European languages.In their experimental result: 63.6% of words are of the length 2.
- Though several segmentation methods are present, they are still imperfect.

Baldwin *et al.* [2] investigated the influences of word order and segmentation on the performance of Japanese-to-English translation retrieval and found that character-based indexing consistently outperforms word-based indexing and order-sensitive metrics outperform bag-of-words metrics when using character indexes. The result showed the usefulness of positional information.

The sequence model is not the first idea to adopt positional information in IR. Murata *et al.* [10] used absolute term positions in a document to decide term weights. Terms appearing in the title are assigned higher weights, and term weights decreases as locations are closer to the end of the document body. Clark *et al.* [4] introduced positional information in deciding term distance, which is used in his extension of boolean model. Frank *et al.* [7] used positional information for keyphrase extraction. Krester *et al.* [6] adopted positional information in the form of term locality. His approach gives interested parts of documents to the user instead of whole documents. Rajaraman *et al.* [13] used term proximities (which is measured by position) to achieve the effect of phrases. Wang [15] used positional information in deciding term distance and in finding the common subsequence between the query and the documents. Lin [9] used positional information in computing editing distance and determining the longest common subsequence. Documents and queries are sequences of terms in its approach. In addition to the measures from term position, it also incorporates term appearance in relevance evaluation.

Positional information of terms in the above approaches serves as a supplementary role in relevance judgement. The positions of term are used either to enhance the existing retrieval approaches or to model the weighting of terms. On the other hand, some approaches try to use positional information as the main role of relevance judgement by treating documents and/or queries as strings or sequences. These approaches assume that the semantic relationship between documents and queries can be measured by the closeness of sequence structure of text strings/sequences. Therefore, these approaches model the similarity between documents and queries as the similarity between strings or sequences.

In summary, term positions provide an alternative to similarity evaluation. Relevance is determined by how terms appeared in the text collection in terms of locations, rather than statistics based on appearances. The use of positional information can vary from the indication of article titles to the measures based on sequences. The experiments of these studies show the effectiveness of using positional information, not only in text retrieval but also in similar domains such as translation retrieval. On the other hand, it should be noted that using positional information has larger cost in terms of index space and computational overhead, but this cost may be supplemented if the effectiveness of retrieval results is more critical.

3 The Sequence Model

We first describe the basic ideas and rules of the sequence model and define our similarity measures for relevance judgement. The model alone does not immediately permit an effective implementation. We then provide further implementation-dependent details in the last part of this section.

3.1 Basic Principles

In the sequence model, documents and queries are seen as represented by sequences of token-position pairs, i.e., tokens indexed with their positions. A token (or primitive term) in a sequence is meant to be a syntactic unit of some language (such as an English word or Chinese character), number, punctuation mark, or special symbol, etc. Visually, we write a sequence S of n token-position pairs as $s_{i_1}, s_{i_2}, \cdots, s_{i_j}, \cdots, s_{i_n}$, where s_{i_j} is the j-th token of S and the index i_j indicates the position of that token (in the corresponding document or query) such that $i_1 < i_2 < ... < i_n$.

3.2 Similarity Measures

The relevance of a document to a query is judged according to the similarity between the representative document sequence and the query sequence. There are quite a few ways for measuring the similarity between two sequences. We propose three similarity measures and suggest that the similarity between between two sequences be calculated by a weighted sum of the three measures. Let $D = (d_{i_1}, d_{i_2}, \cdots, d_{i_j}, \cdots, d_{i_m})$ (of m tokens) and $Q = (q_{i_1}, q_{i_2}, \cdots, q_{i_j}, \cdots, q_{i_n})$ (of n tokens) respectively be the representative sequences of the document and the query under similarity measurement.

