A PROBABILISTIC APPROACH FOR CHINESE INFORMATION RETRIEVAL: THEORY, ANALYSIS AND EXPERIMENTS

Xiangji Huang

A THESIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

at

City University, London

Department of Information Science

August 2001

To my wife, daughter and parents

Contents

List of Tables

List of Figures

Acknowledgements

There are many people I should thank for their concern and encouragement. First of all, I would like to express my sincere gratitude to my supervisor, Professor Stephen Robertson, for his excellent guidance, constant encouragement and useful discussions during my Ph.D. study. I feel very lucky that I had a chance to work with him. Steve's encouraging words and his wealth of knowledge in information retrieval have been a constant source of inspiration and have greatly contributed towards the completion of this dissertation. I thank Steve for recruiting me as his student from China, for having been with me every step of the way in the development of this dissertation, for providing wise suggestions which made the achievements in this work possible, and for supporting me to attend the TREC conferences in Washington D.C.

I am deeply indebted to my parents, Wenfu Liu and Professor Shoumeng Huang, and my sister for their strong support, incredible understanding and encouragement which kept me going for these years. This accomplishment is also theirs. Also, I would like to thank my wife, Dr. Aijun An, who helped me through both the intellectual and emotional development in undertaking this work. She is always willing to listen and share her ideas with me. This dissertation could not have been completed without her love, encouragement and emotional support throughout this long process. I would also like to thank our daughter, Angela, who has been my moral support in the final stage of this thesis. Thanks for her "understanding" when I was away to Cambridge for a conference with the first draft of this thesis a week after she was born.

Many thanks to the following people who have made suggestions to this work: Professor Nick Cercone and Professor Michline Beaulieu. I would like to sincerely

thank Professor Alan Smeaton and Gareth Jones for being my external examiners and for providing constructive comments and suggestions to this dissertation. Thanks Professor Bernie Cohen for serving as the chair of my viva. I would like to express my sincere gratitude to Maria Dexter, Andrew MacFarlane, Maxine Walsh, Sylvia Bugg and Jan Watkins for their help at City University when I was away in Canada. I also would like to thank my officemates Mike Dilworth, Rachel Soper, Simon Powell and Yasemin Kural for their friendship at the early stage of this research. Thanks also go to Denis Kelleher and Maureen Cusick for their friendship and help when I was a student tutor at Walter Sickert Hall of City University. It was very memorable and enjoyable to live there for two years.

This research was supported by ORS award from Committee of Vice-Chancellors and Principals of United Kingdom and Centenary Scholarship from City University. The author would also like to thank University of Waterloo and University of Regina in Canada for providing computer facilities for conducting the experiments reported in the thesis.

Declaration of Copyright

Abstract

Using probabilistic methods to retrieve information has always been a challenging task in the area of information retrieval. A key issue in probabilistic retrieval methods is the design of query term weighting functions. In this thesis, we provide a comprehensive description of the probabilistic retrieval model and propose several new weighting functions, which include both single unit weighting and compound unit weighting functions. Detailed analysis and evaluation of these new weighting functions are also provided.

This thesis provides a large number of empirical results for comparing different weighting methods in Chinese word-based and character-based retrieval systems. The results show that (1) compound unit weighting is useful for improving the system performance; (2) a newly designed single unit weighting function, BM26, contributes to the improvement of Chinese information retrieval; (3) the character-based system outperforms the word-based system in terms of average precision.

The thesis makes three original contributions to modern information retrieval. First, it demonstrates that probabilistic compound unit weighting is useful for Chinese information retrieval systems. Second, it proposes a new probabilistic single unit weighting function, BM26, that considers document lengths when assigning weights to documents, and it demonstrates that the new function outperforms the function that it evolved from. Third, this thesis reports the results of large scale experiments that compare Chinese word-based and character-based retrieval systems.

In summary, the thesis combines a comprehensive description of the probabilistic model of retrieval with some new designs of probabilistic weighting formulae and new systematic experiments on the Chinese TREC Programme material. The experiments demonstrate, for a large test collection, that the probabilistic model is effective and robust for Chinese text retrieval, and that it responses appropriately, with major improvements in performance, to key features of retrieval situations in Chinese text retrieval.

Chapter 1

Introduction

1.1 Research Objectives

Information retrieval (IR) is concerned with the organization and retrieval of information from a large number of documents. The objective of information retrieval is to locate relevant documents based on user input. Different models have been used to model the retrieval process, such as the probabilistic model [85], the vector space model [105] and the regression model [24]. The historical root of the use of probabilistic methods in information retrieval can be traced back as far as the early sixties when Maron and Kuhns first presented the probabilistic approach to IR [70]. However, the early ideas never took hold. It was only in the 1970s that some significant headway has been made with probabilistic methods [83, 85, 86]. Since the 1970s the probabilistic model has been elaborated in different ways, tested and applied, especially in work by Robertson and his colleagues at City University [93, 96]. As implemented in their Okapi system, the probabilistic model has been subjected to heavy testing in the very large evaluation programme represented by the NIST Text REtrieval Conferences (TRECs [1]). However, there has been very little use of the probability theory in modeling Chinese information retrieval. Most of the experimental results obtained so far have been for English. Dealing with searching over other languages, such as Chinese, has rarely been attempted although there is currently much work on retrieval in different languages and across

[1]TREC is an annual conference organised by NIST (National Institute of Standards and Technology), starting in 1992. Its purpose is to support research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies.

languages in the past few years, some of which is based on probabilistic methods. The rapid growth of Chinese texts in Internet and globalization of information communication has made Chinese IR an important research topic. This thesis is intended to investigate the effectiveness of using probabilistic methods in Chinese information retrieval.

The rapid increase in the number of available on-line databases has accentuated the importance of computer-based information retrieval methods. These databases contain a huge variety of information, including the full text of newspaper stories, stock quotations, airline schedules, journal abstracts, descriptions of chemical compounds, art reproductions, instruction manuals, library catalogs, census data, and so on. The majority of this information is in the form of text, and, although the amount of digitized image and video data is growing, text will continue to be critical due to its unique role as a medium for communication. The most interesting thing about text, and the central problem for designers of information retrieval systems, is that the semantics of text is not well represented by surface features such as individual words. Consequently, the performance of retrieval systems that rely on matching exactly a text's surface features to the features specified by searchers will be fair at best. What we focus on in this thesis is textual IR and we do not study any other type of data such as speech, images or multimedia. We largely concentrate on what we regarded as the main tasks in textual IR such as indexing and searching. The indexing task is the process of taking raw text and building an index over that text and the search task is the process of servicing queries to produce a ranked list of documents.

Both indexing and searching tasks require processing of the natural languages in documents and queries. One way to process natural language texts in IR is statistical. In this approach, linguistic units are extracted from the documents and queries. Those units extracted from documents are used to index them and those extracted from a query are weighted according to their degrees of importance. The most highly weighted units are chosen as query terms. An information retrieval system takes the selected query terms, matches the terms with the indexes of documents, calculates a retrieval status value for each matched document using a term-weighting method, and presents the user with potentially relevant documents

from a collection of documents. The performance of the information retrieval system in identifying relevant documents greatly depends on the text processing method and term weighting method.

Our first primary aim is to investigate the effectiveness of different text extraction methods in the context of Chinese information retrieval. It is a well known problem that there is no separator between Chinese words so that Chinese words cannot be easily used to index or search texts as is possible in English. Therefore, some people use characters or n-grams as searchable tokens, instead of words. We discuss two text extraction methods. One extraction method uses words, compound words and phrases in the document and query texts as indexing terms to represent the texts. We refer to this method as the word-based approach. For this approach, text segmentation, which divides both document and query texts into linguistic units, is regarded as a necessary precursor [142]. The other extraction method is based on single Chinese characters, in which texts are indexed by the characters appearing in the texts [16]. By using single character approaches, a search could be conducted for any multi-character word or phrase identified at search time, no matter whether this word or phrase is in the dictionary. In China, almost all practical systems are still based on the word approach. There has been very little discussion on using the character approach for Chinese IR indexing, let alone using probabilistic methods for retrieval. However, both word-based and character-based methods have been used in information retrieval systems outside China. For examples, Cornell's SMART system [10] uses character-based retrieval augmented with character bigrams. The Chinese retrieval system in Berkeley [34] is a purely word-based system, which uses a dictionary of 140,000 words, to segment the Chinese collection and queries. Queens College's PIRCS system [62] applies a combination of dictionary and statistical techniques to detect 2, 3 and occasionally 4 character words. Its aim was to obtain good indexing features rather than "correct" segmentation. We use both word-based and character-based segmentation methods in our Okapi retrieval system [5, 48, 50, 52, 53]. In this thesis, we present the two methods used in Okapi and provide empirical results that compare the two methods in terms of their effectiveness for Chinese text retrieval in Okapi.

Our second primary aim is to investigate the effectiveness of different term

weighting methods in the context of Chinese information retrieval. An important issue in text retrieval is how to make use of the extracted terms in the retrieval process. A usual way is to weight the query terms and calculate a retrieval status value for each matched document based on the terms' weights. Statistical term-weighting methods for IR have traditionally taken two forms: formal approaches, where an exact formula is derived theoretically, and ad-hoc approaches, where formulae are tried because they seem to be plausible. Both categories have had some notable successes [10, 34, 62, 95]. A problem with the formal model approach is that it is very difficult for a model to take into account the wide variety of variables that are thought or known to influence retrieval. The difficulty arises either because there is no known basis for a model containing such variables, or because any such model may simply be too complex to give a usable exact formula. A problem with the ad-hoc approach is that there is little guidance as to how to deal with specific variables. In Okapi, a complex formal probabilistic model is used to take a model which provides an exact but intractable formula, and use it to suggest some much simpler formulae. In this thesis, we describe several probabilistic weighting formulae (such as BM11 and BM25 [96]) used in Okapi, propose some new formulae and provide empirical results that compare these formulae coupled with either word-based or character-based text processing methods in terms of their effectiveness for Chinese retrieval. The probabilistic weighting formulae we proposed concentrate on single unit weighting and compound unit weighting for Chinese terms.[2]

For the experiments presented in this thesis, we use a standard collection of Chinese documents and queries provided by NIST. The document collection and queries have been used by participants in TREC. Evaluation of Chinese information retrieval systems was included at the fifth and sixth TREC conferences (TREC-5 and TREC-6), where a large collection of Chinese documents and two sets of queries (one for TREC-5 and the other for TREC-6) were provided.

In summary, the primary motivation for this thesis is to answer the following questions: (1) Can the probabilistic model be used in Chinese information retrieval successfully? (2) What are the suitable probabilistic weighting formulae for

[2]See chapter 4 for the definitions of single unit and compound unit.

Chinese information retrieval? (3) How should a weighting scheme for Chinese language material deal with compound units? (4) Is there a better weighting formula than BM25 for handling the document length variable on the Chinese TREC-5 and TREC-6 datasets? (5) Are word-based document indexing methods better than character-based document indexing methods in terms of average precision for Chinese information retrieval?

1.2 Main Findings

We list the main findings of this thesis according to the above objectives as follows:

- The probabilistic model interpretations described for English carry over to the Chinese language, even if some special single unit weighting and compound unit weighting formulae need to be designed for Chinese IR systems.
- In terms of single unit weighting, the document length correction factor in BM25 is not designed properly. A new correction factor is proposed and it makes a significant positive contribution to the quality of retrieval compared to using BM25.
- Compound unit weighting is useful for improving the system performance, especially for character-based retrieval systems. However, it is important to determine a good method for weighting compound units.
- Accurate word segmentation is not a pre-requisite for effective Chinese IR. Character-based document processing is better than word-based document processing in terms of retrieval effectiveness.

1.3 Overview of the Dissertation

The thesis is divided into nine chapters and includes appendices and a list of references. These nine chapters are organized as follows: Chapter 2 provides background information on information retrieval and issues related to Chinese text retrieval. Chapter 3 describes the latest probabilistic 2-Poisson retrieval model

and related probabilistic weighting formulae. Chapter 4 proposes an improvement to the 2-Poisson model and some newly designed weighting formulae for single units and compound units. Chapter 5 describes the Okapi retrieval system and gives a detailed description of Chinese text retrieval with Okapi which includes the system architecture, the dictionary, algorithms and detailed designs. Chapter 6 describes the design of Chinese experiments and evaluation measures used in the experiments. Chapter 7 reports the experimental results on the Chinese TREC datasets and comparisons with some other TREC participating systems. Chapter 8 provides detailed analyses and discussions of the experimental results on the following three aspects: (1) comparison between word-based and character-based document processing; (2) comparison of different compound unit weighting methods; (3) comparison of different single unit weighting formulae. Finally, we conclude the thesis in Chapter 9 with a summary of the major contributions of the presented research and with suggestions about the directions for future research.

Chapter 2

Background

2.1 Overview of Information Retrieval

Information Retrieval is concerned with the representation, storage, and accessing of documents [113]. The popularity of online information services such as LEXIS/NEXIS [1] and, more recently, the popularity of the World Wide Web and electronic publishing have resulted in a tremendous increase in the number of documents available in machine-readable form. Consequently, the ability to effectively search for documents relevant to a particular information need has become very important and research in this area has received a lot of attention.

2.1.1 Information Retrieval Problems

Information Retrieval problems can be summarized as: given a user's "information need" in the form of a written query, find those documents amongst a possibly huge and changing electronic collection (or corpus) which satisfy the user's need.

In the past few decades, the availability of cheap and efficient storage devices and information systems has prompted the rapid growth and proliferation of relational, graphical, and textual databases. Information collection and storage become easier, but the effort required to retrieve relevant information has become significantly greater, especially in large-scale databases. This situation is particularly evident for textual databases, which are widely used in traditional library

[1] LEXIS/NEXIS [65] is an online information retrieval system providing access to a wide range of legal, business, and government sources, including full-text and abstracted information from newspapers, news and business magazines etc.

science environments, in business applications (e.g., manuals, newsletters, and electronic data interchanges), and in scientific applications (e.g., electronic community systems and scientific databases). Information stored in these databases often has become voluminous, fragmented, and unstructured after years of intensive use. Only users with extensive subject area knowledge, system knowledge, and classification knowledge are able to maneuver and explore in these textual databases effectively [14].

In conventional information retrieval environments, keywords are manually or automatically assigned to documents and queries are formulated by using terms interconnected by Boolean operators. Although widely used, Boolean query languages have some drawbacks: users find it difficult to formulate their queries using Boolean semantics; the retrieved documents are not ranked in any particular order; and most importantly, the retrieval results are often inadequate [110, 112]. The vocabulary problem in human-computer interactions further confound the keyword-based Boolean retrieval mechanism [17]. In [32], Furnas *et al.* found that in spontaneous word choice for objects in five domains, two people favoured the same term with less than 20% probability. This fundamental property of language limits the success of various design methodologies for keyword-driven interaction.

Many approaches to the IR problem are possible, and some work better than others. As in nearly any problem solving situation, there is a tradeoff that must be considered when designing a system to handle IR: speed and efficiency versus accuracy and performance. Typical Web search engines opt for speed, much to the chagrin of users who must then wade through pages of irrelevant documents. On the other hand, the kinds of systems typically studied by IR researchers do not always scale up nicely to collections as large as the Web, but often perform much better than typical Web search engines.

The typical scenario which unfolds when a user sits down in front of an IR system goes like this:

1. The user has an information need in mind, for example, "What is the proper way to care for orchids?"

2. The user then formulates a written query, the details of which will depend on the particular IR system being used, and submits it to the system. Typical

queries in this case might be "growing orchids" or "orchid or (garden and perennial)".

3. The system "does its magic" and displays to the user a list of documents which may be relevant to the query, in the following fashion:

   - Some summary of each document is shown, such as title, abstract, or some other automatically generated summary. Some capability to access the entire document is often also available.
   - The documents are usually ranked according to how probable it is that each is actually relevant to the query.
   - Only a fixed number of the top-ranked documents are displayed, since the user will most likely not be interested in low-ranked documents.

4. Depending on the system, the user might then be given the option of modifying her original query, or giving feedback as to which documents look promising, thus allowing the system to modify its approach and return a different (or differently ranked) set of documents.

This process is illustrated graphically in Figure 2.1

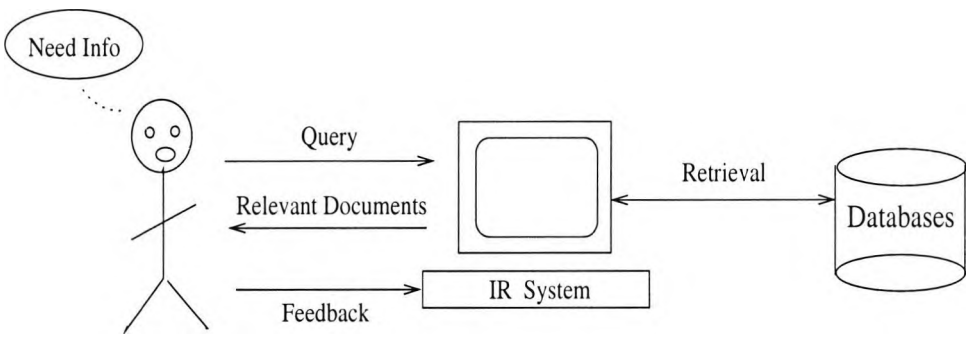


Figure 2.1: The Information Retrieval Process

General Approaches to IR

The number of different approaches to IR is very large and varied. However, most follow a similar general formula. Each document is somehow represented by a number of "features" which attempt to capture the document's semantic content. Typical features include terms (words and morphological variants thereof) and

phrases. Such features are also used to represent the query. A matching function is then used to compare the document's representation to the query, generating a score known as the Retrieval Status Value (RSV). RSVs can be boolean (0 or 1) or real-valued, and may or may not attempt to model the actual probability of relevance. Once RSVs are generated for all available documents, they are used to sort the documents and select the top-ranked ones for presentation to the user.

The variation in IR approaches comes from the selection of (a) which features to use, (b) how to combine them into document and query representations, and (c) what matching function is used to generate RSVs. Section 2.1.5 will describe several specific instances of IR systems.

IR Tasks: Retrospective, Routing, Classification

The above step-by-step description most accurately reflects a particular instance of the IR problem known as the retrospective (or ad hoc) task. The retrospective task is characterized by a static collection of documents, against which new queries are compared. This is typified in the instance of searching the Web using one of the Web "search engines" that allow the user to type in a natural language or possibly structured query.

In contrast, in the routing (also referred to as profiling or filtering) task, the document collection is dynamic and the queries are static. A typical example of routing is when a stock analyst who is interested in a particular financial sector creates a detailed query to run against the newswire in order to keep abreast of information relevant to his particular stocks. In this situation, the scores returned by an IR system are often interpreted as probabilities of relevance because the documents are sometimes immediately discarded. Thus, a closed-end scale is needed so a cutoff value can be used to determine which documents to present to the user.

A third common IR task is categorization (or classification) in which each document must be placed into one or more predefined categories. Categorization is closely related to routing in the following way: both require the creation of a classifier. For routing, the classifier decides whether the document is in the class of documents relevant to the particular query it was designed for. Thus, one technique for doing categorization is to train a fleet of classifiers, one for each

possible category.

Usually, IR research focuses on one of these three tasks. However, the separation is somewhat artificial because the three overlap heavily, and it is rarely the case that one can pay attention to one without also being able to apply a similar technique to the others.

2.1.2 Relevance

Creating new and better IR systems would be impossible if there were no definition for "better." Evaluation of IR systems hinges on the notion of relevance. Because of the nature of the IR problem, relevance is necessarily mostly subjective. In other words, only the user who initially had the information need can truly determine the relevance of any particular document. This creates a problem, since document collections often contain tens of thousands or even millions of documents. It becomes impossible to measure true relevance because no human could possibly read all of the documents and generate what are known as relevance assessments. Another issue with relevance is the degree of relevance. For ease of evaluation reasons, relevance has traditionally been a binary concept: a document is either relevant to a query or not. Current evaluation techniques (see below) do not allow for a continuum, even though such a model is more intuitively appealing.

2.1.3 Evaluation

IR systems are typically evaluated in terms of their performance, not efficiency. By performance, we mean how well they are able to satisfy the user's information need. Efficiency, on the other hand, is a measure of how fast they can do so. Web search engines need to be highly efficient because it seems that the users may not happy to wait for ten seconds to get results, and thus often sacrifice performance (although this is changing rapidly). Traditionally, however, research into IR systems has focused on performance as an evaluation measure.

The two most accepted measures of IR system performance are precision and recall. **Recall** is defined as the ratio of the number of relevant documents retrieved to the total number of relevant documents in the collection, whereas **precision** is defined as the ratio of the number of relevant documents retrieved to the total

	RELEVANT	NON-RELEVANT	TOTAL
RETRIEVED	$A \cap B$	$\neg A \cap B$	B
NOT RETRIEVED	$A \cap \neg B$	$\neg A \cap \neg B$	$\neg B$
TOTAL	A	$\neg A$	D

Table 2.1: The 'Contingency' Table

number of documents retrieved (regardless of relevance) by the system. These two measures are widely used, for example, in the TREC conferences to measure retrieval effectiveness of different retrieval systems. It is helpful at this point to introduce the famous 'contingency' table [84], as shown in Table 2.1, in which D is the collection of documents in the system; A is the set of relevant documents; and B is the set of document retrieved.

The above two measures of performance can be derived from this table as follows:

$$PRECISION = \frac{|A \cap B|}{|B|}$$

$$RECALL = \frac{|A \cap B|}{|A|}$$

where $|A|$, $|B|$ and $|A \cap B|$ are the counting measure of A, B and $A \cap B$ respectively.

Precision is a measure of the quality of retrieval and Recall measures the coverage of retrieval. Maximizing one of these two measures typically minimizes the other. This tradeoff can be seen in what is a common view of IR system performance: the precision versus recall graph. Precision is calculated at different levels of recall, and these scores are then graphed. A typical precision/recall graph is shown in Figure 2.2. Note that high precision corresponds to low recall and vice-versa. The ideal IR system would have a horizontal line across the graph at precision level 1.00, which would correspond to retrieving all and only the relevant documents. It is useful to have a single number which summarizes the performance of an IR system. One such number is the average precision, which corresponds to the area under the precision/recall curve. In practice, average precision is often approximated by averaging precision at some number of recall points.

Using average precision as a single number measure is not without its faults. In some situations, retrieving all of the relevant documents is not critical, only a certain number are needed by the user. Thus, sometimes precision at a certain

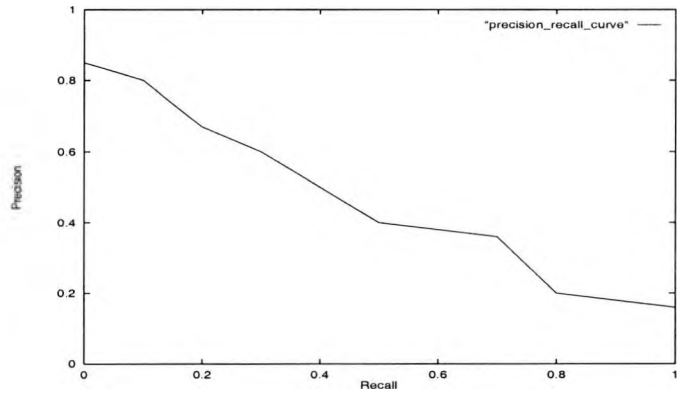


Figure 2.2: Precision versus Recall

number (on the order of 20) of retrieved documents is used, taking into account that the user might only look at the most highly ranked documents. Another measure, most often used in the routing task, is utility. **Utility** takes into account that retrieving relevant documents may have comparatively more or less value than not retrieving irrelevant documents [54]. Utility is calculated as: $U = a * N_{relevant} - b * N_{nonrelevant}$. The parameters a and b can be set to reflect the relative value (e.g. cost) of relevant or irrelevant documents, with typical values in the range from 1 to 3. Thus, utility is a parameterized performance figure, and also depends on the number of documents retrieved.

2.1.4 Relevance Feedback

A user's query is invariably vague and inaccurate for a number of reasons. Once he or she has seen some documents, the user can provide feedback (Step 4 in Section 2.1.1) by marking some of the retrieved documents as relevant or not. IR systems typically make use of this relevance feedback by modifying their internal representation of the user's query, most often by adding terms from the relevant documents, or weighting some terms more heavily than others based on how often they occur in the marked documents. The addition of terms can be done automatically (query expansion) or a list of terms may be presented to the user, who then modifies the query (query refinement). This use of relevance information has been found to almost universally improve IR system performance [8, 30].

Another form of feedback is pseudo-relevance feedback, in which a certain number of the top-ranked documents are assumed to be relevant, and then used to

modify the query [62, 81].

Other IR systems make use of relevance information in less direct ways. These kind of systems use various machine learning techniques to incorporate relevance information [3, 141].

2.1.5 Approaches to Information Retrieval

IR systems can be classified according to which features they use, how they combine these into document and query representations, and their matching function. Every IR system can be described as consisting of a collection of documents, a set of information requests, and some mechanism of determining which, if any, of the documents meets the requirements of the request [105]. Those documents that meet the requirements of the request are called relevant documents. The notion of relevance is central to information retrieval and forms the basis of two measures that are widely used to evaluate retrieval performance: recall and precision as discussed in section 2.1.3. When calculating recall and precision, relevance is assumed to be dichotomous: a document is either relevant or not relevant. There are no in-between states.

Early IR systems were built in a period when computer storage was expensive and as a result, they seldom stored the complete text of each document in the natural language in which it was written. Instead, these retrieval systems stored a representation of the document (i.e. a surrogate) which may be produced either manually or automatically [84]. The process of creating a representation for the documents is known as indexing. As computer storage has become less expensive, it is quite common nowadays for IR systems to store the complete texts of the documents. Nevertheless, indexing is still necessary to create document representations that are more efficient for computer processing. Indexing involves assigning appropriate identifiers capable of representing the content of the documents in the collection, and is a crucial operation required for information retrieval [105]. The indexing operation actually has two components: content analysis that results in the selection of the concepts to represent the document; and expression of the selected concepts in identifiers taken from a vocabulary known as the indexing language [29]. An indexing language is controlled if the indexer can only choose

from a list of approved index terms; and uncontrolled if the index terms come from the texts of the documents [84]. To index a document manually, an indexer who is normally an expert in the subject area selects terms from the indexing language that will describe the information content of the documents in the collection. Manual indexing has been done for many years in standard library environments. As one can see, this is an expensive and time-consuming operation, since the indexer has to be trained and he/she has to scan/read the document in order to assign subject terms. Furthermore, even trained indexers may be inconsistent in their choice of terms and experimental studies have shown that indexers are often in disagreement among themselves over how to index a document, and that they frequently index the same document differently at different times [29]. Indexing consistency is important as similar documents should be identified by comparable indexing entries. Given the problems associated with manual indexing, automatic indexing becomes appealing.