- Token Appearance (TA):

$$TA(D,Q) = \frac{\sum_{j=1}^{n} t(q_{i_j}) * w(q_{i_j})}{\sum_{j=1}^{n} w(q_{i_j})},$$

  where $w(q_{i_j})$ is the weight of j-th query token and $t(q_{i_j})$ indicates its existence in the document sequence (1: yes; 0: no).
  The *idf* of a token appears to be a reasonable choice for its weight.
- Token Ordering (TO):

$$TO(D,Q) = \frac{|LCS(D,Q)|}{(|D| + |Q|)/2},$$

  where LCS(D,Q) is the *longest common subsequence* of D and Q and $|\cdot|$ is the length function.
- Token Consecutiveness (TC):

$$TC(D,Q) = \frac{\sum_{j=1}^{m-1} \frac{1}{rd_j}}{m-1},$$

  where $rd_j = 1 + |(i_{j+1} - i_j) - (pos(d_{i_{j+1}}, Q) - pos(d_{i_j}, Q))|$ where $pos(c,Q)$ is the position of token c in Q.

The above three measures all have a score ranging from 0 to 1. A linear combination (weighted sum) of the measures (which also ranges from 0 to 1) can

be computed from $\alpha_1 TA(D,Q) + \alpha_2 TO(D,Q) + \alpha_3 TC(D,Q)$ with a suitable selection of α_1, α_2, and α_3 such that $\alpha_1 + \alpha_2 + \alpha_3 = 1$. An implementation may allow the user to select the coefficients.

What constitutes the representative sequence of a query is up to the particular instantiation of this model. The entire query string appears to be a reasonable choice. For instance, the representative sequence of a query "資策會", which is an abbreviation of 資訊工業策進會(the Institute for Information Industry), can be $資_1策_2會_3$.

What constitutes the representative sequence of a document is also up to the particular instantiation. But apparently, it should be chosen with respect to the query. Given for example the query "資策會", a document containing "資訊工業策進會" (from position 41) should have a representative sequence containing $資_{41}策_{45}會_{47}$ as a subsequence. A good representative sequence can be selected with heuristics. We will refer to the representative sequence of a query simply as the *query sequence* and that of a document as the *document sequence*.

3.3 Implementation Considerations

When evaluating the similarity between a document sequence and the query, only a small number of (short) fragments of the document is meaningful with respect to the query. It is computationally more efficient to first filter out the meaningful segments of a document before measuring its relevance. As suggested by the model, a representative sequence (or two) of a document should be selected before evaluating the similarity. Precision might degrade a little as a result of the selection process, but the performance gain outweighs the degradation.

Below are the heuristics that we adopted in our implementation. To find a representative document subsequence, we first select the tokens in the document string which also appears in the query token set (from index) and obtain a "document sequence." We then divide the document sequence into segments using a threshold so that the distance between two consecutive tokens in a segment is no larger than that threshold. Finally, we select one of the segments as the representative document subsequence. There are two criteria for segment selection: (1) token cardinality and (2) token number. We compare the segments using Criterion 1; Criterion 2 is used if more than one segments are selected by Criterion 1.

We next consider the impact of common characters on retrieval efficiency. In Chinese, common characters can appear alone as function words, which are useless in retrieval, or appear together with other characters as proper names or other constructs useful for retrieval. A query containing common characters will lead to too many candidate documents generated and evaluated, which degrades retrieval efficiency. To alleviate such efficiency degradation, the system has to filter out "false candidates", which are irrelevant documents but recognized as candidates because of common characters.

Efficiency is very important for a retrieval system, especially when the document set is large. If the query sequence contains some common words and we select all documents that contain any query token as candidates, then almost all

documents will be selected. To alleviate the problem, we need a candidate document selection approach that picks out tokens with a relatively higher weight from the query sequence and uses these tokens to choose candidates. This approach is more efficient, and the recall rate will not degrade too much since most of the relevant documents will contain the tokens with a higher weight. One can also adopt this approach in the selection of the (representative) document sequence.