The starting point of the automatic indexing process may be the complete document text, an abstract, or even the title only, and the output of this process is a representation of the document which the computer can handle [84]. Most automatic indexing efforts are based on Luhn's idea that the frequency of occurrence of individual words can be used to measure their significance [68]. Salton and McGill [105] proposed an automatic indexing process which works as follows:

1. Identify individual words that constitute the documents: Each document is parsed into individual words and rules are applied to handle special situations such as hyphenated words. While this is relatively straightforward for languages such as English where words are delimited by spaces, word identification is not trivial for languages such as Chinese where words are not delimited explicitly.

2. Remove high frequency stop words: Stop words are high frequency function words which cannot possibly be used by themselves to identify document content. Examples of such words are *an, and, by, or, the, he and she.* Stop words contained in the documents are eliminated from the analysis process.

3. Reduce words into word stems by removing suffixes: This process, commonly

known as stemming, reduces morphological variants of words into their root form. For example, the words "developer", "development", "developments" and "developing" are all stemmed to the same word "develop". When stems are used as index terms, a greater number of potentially relevant documents can be identified.

4. Select index terms from stems based on their importance: Salton [103] suggested using the inverse document frequency function to measure term importance. The assumption is that the importance value of a term is inversely proportional to the total number of documents in which the term appears. Words with high importance values can be chosen as index terms.

5. Represent each document by a chosen set of index terms: Index terms that occur in a document are assigned to represent that document, with or without a term weight. In a binary indexing system, each index term that appears in a document is assigned a weight of 1 regardless of how many times it occurs in the document, whereas in a weighted indexing system, a term weight, usually calculated based on the term's frequency of occurrence, is used to reflect its importance.

However, it is not quite clear how these methods could be applied to Chinese information retrieval, in particular the methods for stemming words and the methods for removing stop words. Experiments comparing the retrieval performance of manual indexing and automatic indexing have shown that the latter performs as well, if not better, than conventional manual indexing methods [105, 108]. As the number of documents available online is increasing dramatically and timely access to these documents is becoming critical, automatic indexing becomes necessary as the time required for manual indexing is too long.

The index terms assigned to each document are commonly stored in a file structure known as the inverted file. An inverted file contains a sorted list of index terms with each having a link to a posting list, which contains information about the term's occurrence in the collection (e.g. the identifiers of all the documents assigned to that index term). The use of an inverted file allows efficient implementation of the search operation. Most commercial information retrieval systems are

built upon inverted files.

A retrieval model specifies the representations used for documents and information needs and how they are compared. It is often used as the basis for classifying IR systems. Many commercial retrieval systems, such as LEXIS/NEXIS, Dow Jones News/Retrieval, and World Wide Web search engines such as AltaVista, support what is known as the Boolean model.

In the Boolean model, retrieval is based on the concept of an exact match of a query specification with one or more documents [6]. Queries are expressed as words or phrases combined using the standard Boolean operators (e.g. AND, OR, NOT). Documents containing the words and phrases satisfying the Boolean conditions specified in the query are retrieved, and the retrieved documents are not ranked — there is no distinction made between any of them. For example, to search for documents on the topic of information retrieval, the user will typically enter a query like **"information AND retrieval"**, and all the documents containing both the words will be returned unranked. This model works well with binary indexing, as all the retrieval system needs to know is whether a term appears or not in the document.

Although Boolean systems are popular because efficient implementations are available, the Boolean model has several major drawbacks [7, 140]. First, by treating all matching documents the same way, the unranked results produced by a Boolean retrieval system is often counter-intuitive. For example, when a user enters the query **"A OR B"**, documents that contain only one instance of term A is considered equally relevant as documents that contain many instances of both term A and B. Second, query terms are considered equally important and therefore for a query like "A OR B", regardless of the fact that term A may be more important than term B, the system will treat all documents containing either term A or term B or both the same way . Third, Boolean query formulation is not natural to many users and systems that can process natural language queries are more desirable. Fourth, it is difficult to control the number of documents retrieved. To alleviate some of these problems, some efforts have been made to extend the Boolean model [7, 106], but those efforts have not been tested in large experiments.

Another popular retrieval model, the vector space model, addresses some of

these problems. In the vector space model, documents and queries are represented as vectors in a multidimensional space, where each dimension corresponds to an index term used in representing the texts [6]. During the retrieval process, the similarity between the vector representing the query and the vector representing each document are compared using, for example, the cosine correlation measure. The assumption is that the more similar the document and the query vectors are, the more likely that the document is relevant. The retrieval result is a ranked list of documents based on the similarity values. Information retrieval systems that support the vector space model, such as the SMART retrieval system [104] and Lycos [69], are usually based on a weighted indexing system and documents are represented as a vector of term weights. Experiments have shown that term weights calculated based on each term's within-document frequency (tf) and inverse document frequencies (idf), produce good retrieval performance [109]. With this method (known as $tf \times idf$), each term in a document vector and the query vector is given a weight proportional to its within-document frequency and inversely proportional to the number of documents in the collection containing that term. If the variation in document lengths in the collection is large, the term weights in a vector are often normalized by dividing each of them by a factor calculated based on the length of the document.

The term weighting techniques used in the vector space model are often derived from heuristics and have been criticized because they lack well-substantiated theoretical properties. In contrast, the probabilistic model of information retrieval is based on the Probability Ranking Principle [87, 85] and has a sound theoretical background. The Probability Ranking Principle states that given a query, an information retrieval system should produce a ranked list of documents based on their probability of relevance to the query. Term weights are derived from relevance properties of the documents: the weight of a term is defined as the proportion of relevant documents containing the term divided by the proportion of non-relevant documents containing the term. The calculation of this weight requires knowledge about the occurrence statistics of the term in relevant and non-relevant documents in the collection. In the absence of this information, the term weight can be reduced to a form similar to the inverse document frequency component used in the

vector space model. The probabilistic model can be extended to include within-document term frequencies [93], and the resulting term weights are very similar to those lately used in the vector space model [9].

All the retrieval models rely on some form of keyword matching in order to identify relevant documents: documents are retrieved only if they contain index terms specified in the query (except when the Boolean NOT operator is used). Therefore, regardless of the model used, retrieval performance depends heavily on the query submitted by the user. If the user formulates a query with terms that are different than those used to index the majority of the relevant documents, many relevant documents will not be retrieved. Users with little knowledge about the documents in the collection often have difficulties in selecting a good set of terms to describe their information needs. As a result, a variety of query refinement techniques have been proposed, usually based on an iterative process: A better query is formulated in each step based on the results of previous steps.

The most straightforward query refinement technique is for the user to read the documents retrieved by the initial query, add to the query terms that appear prominent in relevant documents, as well as remove from the original query terms that appear to retrieve a lot of irrelevant documents. With this technique, the information retrieval system provides no assistance to the user, who has to read the retrieved documents carefully and modify the query appropriately.

To partially automate this refinement process, the relevance feedback process was introduced in the mid-1960s. It involves several steps [91, 111]:

1. The user submits an initial query.

2. A set of ranked documents are returned.

3. The user judges each document (or the top ranked documents), and indicates to the system whether each document is relevant or not.

4. Based on the user's relevance judgment, the system modifies the original query by adjusting the weights given to the query terms and by adding important terms from documents identified as relevant.

Relevance feedback was originally developed for the vector space model. Term importance is normally calculated based on the distribution of the term in relevant

and non-relevant documents. If a term appears in many relevant documents and relatively few non-relevant documents, then the term is assumed to be related to the user's information need and is added to the original query, or given a higher weight if it is already in the query. However, this process still requires the user to read the documents retrieved and make relevance judgments. In pseudo relevance feedback, which is a fully automated version of the relevance feedback process, the top ranked documents (e.g. the top 20 documents) retrieved by the initial query are assumed to be relevant and analysis on term importance is done based on the assumed relevance of the documents. Relevance feedback techniques have been shown to be able to improve retrieval performance quite significantly [91, 111].

Incorporating relevance feedback into a Boolean retrieval system is more difficult as standard Boolean queries do not have weights associated with query terms and in addition to adding extra terms to the query, a Boolean operator has to be chosen as well. Some research in this area has been done [107], but more experiments are required to validate their effectiveness.

2.2 New Requirements for Modern IR Systems

The previous section provides a quick overview of information retrieval. Most of the techniques were developed in the 1960's and have been refined many times by different researchers. However, recently trends have emerged or are emerging in information retrieval environments that call for the provision of new features in modern information retrieval systems: e.g. support for full-text retrieval; support for multiple languages; and information retrieval from network. Each requirement is described in more details below.

2.2.1 Full-Text Retrieval

The complete texts of research articles, books, newspaper stories, magazines and so on, are now widely available in machine-readable form. For example, the full-texts of magazines and transactions published by the IEEE Computer Society are accessible through the World-Wide Web. Retrieval systems providing access to the complete texts of documents so that every word in the entire collection

(except designated stop-words) can be searched are known as full-text retrieval systems [130]. Their popularity has increased rapidly in the last 10 to 15 years [130]. Full-text retrieval systems rely on automatic indexing and offer advantages such as timely access to the documents; ease of use for inexperienced searchers as they can enter queries in the authors' natural language; the ability for a user to immediately judge relevance; and the ability to locate valuable information that is peripheral to the main focus of the document [130].

Most of the information retrieval models are originally developed to process abstracts and document surrogates that are much shorter than the complete document. Index terms are treated equally regardless of where they occur in the document (although some Boolean retrieval systems provide proximity operators that, for example, allow the user to search for terms within a paragraph or a sentence or a fixed number of words). For abstracts and other short surrogates this is reasonable as they usually contain terms that are closely related to each other. However, long documents like newswire articles often contain multiple topics, and terms appearing in one part of the document may be totally unrelated to terms appearing in other parts of the document. For example, if both the words "Pentium" and "testing" appear in a full-text document, it is not necessary that the document is about testing of the Pentium chip. The document could be a newswire article where the word "Pentium" appears in the top of the document in a story about the price of computer chips being reduced, while the word "testing" appears in at the end in a story about DNA testing being used in a court case. On the other hand, if both of these words appear in an abstract it is much more likely that the document is related to testing of the Pentium chip. Relevance feedback is particularly problematic in full-text retrieval systems. Query expansion through relevance feedback usually involves adding to the query important terms extracted from relevant documents. The relevance feedback process does not take into account where those terms occur in the document. As a result, terms from non-relevant portions of the documents may be chosen and added to the query, thus degrading the original query. Therefore, it is clear that one should take into account the proximity between the terms in full-text retrieval environments.

2.2.2 Support for Multiple Languages

As electronic processing of documents is becoming more and more popular in every part of the world, there is a need for information retrieval systems that can effectively process documents written in languages other than English. IR is a well-established field especially in the context of English and other Western languages that share the same characteristics of easily distinguishable word and word phrases. Supporting languages similar to English is relatively straightforward, but supporting languages that are very different from English, such as Chinese, is more difficult. Written Chinese texts appear as sequences of characters and punctuation with no delimiters to mark the word boundaries. A character (or ideographic symbol) can itself be a word, or can form multi-character words with adjacent characters. The process of determining the boundaries of words in a Chinese character string is known as text segmentation.[2]

Usually, word segmentation is one of the initial steps in automatic text analysis and indexing. With unsegmented Chinese texts, a decision has to be made on what constitutes a "word". One option is to first segment the text and identify word boundaries. Although there has been much research in segmentation [15, 18, 72, 142, 143], Chinese word segmentation is still considered a difficult problem and even humans often disagree on how to segment a piece of text. Most segmentation software also requires a large dictionary or a large training corpus, and both are expensive to maintain. Another option is to treat individual characters (unigrams) or all bigrams (adjacent overlapping character pairs) as "words". Regardless of the indexing strategy, retrieval systems should be able to maintain a good level of retrieval effectiveness. Support for multiple languages is especially important for World-Wide Web based search engines which have to index a large number of documents written in many different languages.

2.2.3 Information Retrieval from Network

With the Internet growing rapidly in popularity, the availability of information becomes less of a problem as more and more information becomes available online.

[2]In some literatures, text segmentation is also referred to as word segmentation. A detailed description of Chinese text segmentation will be given in section 2.3.3.

In particular, with the World Wide Web (WWW), a large amount of digitally stored information is readily available to anyone who has Internet access. However, what is becoming more of a problem with the growth in the amount of online information is a phenomenon known as information overloading. In other words, there is so much information available such that it becomes increasingly difficult and time-consuming for the user to find the information relevant to his needs.

The explosive growth of unstructured information on the Internet and in particular the WWW has greatly increased the need for information retrieval systems. The significance of IR techniques is clear. Conventional IR techniques has long been emphasized in fundamental research, such as keyword extraction and indexing, full-text searching, term weighting, document ranking, relevance feedback, etc [105, 108]. Having been thoroughly studied, these conventional IR techniques are well established and have been successfully applied in retrieving English. Recently, researchers have begun to move in new directions in exploring more advanced techniques, for instance, Web spiders and Internet searching tools for Networked IR [1, 13, 57, 59, 69, 79, 144], information filtering and adaptation techniques for Intelligent IR [6, 25, 64], and speech and audio retrieval techniques for Multi-media IR [36, 126].

Information retrieval from network is evident from the rapid growth of the number of WWW search engines, such as AltaVista [1], Excite [28], Google [37], Infoseek [55], Lycos [69], Opentext [80], Webcrawler [137] and Yahoo [144] etc. These search engines have been developed for browsing and searching through these collections of highly unstructured and heterogeneous data. Although networked information retrieval is a promising research subject, it is not the focus of this thesis.

2.3 Chinese Information Retrieval

With global networking through the Internet, the number of electronic documents in Chinese and oriental languages has been rapidly increased. These documents are mostly non-structured and usually demand efficient IR techniques for retrieval. There is an increasing need to retrieve large numbers of such documents effectively and intelligently. Unfortunately, it is generally believed that due to the

inherent differences in languages, such as the lack of explicit separators, i.e. blanks or delimiters, in written oriental sentences to indicate word boundaries, the techniques developed for retrieving English documents can not be directly applied to retrieval of oriental language documents.

In this section, we will emphasize the significance of Chinese information retrieval and raise several important research issues which are fundamental and need to be further investigated. These issues include word and character indexing methods, dictionary, Chinese word segmentation, retrieval models and standard text collection for evaluation. In addition, we will point out some problems and requirements which have often been neglected in designing Chinese information retrieval systems. These include the need for adopting non-Boolean models, character-level indexing and best match searching.

2.3.1 Chinese Language

In the Chinese language each character represents at least a complete syllable, rather than a letter as in other languages. Each Chinese character is usually given a unique two-byte machine code. Many characters are also single syllable words. The total number of characters is therefore quite large and somewhat ill-defined. A literate adult typically recognises at least 5-6,000 characters. Various modern Chinese dictionaries usually define between 1,0-17,000 characters. For example, the most famous modern Chinese dictionary "CiHai" [145], which was published in 1989, contains 16,534 Chinese characters[3]. Ancient literatures contain even more Chinese characters, which may rise to approximately 60,000. "JiYun" [27] published in the Song dynasty has 53,525 characters, while the "KangXi Dictionary" [147] collects 47,035 characters.

Most modern Chinese words consist of more than one ideographic character and the number of characters in a word varies. A Chinese word is the minimal linguistic unit that can be used independently. For the convenience of discussion, we define Chinese words and phrases as follows.

Definition 1: A Chinese word $w = c_1c_2...c_m, m \geq 1$, where c_i is a Chinese

[3]Some new Chinese characters, such as the names for the newly-found chemical elements, have been invented since then.

character for $i = 1...m$.

Definition 2: A Chinese phrase $p = w_1 w_2 ... w_n, n \geq 2$, where w_j is a Chinese word for $j = 1...n$.

One estimation of the average length of a Chinese word is: $m \approx 1.5$ [136]. However, many terms (words or phrases), which are often of value in retrieval, tend to be longer[4]. The total number of Chinese words and phrases are hard to estimate. More than 120,000 Chinese words and phrases are included in the "CiHai" dictionary [145].

2.3.2 Word and Character Approaches to Indexing

One possible step during the indexing process is to identify individual Chinese words that constitute the documents. Word identification is non-trivial for languages such as Chinese. Unlike English texts where words are separated by spaces, Chinese texts appear as sequences of characters and punctuation symbols without spaces or explicit word boundaries. A character can itself be a word, or can form a short word with adjacent characters. Short words can be concatenated together to form longer words. Figure 2.3 shows a sample piece of Chinese text[5] taken from an ancient Chinese novel "Three Kingdoms". The reader should notice the lack of word boundaries in the text. Once the indexing units are defined, the rest of the text retrieval process for Chinese is virtually identical to English text retrieval.

There are two common approaches to indexing Chinese texts. One indexing method is to use words and phases in texts as indexing terms to represent the texts. We refer to this method as the word-based indexing. Identifying unknown and ambiguous words is the most outstanding problem for this indexing approach. The other method for indexing Chinese texts is based on single characters, in which texts are indexed by the characters appearing in the texts. This method is referred as to the character-based indexing.

For word-based indexing, text segmentation, which divides text into linguistic units (commonly words and phases), is regarded as a necessary precursor. But

[4]Longer retrieval terms usually have more specific meaning and therefore these terms are more useful for retrieval because they are less ambiguous.

[5]See Appendix *B* for its English translation.

〔明〕罗贯中

词曰：

滚滚长江东逝水，浪花淘尽英雄。是非成败转头空：青山依旧在，几度夕阳红。
白发渔樵江渚上，惯看秋月春风。一壶浊酒喜相逢：古今多少事，都付笑谈中。

第一回

宴桃园豪杰三结义　　斩黄巾英雄首立功

话说天下大势，分久必合，合久必分：周末七国分争，并入于秦；及秦灭之后，楚、汉分争，又并入于汉；汉朝自高祖斩白蛇而起义，一统天下，后来光武中兴，传至献帝，遂分为三国。推其致乱之由，殆始于桓、灵二帝。桓帝禁锢善类，崇信宦官。及桓帝崩，灵帝即位，大将军窦武、太傅陈蕃，共相辅佐；时有宦官曹节等弄权，窦武、陈蕃谋诛之，机事不密，反为所害，中涓自此愈横。

Figure 2.3: A Sample Piece of Chinese Text

Chinese text segmentation, which will be discussed later, is difficult - not even humans will always agree on correct segmentation. However, character-based indexing is a much simpler process. With character-based indexing, texts are divided into fixed-size overlapping blocks of characters and each block is used as an index entry. Character-based indexing is appealing because of its simplicity. Many of the problems involved in text segmentation, such as its inherent ambiguity (two people often disagree on how to segment the same piece of Chinese text), the need for large dictionaries or training samples, and the difficulty with handling new words and proper nouns, can be avoided. However, character-based indexing has many weaknesses, such as the demand for large space overhead, slower retrieval speed, the lack of high-level semantic information and poor retrieval precision, etc.

It has to be pointed out that each Chinese character and character bigram (each pair of adjacent characters) holds more semantic meaning than does a letter in English. In a Chinese document, though the composed words are difficult to correctly segment automatically, it is easy to extract all of the composed characters and character bigrams from the text. These characters and character bigrams hold certain semantics and form the features of the document. For example, there may be a three-character word 英国人 (British) in a document which cannot be correctly segmented, yet its decomposed characters and character bigrams, i.e., 英 (brave and handsome), 国 (country), 人 (people), 英国 (United Kingdom) and 国人 (citizen),

remain relevant semantics of the word. That is, if we use these characters and character bigrams to form the features of the word, then there still remain strong relationships between the word and other relevant keywords such as 中国 (China) and 美国人 (American people). In addition, the word can even be distinguished from irrelevant words, such as 信息 (information), 科学 (science) etc.

Bigram (overlapping character-pairs) indexing is a popular approach used by many probabilistic and vector space retrieval systems developed by those without much expertise in segmentation. For example, some groups participating in the Chinese track at the TREC conferences have no members who understand Chinese, and they simply indexed the Chinese documents as overlapping bigrams and applied their retrieval techniques developed for English [10, 113]. The reason for using bigram indexing is that using unigram indexing is too generic while using trigrams (three-character blocks) is too specific. A query submitted to a bigram indexing system is also divided into overlapping bigrams. These systems accept structured queries containing words linked together by Boolean operators. A string-matching operation involving the use of adjacency and ordering operators is used to search for multiple-character query terms.

It was originally thought that retrieval systems based on segmented texts would perform better than character-based retrieval systems. Kwok [60] recently compared the retrieval performances of unigram indexing, bigram indexing and word indexing using TREC-5 Chinese track data and topics. For word indexing, Kwok used a small dictionary of about 2000 words augmented by additional two- to four-character sequences that are most common in the documents. His results show that retrieval performance with bigram indexing is comparable with that of short-word indexing. Chen et al. [18] experimented with the same data and compared statistical and dictionary-based segmentation against unigram indexing, bigram indexing and trigram indexing. They found that on long queries, bigram indexing performs comparably with statistical segmentation and dictionary-based segmentation. For short queries, bigram indexing performs noticeably better than dictionary-based segmentation, but slightly worse than statistical segmentation. Both Kwok and Chen used relatively simple dictionary-based segmentation algorithms that did not involve syntactic knowledge. It is unclear whether bigram indexing will perform

comparably with more sophisticated segmentation algorithms.

2.3.3 Chinese Text Segmentation

Text segmentation is defined as the segmentation of texts into linguistic units, normally words and phrases. It is usually regarded as a necessary precursor to text retrieval. An English text is segmented into words by using spaces and punctuations as word delimiters. These words can then be used for indexing and retrieval. For text retrieval, what is at issue in any language is the automatic extraction of certain meaningful and content-bearing units, which may be morphemes, words, phrases, or some kinds of combination. But in a Chinese text, these units do not have natural boundaries separating one another. The absence of word boundaries poses a problem for Chinese text retrieval. Indeed, it is generally regarded as one of the biggest obstacles to computer processing of the Chinese language.

An English analogy to this problem occurs with phrases. Text retrieval in English is thought to benefit if suitable noun phrases (particularly those whose meaning bear little relation to the component words, such as "black hole") are identified and indexed. But the lack of obvious phrase boundaries renders this identification difficult. The Chinese problem is analogous in nature but very much more serious, because it applies to words as well as phrases. Various approaches to automatic text segmentation for the retrieval of Chinese texts are organised into a hierarchical tree in Figure 2.4.

Chinese text segmentation has been well-researched [15, 18, 72, 142, 143] and it often involves the use of dictionaries or character-occurrence statistics.

For a word-based Chinese information retrieval system, Chinese texts first need to be segmented into Chinese linguistic units. In order to do this, a Chinese dictionary, containing the segmentation units (usually Chinese words and phrases), needs to be built. It is well-known that, even if for Chinese linguists, it is hard to define every Chinese word and phrase clearly. Furthermore, from the perspective of retrieval system designers, different users have different retrieval concepts. It is not reasonable to ask the users to use the query terms in accordance with those in a dictionary. Therefore, the dictionary construction process is usually very time-consuming and difficult.

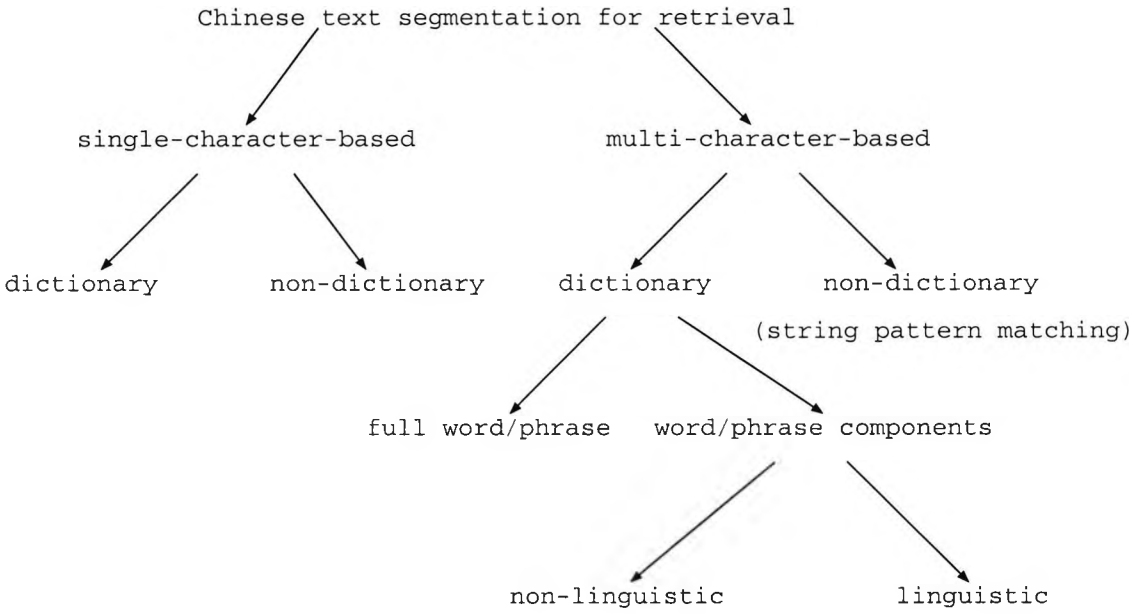


Figure 2.4: Classification of Chinese Text Segmentation

Full Dictionary Method

There are two common approaches used in dictionary-based segmentation. One approach is to first extract a string of n characters ($n = 7$ is commonly used) and to look for a matching entry in the dictionary. If a match is found, the string is marked as a word and the next n characters are processed. If no matching entry is found, the last character in the string is removed and the remaining string is matched against the dictionary entries. If a match is found, the remaining string is marked as a word, otherwise another character is removed and the process is repeated. The other approach is to first extract one character and look for a matching entry in the dictionary. If a match is found, the character next to it is added to the original character and the resulting string is matched against the dictionary entries. Characters are added one by one until the resulting string is not found in the dictionary. The longest string that matches an entry in the dictionary is indexed as a word. These two approaches are often known as *longest match*. There are some other methods such as *shortest match* and *overlap match*. The shortest match means that the shortest matched tokens are selected and the overlap match means that tokens generated from the text can overlap each other.