4 Implementation and Experimental Results

We implemented the sequence model in our text retrieval system SIR (standing for Sequence-based Indexing and Retrieval). The system was designed primarily for documents written in Chinese. However, it is also capable of handling Chinese documents mingled with English words. The system can also handle homophone queries in Chinese, which is enabled by a homophone table consisting of 5000 Chinese frequent words. While processing homophonic retrieval, the system merges the index of the word and its homophones found in the table for similarity measuring.

Document processing in the system consists of two main processes. First, text strings are tokenized. The system indexes Chinese characters, English words, numbers and punctuation marks. Second, the information of a selected token $(DocID, TokID, Pos)$ are coded and stored in the index where $DocID$, $TokID$ and Pos denotes the document ID, token ID and token position, respectively.

The structure of our index is an extension of inverted lists. Section 8.1 of the Modern Information Retrieval [1] introduces different levels of detail of occurrences: document addressing, word addressing and block addressing. Traditional inverted lists map a term (token) to a list of documents, word positions or blocks where the token appears. Extra information such as term frequencies can be recorded if necessary. Take document addressing for example, if we look up the inverted lists with term t, we can obtain $(d_1, d_2, ..., d_k)$ or $((d_1, e_1), (d_2, e_2), ..., (d_k, e_k))$, where d_i is a document number and e_i is the extra information of d_i.

The cases of word addressing or block addressing are alike: the structure of the traditional inverted lists is one-level lists. The structure of the extended inverted lists is two-level lists. It provides two level of detail of the occurrences of tokens: document IDs and the positions of tokens in the documents. If we look up the extended inverted lists with term t, we will get a list of document lists containing in-document occurrence of t. It can be presented by $((d_1, (p_{1,1}, p_{1,2}, ..., p_{1,n_1})), (d_2, (p_{2,1}, p_{2,2}, ..., p_{2,n_2})), ..., (d_k, (p_{k,1}, p_{k,2}, ..., p_{k,n_k})))$.

When the indexer wants to store the information of a token $(DocID, TokID, Pos)$ in the index, it maps $TokID$ to the first-level list and then add Pos to the second-level list of $DocID$. The mapping can be done via hashing or search structure like B-Tree. Here we choose hashing because our original propose is to index Chinese characters only, and the number of Chinese alphabet is fixed (about 15,000). However, the structure can be easily extended to handle

texts mingled with Chinese, English and numbers, by introducing a table records the English word (or number) and its hash number. Note that owing to sequential processing of texts, the values stored in an extended inverted list is monotonically increasing, i.e. for a list $((d1,(p_{1,1},p_{1,2},...,p_{1,n_1})),(d_2,(p_{2,1},p_{2,2},...,p_{2,n_2})),...,(d_k,(p_{k,1},p_{k,2},...,p_{k,n_k})))$, $d_1 < d_2 < ... < d_k$ (if DocIDs are assigned to documents according to their processing order) and $p_{i,1} < p_{i,2} < ... < p_{i,n_i}, \forall 1 \leq i \leq k$. We apply differential and variable-length coding that exploits this property to reduce index space overhead.

For the above list, we store on disk the differences of DocIDs ($d_i - d_{i-1}, \forall 1 < i \leq k$) and positions ($p_{i,j} - p_{i,j-1}, \forall 1 \leq i \leq k, 1 < j \leq n_i$) except the first appearances (d_1 *and* $p_{i,1}, \forall 1 \leq i \leq k$). The differences are stored using variable number of bytes, each of which contains a reserved flag bit indicating whether the next byte is also used to record the value. A position-recording byte contains an additional reserved flag bit indicating whether the current position represents the last occurrence in the document. Additionaly, we have also designed and implemented an incremental update mechanism for the indexer.

Below we show some of our experimental results. We compare our approach with bigram-based approach. We also make the comparison with Rajaraman's approach[13] because it can be thought of as a generalizaion of bigram based approach. It relaxes bigrams-based approach by taking into account two consecutive query tokens appearing non-consecutively, which will be neglected by bigram-based approach. The scores of our approach, bigram-based approach and the generalizaion of bigram based approach are listed in the following: (The score of bigram-based approach is given by $\frac{m}{n-1}$, where m is the number of matched bigrams in the docuemt, and n is the length of the query)

Example 1

Query: 陳總統水扁(陳Chen總統President水扁Shui-Bian)
Result:

Document Summary	SIR		Bigram		G-Bigram	
	Score	Rank	Score	Rank	Score	Rank
...陳總統水扁...	1.0	1	1.0	1	1.0	1
...總統陳水扁...	0.861	2	0.5	2	0.75	2
...陳水扁總統...	0.808	3	0.5	2	0.75	2
...陳水扁參選總統...	0.804	4	0.5	2	0.75	2
...陳水扁...	0.654	5	0.25	5	0.25	5
...總統...	0.616	6	0.25	5	0.25	5

In Chinese, a person name with title can be represented in various ways. The title can be inserted between the given name and the surname, in front of the name, or after the name. In this example, the first three items 陳總統水扁(Chen,President,Shui-bian), 總統陳水扁(President,Chen Shui-bian), and 陳水扁總統(Chen Shui-bian President) are equal in Chinese. The fourth item 陳

水扁參選總統(Chen Shui-bian join the election of president) is talking about the relation between 陳水扁(Chen Shui-bian) and 總統(President), compares to the last two items 陳水扁(Chen Shui-bian) and 總統(President), it should be ranked higher. We obtain the best results when $\alpha_1 : \alpha_2 : \alpha_3 = 1 : 1 : 1$ or $2 : 1 : 1$.

Example 2

Query: 辜振甫與汪道涵(Gu Zhen-fu and Wang Dao-han)
Result:

Document Summary	SIR		Bigram		G-Bigram	
	Score	Rank	Score	Rank	Score	Rank
...辜振甫與汪道涵...	1.0	1	1.0	1	1.0	1
...辜振甫與...汪道涵...	0.968	2	0.833	2	1.0	1
...辜振甫汪道涵...	0.903	3	0.667	3	0.667	3
...汪道涵與辜振甫...	0.79	4	0.667	3	0.667	3
...汪道涵與...辜振甫...	0.787	5	0.667	3	0.667	3
...汪道涵辜振甫...	0.76	6	0.667	3	0.667	3
...辜振甫...	0.614	7	0.333	7	0.333	7
...汪道涵...	0.614	7	0.333	7	0.333	7
...辜汪...	0.33	9	0	9	0	9

This example contains two semantic blocks (two person names, 辜振甫(Gu Zhen-fu) and 汪道涵(Wang Dao-han)) and a coordinator 與(and). All of the first six items are talking about the two person. They should be ranked higher than the next two elements,辜振甫(Gu Zhen-fu) and 汪道涵(Wang Dao-han), which contained only one semantic block (one person name.) The last item 辜汪(Gu Wang) is the abbreviation of the two person. Ideally, it should be ranked higher than the previous two item. But it structure is greatly different to the query string. In this situation, we can use our user feedback mechanism, set 辜汪(Gu Wang) as query expansion, than the similarity score of 辜汪(Gu Wang) will be much higher. In this example, we get the best result when $\alpha_1 : \alpha_2 : \alpha_3 = 2 : 1 : 1$

Example 3

Query	Document Summary	Score		
		SIR	Bigram	G-Bigram
聯合國安理會	...聯合國安全理事會...	0.95	0.6	1
聯合國安全理事會	...聯合國安理會...	0.789	0.43	0.6
臺大	...臺灣大學...	0.875	0	1
臺灣大學	...臺大...	0.541	0	0
資策會	...資訊工業策進會...	0.844	0	1
資訊工業策進會	...資策會...	0.458	0	0
海基會	...海協交流基金會...	0.844	0	1
海峽交流基金會	...海基會...	0.458	0	0

This is an example retrieving full names by abbreviations, and vice versa. We can find the similarity scores of retrieving abbreviations by full names are low. But we can use the user feedback mechanism to rise their similarity scores.