Syntactic Rule Method

More sophisticated segmentation algorithms involve the use of syntactic or grammatical rules. The ACTS (Automatic Chinese Text Segmentation) system developed by Wu and Tseng [143] uses a three step process. First, the input text is matched against a dictionary containing about 17,000 one- and two-character words tagged with syntactic categories, and divided into words with syntactic category(ies) attached. Next, a set of rules — based on linguistic structures or obtained through trial and error — are applied to words tagged with more than one syntactic category to select the most appropriate category. Finally, a parser is used to combine segments into words based on grammatical rules. This method is tested on a small sample of 30 texts. The experimental results show that most of significant words and phrases in these texts can be extracted with a hight degree of accuracy [143].

Most of the word-based indexing methods are based on a full word/phrase dictionary or thesaurus. An alternative to the full word/phrase method is for the dictionary to store word and phrase components, such as: ACTS systems [143]. The full word/phrase approach is easier to implement than the component approach because the former is merely a dictionary look-up procedure while the latter also requires a parsing procedure to combine components in addition to dictionary look-up. The full dictionary-based segmentation requires the use of comprehensive dictionaries. However, practically, it is unlikely that a dictionary can include all the words and phrases. Moreover, the more words/phrases that are in the dictionary, the more ambiguity of text segmentation may appear. Strings of characters not found in the dictionary are either discarded or treated as words. But automatic word extraction from text is quite difficult especially for these unknown words which are not found in the dictionary, such as people's names, locations, translated terms, technical terms, abbreviations, etc. So far, there has been little research on this issue[6]. It is important to note that without an efficient keyword extraction method, many IR applications, cannot obtain satisfactory results [71]. The full dictionary approach also deters its wide application, especially if a text

[6]Huang [43] proposed an algorithm for extracting noun phrases from middle-scale text collection.

segmentation system intends to divide text into not only words but also phrases for the retrieval purpose.

As to the component method, it requires less memory and disk space and leads to a fast dictionary lookup. But its weakness is associated with component parsing. Chinese language parsing is very complex and so are morphological complexities. It is difficult to describe the entire combination behavior of the components. So the more documents the system uses for testing, the more false combinations it may produce. For example, geographical or personal names, abbreviations, and words resulting from the translation of foreign words are problematic in Chinese morphology. They cannot be characterized by any single hypothesis.

Statistical Method

In order to avoid the costs of constructing and maintaining the dictionaries, alternate approaches to segmentation based on statistical techniques have been studied. Chen et al.[18] described how to segment Chinese texts using mutual information between two characters. The mutual information between two characters c_1 and c_2 is defined as

$$I(c_1, c_2) = \log_2 \frac{p(c_1, c_2)}{p(c_1)p(c_2)} \tag{2.1}$$

where $p(c_1, c_2)$ is the probability of the string c_1c_2 occurring in the collection, $p(c_1)$ and $p(c_2)$ are the probabilities of c_1 and c_2 occurring respectively in the collection. Chen et al. estimated these probabilities using occurrence statistics. Let $f(c_1)$ be the number of occurrences of c_1 in the collection; $f(c_2)$ the number of occurrences of c_2; $f(c_1, c_2)$ the number of occurrences of the string c_1c_2; and N the total number of characters in the collection. The probabilities $p(c_1)$, $p(c_2)$ and $p(c_1c_2)$ can be estimated by $\frac{f(c_1)}{N}$, $\frac{f(c_2)}{N}$ and $\frac{f(c_1c_2)}{N}$ respectively. Thus $I(c_1c_2)$ becomes

$$I(c_1c_2) = \log_2 \frac{f(c_1c_2) \times N}{f(c_1) \times f(c_2)} \tag{2.2}$$

When these occurrence frequencies are collected, a sentence can then be segmented using a multi-step process:

1. Compute the mutual information values for all bigrams (adjacent pairs of

characters) in the sentence.

2. Identify as a word the bigram with the largest mutual information value and remove it from the sentence. The removal of this bigram will result in one or two shorter phrases.

3. Repeat step 2 on each of the shorter phrase until the remaining phrases are all of one or two characters long.

While this approach does not require the use of a dictionary, it can only generate one- or two- character words. In addition, adding new documents to the collection is difficult. The mutual information values have to be recalculated, and the entire collection may have to be re-segmented using these new values.

Word ambiguity

Word ambiguity can only appear when we match ambiguous character strings. There are two kinds of ambiguous character strings in Chinese text: combinative ambiguous strings and overlapping ambiguous strings. They can be described as follows. Let $a_m(m = 1,...i), b_n(n = 1,...,j)$ and $c_o(o = 1,...k)$ be Chinese characters and S be the set of segmentation units in a dictionary.

(1) Let a string $C = a_1..a_ib_1..b_j$; if $a_1..a_i$, $b_1..b_j$ and $C \in S$, then C is called combinative ambiguous string.

(2) Let a string $C = a_1..a_ib_1..b_jc_1..c_k$; if $a_1..a_ib_1..b_j$ and $b_1..b_jc_1..c_k \in S$, then C is called an overlapping ambiguous string.

So far, none of the aforementioned methods have resolved the ambiguous problem completely. Some algorithms can solve this problem to some extent with the help of a human being. But it is just a result based on that person's knowledge. The result may be different if segmented by a different person, or even for the same person the result may be different at another time.

2.3.4 Retrieval Models

An important issue in Chinese information retrieval is the model used in the retrieval systems. Conventional Chinese information retrieval approaches are primarily designed for exact match searching and supporting Boolean queries. Most

commercial and practical Chinese information retrieval systems still rely on conventional inverted index and Boolean model, such as the full text retrieval system used by the Peoples' Daily Newspaper [82] and Sohoo [121]. Sohoo is the most famous search engine that has been designed to match the particular cultural tendencies of the Chinese user. Its classification hierarchy is divided into 18 sections and houses over 100,000 links. However, in spite of its ability to process structured queries, the Boolean model has been criticized for its inability to provide ranked output as all retrieved documents are considered equally important. Now a variety of alternatives to the Boolean model have been proposed and implemented for Chinese text retrieval, such as the Okapi system [96], SMART system [10], INQUERY system [12], the Berkeley's system and Queens College's PIRCS system. Okapi used a famous probabilistic model proposed by Robertson and Sparck Jones. The SMART system adopted a vector model and INQUERY is a probabilistic information retrieval system based upon a Bayesian inference network model.

2.3.5 Standard Test Collection

For research and development of computer systems, standards for measurements are essential in system evaluation. Information retrieval systems are no exception. Standard test collections are vital for research and development of information retrieval systems and comparison of the effectiveness of various retrieval models and approaches in the same evaluation environment. It is important to use standard test collections and measures to evaluate working systems proposed by different agencies.

The TREC conference [39], co-sponsored by ARPA and NIST, is an ongoing conference dedicated to encouraging research in retrieval of large-scale test collections and to increasing cooperation among research groups from industry and academia. A standard text collection of Chinese documents and 54 topics have been provided by TREC-5 and TREC-6. Evaluation of Chinese information retrieval systems was also included at the fifth and sixth TREC conference. The TREC conference brings together IR researchers to discuss system performance on standard test collections and is very effective in promoting the development of IR techniques. In this thesis, we use the test collection of Chinese documents and

the topics provided by TREC to evaluate several term weighting and document indexing methods.

Other major reported test collections in Europe and the United States are provided by CACM, MEDLARS, TIPSTER etc. In Japan, the NTCIR workshops have been held since 1999. NTCIR [76, 77] (NII-NACSIS Test Collection for IR Systems) is a project constructing large-scale test collections, which will be available for research purposes, based on some databases having been compiled and serviced by NII/NACSIS. NACSIS Test Collection 1 (NTCIR-1) contains more than 330,000 documents selected from NACSIS Academic Conference Papers Database, which is a collection of summaries of papers presented at conferences hosted by 65 Japanese academic societies in various subject fields; more than half are Japanese-English paired documents. NTCIR-1 contains 83 search topics and IR relevance judgments. Relevance judgments were done in three grades; Relevant, Partially relevant, Non-relevant. The second round, NTCIR-2, is also using Chinese language material. However, this collection was not available at the time the experiments for this thesis were run.

Chapter 3

2-Poisson Model for Probabilistic Weighted Retrieval

3.1 Introduction

Statistical approaches to information retrieval have traditionally taken two forms: formal approaches, where an exact formula is derived theoretically; and ad-hoc approaches, where formulae are tried because they seem to be plausible. Both categories have had some notable successes [10, 34, 62, 96].

One problem with the formal model approach is that it is very difficult to take into account the wide variety of variables that are thought or known to influence retrieval. The difficulty arises either because there is no known basis for a model containing such variables, or because any such model may simply be too complex to give a usable exact formula. One problem with the ad-hoc approach is that there is little guidance as to how to deal with specific variables.

The probabilistic weighted approach described in this chapter takes a model which provides an exact but intractable formula, and uses it to suggest a much simpler formula. The simpler formula can then be tried in an ad-hoc fashion. The variables we have included in this chapter are: within-document term frequency, document length, and within-query term frequency. The formal model which is used to investigate the effects of these variables is the 2-Poisson model [88].

3.2 Basic Probabilistic Weighting Model

The probabilistic retrieval model was first developed by Stephen Robertson and Karen Sparck Jones in the 1970s. The basic formula for a term-presence-only weight is as follows [85]:

$$w = \log \frac{p(1-q)}{q(1-p)} \tag{3.1}$$

where

$p = P(\text{term present} \mid \text{document relevant})$,

$q = P(\text{term present} \mid \text{document not relevant})$.

With a suitable estimation method, this becomes:

$$w = \log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} \tag{3.2}$$

where

N is the number of indexed documents in the collection;

n is the number of documents containing a specific term;

R is the number of known relevant documents for a specific topic; and

r is the number of known relevant documents containing a specific term,

This formula approximates to inverse collection frequency (ICF) shown in equation 3.3 when there is no relevance information. We will refer to this equation as $w^{(1)}$ in the latter part of this chapter.

$$ICF = \log \frac{N-n+0.5}{n+0.5} \tag{3.3}$$

If we deal with within-document term frequencies rather than only presence and absence of terms, then the formula corresponding to equation 3.1 would be as follows:

$$w = \log \frac{p_{tf} q_0}{q_{tf} p_0} \tag{3.4}$$

where

p_{tf} = P(term present with frequency tf | document relevant);

q_{tf} = P(term present with frequency tf | document not relevant);

p_0 = P(term absence | document relevant); and

q_0 = P(term absence | document non relevant),

3.3 The 2-Poisson Model

The basic extension to the binary Robertson/Sparck Jones model used for exploring the effects of other variables is the 2-Poisson model. Harter presents an indexing model based on an assumption that within-document term frequencies are distributed as a mixture of two Poisson distributions [41]; Robertson *et al.* use a similar idea to develop a probabilistic searching model [88].

Actually, the 2-Poisson model is a specific distributional assumption based on the eliteness hypothesis discussed in [93]. The assumption is that the distribution of within-document frequencies is Poisson for the elite documents,[1] and also for the non-elite documents.

We can assume that each term has associated with an elite set and that the distributions of numbers of occurrences of the terms in the two sets (elite and not-elite) are different (in particular, we assume that both are Poisson). We assume further that the elite sets for the query terms are correlated with relevance to the query, in a manner to be specified below.

Let us suppose that a, b, c are three events; then the following statements are equivalent:

$P(\mathrm{a}, \mathrm{b}|\mathrm{c}) = P(\mathrm{a}|\mathrm{c})P(\mathrm{b}|\mathrm{c})$

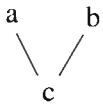
$P(\mathrm{a}|\mathrm{b}, \mathrm{c}) = P(\mathrm{a}|\mathrm{c})$

$P(\mathrm{b}|\mathrm{a}, \mathrm{c}) = P(\mathrm{b}|\mathrm{c})$

All the statements say that a and b are independent, given c; or equivalently, that

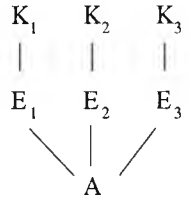
[1] An 'elite' document is one that is in some sense really about the concept represented by the term in question (see [88])

the only relation between a and b is implied by c. Diagrammatically, one might represent the situation under such an assumption as:



That is, it is assumed that there is no direct relation between a and b.

Returning to the 2-Poisson model, we can represent our independence assumptions as follows: Suppose that there is a query with relevance property A, with terms t_1, t_2, t_3 ... each with eliteness E_i and number of occurrences K_i; then we assume:



The various assumptions embedded in this model are:

1. The number of occurrences of a term depends only on the property of eliteness for that term, not on eliteness for any other term or on relevance.

2. Eliteness for a given term depends only on relevance, not on eliteness for any other term.

Based on the above discussion, we can get the following formula:

$$
\begin{aligned}
& P(K = k|A = 1) \\
= \; & \frac{P(K = k \wedge A = 1)}{P(A = 1)} \\
= \; & \frac{P(K = k \wedge A = 1 \wedge E = 1) + P(K = k \wedge A = 1 \wedge E = 0)}{P(A = 1)} \\
= \; & \frac{P(K = k|A = 1 \wedge E = 1)P(A = 1 \wedge E = 1) + P(K = k|A = 1 \wedge E = 0)P(A = 1 \wedge E = 0)}{P(A = 1)} \\
= \; & P(K = k|A = 1 \wedge E = 1)P(E = 1|A = 1) + P(K = k|A = 1 \wedge E = 0)P(E = 0|A = 1) \\
= \; & P(K = k|E = 1)P(E = 1|A = 1) + P(K = k|E = 0)P(E = 0|A = 1)
\end{aligned}
$$

As with the binary independence model, we need two parameters to describe the relation between eliteness for a term and relevance. There are as follows (for a given term):

$$p\prime = P(E = 1|A = 1)$$
$$q\prime = P(E = 1|A = 0)$$

Now we have:

$$\begin{aligned} P(K = k|A = 1) &= P(K = k|E = 1)P(E = 1|A = 1) + P(K = k|E = 0)P(E = 0|A = 1) \\ &= \frac{1}{k!}(p\prime exp(-\lambda)\lambda^k + (1 - p\prime)exp(-\mu)\mu^k) \end{aligned}$$

According to formula 3.4: $w = \log \frac{p_{tf} q_0}{q_{tf} p_0}$
where

$$p_{tf} = p\prime exp(-\lambda)\lambda^k + (1 - p\prime)exp(-\mu)\mu^k$$
$$q_{tf} = q\prime exp(-\lambda)\lambda^k + (1 - q\prime)exp(-\mu)\mu^k$$
$$p_0 = p\prime exp(-\lambda) + (1 - p\prime)exp(-\mu)$$
$$q_0 = q\prime exp(-\lambda) + (1 - q\prime)exp(-\mu)$$

Hence, we obtain the following weight for a term t:

$$w = \log \frac{(p\prime\lambda^{tf}e^{-\lambda} + (1 - p\prime)\mu^{tf}e^{-\mu})(q\prime e^{-\lambda} + (1 - q\prime)e^{-\mu})}{(q\prime\lambda^{tf}e^{-\lambda} + (1 - q\prime)\mu^{tf}e^{-\mu})(p\prime e^{-\lambda} + (1 - p\prime)e^{-\mu})} \tag{3.5}$$

where λ and μ are the Poisson means for tf in the elite and non-elite sets for t respectively, $p\prime = P(document\ elite\ for\ t|R)$, and $q\prime$ is the corresponding probability for $\overline{R}$.

The estimation problem is very apparent from 3.5, in that there are four parameters for each term, for none of which are we likely to have direct evidence. This consideration leads directly to the approach taken in the next section. This approach is proposed in [93] for solving the above estimation problem.

3.4 A Rough Model for Term Frequency

In fact, equation 3.5 has the following characteristics: (a) It is zero for tf=0; (b) it increases monotonically with tf; (c) but to an asymptotic maximum; (d) which approximates to the Robertson/Sparck Jones weight that would be given to a direct indicator of eliteness.

Only in an extreme case, where eliteness is identical to relevance, is the function linear in tf. These points can be seen from the following re-arrangement of equation 3.5:

$$w = \log \frac{(p\prime + (1-p\prime)(\frac{\mu}{\lambda})^{tf} e^{\lambda-\mu})(q\prime e^{\mu-\lambda} + (1-q\prime))}{(q\prime + (1-q\prime)(\frac{\mu}{\lambda})^{t} f e^{\lambda-\mu})(p\prime e^{\mu-\lambda} + (1-p\prime))} \tag{3.6}$$

where μ is smaller than λ. As $tf \to \infty$, $(\mu/\lambda)^{tf}$ goes to zero. $e^{\mu-\lambda}$ is small, so the approximation is:

$$w \approx \log \frac{p\prime(1-q\prime)}{q\prime(1-p\prime)} \tag{3.7}$$

Instead of estimating the above parameters directly, a simple tf-related weight function that has the characteristics (a)-(d) listed above is constructed in [93]. Full details of this function and models are presented by Robertson and Walker [93]. This weighting function can be expressed as follows:

$$w = \frac{tf}{(k_1 + tf)} w^{(1)} \tag{3.8}$$

where k_1 is an unknown constant. The effect of varying k_1 is to determine the extent to which tf affects the weight. For small k_1, tf has little effect; for large k_1, the effect of tf is almost linear. In between, the rate of increase of w with tf declines as tf increases, so that asymptotically approaches $w^{(1)}$.

The model tells us nothing about what kind of value to expect for k_1. Robertson *et al* have tried out various values of k_1 for the TREC data [96]. We will have more discussions about it later in the empirical chapters.

3.5 Document Length

The 2-Poisson model assumes that all documents are of equal length. Document length is a variable whose inclusion in various term-weighting schemes apparently brings benefits, and in fact the document length in the TREC collections for both English and Chinese are of extremely variable length.

A very rough model [93] is described in the following two functions: first, a modification to the formula 3.8:

$$w = \frac{tf}{(\frac{k_1 \times dl}{avdl} + tf)} w^{(1)} \tag{3.9}$$

This modification has the effect of normalising tf for document length, rather than using the absolute number, as does formula 3.8.

Second, the analysis suggests a correction factor to be added to the matching value of each document, as follows:

$$correction\ factor = k_2 \times nq \frac{(avdl - dl)}{(avdl + dl)} \tag{3.10}$$

where

nq = the number of terms in the query;
dl = the length of the document;
$avdl$ = the average document length; and
k_2 = another unknown constant.

In effect, this correction factor downweights long documents and upweights short documents, irrespective of the matching terms; it might be described as a "global" document length correction. We will give more detailed discussion about the document length correction factor in chapter 4.

3.6 Query Term Frequency

This natural symmetry of the retrieval situation as between documents and queries suggests that we could treat within-query term frequency (qtf) in a similar fashion to within-document term frequency [94]. Robertson suggests [96] a weight function for longer queries:

$$w = \frac{qtf}{(k_3 + qtf)} w^{(1)} \tag{3.11}$$

where k_3 is another unknown constant. k_3 exercises the same control over the effect of qtf as does k_1 over the tf effect.

3.7 Weighting Functions

The weighting functions described above have been implemented as BM15 (the model using equation 3.8 for the document term frequency component), BM11 (using equation 3.9) and BM25 [95, 96]. Both BM11 and BM15 incorporate the document length correction factor of equation 3.10. These functions were compared with baseline $w^{(1)}$ weighting (BM1 with $k_3 = 0$) and with a simple coordination-level model BM0 in which terms are given equal weightings.

The various weighting functions used are summarized below:

$$BM0: \quad w = 1$$

$$BM1: \quad w = \log \frac{N - n + 0.5}{n + 0.5} * \frac{qtf}{k_3 + qtf}$$

$$BM15: \quad w = \frac{tf}{k_1 + tf} * \log \frac{N - n + 0.5}{n + 0.5} * \frac{qtf}{(k_3 + qtf)} \oplus \quad k_2 * nq * \frac{(avdl - dl)}{(avdl + dl)}$$

$$BM11: \quad w = \frac{tf}{(k_1 * dl/avdl + tf)} * \log \frac{N - n + 0.5}{n + 0.5} * \frac{qtf}{(k_3 + qtf)} \oplus \quad k_2 * nq * \frac{(avdl - dl)}{(avdl + dl)}$$

$$BM25: \quad w = \frac{(k_1 + 1) * tf}{K + tf} * \log \frac{N - n + 0.5}{n + 0.5} * \frac{(k_3 + 1) * qtf}{k_3 + qtf} \oplus \quad k_2 * nq * \frac{(avdl - dl)}{(avdl + dl)}$$

where

N is the number of indexed documents in the collection,

n is the number of documents containing a specific term,

R is the number of known relevant documents for a specific topic,

r is the number of known relevant 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 topics and are empirically determined,[2]
K equals to $k_1 * ((1 - b) + b * dl/avdl)$, and
the $\oplus$ in the formula indicates that the following component is added only once per document, rather than for each term.

The formulae BM11, BM15 and BM25 now define the weight of a term (that is, the contribution of that term to the total score for the document) in the context of an individual document. Actually, the BM25 function is a mix of the two functions BM15 and BM11 with different values of b (b=0 gives BM15 and b=1 gives BM11).

A large number of experiments for these weighting functions were performed on the complete training English database (TREC-1, TREC-2, TREC-3, TREC-4 and TREC-5). Routing experiments show that BM1 was slightly better than BM15, but BM11 was substantially better than either [96]. Ad hoc experiments show: for the shorter queries (title + concepts) BM15 appears somewhat better than BM1 (the difference is greater when a document length correction is added). But its performance on the long queries is very poor. However, BM11 gives a very marked improvement over the baseline, particularly for the long queries [96]. The modified weighting function BM25 seems able to give slightly improved English retrieval results comparing to BM11, at the cost of another parameter to be guessed. $b < 1$ can give some improvement. Values around 0.75 were used, which usually give the best results [95].

According to the good ideas used in the English experiments, we also conducted a large number of ad hoc experiments on the complete training Chinese database (TREC-5 and TREC-6) for weighting functions BM11 and BM25. The Chinese results show that BM25 is able to give much more improvement, not just slightly, over the BM11 on both TREC-5 and TREC-6 Chinese datasets. Some new weighting functions, which are based on BM25, are designed and implemented for Chinese ad hoc experiments. A detailed description will be given in Chapter 4.

[2]For our experiments, the k_is' values will be given in Chapter 7 and 8

Chapter 4

Improvements to the 2-Poisson Model

4.1 Motivation

In English text retrieval, documents are usually indexed by words, however, the keywords extracted from a query can be either words or phrases[1]. Whether a document matches with a phrase in the query can be determined at search time using position information in the index file. Many retrieval systems make use of both words and phrases in the keywords and conduct both word and phrase searching at retrieval time. Since a match of a phrase in the query with a phrase in a document usually indicates that the document is more relevant than a document matching only part of the phrase, a matched phrase should be given a higher weight than the words that constitute the phrase. Therefore, phrase weighting should be designed differently from word weighting.

In Chinese text retrieval, situations may be different from English text retrieval because documents can be indexed by either words or characters. When the documents are indexed by words, the retrieval process in terms of word and phrase weighting is similar to the English text retrieval. However, if the documents are indexed by characters, any words of more than one character as well as phrases of more than one word in the query are considered as "phrases" in the sense that position information need to be considered when searching for a word or phrase in a document. Therefore, for character-indexed document retrieval, a "phrase" weighting function should be used to weight the multi-character words in the query.

[1]In this thesis, "phrases" are always taken as contiguous units of text. This is standard for IR but not for NLP (Natural Language Processing), where, e.g., to "take away" might be regarded as a phrase even in the form "I want to take this box away".

To avoid confusion caused by these terms, we propose to use the terms 'single unit weighting' and 'compound unit weighting'.

Keyword weighting can be classified into single unit weighting and compound unit weighting. A single unit weighting function is used for weighting a single linguistic unit in a keyword. A single linguistic unit is the linguistic unit that is used to build the index of documents. For example, if documents are indexed using words, a word is a single linguistic unit; if documents are indexed using characters, a character is a single linguistic unit. A compound unit weighting function is used for weighting a compound linguistic unit that consists of two or more single units. For example, if the documents are indexed using words, then a phrase is a compound linguistic unit; if the documents are indexed using characters, then both multi-character words and phrases are compound units.

The work described in this chapter is motivated by observations of the performance of a retrieval system with respect to both single-unit and compound-unit weighting.

4.1.1 Correction Factor in Single Unit Weighting

The performance of a retrieval system greatly depends on the weighting functions used on query terms. In the last chapter, we have described functions BM11, BM15 and BM25. Experiments on both standard English and standard Chinese collections showed that function BM25 leads to better results than functions BM11 or BM15. The experiments also showed that the best results were obtained when the parameter k_2 in BM25 was set to be zero, which means that the following *correction factor* was ignored from the whole formula:

$$k_2 \times nq \frac{avdl - dl}{avdl + dl} \tag{4.1}$$

where dl is the length of the document, $avdl$ is the average document length, nq is the number of query terms, and the k_2 are tuning constants. This correction factor was designed to take into account the length of a document in calculating the matching value for the document. The experimental results indicate that this factor was not designed properly, otherwise the factor would not have been ignored

to obtain the best results. By close examination of the factor, we found that the correction factor (as defined in equation 4.1) decreases with dl, from a maximum as $dl \rightarrow 0$, through zero when $dl = avdl$, and to a minimum below zero as $dl \rightarrow \infty$, as shown in Figure 4.1.

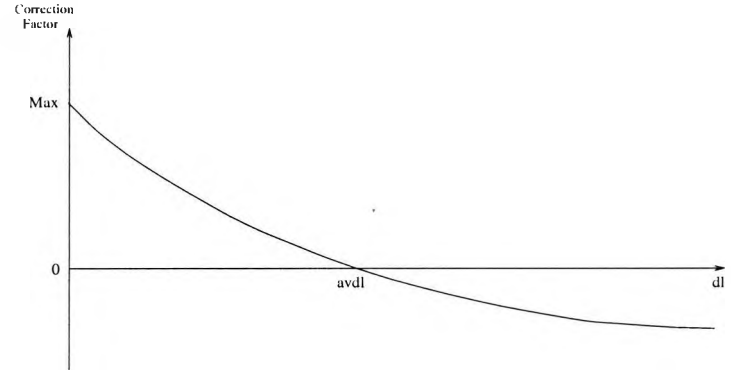


Figure 4.1: Curve for the Correction Factor with respect to Document Length

This definition of the correction factor simply downweights long documents and upweights short documents. It gives highest values to documents whose lengths approach zero. However, the Chinese TREC-5 evaluation results indicate that very short documents, such as titles, are usually considered irrelevant by human appraisers.[2] By using BM25, these very short documents, usually titles containing only one sentence, are retrieved as highly relevant documents. Therefore, the definition of the correction factor in BM25 needs to be modified to consider very short documents as irrelevant.

In this chapter, we present a new single-unit weighting function which is a modified version of BM25. The modification is done by redefinition of the correction factor. The new weighting function is referred to as BM26. The correction factor in BM26 will give a low value for a short document, a high value for a middle-sized document and a low value for a large document.

4.1.2 Compound Unit Weighting

The other objective of this chapter is to investigate compound unit weighting functions for Chinese document retrieval. A compound unit is a phrase in the

[2]This observation also holds for other TREC evaluations. Therefore, it is reasonable to say that very short documents, such as titles, are in general considered irrelevant.

query if documents are indexed by words, or it can be either a word or a phrase if documents are indexed by characters. A number of formulas for compound unit weighting, usually called phrase weighting, have been designed and evaluated. However, the evaluation results do not show phrases weighting has much advantage over single unit weighting for the English language [95]. Since a match of a phrase in the query with a phrase in a document usually indicates that the document is more relevant than a document matching only part of the phrase, a good phrase weighting formula should result in better results than using only single unit weighting.

In this chapter, we propose a number of compound unit weighting functions and describe each formula and the rationale behind it. We are looking at combining these compound unit weighting functions with probabilistic approaches to text retrieval. We also conduct a large number of experiments and performance analysis on Chinese TREC data set so that we can better understand the use of phrase weighting in Chinese text retrieval. This may give us some good ideas for phrase weighting in a language independent system.

4.2 Document Length

As we have discussed in section 3.7, ad hoc English experiments showed that BM15 appears somewhat better than BM1 for the shorter queries. BM11 gives a very marked improvement over BM1 with $k_3 = 0$ [96]. The modified weighting function BM25 is able to give slightly improved retrieval results comparing to BM11.

For BM15, BM11 and BM25, there is a correction factor which is designed to take into account the length of a document. The correction factor, shown in expression 4.1, assumes: the shorter the document is, the more value the correction factor should have – that is the more possible the document is relevant. This assumption is reasonable in some cases, for example a short article on the first page of a newspaper may be more important than a longer article on another page. However, the human appraisers for Chinese TREC-5 and TREC-6 usually considered the very short documents as irrelevant. Therefore, it seems likely that the correction factor can be improved for the TREC-5 and TREC-6 datasets.

4.2.1 Assumptions for BM26

To tackle the correction factor problem, we propose a new weighting function, referred to as BM26. BM26 is a modified version of BM25 and is designed on the basis of the following assumptions:

Assumption 1: too short documents are not relevant

Assumption 2: the function curve for correction factor should be consistent with the distribution of relevant documents in the TREC-5 and TREC-6 datasets.

4.2.2 Chinese TREC Dataset Analysis

In order to justify the first assumption, let us look at some statistics in the Chinese TREC-5 and TREC-6 datasets. The document collection used in TREC-6 Chinese track was identical to the one used in TREC-5. Table 4.1 describes the Chinese collection for the TREC-5 and TREC-6 experiments in terms of the minimum, maximum, average length of documents and the total number of documents. The length of documents is calculated on the basis of Chinese texts in documents only. The SGML tags (such as "⟨DOC⟩", "⟨/DOC⟩", "⟨DOCNO⟩", "⟨/DOCNO⟩", "⟨HL⟩", "⟨/HL⟩", "⟨TEXT⟩" and "⟨/TEXT⟩") and document IDs are not counted. For this reason, there are two documents "pd9301-1034" and "pd9311-1495" with 0 byte document length. These two documents only contain document IDs and SGML tags without even one Chinese character inside. Figure 4.2 depicts the frequency distribution of the documents in the collection with respect to the length of documents. In the figure, the document length is discretized by using equal-interval binning, where the length of each interval is 500 bytes. Detailed information for discretized document lengths are given in Appendix C.

Min. length	0 byte
Max. length	294056 bytes
Total Number of documents	164768 documents
Average Length	891 bytes

Table 4.1: Whole Chinese TREC Dataset

Table 4.2 shows the statistics on the relevant documents for the TREC-5 and TREC-6 queries. The frequency distribution of the TREC-5 and TREC-6 relevant

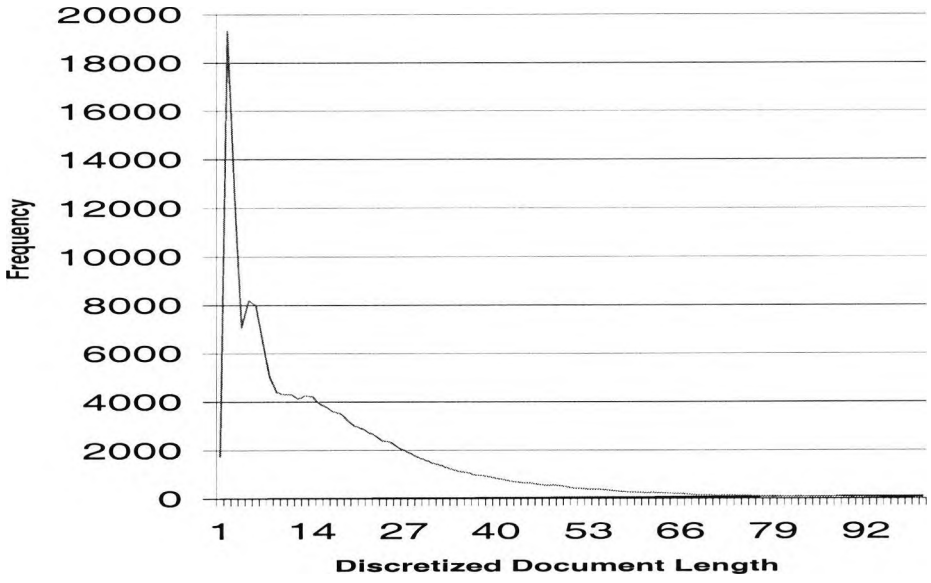


Figure 4.2: Distribution Curve for the Whole Chinese TREC Dataset

documents with respect to the document length is depicted in Figure 4.3.

	TREC-5	TREC-6
Min. length	58 bytes	60 bytes
Max. length	22718 bytes	294056 bytes
Average Number of Relevant Documents per Query	83.9 documents	105.6 documents
Average Length	1399 bytes	1987 bytes
Standard Deviation	1675.31 bytes	6194.5 bytes

Table 4.2: Relevant Datasets for TREC-5 and TREC-6

From these statistics we can observe that

1. The average length of relevant documents for both TREC-5 and TREC-6 is longer than the average length of the documents in the whole TREC document collection.

2. On average only 0.05% - 0.064% of the documents in the collection are relevant for a query.

3. Since only 0.05-0.064% of whole documents are relevant, Figure 4.2 can be roughly regarded as the distribution curve of non-relevant documents.

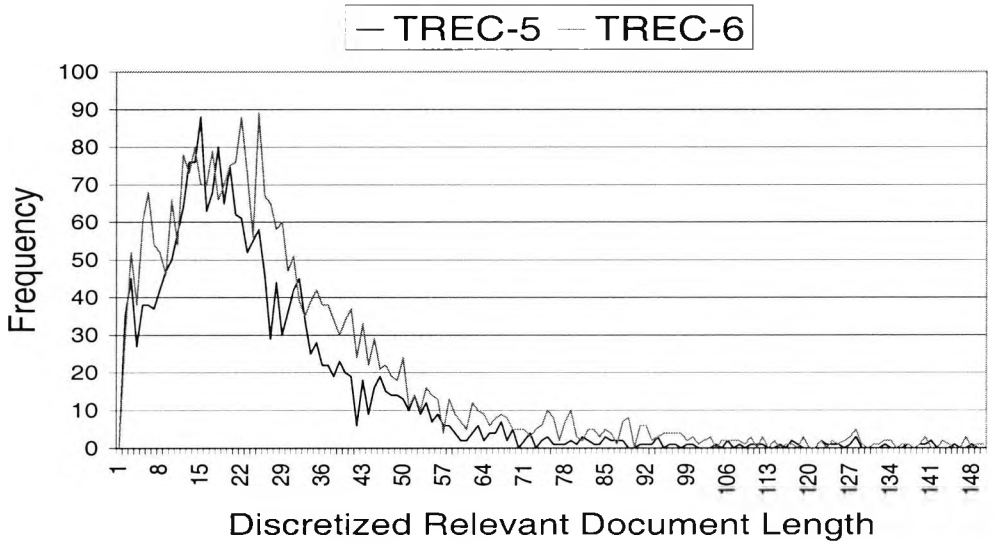


Figure 4.3: Distribution Curve for the Relevant TREC-5 and TREC-6 Datasets

Obviously the curve for the correction factor of BM25, shown in Figure 4.1, matches the distribution curve for the whole dataset (Figure 4.2) better than the distribution curves shown in Figure 4.3. This means that Figure 4.1 matches the distribution curve of document length for the non-relevant dataset and does not match the distribution curves of relevant documents for TREC-5 and TREC-6 shown in Figures 4.3. Therefore, we can conclude that the BM25 correction factor function is suitable for the non-relevant dataset, but not for the relevant dataset.

4.2.3 Design of the BM26 Correction Factor

From Figure 4.3 we can see, the distribution curves of relevant documents for TREC-5 and TREC-6 are different from the distribution curve of the documents in the whole collection. It is reasonable to consider that documents whose lengths are close to the average length of relevant documents are more likely to be relevant than the documents whose length is further away from the average length. Based on this consideration, we modify the correction factor in BM25 so that the trend of this correction factor with respect to document length matches the distribution of relevant documents. The function for this new correction factor is defined as

follows:

$$y = \begin{cases} \ln(\frac{dl}{avdl}) + \ln(x_1) & \text{if } 0 < dl <= rel\_avdl; \\ (\ln(\frac{rel\_avdl}{avdl}) + \ln(x_1))(1 - \frac{dl - rel\_avdl}{x_2 * avdl - rel\_avdl}) & \text{if } rel\_avdl < dl < \infty. \end{cases} \quad (4.2)$$

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 curve of this new function is shown in Figure 4.4. As can be seen, y is negative when the document length is very small; it becomes positive when the document length becomes larger; it reaches a maximum as $dl \rightarrow rel\_avdl$; and then it decreases through zero when $dl = x_2 * avdl$, and becomes negative again as $dl \rightarrow \infty$.

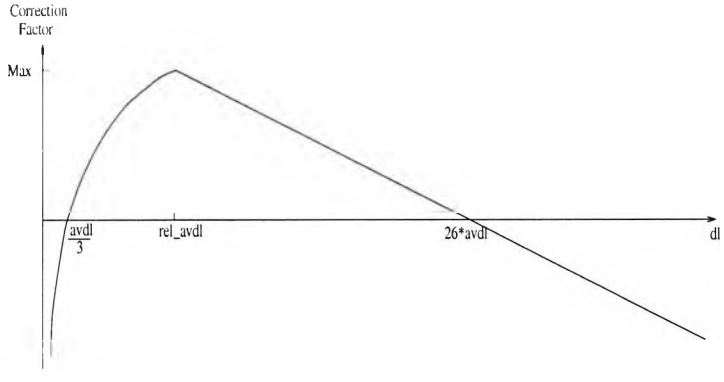


Figure 4.4: Curve for the New Correction Factor with respect to Document Length

This design of the BM26 correction factor is based on the following consideration: the trend of this correction factor with respect to document length should match the distribution curve of relevant documents. The distribution curve of relevant documents come from the distribution curves for the TREC-5 and TREC-6 relevant datasets which satisfy the following:

1. The frequencies of documents reach a maximum as the length of document increases from 0 to a certain length which is usually not small.

2. The frequencies of documents decrease from maximum to 0 smoothly compared to a sharp decrease in Figure 4.2.

It is reasonable to assume that the distribution curve of relevant documents for a well-chosen dataset satisfies the two above characteristics. Hence, under this assumptiop, we can say that the new design correction factor for BM26 can be generally applied to other datasets, not just to the TREC Chinese dataset.

An alternative to the above proposed correction factor in BM26 is a regression function that approximates the probability distribution of relevant documents. Comparison of this alternative with the proposed correction factor is an interesting topic for future work.

4.2.4 BM26 Weighting Function

The weighting function we use for Okapi TREC-6 Chinese experiments is given as follows. This function is extended from the BM25 function [7]. We refer to this weighting function as BM26, defined as follows:

$$w = \frac{(k_1+1)*tf}{K+tf} * \log\frac{N-n+0.5}{n+0.5} * \frac{(k_3+1)*qtf}{k_3+qtf} \oplus k_d * y \qquad (4.3)$$

where

N is the number of indexed documents in the collection,
n is the number of documents containing a specific term,
R is the number of known relevant documents for a specific topic,
r is the number of known relevant documents containing a specific term,
tf is within-document term frequency,
qtf is within-query term frequency,
the k_is are tuning constants, which depend on the database and possibly on the nature of the topics and are empirically determined,[3]
K equals to $k_1 * ((1-b) + b * dl/avdl)$,
the $\oplus$ in the formula indicates that the following component is added only once per document, rather than for each term, and
y is defined in equation 4.2.

The difference between BM26 and BM25 is in the y component. In BM25, y equals to $nq * \frac{avdl-dl}{avdl+dl}$, where nq is the number of query terms. This means y

[3]For our experiments, the k_is' values will be given in Chapter 7 and Chapter 8

decreases with dl, from a maximum as $dl \rightarrow 0$, through zero when $dl = avdl$, and to a minimum as $dl \rightarrow \infty$. In BM26, y will reach a maximum as $dl \rightarrow rel\_avdl$, through zero when $dl = avdl/x_1$ (or $dl = x_2 * avdl$), and to a minimum as $dl \rightarrow 0$ (or $dl \rightarrow \infty$). Therefore, a new y component is designed for BM26, where x_1 and x_2 were set to 3 and 26 respectively in our experiments. The reason why we chose 3 and 26 for x_1 and x_2 are: we assume that we should add a negative weight to the document whose length is smaller than one third of $avdl$ or 26 times bigger than $avdl$. We may get better results by setting other values for x_1 and x_2. Detailed analysis and experiments for the BM26 weighting function will be given in Chapter 8.

4.3 Compound Unit Weighting

Compound unit weighting has been studied under the name of "phrase weighting" for English text retrieval. Throughout the last three decades in the IR, computer science, and other literature, there have been repeated attempts to incorporate phrases, i.e. combinations of words, in the automatic indexing and retrieval processes for full text applications [26, 56, 95, 105]. The improvements gained through the use of phrases have counter-intuitively always been quite small. In some cases they have produced no improvement to the retrieval process at all. In very small message collections, with narrow, well-defined domains of discourse and small vocabularies, NLP methods using linguistically parsed phrases produce significant retrieval performance improvements [56]. But for larger unrestricted collections, most information retrieval systems which incorporate phrases in indexing and retrieval have not had much success.

When a user searches for a number of words, and those words are found close together in a document, that document should be assigned a higher weight than the one in which the words appear scattered farther apart. In the document in which the words are found close together, it's more likely that they are being used in the same context as the user meant. For example, the query "Artificial intelligence is a research area" is more likely to be answered by a document including the phrase "The research in artificial intelligence" or "artificial intelligence reseach" than a document in which the three words appear widely separated.

In this section we describe the phrase weighting functions used in our Chinese Okapi system. Later in the experimental chapter we will compare character-based and word-based document processing methods in terms of these weighting functions. To reflect the use of phrase weighting more exactly with both word-based and character-based Chinese text processing, we refer to phrase weighting as compound unit weighting since both words and phrases need to be treated as "phrases" when documents are indexed by characters.[4]

Our basic approach to dealing with compound units within a probabilistic weighting scheme is to have some consistent way of using the information about the single units that make up the compound unit, together with the compound unit information. If the single units in a compound unit are not to be used at all, then the compound unit can be treated as if it were a single unit. For single unit weighting, we can use BM1, BM11, BM15, BM25 or BM26, etc.

4.3.1 Assumption for Compound Unit Weighting

Compound units (phrases or words) contain more information than single units (words or characters). It is reasonable to consider that searching for compound units during retrieval is one of the most powerful combination techniques that could improve retrieval performance. But how can we make this possible?

Compound units may be handled in a number of ways, which we may briefly summarise as follows. They may be defined at file time; alternatively, the file-time indexing may be done with single units, and any compound unit may be identified only at search time. A compound unit may be treated as an undecomposable unit; alternatively the compound unit and its component single units may be regarded as indexable and/or searchable terms, in which the compound unit is treated as a decomposable unit.

An undecomposable unit can be dealt with in a very straightforward way in the probabilistic model and it can be treated as strictly equivalent to a single unit or stem. However, there is a substantial problem with the interpretation of the

[4]Documents can be indexed by words or characters and query keywords can be words or phrases. For word-indexed document retrieval, a phrase in keywords is a compound unit. For character-indexed document retrieval, both words and phrases are compound units.

probabilistic model in the context of decomposable units.

Our method for dealing with phrases in Chinese is as follows. If a word or phrase is identified at indexing time, it is indexed as a single unit, and the component characters of the word or the component words of the phrase are not normally indexed. If it is identified only at search time, it may be searched as a compound unit by means of an adjacency operator.

Though we have assumed that in practice terms will be single unit words or stems, we have so far accepted that the general probabilistic model may be applied to any type of index term that is viewed as an integral unit [128]. From this point of view, compound units can be weighted using a single-unit weighting function, such as BM25 or BM26. However, we may allow partial matching, which means we treat the compound unit as a decomposable. That is, a query can match a document on the whole compound or only on one or more components of it. Now there are two matching situations: (1) the matched components are adjacent to each other in the document and (2) the matched components are not adjacent. It is reasonable to assume that, for a compound unit consisting of two single terms $t_1 t_2$,

$$w(t_1), w(t_2) < w(t_1 \wedge t_2) = w(t_1) + w(t_2) < w(t_1 \ adj \ t_2), \tag{4.4}$$

where $\wedge$ is the *and* operator and *adj* is the adjacency operator. The equation in the middle represents the usual scoring method for the $\wedge$ (and) operator: the score assigned to a document is the sum of the weights of the matching terms.[5] The assumption is that preference will normally be given to matches on the two adjacent units because the compound will have a higher weight than either component, so a document containing the adjacent unit or phrase will be ranked higher than one with just a member term, or than one containing both terms but not in the phrasal relationship. This preference is simply an automatic consequence of the nature of the data used for weighting.

[5]The additivity is implied by the independence assumptions that we should add the weights of terms and it comes before the ICF. The formal proof can be found in reference [85, 84].

4.3.2 Analysis for Compound Unit Weighting

Before we design weighting functions for the compound unit, let us investigate whether the assumption 4.4 can be satisfied when we use ICF (see the definition in Section 3.2) as the single unit weighting function. First, we will prove that under a certain condition the assumption 4.4 can be satisfied. However, the assumption 4.4 may not be satisfied in other circumstances, but downgrading weight $w(t_1 \wedge t_2)$ may help.

Let t_1 and t_2 be two single unit terms; t_1 *adj* t_2 is an adjacent compound unit; $w(t_1)$, $w(t_2)$, $w(t_1 \wedge t_2)$ and $w(t_1$ *adj* $t_2)$ are the weights for t_1, t_2, $t_1 \wedge t_2$ and t_1 *adj* t_2 respectively. If we treat $t_1 \wedge t_2$ as an integral unit, we should (for consistency) weight it according to $\#(t_1 \wedge t_2)$.[6] Although the basic probability model does not normally do this, it is at least plausible to argue that $t_1 \wedge t_2$ should relate to $\#(t_1 \wedge t_2)$. In the following analysis, we can infer something about how the normal weight of $ICF(\#(t_1)) + ICF(\#(t_2))$ relates to the possible alternative of $ICF(\#(t_1 \wedge t_2))$ by this argument.

Lemma 4.1: For any two terms t_1 and t_2, if the two terms are independent given that both R and r equal to 0 and ICF is used to calculate $w(t_1)$, $w(t_2)$ and $w(t_1$ *adj* $t_2)$, then

$$w(t_1 \text{ adj } t_2) \geq w(t_1) + w(t_2)$$

Proof:

Let $\#(t)$ be the number of documents containing the term t and N is the total number of documents in the collection. Assume that $R = r = 0$. If the two terms t_1 and t_2 are independent, we have:

$$P(t_1 \wedge t_2) = P(t_1) * P(t_2)$$

$$\Longleftrightarrow \quad \frac{\#(t_1 \wedge t_2)}{N} = \frac{\#(t_1)}{N} * \frac{\#(t_2)}{N}$$

$$\Longleftrightarrow \quad \#(t_1 \wedge t_2) = \frac{\#(t_1) * \#(t_2)}{N}$$

[6] Under the independence assumptions we use in the probabilistic model, $w(t_1 \wedge t_2) = log\frac{N - \#(t_1 \wedge t_2)}{\#(t_1 \wedge t_2)}$ when R and r are set to be 0 (see section 3.2 for more details).

Since $\#(t_1\ adj\ t_2) \leq \#(t_1 \wedge t_2)$, we have

$$\#(t_1\ adj\ t_2) \leq \frac{\#(t_1) * \#(t_2)}{N} \tag{4.5}$$

$$\iff \frac{1}{\#(t_1\ adj\ t_2)} \geq \frac{N}{\#(t_1) * \#(t_2)} \tag{4.6}$$

We have

$$w(t_1\ adj\ t_2) - w_{t_1} - w_{t_2} = log\frac{(\frac{N}{\#(t_1\ adj\ t_2)} - 1)}{(\frac{N}{\#(t_1)} - 1)(\frac{N}{\#(t_2)} - 1)}.$$

Let $D_1 = \frac{N}{\#(t_1\ adj\ t_2)} - 1$ and $D_2 = (\frac{N}{\#(t_1)} - 1)(\frac{N}{\#(t_2)} - 1)$.
Then

$$D_1 - D_2 = N(\frac{1}{\#(t_1\ adj\ t_2)} + \frac{1}{\#(t1)} + \frac{1}{\#(t_2)} - \frac{N}{\#(t_1) * \#(t_2)}) - 2. \tag{4.7}$$

Based on 4.6 and 4.7, we have

$$D_1 \geq D_2.$$

Hence,

$$w(t_1\ adj\ t_2) \geq w(t_1) + w(t_2)$$

□

Since we can prove that $w(t_1 \wedge t_2) = w(t_1) + w(t_2)$ when t_1 and t_2 are independent given relevance [85, 84], we can conclude that $w(t_1\ adj\ t_2) \geq w(t_1 \wedge t_2)$ when t_1 and t_2 are independent given that $R = r = 0$.

Lemma 4.2: For any two terms t_1 and t_2, if BM25 or BM26 is used to calculate $w(t_1)$, $w(t_2)$ and $w(t_1\ adj\ t_2)$, and

$$w(t_1\ adj\ t_2) < w(t_1) + w(t_2),$$

then

$$w(t_1 \wedge t_2) < w(t_1) + w(t_2)$$

Proof:

Let $\#(t)$ be the number of documents containing the term t and N is the total number of documents in the collection. Assume that $R = r = 0$.

According to the **Lemma 4.1**, if t_1 and t_2 satisfy

$$\#(t_1 \wedge t_2) \leq \frac{\#(t_1) * \#(t_2)}{N},$$

then

$$w(t_1\ adj\ t_2) \geq w(t_1) + w(t_2).$$

Hence, if

$$w(t_1\ adj\ t_2) < w(t_1) + w(t_2),$$

then

$$\#(t_1 \wedge t_2) > \frac{\#(t_1) * \#(t_2)}{N} \tag{4.8}$$

The above inequality is equivalent to

$$\frac{N}{\#(t_1 \wedge t_2)} < \frac{N}{\#(t_1)} * \frac{N}{\#(t_2)} \tag{4.9}$$

According to the definitions of BM25 and BM26, we have

$$w(t_1 \wedge t_2) \approx \frac{(k_1+1) * tf_{t_1 \wedge t_2}}{K + tf_{t_1 \wedge t_2}} * (log \frac{N}{\#(t_1 \wedge t_2)}) * \frac{(k_3+1) * qtf_{t_1 \wedge t_2}}{k_3 + qtf_{t_1 \wedge t_2}}$$

$$w_{t_1} \approx \frac{(k_1+1) * tf_{t_1}}{K + tf_{t_1}} * (log \frac{N}{\#(t_1)}) * \frac{(k_3+1) * qtf_{t_1}}{k_3 + qtf_{t_1}}$$

and

$$w_{t_2} \approx \frac{(k_1+1) * tf_{t_2}}{K + tf_{t_2}} * (log \frac{N}{\#(t_2)}) * \frac{(k_3+1) * qtf_{t_2}}{k_3 + qtf_{t_2}}$$

in which we assume that the correction factor for document length is 0

Let $f(tf_t) = \frac{(k_1+1)*tf_t}{K+tf_t}$ and $g(qtf_t) = \frac{(k_3+1)*qtf_t}{k_3+qtf_t}$.

Based on 4.9, we get the following inequality,

$$(\frac{N}{\#(t_1 \wedge t_2)})^{f(tf_{t_1 \wedge t_2}) * g(qtf_{t_1 \wedge t_2})} < (\frac{N}{\#(t_1)} * \frac{N}{\#(t_2)})^{f(tf_{t_1 \wedge t_2}) * g(qtf_{t_1 \wedge t_2})} \tag{4.10}$$

Since

$$f(tf_{t_1 \wedge t_2}) \leq f(tf_{t_1}), f(tf_{t_1 \wedge t_2}) \leq f(tf_{t_2}),$$

$$g(qtf_{t_1 \wedge t_2}) \leq g(qtf_{t_1})\ and\ g(qtf_{t_1 \wedge t_2}) \leq g(qtf_{t_2})$$

We have

$$(\frac{N}{\#(t_1 \wedge t_2)})^{f(tf_{t_1 \wedge t_2}) * g(qtf_{t_1 \wedge t_2})} < (\frac{N}{\#(t_1)})^{f(tf_{t_1}) * g(qtf_{t_1})} * (\frac{N}{\#(t_2)})^{f(tf_{t_2}) * g(qtf_{t_2})} \quad (4.11)$$

$$f(tf_{t_1 \wedge t_2}) * log\frac{N}{\#(t_1 \wedge t_2)} * g(qtf_{t_1 \wedge t_2}) < f(tf_{t_1}) * log\frac{N}{\#(t_1)} * g(qtf_{t_1}) + f(tf_{t_2}) * log\frac{N}{\#(t_2)} * g(qtf_{t_2}) \quad (4.12)$$

Hence,

$$w(t_1 \wedge t_2) < w(t_1) + w(t_2)$$

□

Lemma 4.2 implies that the usual scoring method for $t_1 \wedge t_2$, which is $w(t_1 \wedge t_2) = w(t_1) + w(t_2)$, gives too much weight if $w(t_1\ adj\ t_2) < w(t_1) + w(t_2)$. A lower weight should be assigned to $t_1 \wedge t_2$. The conclusions we obtain from the above analyses are as follows:

1. for any two terms t_1 and t_2 if they are independent given that $R = r = 0$ and ICF is used to calculate $w(t_1)$, $w(t_2)$ and $w(t_1\ adj\ t_2)$, then

   $$w(t_1\ adj\ t_2) \geq w(t_1) + w(t_2).$$

   which means that the assumption 4.4 for the compound unit weighting is satisfied.