Example 4

Query: 南亞的海嘯(The South Asia Tsunami)
Result:

Document Summary	SIR		Bigram		G-Bigram	
	Score	Rank	Score	Rank	Score	Rank
...南亞的海嘯...	1.0	1	1.0	1	1.0	1
...南亞大海嘯...	0.87	2	0.5	2	0.5	2
...環遊南亞遇海嘯...	0.87	2	0.5	2	0.5	2
...南亞地震海嘯...	0.813	4	0.5	2	0.5	2
...南亞海嘯...	0.813	4	0.5	2	0.5	2

In Example 1, we can see that bigram-based methods are not suitable when semantic units can be reordered without changing the meaning. This is because changing token order will result in different bigrams. This influence is even greater if longer n-grams are used. Bigrams also suffer from name contractions. Example 2 shows the retrieval of full names given abbreviations. In these cases, there is no bigram match between full names and abbreviations, so bigram-based methods cannot deal name contractions. The approach in [13] is superior to bigrams only when using the abbreviation to retrieve the full name (Example 3), where two consecutive tokens in the query can be found in the documents with a distance larger than 1. Example 4 shows the fact that bigrams cannot distinguish between the omission of function word "的" and the insertion of modifiers between main semantic units . The two situations are semantically different, but bigram-based methods treat them equally.

5 A Comparative Summary and Discussion

In this section, we first compare our model with the classical ones and explain the advantages of our model. We then make a comparison between our model and other approaches that are designed to handle Chinese.

Implementing a statistical model faces many difficulties. It requires a large amount of training data, a lot of tuning time to become stable, and has to be adjusted for different environments. In contrast, implementing a system with the sequecne model is easier. The implementation of the sequence model can apply in any environments as long as the query and document can be transformed into tokens. The effectiveness of a statistical-based system is highly dependent on the quality of training data. Therefore, it is hard to compare the retrieval effectiveness between the two-type of system. However, according to our experimental results, our system can provide at least "acceptable" retrieval effectiveness.

In the vector-space model (VSM), the design of feature vectors is a difficult job. If we choose too few elements, it is hard to reflect the feature of a document. But, if we choose too many elements, the feature vector of a document might be very sparse. It wastes a lot of storage space and processing time. Although we can use the Latent Semantic Indexing(LSI) to reduce the dimension of vector. The VSM is still not perfect because the implementataion of VSM needs to design different feature vector for different data-sources. A system implemented with the sequence model will not meet these difficulties. And theoretically can archive better retrieval effectiveness because it adopts more information (the order and proximity of terms) than vector model.

We now present an overall comparison between our approach and appearance-based Chinese text retrieval approaches (approaches using unigrams, bigrams, and words). To make this comparison, we simulate unigram-based and bigram-based indexing, and the index size overheads of the two type indexing approaches are about 30% and 120%, respectively. Note that the simulation results are given without compression techniques applied, where one occurrence (as document ID) is recorded using 4 bytes. If compression is applied, the overheads can be reduced to the half (estimated). The index size of our approach is larger than those of the other approaches (if compression techniques are not applied), for each occurrence in the documents should be recorded. The other approaches do not keep track of each occurrence, and repetitive appearances of tokens in a document are not indexed. We can say our indexing approach is exhaustive and the others' are lossy in terms of the portion of the information in the document collection being recorded.

The retrieval effectiveness is better than bigrams and generalized bigrams in the above testings. We did not conduct experiments on large test collections to see the our performance in terms of commonly used indicators such as precision and recall, but we expect our approach to perform at least comparable to words and bigrams.