2. for any two terms t_1 and t_2 if BM25 or BM26 is used to calculate $w(t_1)$, $w(t_2)$ and $w(t_1\ adj\ t_2)$, and

   $$w(t_1\ adj\ t_2) < w(t_1) + w(t_2),$$

   then

   $$w(t_1 \wedge t_2) < w(t_1) + w(t_2),$$

   which means that the weight "$w(t_1) + w(t_1)$" we usually assign to $w(t_1 \wedge t_2)$ is overweighting.

The above conclusions give us some suggestions on designing a compound unit weighting mechanism within probabilistic weighting models. In order to assign a weight to a compound unit which satisfies the assumption 4.4 and to achieve better retrieval results, we can downgrade the assigned weight "$w(t_1) + w(t_1)$" for $w(t_1 \wedge t_2)$ or the other way, we can assign an extra boost-weight to $w(t_1\ adj\ t_2)$ which is given by the single unit weighting function BM25 or BM26.

4.3.3 Probabilistic Formulation

The general problem of dealing with compound units within a probabilistic weighting scheme is to have some consistent way of using the information about the single unit terms that make up the compound unit, together with the compound unit information. If the single unit terms are not to be used at all, then the compound unit can be treated as if it were a single unit term, without any danger of inconsistency. But that would lose one of the main advantages of probabilistic weighting schemes – the automatic relaxation of the exact query specification.

In this section, a general principle for how we should weight a compound unit is proposed. The idea for this principle originally comes from [97]. Although there may be considerable practical difficulties, the principle would actually be quite widely applicable. The principle can be described as follows:

If we have information $I(q,d)$ relating a document to a query which implies some other such information $J(q,d)$, that is

$$I(q,d) \Longrightarrow J(q,d).$$

Then the appropriate way to formulate the probabilistic model is to estimate in the usual way the log-odds of relevance for the less specific condition $J(q,d)$, as

$$log \frac{P(J|R)P(\bar{J}|\bar{R})}{P(\bar{J}|R)P(J|\bar{R})}. \tag{4.13}$$

and then to boost this estimate according to the additional information provided by $I(q,d)$, in other words according to estimates of such quantities as:

$$P(I|R,J).$$

or alternatively $I(q,d)$ can be treated as if it were a single unit term which is estimated as

$$log\frac{P(I|R)P(\bar{I}|\bar{R})}{P(\bar{I}|R)P(I|\bar{R})}. \tag{4.14}$$

In this case, J should be ignored if we know I and then boost equation 4.14 according to the following

$$P(I|R).$$

The simplest compound unit situation to which we could apply it would be to a compound unit of two single unit terms t_1t_2[7], each of which was present in the query in its own right. This would involve the implication

$$t_1\ adj\ t_2 \Longrightarrow t_1 \wedge t_2$$

The implications $t_1 \wedge t_2 \Longrightarrow t_1$ and $t_1 \wedge t_2 \Longrightarrow t_2$ are already dealt with by the independence assumptions. Thus $t_1 \wedge t_2$ would be given its usual weight as the sum of the weights of t_1 and t_2 or $t_1\ adj\ t_2$ treated as if it were a single unit term; the boost provided by the phrase would depend on such quantities as

$$P(t_1\ adj\ t_2|t_1 \wedge t_2)$$

If we go back to the first principle in the probabilistic model, we can arrive at an exact formulation of the boost-weight. Within the set of documents defined by the implied conditions J, the presence or absence of I can be weighted in accordance with the original presence-absence formulation of RSJ. This would give the presence of I a weight of:

$$log\frac{P(I|R,J)}{P(I|\bar{R},J)} \tag{4.15}$$

and the absence of I a weight (which would normally be negative) of:

$$log\frac{P(\bar{I}|R,J)}{P(\bar{I}|\bar{R},J)} \tag{4.16}$$

The difference of these two formula gives a limited version of the usual log-odds weight for presence, which would be the net effect of the presence of I on

[7]Term "t_1t_2" is the same as term "$t_1\ adj\ t_2$"

the evidence for relevance, given J. However, in the present case the score of documents in this set would need to be comparable with other documents not containing J. This means that strictly, we should be applying presence-absence weights: that is, when we boost the documents containing the compound unit, we should also downgrade those documents in the $\wedge$ set which do not contain the compound unit.

For example, if the compound unit t_1 *adj* t_2 is weighted as a single unit term by using either weighting function BM25 or BM26, then we need to design a boost weight for the presence of the compound unit t_1 *adj* t_2 in the documents; we also need to design one more extra boost weight for the absence of the compound unit t_1t_2. By using this extra boost weight for the absence of the compound unit, we do not need to downgrade those documents in the $\wedge$ set which do not contain the compound unit t_1t_2. However, if t_1t_2 is not treated as a single unit term, a weight downgrade for the above $\wedge$ set document may still be needed.

So the boost-weight $bow_{I|J}$ should be:

$$bow_{I|J} = log\frac{P(I|R,J)P(\bar{I}|\bar{R},J)}{P(\bar{I}|R,J)P(I|\bar{R},J)} \tag{4.17}$$

which is precisely the RSJ weight[8] for I given the limit set defined by J. The above boost-weight equation can also be expressed as follows:

$$bow_{I|J} = log\frac{p'(1-q')}{q'(1-p')} \tag{4.18}$$

where p'=$P(I|R,J)$ and q'=$P(I|\bar{R},J)$.

The equation 4.18 is the formula for boosting equation 4.13. To boost equation 4.14, J should be ignored.

4.3.4 Approaches to estimating $bow(I|J)$

The difficulty of estimating the boost weight by using equation 4.17 in a consistent manner is very considerable, particularly since the score of the **AND** or "$\wedge$" set is not derived directly, but indirectly via the weights of the individual terms.

[8]RSJ weighting can be expressed as $w = logO(t\ present|Rel) - logO(t\ present|notRel)$, where O is odds, that is, as a difference in log-odds.

For example, the score of downgrade documents in the **AND** set should almost certainly remain higher than those containing *only* one of the constituent terms; but the scheme to do this consistently would be very hard to construct.

This simplest way would be to use some flat bonus weight [97], independent of any other factors. Given the likely difficulty of any alternative method, this method is actually quite an attractive idea to try since we do not need to be concerned about the complexity of the phrase processing done for the practical application.

One major advantage of the flat bonus is that it does not require determination of the next-most-specific implied condition and its simplicity. A major disadvantage of the flat bonus is that it does not suggest any way of using relevance feedback information. The obvious alternative is to treat $bow_{I|J}$ in the same sort of way as a normal RSJ weight, with a formula which would have a sensible value in the absence of relevance information, but would be refined as such information was obtained. This alternative method suggests a way of using relevance feedback information. However, we are now proposing the use of RSJ weighting within a very much smaller set of documents, namely those retrieved by J. In these circumstances, the point-5 formula would frequently give negative boost weights. This would not be acceptable.

The following two tables 4.3 and 4.4 show some possible approaches to estimating the boost weight 4.17 of a compound unit in a very simple way. In these two tables, the boost weight is divided into two parts. The first part is for the presence of I (such as the presence of $t_1\ adj\ t_2\ adj \cdots adj\ t_j$). And the other part is for the absence of I. In Table 4.3, the compound unit $t_1\ adj\ t_2\ adj \cdots adj\ t_j$ is treated as a single unit term and BM25/BM26 is used to calculate its weight. By using these two boost weights in Table 4.3, the assumption 4.4 for the compound unit is always satisfied. In Table 4.4, the compound unit $t_1\ adj\ t_2\ adj \cdots adj\ t_j$ is calculated as the sum of term weights for $t_1 \wedge t_2 \wedge \cdots \wedge t_j$ and the second part boost weight (the first part boost weight is always set to 0). By calculating the weight for compound unit in this way, the assumption 4.4 for the compound unit weighting can always be satisfied too.

From the Tables 4.3 and 4.4 we can see that if we design a second boost weight for the absence of the compound unit in the documents, we may not need to

Boosting methods	First boost weight (4.15)	Second boost weight (4.16)
$Method_1$	$\sum_{i=1}^{j} w_{t_i}$	$\sum_{i=1}^{j} w_{t_i}$
$Method_2$	$\sum_{i=1}^{j} w_{t_i}$	j^k
$Method_3$	$\sum_{i=1}^{j} w_{t_i}$	0
$Method_4$	$\sum_{i=1}^{j} w_{t_i}$	$log\frac{\#(t_1 \wedge t_2 \wedge \cdots \wedge t_j)}{\#(t_1\ adj\ t_2\ adj \cdots adj\ t_j)}$
$Method_5$	$\sum_{i=1}^{j} w_{t_i}$	$w_{t_1 t_2 \ldots t_j} * (log(j) - 1)$
$Method_6$	$\sum_{i=1}^{j} w_{t_i}$	$\frac{\sum_{i=1}^{j} w_{t_i}}{d}$
where $\#(t)$ indicates the number of documents containing the term t and $k \in [0, 2]$ and d are tuning constants		

Table 4.3: Boost Weight for Equation 4.14

Boosting methods	First boost weighting (4.15)	Second boost weighting (4.16)
$Method_7$	0	$\frac{w_{t_1 t_2 \ldots t_j}}{d}$
$Method_8$	0	j^k
$Method_9$	0	$log\frac{\#(t_1 \wedge t_2 \wedge \cdots \wedge t_j)}{\#(t_1\ adj\ t_2\ adj \cdots adj\ t_j)}$
$Method_{10}$	0	$(j-1) * \sum_{i=1}^{j} w_{t_i}$
where $\#(t)$ indicates the number of documents containing the term t and $k \in [0, 2]$ and d are tuning constants		

Table 4.4: Boost Weight Function for Equation 4.13

downgrade those documents in the $\wedge$ set which do not contain the compound unit. Also, if single unit weighting functions BM25 or BM26 are used to calculate the weight for compound unit $t_1\ adj\ t_2\ adj \cdots adj\ t_j$, the computational complexity will increase.

4.3.5 Compound Unit Weighting Functions

In this section, a more precise description of the compound unit weighting functions derived from the probabilistic formulation 4.17 is given. All these designed compound unit weighting functions are based on the analysis and proofs in the previous two sections.

Suppose that we have a sequence of j adjacent units $t_1 t_2 ... t_j$ (characters or words) constituting a single larger unit (word or phrase). Each unit (large or small) has a "natural" weight, given by a single unit weighting formula; let these be w_{t_i} and $w_{t_1 t_2 \ldots t_j}$ respectively. Table 4.5 gives us ten compound unit weighting functions

(denoted as $Weight_1$, $Weight_2$, $\cdots$ and $Weight_{10}$). Only $Weight_1$, $Weight_2$, $Weight_3$, $Weight_4$, $Weight_5$ and $Weight_6$ are used in our Chinese experiments[9], each of which is coupled with BM25 or BM26 for single unit weighting. Detailed empirical results and analysis for these boosting weight methods will be given in Chapter 7.

Weight methods	$w(t_1\ adj\ t_2\ adj \cdots adj\ t_j)$	$w(t_1 \wedge t_2 \wedge \cdots \wedge t_j)$
$Weight_1$	$w_{t_1 t_2 \ldots t_j} + 2\sum_{i=1}^{j} w_{t_i}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_2$	$w_{t_1 t_2 \ldots t_j} + \sum_{i=1}^{j} w_{t_i} + j^k$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_3$	$w_{t_1 t_2 \ldots t_j} + \sum_{i=1}^{j} w_{t_i}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_4$	$w_{t_1 t_2 \ldots t_j} + \sum_{i=1}^{j} w_{t_i} + log\frac{\#(t_1 \wedge t_2 \wedge \cdots \wedge t_j)}{\#(t_1\ adj\ t_2\ adj \cdots adj\ t_j)}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_5$	$w_{t_1 t_2 \ldots t_j} * log(j) + \sum_{i=1}^{j} w_{t_i}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_6$	$w_{t_1 t_2 \ldots t_j} + \sum_{i=1}^{j} w_{t_i} + \frac{\sum_{i=1}^{j} w_{t_i}}{d}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_7$	$\sum_{i=1}^{j} w_{t_i} + \frac{w_{t_1 t_2 \ldots t_j}}{d}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_8$	$\sum_{i=1}^{j} w_{t_i} + j^k$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_9$	$\sum_{i=1}^{j} w_{t_i} + log\frac{\#(t_1 \wedge t_2 \wedge \cdots \wedge t_j)}{\#(t_1\ adj\ t_2\ adj \cdots adj\ t_j)}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_{10}$	$j * \sum_{i=1}^{j} w_{t_i}$	$\sum_{i=1}^{j} w_{t_i}$
where $\#(t)$ indicates the number of documents containing the term t and $k \in [0, 2]$ and d are tuning constants		

Table 4.5: Weight Methods

4.4 Discussion

Using probabilistic weighting methods to retrieval information has always been a challenging task in the area of information retrieval. In this chapter, a new single unit weighting method BM26 and ten new compound unit weighting methods, have been proposed. All the above analysis and newly designed weighting functions provide a theoretical basis for our next step experiment and evaluation. These experiments will compare different weighting methods on Chinese word and single character based retrieval systems. Our objective is to determine (1) whether compound unit weight weighting is useful for improving the system retrieval performance; and (2) whether BM26 can give positive contribution to the improvement of Chinese information retrieval.

[9] $Weight_7$, $Weight_8$, $Weight_9$ and $Weight_{10}$ were not used in our official experiments because their initial results are not good.

The other thing that needs more investigating is to find out a good method to estimate the boost weight according to 4.17. We should boost the documents containing the phrase and also should downgrade those documents in the AND set which do not contain the phrase. Our new compound unit weighting functions can guarantee to boost the the documents containing the phrase, but not to downgrade those documents in the AND set which do not contain the phrase. We may need to figure out a solution to this issue in the future.

Chapter 5

Chinese Information Retrieval with Okapi

To familiarize the reader with the Okapi retrieval system, a detailed description of the system, as well as the Chinese text retrieval system based on Okapi, is given in this chapter.

5.1 Okapi Retrieval System

Okapi is an experimental information retrieval system, developed in the Centre for Interactive Systems Research in the Department of Information Science at City University. The Okapi retrieval system is based on the probabilistic retrieval model. This model was first developed by Steve Robertson and Karen Sparck Jones in the 1970s [85] and has been used extensively with the Okapi experimental retrieval system. Okapi represents more than twelve years of research in probabilistic retrieval [1] and is one of the most advanced retrieval systems in the world.

5.1.1 The Okapi Projects

The original Okapi system was constructed in 1982-1984 as part of a project to develop an experimental online catalogue search system at the Polytechnic of Central London (now the University of Westminster). A number of other projects and versions of the system followed, investigating the effect of automatic stemming,

[1] See the special edition of the *Journal of Documentation* Volume 3, Issue 1, January 1997, ISSN 022 0418, which contains eight papers and three research briefs on various aspects of the Okapi system and Okapi-based projects.

automatic cross-referencing, user-aided spelling correction and automatic query expansion.

In 1988 the projects moved to City University, with the intention of building a flexible tool for investigating aspects of interactive retrieval of bibliographic references and other textual material. Further experiments with automatic query expansion in live searching of both catalogue and abstracting and indexing databases followed.

5.1.2 A Typical Okapi System

Most Okapi systems are designed as search systems for end users who are not expert searchers. What the user sees first is an invitation to enter a query in a free-text form. This free-text query is then parsed into a list of single word-stems; each stem is given a weight based on its collection frequency. The system then produces a ranked list of documents according to a best-match function based on the term weights, and shows the user titles of the top few items in the list. The user can scroll the list and select any title for viewing of the full record. Having seen the full record, he or she is asked to make a relevance judgement ('Is this the kind of thing you want?') in a yes/no form.

Once the user has marked a few items as relevant, he or she has the opportunity to perform a relevance feedback search ('More like the ones you have chosen'). For this purpose, the system extracts terms from the chosen documents and makes up a new query from these terms. This is normally referred to as query expansion, although the new query may not necessarily contain all the original terms entered by the user. The new query is run and produces a ranked list in the usual fashion, and the process can iterate.

5.1.3 The Probabilistic Retrieval Model

The basic weighting-ranking-and-relevance-feedback mechanism of Okapi is based on a probabilistic model of information retrieval, which leads to a search term weighting formula and a match function for documents. The original model, proposed in 1976 [85], took account of term presence only in the requests and documents (that is, both were regarded simply as lists of terms). It has since been

extended to take account of within-document and within-query term frequency and document-length, and may also make some use of information specifying term position in the document. A simple presentation of some of the weighting mechanisms in current use is given in Robertson and Sparck Jones [98]. A more detailed description about the currently used formula in Okapi is given in chapter 3.

5.1.4 Structure of Okapi

The current structure of the Okapi system could be summarised as follows:

- indexing routines: these are not highly developed, in that (for example) there is no updating system - if a collection is altered it has to be re-indexed from scratch. The indexing routines generate inverted files of a relatively conventional two-stage kind, but normally containing full position information;
- a search engine, the BSS (Basic Search System). The Okapi Basic Search System, which has been used in all Okapi TREC experiments, is a simple and robust set-oriented ranked output system based on a generalised probabilistic model with facilities for relevance feedback, but also supporting a range of deterministic Boolean and quasi-Boolean operation (such as proximity and limit operations). This search engine provides efficient low-level functionality for weighting and ranking searches;
- various interface systems, providing the sort of variant facilities discussed above.

Figure 5.1 gives in diagrammatic form an outline of the structure of Okapi. It indicates, for example, that in at least some versions the query model [2] may make reference to a thesaurus; that the query model is also generally responsible for logging searches for evaluation purposes; and that batch searching (for some experiments) is usually accomplished by means of scripts which access the BSS directly.

[2]Query model supports the maintenance and use of a model of the current query, including for example the list of items currently marked relevant, and the list of candidate terms for query expansion.

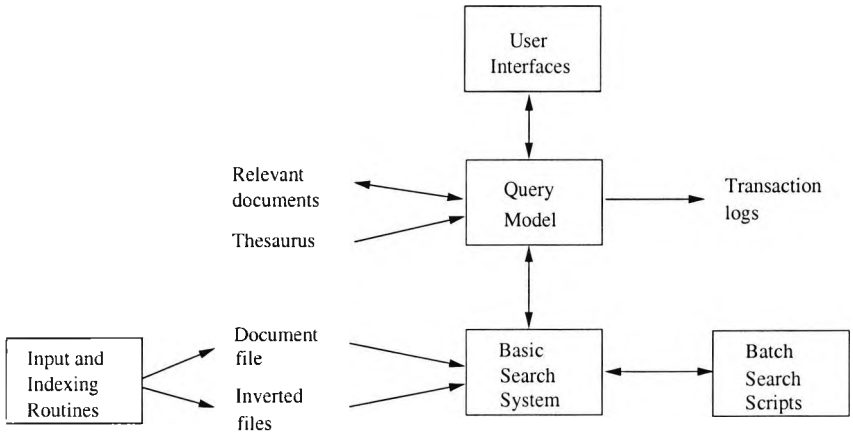


Figure 5.1: Overview of the Structure at Okapi

5.2 Chinese Text Retrieval with Okapi

The Okapi-based Chinese text retrieval system(s) *C-Okapi*, which has been used in TREC-5 and TREC-6 experiments, is a simple and efficient Chinese text retrieval system based on an enhanced probabilistic model for Chinese text retrieval. *C-Okapi* is designed for Chinese laboratory experiments so that we can try out ideas, to evaluate different methods, and to aid in the development of retrieval theories for Chinese text retrieval.

In order to give a picture of the *C-Okapi*, it is appropriate to describe one typical version containing some of the features that have become central to our Okapi-based Chinese systems. In this section, first we will give the system architecture of the *C-Okapi* system; second we will describe the dictionary we use for word segmentation; third we will describe the word segmentation algorithm we used in the *C-Okapi* system; finally, design and implementation details of the system, which include indexing and searching, will be given.

5.2.1 System Architecture

The C-Okapi system consists of four functional components: dictionary construction, document indexing, query processing and retrieval. Figure 5.2 illustrates the general structure of C-Okapi. The dictionary construction component builds a word dictionary from a manually-constructed collection of words[3]. The indexing

[3]The *word* dictionary may contain other terms than words, such as phrases, depending on what is in the manually-constructed collection of *words*. For convenience, we assume all the terms in

component takes a document collection (written in natural language) and generates an index file of documents. It may use the word dictionary for indexing purposes, depending on whether the index file is word-based or character-based. The query processing component processes a natural language query according to the word dictionary and outputs a ranked list of query terms. A detailed description of query processing will be given in section 7.1.5. The retrieval component searches for relevant documents according to the query terms and the index file, and generates an ordered list of documents.

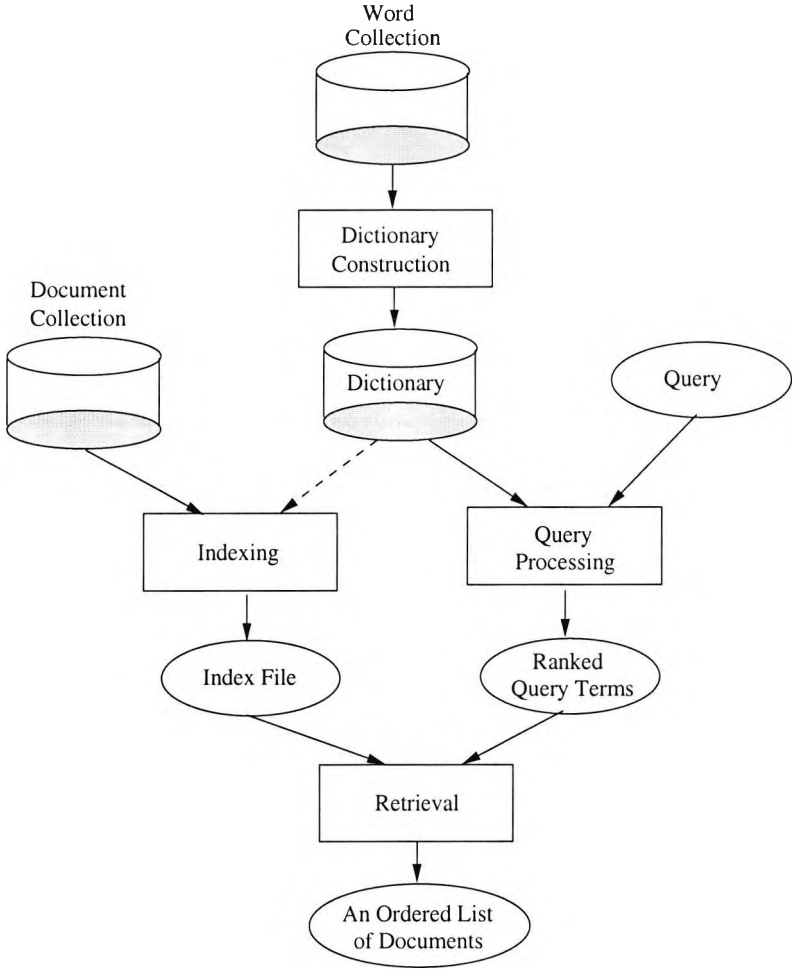


Figure 5.2: System Architecture for Chinese Retrieval System

Detailed descriptions of the last three functional components are illustrated in Figure 5.3 (for the word-based C-Okapi system) and Figure 5.4 (for the character-based C-Okapi system). The indexing component is decomposed into text segmentation for each document and sorting and merging of all the segmented terms

this collection are words in order to distinguish between our word-based and character-based methods.

from different documents in the collection. The query processing component is decomposed into text segmentation of the query and combination and ranking of the query terms. The retrieval component weights these query terms, calculates the weights for each document in the collection, ranks the documents according to the document weights and outputs an ordered list of documents.

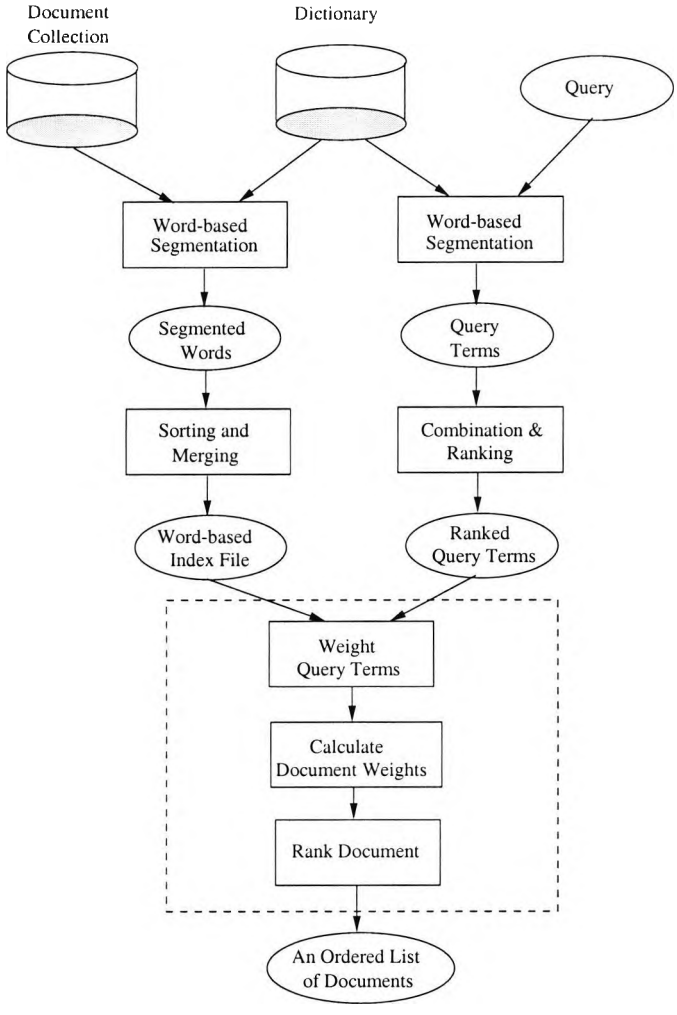


Figure 5.3: System Architecture for Word-based Chinese Retrieval System

5.2.2 Dictionary

The Chinese dictionary we use for our word approach retrieval system contains 69,353 Chinese words and phrases. This dictionary was manually constructed in China and has been used as a standard dictionary for Chinese text segmentation in the 1990s. The project for constructing this Chinese dictionary was supported by the National AI Lab in Beijing and it took about two years from 1988 to 1990

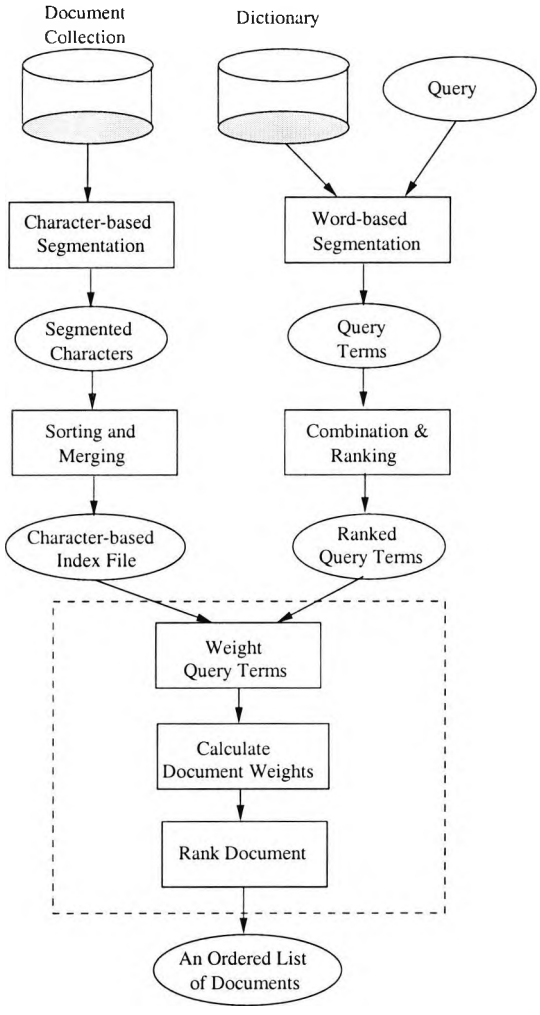


Figure 5.4: System Architecture for Character-based Chinese Retrieval System

to finish [43].

The dictionary is an extended version of a commonly-used Chinese word dictionary [67]. The terms that are not in the commonly-used dictionary were selected by a group of Chinese linguistic experts according to the national standard GB13715 [33] for Chinese text segmentation. The commonly-used Chinese word dictionary consists of 7,055 first-level commonly-used words and 29,355 second-level commonly-used words plus 2,606 single character words totaling 39,016 words which all satisfy the national standard GB13715. These 39,016 commonly-used words were selected from the top of an ordered frequency list that contained 77,482 different Chinese words. Detailed results for the coverage of these 77,482 words are shown in Table 5.1. All of these 77,482 Chinese words in Table 5.1 were extracted from a specific corpus with more than 20 million words in total [67].

It can be observed from Table 5.1 that the percentage coverage increase is very

Number of Words	*Coverage*	Number of Word Increase	Coverage Increase
500	53.3116%		
1500	70.2003%	1000	16.8887
3500	81.7676%	2000	11.5673
5500	86.8745%	2000	5.1069
7500	89.9086%	2000	3.0341
10500	92.6994%	3000	2.791
15000	95.1050%	4500	2.1056
49065	99.0000%	14065	3.8950
60000	99.9188%	10835	0.9188
77482	100%	17482	0.0812

Table 5.1: Coverage of Chinese Words with Respect to the Number of Words

small after the number of words reaches 60,000. Due to this fact, the extended version of the commonly-used word dictionary which we use in our *C-Okapi* system contains only 69,353 words and phrases, which is not as large as some dictionaries used by other TREC Chinese systems, such as the one used by Berkeley [42]. The Berkeley dictionary contains 137,659 Chinese words and phrases. Most of them were extracted from several Chinese on-line dictionaries. The other reason why we do not use a very large dictionary is: the more words collected in the dictionary, the higher the likelihood of resulting ambiguous words from segmentation [43]. For instance, a Chinese character string, ABC, may be segmented differently to yield A/B/C, AB/C, A/BC or ABC. Such combinations are potential words in the complete dictionary! Increasing the size of the dictionary is not an effective way to increase IR performance. Similar results have also been obtained by [75].

5.2.3 Algorithms for Chinese Word Segmentation

There are usually three alternative matching methods for Chinese word segmentation. The first one is called longest match, in which the longest matched strings are taken as indexing and search tokens and shorter tokens within the longest matched strings are discarded. The second word segmentation method is shortest match method, for which text is sequentially scanned to match the dictionary and the first matched tokens are selected. The third method is the overlap match, for which tokens generated from the text can overlap each other across

the matching boundary. The longest match generates less tokens with more specific meaning; while the shortest match generates more tokens with less specific meaning. These three methods can be implemented by "forward" or "backward" scanning. The "forward" scan means to start from the beginning of the string and the "backward" scan means to start from the end of the string. There are some other word segmentation methods such as statistical methods, rule-based methods or hybrid methods, but these methods are not widely used. The most successful word segmentation method is the longest matching by forward scanning, which usually produces better segmentation results than some other methods [19, 43].

In our *C-Okapi* system, we use the longest matching from beginning which is shown in Figure 5.5 as our word segmentation algorithm. For example, this algorithm is to extract a string of a certain number k (usually $k = 7$) of characters and to search for it in the dictionary. If it exists in the dictionary, the process continues with the next k-character text string. If it doesn't, the last character in the string is removed and the new string is then searched for in the dictionary. Figure 5.5 shows the longest match algorithm. According to the word segmentation produced by this algorithm, the Chinese TREC texts consist of approximately 43.6 million segmented units. Some simple rules have been implemented in our longest matching algorithm such as segment the "2000" in Chinese as a single word.[4]

5.2.4 Algorithm for Sorting and Merging Segmented Results

The TREC Chinese collection was segmented into two files for use with *C-Okapi*. One file is for the character-based approach with the size of about 1 giga-byte and the other file is for the word-based approach with the size of about 0.6 giga-byte. It is obvious that we can not load such a big file of word-based or character-based system into memory at one time and sort them. The reason is that this size of file is much bigger than the memory of any computer used in the experiments reported here. For this reason, a well-designed algorithm for sorting and merging a very large-size file is necessary. This algorithm for the word-based

[4] "2000" in Chinese is not included in our dictionary for word segmentation

ALGORITHM: The Longest Match
INPUT: A string of Chinese characters $a_1 a_2 \cdots a_k$;
OUTPUT: A set of Chinese words and phrases
begin
    let i be 1 and j be k
    while $(i <= k)$
    begin
        if a_i is a digital character or English character
        then call special_string_processing($a_i \cdots a_j$);
        else if there is no character a_i in the index file of word segmentation
                dictionary or i equals to j
            then put a_i into segmentation result file;
                increase i by 1;
            else if there is a string $a_i \cdots a_j$ in the lexicon file of word
                segmentation dictionary
                then if there are possible ambiguous segmentation for the
                    the string $a_i \cdots a_j$
                    then call ambiguity_processing($a_i \cdots a_j$);
                    else put $a_i \cdots a_j$ into segmentation result file;
                        let i be j;
                else decrease j by 1;
    endwhile ;
end

Figure 5.5: The Longest Match Algorithm

ALGORITHM: Sorting and Merging for Word Approach
INPUT: A collection of Chinese documents;
OUTPUT: Word-based indexing file
Step 1
begin
for $(i = 1st\ doc;\ i <= last\ doc;\ i++)$
begin
1. read document;
2. segment document into words;
3. output word, document_no, paragraph_no, sentence_no and character_no;
4. store the string word, document_no, paragraph_no, sentence_no and character_no into memory until the memory is full;
5. sort them in the memory by word, document_no, paragraph_no, sentence_no and character_no;
6. output to $temp_i$ file;

endfor ;
end

Step 2
begin
1. merge $temp_1$, ..., $temp_n$ into a final output merged file;
2. generate the index files 'word_index' and 'word_invert';

end

Figure 5.6: The Sorting and Merging Algorithm for Word Approach

system, which is pretty similar to the sorting and merging algorithm for character-based system[5], is shown in Figure 5.6. The basic idea of this algorithm is to divide this big inverted file into many small files. We sort these small files, merge all these sorted small files into one large sorted file and then generate the index file of the test collection based on this large sorted file.

5.2.5 Algorithms for Retrieval

The objective of retrieval is to retrieve from the document collection those documents that are relevant to the query. In C-Okapi, we assign a weight to each document to measure the probability of relevance of each document with respect to a set of query keywords obtained from the query processing module, and output

[5]The only difference for character-based algorithm is that we need replace the "word" with the "character" in the word-based algorithm

the 1000 documents with the highest weights. Figure 5.7 and Figure 5.8 describe the algorithms for C-Okapi's character-based retrieval and word-based retrieval, respectively. In character-based retrieval, the algorithm first calculates a number of frequencies and numbers used for calculating the weight of each query keyword, and then uses one of the six compound unit weighting formulae to compute a weight for each query keyword. Each query keyword is selected from the segmented terms and the adjacent pairs of these segmented terms. This means that one selected query keyword may be part of several other keywords. A more detailed description will be presented in section of 7.1.5. In computation of a compound unit weight, the weight for a single unit inside the compound unit is calculated using BM26, which is defined in equation 4.3, without the correction factor. The algorithm then calculates the weight for each document by first summing up the weights of the keywords that are contained by the document and then adding the value of the correction factor in BM26 to the sum. The last step of the algorithm ranks the documents in decreasing order of their weights and then outputs the top 1000 documents as the retrieval result.

The algorithm for the word-based retrieval takes similar steps except the following:

1. At the beginning of the algorithm, each query keyword is segmented into single units if it is a compound unit. These single units will be used in calculation of the compound unit weight and they themselves will also contribute to the calculation of the document weight as if they were query keywords.

2. In calculation of the weight for a query keyword, either single unit or compound unit weighting is used depending on whether the keyword is a single unit or a compound unit.

5.2.6 Design and Implementation

The original version of *C-Okapi* was implemented from June 1995 to October 1995[6]. Before that, a very basic graphical user interface had also been implemented.

[6]Two Chinese software packages ZWDOS and CXTERM, which can run in a networked environment, were obtained from the Internet and used as a basic environment for the implementation

ALGORITHM: Character-based Retrieval
INPUT: A ranked list of retrieval keywords with weights;
OUTPUT: A ranked list of top 1000 documents
begin

let C be the set of distinct characters in the list of query keywords;
let K be the set of distinct keywords in the list of query keywords;
let D be the set of distinct candidate documents in which at least one keyword appears;

1. calculate the frequency of each character $c \in C$;
2. calculate the frequency of each keyword $k \in K$;
3. calculate the number of candidate documents in D;
4. calculate the number of distinct documents for each character in C within the total collection;

for $(i = 1st\ doc\ in\ D;\ i <= last\ doc\ in\ D;\ i{+}{+})$
begin
5.1. calculate the frequency of each query keyword for every document in D;
5.2. calculate the frequency of each character in C for every document in D;
endfor ;

6. calculate the weight of each keyword in K by using compound unit weighting formula $Weight_1$, ..., or $Weight_6$, in which the weight for each single unit (including the whole keyword and each character in the keyword) is calculated using BM26 without the correction factor;
7. calculate the weight of each document in D by summing up the weights of all the keywords contained by the document and adding the correction factor in BM26;
8. rank the documents according to their weights and output the top 1000 documents;

end

Figure 5.7: The Character-based Retrieval Algorithm.

ALGORITHM: Word-based Retrieval
INPUT: A ranked list of retrieval keywords;
OUTPUT: A ranked list of top 1000 documents
begin

let K be the set of distinct keywords in the list of query keywords and S be the set of distinct segmented words from K;
let D be the set of distinct candidate documents in which at least one keyword appears;

1. segment each keyword $k \in K$;
2. calculate the frequency of each term $t \in K \bigcup S$;
3. calculate the number of candidate documents in D;

for $(i = 1st\ doc\ in\ D;\ i <= last\ doc\ in\ D;\ i{+}{+})$
begin
4. calculate the frequency of each term $t \in K \bigcup S$ for every document in D;

endfor ;

5. calculate the weight of each term $t \in K \bigcup S$ by using BM26 without the correction factor or compound unit weighting formula $Weight_1$, ..., or $Weight_6$, depending on whether t is a single unit or a compound unit;
6. calculate the weight of each document in D by summing up the weights of all the terms in $\in K \bigcup S$ and adding a correction factor in BM26;
7. rank the documents according to their weights and output the top 1000 documents;

end

Figure 5.8: The Word-based Retrieval Algorithm.

The initial version of *C-Okapi* is a single program incorporating search engine, indexing, merging and topic processing etc.

A lot of improvements have been made since 1996. Now we have implemented a Chinese text retrieval system based on the enhanced probabilistic methods, modeled on the Okapi system. It supports Chinese text segmentation, weighted searching, sorting and output ranked documents. It also supports constructing dictionary for Chinese word segmentation and indexing databases for both the word-based systems and character-based systems. *C-Okapi* can support both word-based retrieval and character-based retrieval. But there are no supports in *C-Okapi* for relevance feedback and query expansion. All these programs were written in C under the UNIX environment.

Data Structure for the Index of Dictionary

As we know, the objective of information retrieval is to find from a data collection a set of relevant documents based on a user's query. Since in most cases, the query and the documents are written in natural languages, usually the first thing we need to do is text extraction from the documents and from the query. As we have discussed in the previous chapter, the Chinese words in a text are not separated by spaces, a word segmentation program is usually needed for word-based systems. Our word segmentation program is based on a dictionary with 69,353 Chinese terms.

The data structure we designed for the index of dictionary is shown in Figure 5.9. This data structure is stratified into two levels: index file level and lexicon file level. The index file is organized character by character which means one line per Chinese character in increasing order of the Chinese character's internal code. The lexicon file is organized word by word which means one line per Chinese word in increasing order of the Chinese word's internal code. The index file consists of three fields "character", "begin" and "end". The "begin" field is the pointer that points to the beginning of a list of Chinese words in the lexicon file that start with the corresponding Chinese character and the "end" field is the pointer

---

at the beginning. ZWDOS and CXTERM provide the ability of input and display Chinese on PC and UNIX workstation respectively.

that points to the end of this list. The lexicon file also consists of three fields "word", "knowledge" and "length". The knowledge field in the lexicon file is set to 0 or 1. "0" means that there is no rule associated with this word and "1" means that there are some segmentation rules associated with it. For example, the Chinese character "安" in the index file is associated with a list of words which start with the character "安" in the lexicon file (126 words in total). The motivation for designing the rule base module in Figure 5.9 is to try to solve the ambiguity problem in Chinese segmentation.

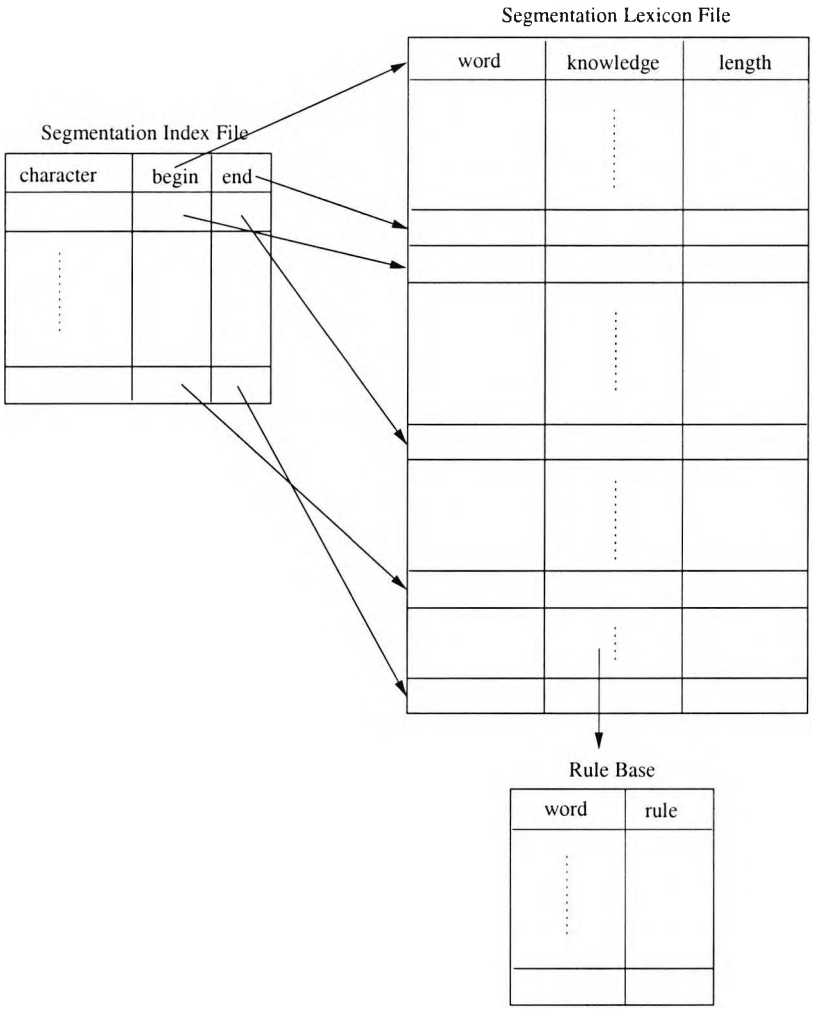


Figure 5.9: Data Structure for Segmentation Dictionary's Index File

Data Structures for the Word and Character Indexes

Indexing is a pre-requisite for word-based and character-based retrieval. The function of the indexing module is to create a database from a text collection. The indexing module can be run including segmentation, so that the index is

word-based, or without, so that the index is character-based. The data structures we designed for the word-based index and character-based index are shown in Figure 5.10 and 5.11 respectively. The data structures for word-based index and character-based index are stratified into two levels too: index file level and inverted file level. The index files are organized as one line per Chinese word for word-based approach and one line per Chinese character for character-based approach. The index file consists of four fields "keyword", "knowledge", "begin" and "end". The keyword field is either Chinese word for word-based approach or Chinese character for character-based approach. The knowledge field is set to 0 or 1. "0" means that there is nothing associated with the keyword field and "1" means that there is a thesaurus or rule base associated with the keyword field, which can be used in retrieval. The "begin" field is the pointer that points to the beginning of a list documents that the corresponding keyword (word or character) occurs. The inverted file consists of four fields "document_no", "paragraph_no", "sentence_no" and "character_no" and lists all the documents containing the corresponding keyword (word or character). Thus the inverted file keeps track of the position information of the Chinese word or character in the documents. Again, there is a thesaurus module for word-based indexing and a rule based module for character-based indexing in our original design.

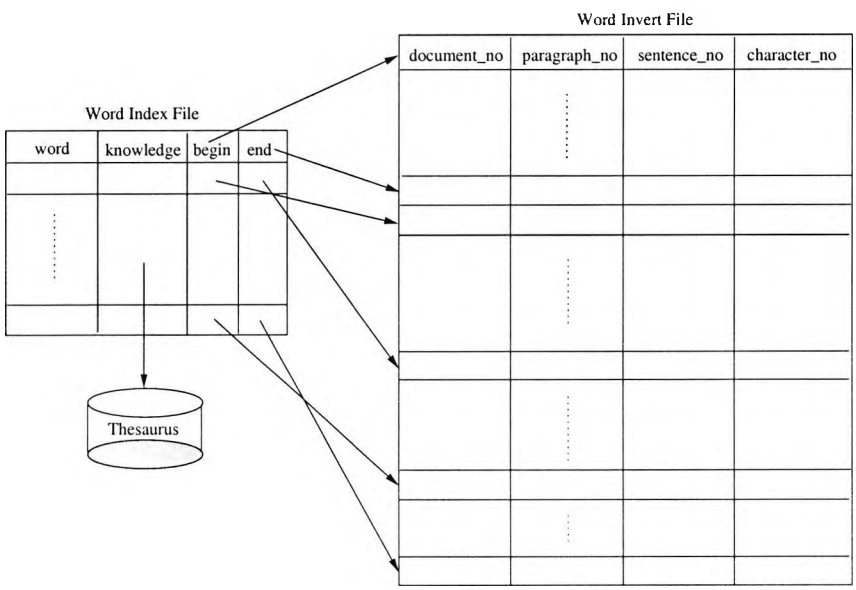


Figure 5.10: Data Structure for Word-based System's Index File

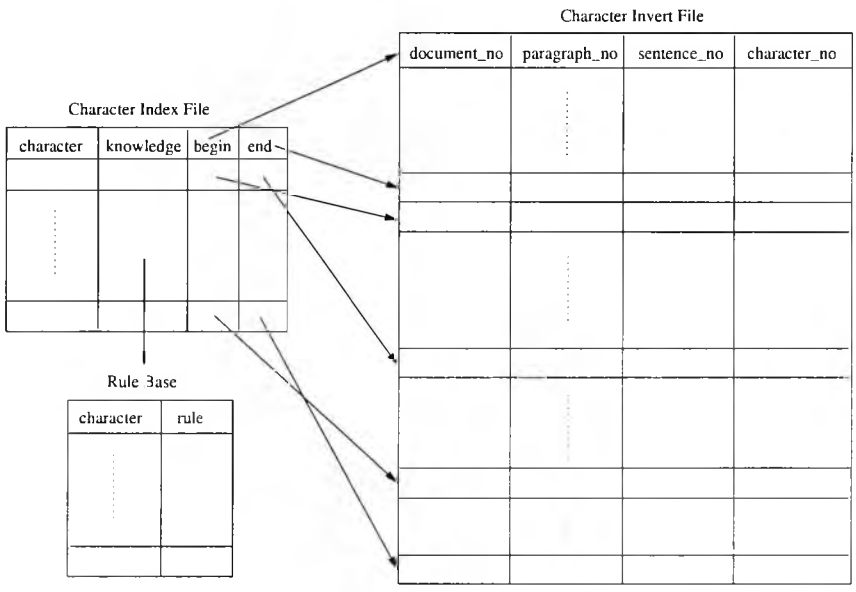


Figure 5.11: Data Structure for Character-based System's Index File

Data Structures for Retrieval

The data structures for the search module are the most important part in our system design and implementation. In this section, we will concentrate on the character-based system. Two important data structures for the character-based system are described in Figure 5.12 and Figure 5.13. Figure 5.12 is the data structure designed for each character appearing in query and Figure 5.13 is the data structure designed for each document in the candidate document set D. By using these two data structures, our search programs can retrieve documents from the database based on either word approaches or single character approaches, with a range of different weighting methods. Text segmentation can be applied to the queries for searching the word index, or for searching the character index using the adjacency operator. Alternatively, no segmentation need be applied: characters may be searched singly, and/or with the adjacency operator applied to pairs.

By using the data structure shown in Figure 5.13, we can calculate the document weight for each candidate document and rank these candidate documents. In order to calculate the weight for these candidate documents, we may need to use the data structure shown in Figure 5.12 to compute the weight of each character appearing in the retrieval keywords[7]. The algorithm for calculating

[7]For example, we need to use both of these two data structures shown in Figure 5.12 and 5.13 to obtain all the necessary information for calculating the compound unit weighting function

the document weight using $Weight_1$ and $Weight_2$ is shown in Figure 5.14. We can observe from Figure 5.14 that for $Weight_1$ we calculate the "z" by using "$retrieved\_document[i].sum\_hz\_weight[j]$" and for $Weight_2$ we calculate the "z" by using "$MULTIPLY * pow(x, y)$".

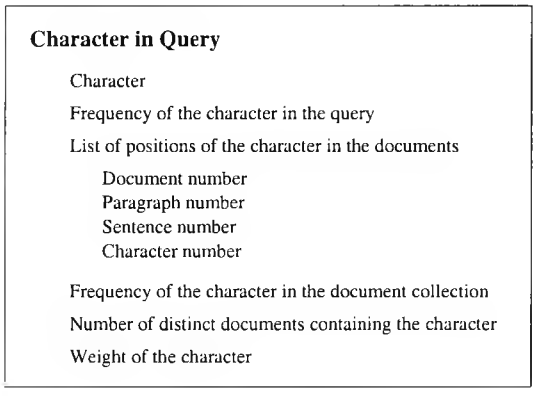


Figure 5.12: Data Structure for Retrieved Character

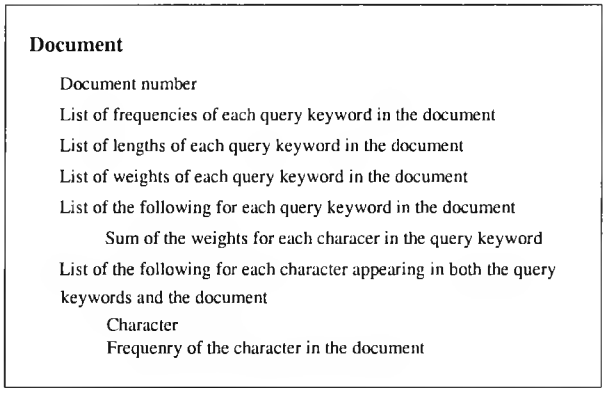


Figure 5.13: Data Structure for Retrieved Document

The data structure designed for the word-based system is pretty similar and only the first four items in Figure 5.13 are used. These four items are "Document number", "List of frequencies", "List of lengths" and "List of weights" of each query keyword in this document. The set of possible values for the "Document number" attribute is 1, 2, ..., 164768. Actually, the real values for the "Document number" are strings such as "CB058029-BFW-611-752" and we convert each of the document number strings to an integer number for convenience. From the above discussion it can be observed that: first, both of the data structures described in Figure 5.12 and Figure 5.13 are needed in order to compute the document

---

$Weight_1$ and $Weight_6$

ALGORITHM: Calculating the Weight of Each Candidate Document
INPUT: $retrieved\_document[Max\_Retrieved\_Doc]$ and $parameter\_k_2$;
OUTPUT: A ranked list of top 1000 documents
begin

```
    Initialize;

    let size_of_document be the number of distinct candidate
      documents in which at least one keyword appears;
    let size_of_query be the number of distinct keywords in
      the list of query keywords;
    for  (i = 0; i < size_of_document; i++)
    begin
      for  (j = 0; j < size_of_query; j++)
      begin
        if (retrieved_document[i].frequency[j] > 0)
        then
        begin
          x = retrieved_document[i].word_length[j];
          y = parameter_k2;
          if Weight2 is used
          then z = pow(x, y);
          else if Weight1 is used
          then z = retrieved_document[i].sum_hz_weight[j]

          document_weight = retrieved_document[i].word_weight[j] + z;
        endthen
      endfor ;
    endfor ;

    Calculate the correction factor for each candidate document
    by using equation 4.2;
    Sort and rank all the candidate documents;
    Output the top 1,000 candidate documents;
```

end

Figure 5.14: Algorithm for Calculating the Weight of Each Candidate Document

weight for character-based systems using $Weight_1$ and $Weight_6$ as compound unit weighting functions; second, for word-based systems and character-based systems using $Weight_2$ and $Weight_5$ as compound unit weighting functions, only the first four items in Figure 5.13 are needed. Obviously, a lot of memory resource can be saved in this way.