The advantage of word-based approaches lies in the semantics carried by words. Therefore, it is easier and more reasonable to perform term weighting, thesaurus processing and other kinds of processing based on semantics. Cross-language text retrieval is also easier when using words. However, it suffers from the main shortcomings of segmentation such as the processing overhead before indexing and the imperfection of segmentation results.

The main disadvantage of our method is retrieval time. Most of the computation of other approaches is completed during document processing (indexing plus other processing such as segmentation and term statistics calculation). Our approach spends longer time to complete the request compared to these approaches. However, it has another advantage over other approaches: flexibility. Because each occurrence in the document collection is recorded, retrieval method can be modified without re-indexing (as long as the modified approach still use sequences of tokens and positions). The comparison is briefed in table 1.

	Words	Bigrams	SIR
Index Size	30–120%	about 120%	about 100%
Indexing Exhaustiveness	lossy	lossy	exhaustive
Retrieval Effectiveness	acceptable	acceptable	acceptable
Requiring Segmentation	yes	no	no
Semantic Processing	moderate	difficult	difficult
Retrieval Time	fast	fast	moderate
Flexibility	low	low	high

Table 1. A brief comparison between our approach and word-based, unigram-based, and bigram-based approaches

6 Conclusions and Future Work

Positional information of terms is useful and yet has only been utilized to a limited extent in text retrieval. In this paper, we proposed an approach to text retrieval that fully explores the use of term positions. The approach views documents and queries as represented by sequences of token-position pairs and the relevance of a document to a query is judged by the similarity between their respective sequences. We defined sequence similarity measures that capture term ordering and proximity. The main cost of incorporating positional information into an text retrieval system is a larger index space overhead because of the near lossless preservation of token occurrences. However, this cost is compensated by better retrieval results.

We have focused on the application of our approach in Chinese text indexing and retrieval. Classical models encounter several problems when applied to Chinese text retrieval, for example, the word segmentation problem, the training process of the statistical approach, the efforts in designing feature vectors in the vector model, the discovery of new words in the dictionary-based approach, and the lack of ranking mechanism in the boolean model. A retrieval system based on the sequence model avoids most of these problems. We believe that our approach can complement other classic models as well. A system that implements a combination of the boolean model and the sequence model is currently being developed.

Our approach can in principle be applied to IR of other languages, including other oriental languages, Indo-European languages, and even pictographs. We actually have tried it with a collection of Chinese-English text where a document is mainly in Chinese but may contain English words. It remains to be investigated whether our approach can perform as well in other languages.

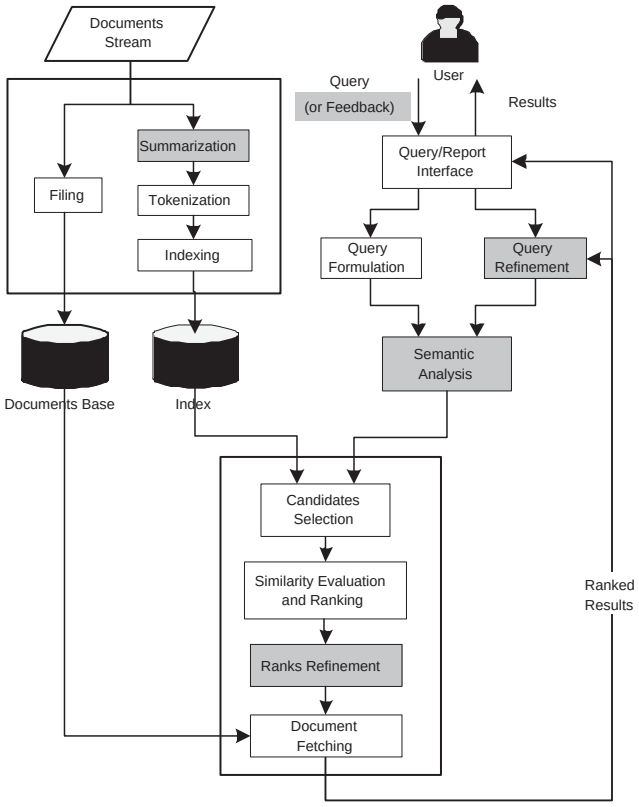

Fig. 1. The SIR system; the shaded areas represent future extensions.