Chapter 6

Chinese Experiments with Okapi

The experimental evaluation of information retrieval systems in a laboratory context now has a history of about forty years. The basic ideas were originally formed in the Cranfield projects in the late fifties and early sixties [99]. The major experimental programme of recent years has been TREC, the Text Retrieval Conference. This chapter gives an overview the development of IR evaluation ideas from Cranfield to TREC and describes the design of the Chinese TREC experiments in this thesis.

6.1 Laboratory Testing in IR

The general model which was developed over the two main Cranfield experiments involved the following components [23, 89]:

- a system: a set of methods and procedures (whether human or machine) for indexing and searching;
- a collection of documents;
- a collection of requests representing information needs;
- an experimental design;
- a basic output evaluation process providing relevance judgements on documents in relation to request/needs;
- performance measures derived from the relevance information.

Despite a continuing tradition of debate and argument about this model, it remains as a generally accepted model for evaluation experiments in information retrieval.

6.1.1 Cranfield

The Cranfield projects pioneered the idea that we may undertake experimental tests of the principles of information retrieval system design. For Cranfield, as for many subsequent experiments, a collection of test materials (that is, documents, request and relevance judgements) was purpose-built. For many researchers in the field a purpose-built collection for their own experiments would have been impossibly expensive to construct. It rapidly became clear that the Cranfield collection itself (and indeed some other collections constructed for specific experiments) could in fact be re-used by other researchers. Such collections became widely distributed among the research community, and were used for an extraordinary range of experiments, far beyond those for which they were originally designed.

In the mid-seventies, about a decade after the process first started, the UK information retrieval community began to discuss the possibility of designing and building a large, general-purposed test collection [122]. This became known as the 'ideal' test collection. Unfortunately, this project was shelved, partly because of the resource commitment it would have required.

It took more than another decade for the project to be resurrected. The resurrected project had a new home (the United States) and a new name (TREC – the Text Retrieval Conference), but retained as the core of its methodology that proposed for the 'ideal' collection.

6.1.2 The Text REtrieval Conference (TREC)

For a long period of time, there had been two missing elements from information retrieval research [38]. First, although there has been a vast amount of work done in the this field since the early 1960's, each research group often used different test collections, different queries and different evaluation techniques. As a result, it was difficult, if not impossible, to compare the performance of various retrieval techniques used by different research groups. Second, many retrieval experiments

were conducted on collections many times smaller than what is common in the commercial world and hence it is unclear how well the experimental techniques would perform in real-life information retrieval environments.

Objectives

In order to address these two missing elements, the Text REtrieval Conference (TREC), co-sponsored by the National Institute of Standards and Technology and the Defense Advanced Research Projects Agency, was organized and has been held annually since 1992[1]. TREC involves participating institutions around the world, and successive cycles of testing under changing data or retrieval conditions [38]. For example, the sixth Text REtrieval Conference (TREC-6) took place in November 1997. The number of participating groups has grown from 25 in TREC-1 to 51 in TREC-6, and includes many information retrieval software companies and most of the universities involved in information retrieval research. Each round of TREC involves the distribution of a large quantity of textual data and a substantial number of requests ('topics' in TREC jargon) to the participants. Each group then undertakes a series of searches and returns the resulting output to NIST for relevance assessment and performance evaluation. As quoted from Voorhees and Harman [133], the main goals of TREC are:

- to encourage research in text retrieval based on large test collections;
- to increase communication among industry, academia and government by creating an open forum for the exchange of research ideas;
- to speed the transfer of technology from research labs into the commercial world by demonstrating substantial improvements in retrieval methodologies on real-world problems;
- to increase the availability of appropriate evaluation techniques more applicable to current systems.

[1]The TREC programme is an extended evaluation project, following a general model of evaluation for other tasks, such as speech recognition and message understanding.

Routing and Ad hoc Tasks

Each of the TREC conferences is organized around two traditional information retrieval modes: the routing task and the ad hoc task. The routing task involves using a stable set of information requests to search new document collections, and is similar to what is required by news clipping services and library profiling system. Participants are given a set of topics, which are natural language descriptions of the users' information requirements, and a document collection (the "training set") that includes known relevant documents for those topics. Based on the given topics, each participating group creates a set of queries that are used as inputs to their retrieval system. By running the queries against the training set and evaluating performance, the participants can adjust, for example, the queries themselves, weighting functions or term weights. When they are satisfied with the queries and the system, the queries are run against a new document collection, and the list of documents retrieved is submitted to the TREC organizers for judging. The participants are required to submit a list of 1000 documents for each topic, with those that are most likely to be relevant ranked at the top.

The ad hoc task involves searching a stable document collection with new topics, and is similar to how a researcher typically uses a library. Participants are given 50 new topics and they have to formulate queries that are run against a document collection of approximately two gigabytes in size. As with the routing task, participants are required to submit a ranked list of 1000 documents per topic. The results are sent to the TREC organizers for judging.

Query Formulation

Query construction methods are classified into two categories: manual or automatic. In automatic query construction, queries are derived automatically from the topic statements with no manual intervention. In manual query construction any other methods are allowed, including interactive feedback, where users modify the queries based on looking at documents retrieved by the initial queries.

Evaluation

The conference is a vehicle for standardizing evaluations of IR systems. The evaluation metrics are versions of precision and recall. One of the issues TREC addresses is the difficulty of obtaining relevance assessments from humans. TREC uses the "pooling" technique [39] to gather relevance assessments. Pooling is based on the assumption that it is highly likely that any truly relevant document will be identified by at least one of the participating systems. Thus, for a given topic, the top 100 documents retrieved in each submitted run for every system are considered candidate relevant documents and are reviewed by the person who originally constructed the corresponding query. He or she labels each as either relevant or irrelevant. Naturally, some relevant documents are missed this way, but it is hopefully a small fraction. After a list of relevant documents for each topic is compiled, precision and recall figures are calculated for each submitted run. The exact evaluation measures used and how they are calculated are shown in a later section.

The TREC collections have become the de-facto standard and yardstick used by IR researchers. This is not to say they do not have shortcomings. However, they represent a significant step forward from previous test collections, presenting more realistic collections in terms of both size and content than previous benchmark corpora. For the tasks of doing IR on long natural language queries against relatively homogeneous, large collections, TREC has no equal.

6.2 Chinese Track at TREC

Beginning in TREC-4, a set of secondary tasks which focus on particular subproblems of text retrieval were introduced. One of them was a multi-lingual track in which participants conducted retrieval experiments on Spanish documents. In TREC-5, Chinese was added to the multi-lingual track. It was found that many of the traditional techniques used on English can also be successfully applied to Spanish, and for this reason the Spanish portion of the multi-lingual track was discontinued and the multi-lingual track became the Chinese track in TREC-6. The unsegmented nature of Chinese texts provided new challenges and opportunities

for extra experimentation.

Since the Chinese track of TREC conferences provides a collection of Chinese documents, a collection of topics, and an output evaluation process that provides relevance judgement on documents in relation to topics, it is an ideal test-bed for our Chinese information retrieval systems. For this reason, we participated in both Chinese TREC-5 and TREC-6.

6.2.1 The Topics and the Document Collections

Chinese TREC topics

The retrieval task for the Chinese track is identical to the *ad hoc task*. Participants at the TREC-5 Chinese track were given 28 topics in both English and Chinese and while at TREC-6 they were given 26 topics. All the 54 Chinese topics are prepared by expert assessors (retired news analysts) who will also make relevance judgements. The task for the Chinese track at TREC-5 and TREC-6 was to retrieve a ranked list of 1000 documents for each topic. The topics are mostly on current affairs and an example is shown in Figure 6.1.

Appendix A lists all the 54 topics used in TREC-5 and TREC-6. Each topic has a title field, a description field and a narrative field. The title field is a short statement of what the topic is about; the description field contains additional terms related to the topic; and the narrative field contains a more detailed description of relevance criteria. As with the main TREC tasks, runs submitted are classified into two categories: those that use manually constructed queries and those that use automatically constructed queries. Relevance assessment is done by the same pooling method used for the main tasks.

Chinese TREC Collections

Both Chinese tracks used the same document collection, which is about 175MB in size and consists of 139,801 articles selected from the People's Daily newspaper and 24,988 articles selected from the Xinhua newswire. Therefore, there are 164,789 articles in total [2]. Chinese TREC collections are made up from two main

[2]K.L Kwok found 10 duplicate document numbers in the Xinhua collection and Donna Harman asked us to remove them before indexing for retrieval purpose. There are total 21 duplicate

```
<top>
<num> Number: CH25

<E-title> China's Protection of Pandas
<C-title> 中国对熊猫的保护

<E-desc> Description:
Ecoprotection, panda, nature preserve, endangered species

<E-narr> Narrative:
A relevant document discusses China's protection of pandas, such as how the Government
sets up nature preserves for pandas, existing nature preserves, the nature preserve
environment, the total number of pandas in China, or increases in the panda population.
An irrelevant document covers panda sighting, without any details about protective
measures, like how the Government is helping pandas to reproduce.

<C-desc> Description:
生态保护，熊猫，保护区，濒临灭绝

<C-narr> Narrative:
相关文件应提到中国对熊猫的保护，比如中国政府如何设立熊猫的保护区，目前熊猫的保护区包括那
些地区；熊猫的生态环境如何；目前中国的熊猫总数大约有多少；以及受到保护后熊猫数量的增长.
不相关文件则包括新闻中只提到在某个地区看到熊猫，但是没有提出具体的保护方法，诸如政府如何设
立保护区来帮助熊猫的繁殖.
</top>
```

Figure 6.1: Topic 25 from TREC-5

sources, including a large amount of news items and newswire material, some scientific abstracts, and some very long government reports (e.g. the document "pd9101-445" from peoples-daily collection contains more than 8,000 lines). The documents are tagged using SGML [3] and contain Chinese characters encoded in GB format, a common encoding standard used in China and Singapore. An example of documents from Xinhua news collection is shown in Figure 6.2.

6.2.2 Evaluation and the Relevance Judgements

When the output lists are returned to NIST after searching has been completed, they are merged for evaluation. That is, for each topic, the top-ranked documents

documents for these 10 duplicate document numbers in Xinhua collection [40].

[3]SGML (Standard Generalized Markup Language) is a standard for how to specify a document markup language or tag set. Such a specification is itself a document type definition (DTD). SGML is not in itself a document language, but a description of how to specify one. It is a metalanguage. The language that most of the Web browsers use, Hypertext Markup Language (HTML), is an example of an SGML-based language. SGML is based somewhat on earlier generalized markup languages developed at IBM, including General Markup Language (GML) and ISIL.

```
<DOC>
<DOCID> CB019021.BFJ (  355)     </DOCID>
<DOCNO> CB019021-BFJ-355-87 </DOCNO>
<DATE>     1995-09-21 12:46:44 (4) </DATE>
<TEXT>
<headline> 法首例不开胸心脏手术成功 </headline>
<p>
<s> 新华社巴黎９月２０日电（记者杨京德）法国首例不开胸心脏搭桥手术１９日在巴黎皮及埃医院
成功进行。 </s>
<s> 这种手术在欧洲尚属首次。 </s>
</p><p>
<s> 手术由法国医生伊拉吉·甘吉巴克主刀，病人是一名５０岁男子。 </s>
<s> 手术方法是在病人左胸和腋下开２个直径１厘米的小孔，分别插入微型摄像机镜头和手术器械，
主刀医生看着电视画面施行手术。 </s>
<s> 闭胸心脏搭桥手术的特点是病人心脏一直跳动，不需要血液体外循环。 </s>
</p><p>
<s> 整个手术持续了３个时，相当于开胸心脏搭桥手术的时间。 </s>
<s> 据甘吉巴克说，以后手术时间将随着熟练程度的提高而缩短。 </s>
<s> 闭胸心脏搭桥手术的另一个优点是病人痛苦小，住院时间缩短一半，康复时间也大幅度减少。
</s>
<s> 这名病人手术后几小时便能坐起。 </s>
</p><p>
<s> 法国每年大约有２．５万人需要接受心脏搭桥手术。 </s>
<s> 闭胸心脏搭桥手术的成功为这些病人带来了福音。 </s>
<s> （完） </s>
</p>
</TEXT>
</DOC>
```

Figure 6.2: An Example of Documents from Xinhua News Collection

from all participating teams are merged (pooled) into a single set, and given to the assessor for relevance evaluation. While it is clearly likely that some relevant documents are missed in this way (full evaluation of the 164,789 Chinese documents is clearly out of the question), it is at least plausible that most of the relevant documents in the Chinese TREC collections have been found. The results for each system are then subjected to a standard analysis program which generates a variety of performance measures of the recall-precision type. Essentially, these measures all address the ability of the system to retrieve early in its ranked output list those documents that are officially judged relevant to the topics. Thus in some sense all the measures are about the same thing, but they measure it in a variety of different ways. More information about relevant datasets for TREC-5 and TREC-6 are shown as follows in Table 6.1 and Table 6.2

Min. length	58 bytes
Max. length	22718 bytes
Average Number of Relevant Documents per Query	83.9 documents
Average Length	1399 bytes
Total Number of Relevant Documents at TREC-5	2182 documents

Table 6.1: Relevant Documents Information for TREC-5

Min. length	60 bytes
Max. length	294056 bytes
Average Number of Relevant Documents per Query	105.6 documents
Average Length	1987 bytes
Total Number of Relevant Documents at TREC-6	2958 documents

Table 6.2: Relevant Document Information for TREC-6

6.3 The Chinese Experimental Design

In this section, we are going to describe the Chinese experimental design for our probability-based text retrieval system. The system is called *C-Okapi*. The system was designed to retrieve Chinese documents. But it can also be adapted to English text retrieval. Our focus here is on Chinese text processing and retrieval.

6.3.1 Objectives

A typical Chinese information retrieval process is illustrated graphically in Figure 6.3. In this diagram we have a collection of documents and a query. The objective of information retrieval is to find from this collection a set of relevant documents based on a user's query. Since in most cases, the query and the documents are written in natural languages, the first thing we need to do is text extraction from the documents and from the query. The information extracted from the documents forms an index of the documents and the information extracted from the query forms query terms, sometimes called keywords. The retrieval process includes weighting these query terms according to the information from the documents, calculating document score for each document that contains one or more query terms, and then ranking the documents according to the document scores. Finally, the process presents the user with an ordered list of relevant documents.

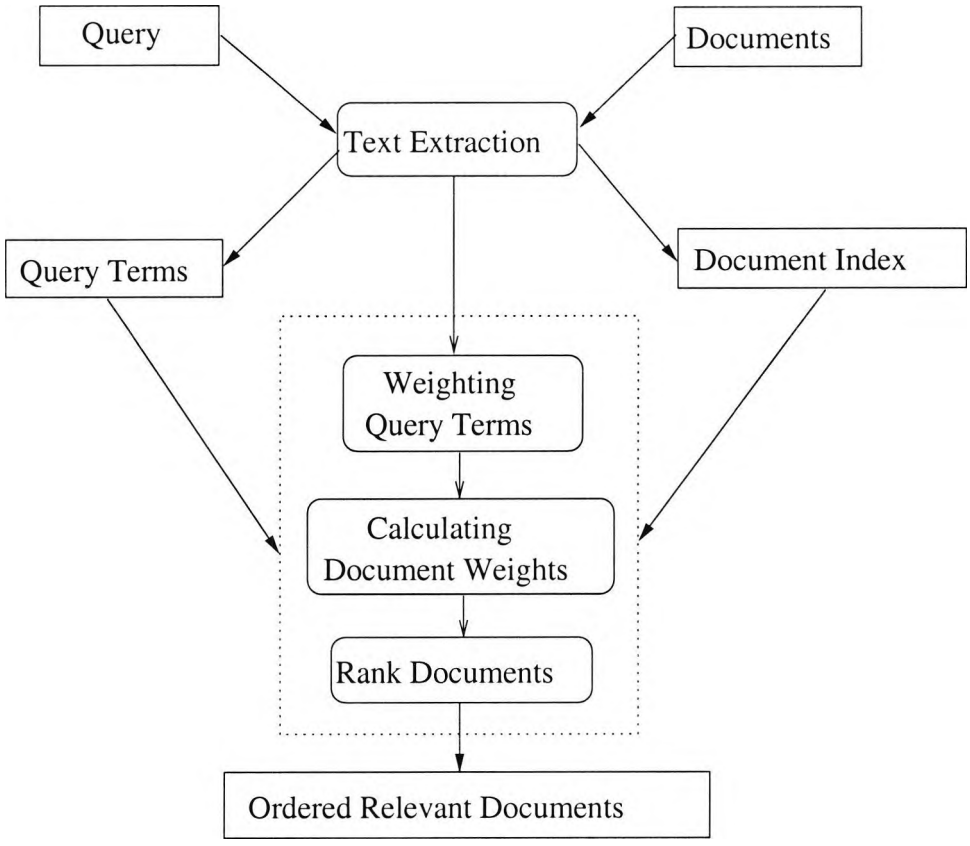


Figure 6.3: A Typical Information Retrieval Process

From this process we can observe that text extraction and query term weighting are two important steps in information retrieval. The objectives of our Chinese experimental design are related to these two important steps. First, we want to investigate the effectiveness of different text extraction methods for Chinese text retrieval. And secondly, we want to investigate the effectiveness of different term weighting methods. The text extraction methods we investigate include a word-based method and a character-based method. The term weighting methods we want to investigate include single unit weighting and compound unit weighting methods. To evaluate the different text extraction methods and term weighting methods, four major experiments are designed as follows for our Chinese text retrieval systems.

1. investigate the effect of word-based and character-based document processing on Chinese text retrieval.

2. investigate the effect of different weighting formulae and of varying their parameters.

3. Compare single unit weighting methods BM11, BM25 (see section 3.7) and BM26 (see section 4.2).

4. Compare different compound unit weighting formulae (see section 4.3).

6.3.2 Chinese Text Processing in Okapi

Chinese text processing in the *C-Okapi* retrieval system includes document processing and query processing. For documents, we use word-based segmentation and character-based segmentation independently, which means that we have two versions of Chinese Okapi. One uses word segmentation for processing documents; the other uses character-based segmentation. The word-based segmentation is conducted based on a dictionary containing 70,000 Chinese words. We use the longest matching method to segment words and these words are used to index documents. In character-based segmentation, single characters that appear in a document are used to index the document. Since word-based segmentation and character-based segmentation methods are used for indexing independently, we can compare these two different text extraction methods. For query processing, we concentrate on the word-based segmentation method. This word-based method uses both segmented terms and pairs of the adjacent segmented terms as potential retrieval keywords. Detailed descriptions will be given in Section 7.1.5.

6.3.3 Probabilistic Keyword Weighting

Our keyword weighting methods for evaluation are classified into single unit weighting and compound unit weighting. As we have discussed in Chapter 4, a single unit is a linguistic unit that is used to build the index of documents and a compound unit consists of two or more single units. For example, if documents are indexed by words, a word is a single unit and a phrase is a compound unit; if documents are indexed by characters, a character is a single unit, while both words and phrases are compound units. Several single unit weighting methods are used in our designed experiments. For single unit weighting experiments, we will concentrate on evaluating weighting functions BM25 (see section 3.7 for definition) and BM26 (defined in equation 4.3). For compound unit weighting experiments,

we will concentrate on evaluating different compound unit weighting functions designed in Chapter 4. All these experiments and evaluation will be conducted on TREC Chinese data sets and topics.

6.3.4 Comparison with Other Systems

To see how our system performs, we are going to compare our results with the automatic run results from other systems participating in TREC-5 and TREC-6 experiments. Since we only have 19[4] TREC-5 topics' evaluation results for other participating systems, our comparison on the TREC-5 queries is based on these 19 topics. The comparison on TREC-6 is based on all the TREC-6 topics. From the performance statistics, we can see how well the Chinese systems from City University work compared to other systems reported at TREC.

6.4 Evaluation Measures used at Chinese TREC

The evaluation measures used at TREC for the ad hoc and routing main tasks as well as the Chinese tracks are based on *precision* and *recall*. Precision measures a system's ability to retrieve only relevant documents:

$$\text{Precision} = \frac{\text{number of relevant documents retrieved}}{\text{total number of documents retrieved}}$$

Recall measures a system's ability to retrieve all relevant documents:

$$\text{Recall} = \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents in the collection}}$$

Various measures of retrieval effectiveness based on precision and recall are reported for each run in a tabular form.

1. Recall Level Precision Averages Table

   Table 6.3 is a sample "Recall Level Precision Averages" table.

   (a) Precision averages at 11 standard recall levels

[4]Only the results for 19 topics were made available by TREC-5.

Recall Level Precision Averages	
Recall	Precision
0.00	0.8626
0.10	0.6618
0.20	0.5592
0.30	0.4852
0.40	0.4117
0.50	0.3459
0.60	0.2694
0.70	0.1930
0.80	0.1326
0.90	0.0688
1.00	0.0112
Average precision over all relevant docs	
non-interpolated	0.3507

Table 6.3: Sample "Recall Level Precision Averages"

The *precision averages* at 11 standard recall levels (0.0, 0.1, 0.2, ..., 0.9, 1.0) are used to compare the retrieval performance of each run. Each of those values is computed by summing the precision values at the specified recall level and then dividing the sum by the number of topics. Let P_{λ_i} be the precision at recall level λ for topic i. The precision average over all topics at this recall level is calculated by:

$$\frac{\sum_{i=1}^{N} P_{\lambda_i}}{N}$$

where N is the total number of topics.

- Interpolating recall-precision

  Standard recall levels are used for easy reporting and plotting of retrieval results. One often needs to use interpolation to derive these values, as it may be impossible to accurately determine the precision value at a particular standard recall level. For example, if there are 35 relevant documents for a topic, to calculate the precision value at recall level 0.1 one needs to know the precision when $35 \times 0.1 = 3.5$ relevant documents are retrieved, and this value can only be estimated. At TREC, interpolation is used to estimate the precision

values at the standard recall levels. The interpolated precision at the ith recall level (R^i) is defined to be the maximum precision at all points p such that $R^{i-1} \leq p \leq R^i$.

For example, suppose there are only 3 relevant documents, and they are ranked at positions 2, 8 and 20. The exact recall points are 0.33, 0.67 and 1.0. The precisions at the true recall values are 0.5 at recall level 0.33, 0.25 at recall level 0.67, and 0.15 at recall level 1.0. Using the rule described in the previous paragraph, the interpolated precision at recall points 0.0, 0.1, 0.2 and 0.3 are 0.5, the precision at recall points 0.4, 0.5 and 0.6 is 0.25, and the precision at recall points 0.7, 0.8, 0.9 and 1.0 is 0.15.

(b) Average precision over all relevant documents, non-interpolated

In addition to the interpolated precisions, the non-interpolated average precision over all relevant documents is also shown. This value is obtained by averaging the precision value obtained after each relevant document is retrieved. With the example in the previous paragraph, the non-interpolated average precision over all relevant documents is $\frac{0.5+0.25+0.15}{3} = 0.3$.

2. Document Level Averages Table

Table 6.4 is a sample "Document Level Averages" table.

(a) Precision at 9 document cutoff values

The precision average after a given number of documents are retrieved is calculated by summing the precisions after that number of documents are retrieved and dividing by the number of topics. It mirrors how a user may measure system performance.

(b) R-Precision

This value is defined as the precision after R documents have been retrieved, where R is the number of relevant documents for the topic. The average R-Precision for a run is computed by averaging the R-Precision values for all the topics.

Document Level Averages	
	Precision
at 5 docs	0.6593
at 10 docs	0.6352
at 15 docs	0.6160
at 20 docs	0.5944
at 30 docs	0.5463
at 100 docs	0.3741
at 200 docs	0.2591
at 500 docs	0.1302
at 1000 docs	0.0780
R-Precision (precision after R docs retrieved, where R is the number of relevant documents)	
Exact	0.4323

Table 6.4: Sample "Document Level Averages"

Chapter 7

Empirical Evaluation on Chinese TREC Datasets

The experiments reported in this chapter were conducted as part of Okapi research group's participation at TREC-5 and TREC-6 Chinese track.[1] In the TREC-6 Chinese track, a new set of 26 topics were evaluated against the existing document collection used by Chinese TREC-5 experiments. Some of the results have been published in [5, 48, 49] for TREC-5 and [50, 51, 53] for TREC-6.

In this chapter we report our experimental results for the 28 TREC-5 queries and 26 TREC-6 queries. For TREC-5, a number of versions of our Chinese Okapi retrieval system are tested, which use different document processing methods and four different compound unit weighting methods. The single unit weighting method used in TREC-5 is BM25. We do not use BM26 for the TREC-5 queries because BM26 requires a parameter, the average relevant document length, to be calculated from previous queries and there were no previous queries for this Chinese document collection before TREC-5. For TREC-6, a number of new versions of our Chinese Okapi retrieval system are also tested, which use six new designed compound unit weighting methods. For these TREC-6 new queries we use BM26 as the single unit weighting method since we can set the average relevant document length parameter (rel_avdl) based on the TREC-5 results.