The sequence model addresses only the core function of an IR system. We did not consider text analysis such as text summarization, token normalization, document collection statistics, nor did we consider query expansion. How to perform such processes in a sequence-based IR approach and their improvements to and impact on the approach are also problems worthy of further investigation.

Acknowledgment

Part of this paper has been derived from the Masters thesis of Ching-Ling Yu [16], which was supervised by the first author. His contributions are gratefully acknowledged.

References

1. Ricardo Baeza-Yates and Bertheir Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

2. Timothy Baldwin and Hozumi Tanaka. The effects of word order and segmentation on translation retrieval performance. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 35–41, 2000.
3. Aitao Chen, Jianzhang He, Liangjie Xu, Fredric C. Gey, and Jason Meggs. Chinese Text Retrieval Without Using a Dictionary. In *Proceedings of the 20th Annual International ACM SIGIR Conference*, pages 42–49, 1997.
4. Charles L. A. Clarke and Gordon V. Cormack. Shortest-Substring Retrieval and Ranking. *ACM Transaction on Information Systems*, pages 44–78, 2000.
5. Yubin Dai and Teck Eh Loh. A New Statistical Formula for Chinese Text Segmentation Incorporating Contextual Information. In *Proceedings of the 22th Annual International ACM SIGIR Conference*, pages 82–89, 1999.
6. Owen de Krester and Alistair Moffat. Effective Document Presentation with a Locality-Based Similarity Heuristic. In *Proceedings of the 22th Annual International ACM SIGIR Conference*, pages 113–120, 1999.
7. E. Frank, G.W. Paynter, I.H. Witten, C. Gutwin, and C.G. Nevil-Manning. Domain-specific keyphrase extraction. In *Proceedings of 16th International Joint Conference on Artificial Intelligence IJCAI-99*, pages 668–673, 1999.
8. K. L. Kwok. Comparing Representations in Chinese Information Retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference*, pages 34–41, 1997.
9. Lung-Chi Lin. A Preliminary Study of Text Retrieval Techniques Utilizing Character/Word Positions. Master's thesis, Department of Information Management, National Taiwan University, 2000.
10. Masaki Murata, Qing Ma, Kiyotaka Uchimoto, Hiromi Ozaku, Masao Utiyama, and Hitoshi Isahara. Japanese Probabilistic Information Retrieval Using Location and Category Information. In *Proceedings of the 23th Annual International ACM SIGIR Conference*, pages 81–88, 2000.
11. Jian-Yun Nie, Jiangfeng Gao, Jian Zhang, and Ming Zhou. On the Use of Words and N-grams for Chinese Information Retrieval. In *Information Retrieval with Asian languages*, pages 141–148, 2000.
12. Jian-Yun Nie and Fuji Ren. Chinese Information Retrieval: Using Characters or Words? *Information Processing and Management*, pages 443–462, 1999.
13. K. Rajaraman, Kok F. Lai, and Y. Changwen. Experiments on Proximity Based Chinese Text Retrieval in TREC 6. In *Text REtrieval Conference (TREC-5)*. NIST, 1997.
14. Gerard Salton. *Automatic Text Processing: the Transformation, Analysis, and Retrieval of Informatio by Computer*. Addison-Wesley, Reading, MA, 1989.
15. Hsing-Yung Wang. Applying Pattern Similarity with Word Proximities of Common Subsequences in Information Retrieval. Master's thesis, Department of Computer and Information Science, National Chao Tung University, 2000.
16. Ching-Lin Yu. Sequence-Based Text Retrieval : Design and Implementation. Master's thesis, Department of Information Management, National Taiwan University, June 2002.