Since participating groups processed queries differently, it is difficult to conclude whether differences in retrieval performances among the participants were due to

[1]The author's role in the Okapi research group at TREC is to evaluate OKAPI on the Chinese collection of TREC and report the evaluation results for TREC-5 and TREC-6.

better query-processing methods or better retrieval ranking techniques. What we are planning to do in this chapter is concentrating on the comparisons between our own different runs for word-based and character-based document processing methods. All of these runs are using the same sets of query terms, but different indexing methods and different probabilistic weighting methods. At the end of this chapter, we will briefly compare our experimental results with some other Chinese systems.

7.1 Experiment Setup

7.1.1 Chinese Coding Schemes

The most popular coding schemes for HanZi (Chinese characters) are GB, HZ and Big5. GB (GuoBiao) is the coding standard in Mainland China, Singapore and where simplified Chinese (JianTi) is used. HZ is an enhanced version of GB, designed/specified by some experts over the Internet, especially, Dr. Fung-Fung Lee, to circumvent the limitations imposed by conventional Internet email and Usenet systems. The HZ coding scheme is used all over the world. Big5 is the de facto coding standard in Taiwan, Hong Kong and where traditional Chinese characters (FanTi) are used. All the document collection and topics used in our Chinese experiments were transformed from GB format (binary codes) into HZ format (printable ASCII codes).

7.1.2 The Topics and the Relevance Judgements

The 28 TREC-5 Chinese track topics (topics 1-28 in Appendix A) and the 26 TREC-6 Chinese track topics (topics 29-54 in Appendix A) were used. The format of the topics, as well as an example, have been described in Chapter 6. The relevance judgements were provided by the track organizers and were obtained using the pooling method described in Chapter 6. For each topic, the top 100 documents from each of the 26 submitted runs were included in a pool shown to the human assessors who read and determined the relevance of each document.

7.1.3 The Document Collection

The TREC Chinese track document collection was used in our experiments. The document collection used in TREC-6 Chinese track was identical to the one used in TREC-5. As described in Chapter 6, this compressed collection, if encoded in HZ, is about 175MB [2] in size. It consists of 139,801 articles selected from the People's Daily newspaper and 24,988 articles selected from the Xinhua newswire. All the original articles were tagged using SGML. Chinese characters inside these articles were encoded using the GB (GuoBiao) coding scheme, which is the coding standard in mainland China and Singapore.

7.1.4 Index Construction

With the GB coding scheme, two bytes are used to encode each Chinese character in binary code. This presented a problem as the original implementation of the Chinese version Okapi system is designed to index printable ASCII character strings (HZ code). Thus, a conversion utility written in C was used to first convert each two byte Chinese character encoded in GB into printable ASCII format encoded in HZ. The conversion utility read each Chinese character and output its hexadecimal value prefixed by "xx". For example, the word "英国" [3] (United Kingdom) was converted to "xx5322 xx397A" [4] by setting the highest 7th bit in each byte of its internal code to "0". Each Chinese character became a "word" after the conversion process. Any English words (including SGML tags) were left unchanged during the process.

The size of the uncompressed document collection tripled to about 500MB after conversion. The resulting documents were then indexed in the normal manner. In our TREC-5 and TREC-6 experiments for Chinese text retrieval, we use both word-based and character-based methods to process documents for indexing purposes. For the word-based method, we used the longest match segmentation algorithm to segment Chinese texts. The reason we use the longest match is that, from our previous experiments [43], the longest match produces the best results among

[2]The size of original Chinese TREC collection, which is encoded in GB, was about 158MB
[3]The internal code for "英国" is "xxD3A2 xxB9FA"
[4]the corresponding ASCII character is: S"9z

other matching methods in terms of average precision of retrieval. This was also confirmed by other research conducted by Chen [19]. By applying this algorithm to the Chinese collection with which we conduct experiments, approximately 43.6 million tokens were identified. These segmented tokens are used to index the documents in the collection for retrieval purposes. For the character-based method, single character-based segmentation is a purely mechanical procedure that segments Chinese texts into single characters. A huge inverted file is generated to index each Chinese character inside documents. The size of the character-based index built for Chinese TREC collection was about one gigabyte, which was about twice the size of the document collection. The size doubled since Chinese version Okapi stored positional information about each term's occurrences. More detailed information is showed in Table 7.1

	Method	Size
word_index	word approach	1,691,575 bytes
word_invert	word approach	678,257,616 bytes
index	character approach	139,734 bytes
invert	character approach	1,077,393,536 bytes

Table 7.1: Size of Index Files for Word and Character-Based Approaches

7.1.5 Query Formulation

Both character-based and word-based segmentations for query processing can be used to process queries. The character-based method uses characters, character pairs and multi-character adjacencies as retrieval keywords. Character pairs and multi-character adjacencies are similar to the bigrams and n-grams investigated by some other researchers [21, 139]. The word-based method uses similar techniques that allow phrases to contribute to the matching.

As with the Chinese track at TREC-5 and TREC-6, submitted results are classified into two categories: those that use manually constructed queries and those that use automatically constructed queries. In this chapter we report the experimental results for a word-based automatic query processing method[5]. This

[5]We have done some experiments that used a character-based method for query processing

word-based method uses segmented terms (regarded as words) and pairs of the adjacent segmented terms (regarded as potential phrases) as retrieval keywords. After words and phrases are extracted from the query text, they are weighted by using an approximation to inverse collection frequency (ICF) (Sparck Jones 1979) as follows:

$$w_{qt} = log\frac{N - n + 0.5}{n + 0.5},$$

where N is the number of indexed documents in the collection and n is the number of documents containing a specific term. All these segmented terms and potential phrases are then ranked by values of their weights multiplied by within-query frequencies. The top 19 terms[6] are selected as keywords for searching the word index and for searching the character index.

```
<top>
<num> Number: CH54

<C-title> 中国关于美国政府向台湾出售F－16战斗机的反应

<E-title> China's Reaction to U.S.  Sale of F-16 Fighters to Taiwan

<C-desc> Description:
中国，美国，台湾，F－16战斗机，出售

<E-desc> Description:
China, U.S., Taiwan, F-16 fighter, sale

<C-narr> Narrative:
相关文件应提到中美“八·一七”联合公报中对美国向台湾出售武器之决定，以及为何中央认为布什
总统决定售予台湾F－16战斗机是违反中美“八·一七”联合公报之精神并损害中美关系.

<E-desc> Description:
A relevant document should discuss the resolution concerning U.S. weapon sales to Taiwan
in the Sino-American "8-17" Joint Communique and why the Chinese consider President Bush
's decision to sell F-16 fighters to Taiwan to be in violation of the spirit of the  Sin
o-American "8-17" Joint Communique and to be damaging to Sino-American relations.

</top>
```

Figure 7.1: Topic 54 from TREC-6

with character segmentation for indexes, but the results were far worse than using word-based query processing.

[6]We chose the number of 19 because it gives the best result among the three numbers we tried in our experiments. There could be another number that gives better results than 19. This is not a perfect way to do it. But it does not affect our results.

An example of the above automatic query processing method is given as follows. First, initial topic processing deletes terms such as "document", "describe", "relevant" and "cite" from any description and narrative summary field. Then, the remaining part is processed in the same way as for segmenting documents. Third, adjacent pairs of segmented terms from the topic statement are produced. Topic 54 is shown in Figure 7.1. After deleting some irrelevant and meaningless terms[7], an ordered list of terms from topic 54 is presented in Table 7.2, which contains 28 segmented words and 17 generated phrases. The 17 generated phrases are "16战斗机", "台湾出售", "一七", "台湾F", "F16", "售予", "决定售", "损害中美", "违反中美", "总统决定", "联合公报中", "出售F", "出售武器", "什总统", "中美关系", "布什" and "美国政府". The 28 segmented words are "战斗机", "F", "台湾", "出售", "中美", "联合公报", "16", "美国", "予", "决定", "反应", "损害", "售", "违反", "武器", "什", "七", "八", "布", "总统", "精神", "中国", "关系", "中央", "政府", "认为", "中" and "一". Since the term for President Bush's Chinese name "布什" is not stored in our segmentation dictionary, it is segmented into two terms "布" and "什". An adjacent pair of these two segmented terms "布什" is produced as a potential phrase for retrieval. The top 19 terms, which include 12 phrases and 7 words, are selected from this ordered list for use as the retrieved keywords of topic 54. One of the 7 selected words is "16" which is not stored inside the segmentation dictionary and is selected according to a simple rule[8] encoded in the program.

7.1.6 Weighting Functions used in the Experiments

In our experiments, documents are indexed by single units (words or characters) and a query keyword could be either a single unit (words) or a compound unit (words or phrases). Whether a document matches with a compound unit is determined at search time using position information in the index file. Compound units (phrases or words) contain more information than single units (words or characters). It is reasonable to consider that search for compound units during retrieval

[7]Such as "相关"(relevant), "文件"(document), "提到"(cite), "是"(is) and "之"(of) etc.

[8]This simple rule is: if the current segmented unit is a digit (either Arabic or Chinese), the next unit is checked to see if it is also a digit. This process is repeated until a non-digit unit is met. The sequence of the digits is selected as a query term. The same rule is also used for word-based document processing.

ranking	terms	weight	freq	*freq * weight*	ranking	terms	weight	freq	*freq * weight*
1	1 6战斗机	766	3	2298	24	予	538	1	538
2	战斗机	603	3	1809	25	布什	518	1	518
3	F	548	3	1644	26	美国政府	518	1	518
4	台湾出售	736	2	1472	27	决定	242	2	484
5	台湾	361	4	1444	28	反应	468	1	468
6	出售	472	3	1416	29	损害	462	1	462
7	一七	707	2	1414	30	售	462	1	462
8	中美	471	3	1413	31	违反	447	1	447
9	联合公报	594	2	1188	32	武器	436	1	436
10	台湾F	1160	1	1160	33	什	424	1	424
11	F 1 6	1160	1	1160	34	七	171	2	342
12	售予	1109	1	1109	35	八	169	2	338
13	决定售	1075	1	1075	36	布	305	1	305
14	损害中美	894	1	894	37	总统	269	1	269
15	违反中美	894	1	894	38	精神	268	1	268
16	总统决定	877	1	877	39	中国	77	3	231
17	1 6	291	3	873	40	关系	221	1	221
18	联合公报中	859	1	859	41	中央	208	1	208
19	出售F	831	1	831	42	政府	172	1	172
20	出售武器	759	1	759	43	认为	141	1	141
21	美国	220	3	660	44	中	36	1	36
22	什总统	633	1	633	45	一	13	2	26
23	中美关系	589	1	589					

Table 7.2: Sorted Terms for Topic 54

is one of the most powerful combination techniques that could improve retrieval performance. In our system, query keywords may include both compound terms and all or some of their constituent elements, e.g., a word pair and both or one of the member single words, depending on whether all or only some of constituent elements are selected during query processing. Therefore, both documents containing the compound unit and documents containing any of its member terms or containing all the member terms but not in the phrasal relationship could match with the query. Intuitively, preference should be given to matches on the compound unit since compound units contain more information than their member single units. This preference should be reflected in designing weighting functions for compound units, i.e., it is reasonable to assume that, for a compound unit consisting of two single terms $t_1 t_2$,

$$w(t_1), w(t_2) < w(t_1 \wedge t_2) = w(t_1) + w(t_2) < w(t_1 \ adj \ t_2), \tag{7.1}$$

where $\wedge$ is the *and* operator and *adj* is the adjacency operator. The equation in the middle represents the usual scoring method for the $\wedge$ (and) operator: the score assigned to a document is the sum of the weights of the matching terms.

The assumption is that two adjacent units carry a higher score than two separate terms.

The problem of devising such a method consistent with the probabilistic model has not generally been tackled in text retrieval in English. But for text retrieval in Chinese, the problem is likely to be more serious than it is in English. This would be so in a word-based system, since there are likely to be considerable differences between Chinese speakers as to whether a given combination of characters is considered to be a single word or a phrase. But it is even more serious in a character-based system, where one would want a match on a complete word or phrase to carry a higher score than matches on the component characters.

Suppose that we have a sequence of j adjacent single units t_1, t_2, ..., t_j (characters or words) constituting a larger compound unit $t_1t_2...t_j$ (word or phrase) and that each single unit and the compound unit are included in the selected list of query keywords. Each unit (large or small) has a "natural" weight, given by a single unit weighting formula (such as BM25 or BM26). Let w_{t_i} be the natural weight for term t_i $(i = 1, ..., j)$ and $w_{t_1t_2...t_j}$ be the natural weight for the whole unit $t_1t_2...t_j$. If both the compound unit and its constituent single units are given weights in the usual fashion, we have

$$\begin{aligned} w(t_1 \wedge t_2 \wedge \cdots \wedge t_j) &= \sum_{i=1}^{j} w_{t_i} \\ w(t_1 \ adj \ t_2 \ adj \cdots adj \ t_j) &= w_{t_1t_2...t_j} + \sum_{i=1}^{j} w_{t_i} \end{aligned}$$

The weight for $w(t_1 \ adj \ t_2 \ adj \cdots adj \ t_j)$ contains both $w_{t_1t_2...t_j}$ and $\sum_{i=1}^{j} w_{t_i}$ because $t_1 \ adj \ t_2 \ adj \cdots adj \ t_j$ implies $t_1 \wedge t_2 \wedge \cdots \wedge t_j$. This natural assignment of weights satisfies the above assumption expressed in (7.1). Here, $\sum_{i=1}^{j} w_{t_i}$ can be considered as the first part "boost weight". Furthermore, for consistency we could also reduce the scores of those documents which contain the component single units but not the compound unit, e.g., by giving a small negative weight to the logical conjunction of the component units (i.e., reducing $w(t_1 \wedge t_2 \wedge \cdots \wedge t_j)$). However, design of this negative weight with the restriction to satisfy the left inequality $(w(t_1), w(t_2) < w(t_1 \wedge t_2))$ in the assumption in (7.1) is not an easy task. To avoid this difficulty, we can add an extra boost weight to $w(t_1 \ adj \ t_2 \ adj \cdots adj \ t_j)$ and let

$w(t_1 \wedge t_2 \wedge \cdots \wedge t_j)$ remain naturally designed. Based on this consideration, we suggest a number of weighting functions which satisfy the condition specified in (7.1). Table 7.3 gives six of such functions (denoted as $Weight_1$, $Weight_2$, ..., and $Weight_6$). In each set of the functions, the formula for $w(t_1\ adj\ t_2\ adj \cdots adj\ t_j)$ contains an extra boost weight, such as j^k in $Weight_2$. Among the six formulas, $Weight_1$ and $Weight_5$ are given the biggest extra boost, $Weight_3$ has no extra boost weight[9], and the others in between. All these formulas are used in our experiments, each of which is coupled with BM25 or BM26 for single unit weighting.

Weight methods	$w(t_1\ adj\ t_2\ adj \cdots adj\ t_j)$	$w(t_1 \wedge t_2 \wedge \cdots \wedge t_j)$
$Weight_1$	$\sum_{i=1}^{j} w_{t_i} + j^k * w_{t_1 t_2 \ldots t_j} + \sum_{i=1}^{j} w_{t_i}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_2$	$\sum_{i=1}^{j} w_{t_i} + w_{t_1 t_2 \ldots t_j} + j^k$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_3$	$\sum_{i=1}^{j} w_{t_i} + w_{t_1 t_2 \ldots t_j}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_4$	$\sum_{i=1}^{j} w_{t_i} + w_{t_1 t_2 \ldots t_j} + log \frac{\#(t_1 \wedge t_2 \wedge \cdots \wedge t_j)}{\#(t_1\ adj\ t_2\ adj \cdots adj\ t_j)}$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_5$	$\sum_{i=1}^{j} w_{t_i} + w_{t_1 t_2 \ldots t_j} \times log_2 j$	$\sum_{i=1}^{j} w_{t_i}$
$Weight_6$	$\sum_{i=1}^{j} w_{t_i} + w_{t_1 t_2 \ldots t_j} + \frac{\sum_{i=1}^{j} w_{t_i}}{d}$	$\sum_{i=1}^{j} w_{t_i}$
where $\#(t)$ indicates the number of documents containing the term t and k ($k \in [0, 2]$) and d are tuning constants.		

Table 7.3: Compound Unit Weighting Methods

7.2 Experimental Results and Performance Evaluation

Extensive experiments have been done on an SGI Challenging L machine to investigate: 1. the effect of probabilistic approach on Chinese text retrieval; 2. the difference between word approach and character approach; 3. the effect of different single unit weighting and compound unit weighting functions and of varying their parameters. All the results reported here are from Chinese ad-hoc experiments on the various TREC collections.

In our experiments for TREC-5 and TREC-6 queries, the relevance judgments for each query come from the human assessors of NIST. Statistical evaluation was done by means of the latest version of the TREC evaluation program [10]. The retrieval results are evaluated according to Average Precision, R Precision and Precision at 10 documents described in Chapter 6.

[9]That is $Weight_3$ only has the first part boost weight (see Table 4.3 and Table 4.4 for details)

7.2.1 Experimental Results for TREC-5 Queries

In this section we report our test results for the 28 TREC-5 queries. A number of versions of our Chinese Okapi retrieval system are tested, which use different document processing methods and different compound unit weighting methods. First we will briefly present our evaluation results of the official TREC-5 runs. Then we will present the full experimental results for the compound unit weighting methods defined in Table 7.3 and the single unit weighting method BM26.

Two Chinese TREC runs from City are submitted for evaluation at the beginning, which are city96c1 for the word approach and city96c2 for the character approach. These two runs were actually made with the BM11 function, which are equivalent to runs on the BM25 functions with b=1 by choosing k_2 appropriately. For these two submitted runs, the values for k_1, k_2 and k_3 in the BM11 function were set to 2.0, 1.5 and 5.0 respectively. The reason why the initial TREC runs were using the BM11 function is that we thought it could produce the best results. So we only implemented the BM11 weight function in our retrieval system at that time. After we obtained the evaluation results, we found that some documents in TREC collection were very short which merely contained titles of other relevant documents. From the evaluation results we noticed that the human assessors usually do not consider these very short documents as relevant. However, by using the BM11 function with k_2=1.5, these very short documents were ranked in front of the relevant documents of which they serve as the titles. These two runs, "city96c1" and "city96c2", made with the BM11 function are shown in Table 7.4 and Table 7.5. The value for parameter k in Table 7.3 for $Weight_1$ was set to -0.5 for these two runs and the comparison in Table 7.5 is made in terms of average precision. The results of both runs were based on the whole 28 topics and were ranked median among the groups participating in the TREC-5 Chinese task.

Run	Average Precision	Indexing	Compound	Single
*city*96c1	0.3256	word	$Weight_1$	BM11
*city*96c2	0.3336	character	$Weight_1$	BM11

Table 7.4: Official TREC-5 Chinese Results

In the above official Chinese TREC experiments, we chose two retrieval results

Run	Best	> median	= median	< median
*city*96*c*1	4	9	3	12
*city*96*c*2	0	11	7	10

Table 7.5: Comparative Chinese Results

which use $Weight_1$ as compound weighting method for evaluation. In our new experiments we also evaluate other compound unit weighting functions. In addition, to improve the results and to evaluate the BM25 function, we implemented the BM25 formula. Table 7.6 illustrates the TREC-5 results for BM25 and compound unit weighting methods defined in Table 7.3 in terms of average precision, total number of relevant documents retrieved, R precision and precision at 10 docs. Average precision, R precision and precision at 10 docs are the average numbers over the 28 TREC-5 queries; and total number of relevant documents is the summation over the 28 queries of the number of relevant documents in the first 1000 retrieved documents for each query. All the numbers were calculated by the TREC evaluation program. The four runs, T5w1.BM25, T5w2.BM25, T5w3.BM25 and T5w4.BM25, use the word-based method and use $Weight_1$, $Weight_2$, $Weight_3$ and $Weight_4$ as the compound unit weighting formula respectively . The four runs, T5c1.BM25, T5c2.BM25, T5c3.BM25 and T5c4.BM25, use the character-based method and use $Weight_1$, $Weight_2$, $Weight_3$ and $Weight_4$ as the compound unit weighting formula respectively. The values of k_1, k_2, k_3 and b for the above runs are 2.0, 0, 5.0 and 0.75 respectively. For all the above results, the value of parameter k in $Weight_1$ in Table 7.3 is set to 0.

Run	Document Processing	Compound unit Weighting	Average Precision	Total Rel Retrieved	R Precision	Precision at 10 docs
T5w1.BM25	Word	$Weight_1$	0.3691	1995	0.3873	0.5500
T5w2.BM25	Word	$Weight_2$	0.3775	2003	0.3865	0.5607
T5w3.BM25	Word	$Weight_3$	0.3762	2002	0.3860	0.5607
T5w4.BM25	Word	$Weight_4$	0.3773	2005	0.3864	0.5643
T5c1.BM25	Character	$Weight_1$	0.3475	2004	0.3611	0.4607
T5c2.BM25	Character	$Weight_2$	0.4126	2056	0.4251	0.5607
T5c3.BM25	Character	$Weight_3$	0.3795	1986	0.3963	0.5429
T5c4.BM25	Character	$Weight_4$	0.3863	2011	0.4017	0.5429

Table 7.6: Results for the TREC-5 Queries

Table 7.7 and 7.8 present more detail evaluation for the four runs T5w3.kd0,

T5c3.kd0, T5w2.kd0 and T5c2.kd0. From these two tables, we can observe that, (1) if $Weight_3$ is used for compound unit weighting, the character approach has a slight advantage over the word approach; (2) if $Weight_2$ is used for compound unit weighting, the character approach has a significant advantage over the word approach; and (3) the use of $Weight_2$ for compound unit weighting leads to better average precision than using $Weight_3$.

Run	Average Precision	Indexing	Compound	Single
$T5w3.kd0$	0.3762	word	$Weight_3$	BM25
$T5c3.kd0$	0.3795	character	$Weight_3$	BM25
$T5w2.kd0$	0.3775	word	$Weight_2$	BM25
$T5c2.kd0$	0.4126	character	$Weight_2$	BM25

Table 7.7: TREC-5 Chinese Ad-hoc Results

Run	Average Precision
$T5w3.kd0$	0.3762
$T5c3.kd0$	0.3795 (+0.9%)
$T5w2.kd0$	0.3775 (+0.3%)
$T5c2.kd0$	0.4126 (+9.68%)

Table 7.8: TREC-5 Chinese Ad-hoc Results Comparison

Since BM26 requires a parameter to be calculated from previous queries and there are no previous queries with evaluation results for TREC-5, we could not test BM26 formula. However, a default value such as 1200 can be set for the average relevant document length in BM26. The experimental results for using BM26 with word-based and character-based document processing are presented in Appendix D.

The results in Appendix D indicate that $Weight_2$ is the best compound unit weighting formula among all the tested formulae for both word-based and character-based Chinese retrieval and also that the character-based method is usually better than the word-based method with the tested compound unit weighting formulae $Weight_2$ and $Weight_4$[10]. In the TREC-5 experiments, the values of parameter k

[10]These two compound unit weighting formulae produce better results than all the other methods.

in Table 7.3 for $Weight_1$ and $Weight_2$ are set to 0 and 1 respectively. We are not going to give specific comparisons in Appendix D. Obviously these two big tables in Appendix D are too unwieldy to understand. Detailed comparisons and analysis about what we can learn from these comparisons will be given in Chapter 8.

7.2.2 Experimental Results for TREC-6 Queries

We also ran different versions of *C-Okapi* on the 26 TREC-6 Chinese queries. For these queries we use BM26 as the single unit weighting method since we can set the average relevant document length parameter (*rel_avdl*) based on the TREC-5 results. 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 compound unit weighting methods we evaluate in our experiments are defined in Table 7.3. The value of k in the compound unit weighting function $Weight_2$ is set to be 1. Table 7.9 shows the results for using word-based document processing.

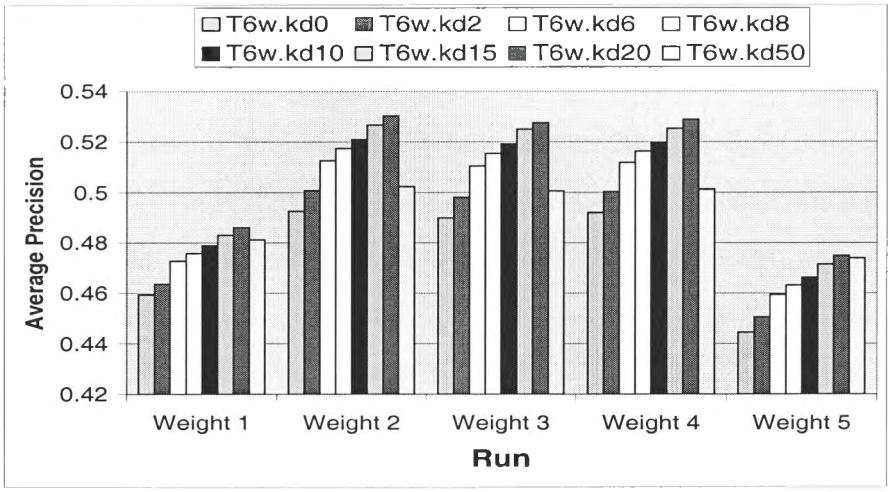


Figure 7.2: Comparison of Single Unit Weighting Functions Using Word Methods

Table 7.10 shows the results for using character-based document processing. Figure 7.2 illustrates in bar plots a comparison of single unit weighting functions

	Run	Weighting Method	k_d	Average Precision	Total Rel Retrieved	R Precision	Precision at 10 docs
1	T6w1.kd0	$Weight_1$	0	0.4594	2516	0.4740	0.7077
2	T6w2.kd0	$Weight_2$	0	0.4927	2546	0.5127	0.7423
3	T6w3.kd0	$Weight_3$	0	0.4900	2542	0.5106	0.7423
4	T6w4.kd0	$Weight_4$	0	0.4921	2548	0.5124	0.7462
5	T6w5.kd0	$Weight_5$	0	0.4446	2502	0.4635	0.6654
6	T6w1.kd2	$Weight_1$	2	0.4636	2528	0.4772	0.7192
7	T6w2.kd2	$Weight_2$	2	0.5008	2557	0.5197	0.7462
8	T6w3.kd2	$Weight_3$	2	0.4982	2551	0.5202	0.7423
9	T6w4.kd2	$Weight_4$	2	0.5004	2558	0.5200	0.7500
10	T6w5.kd2	$Weight_5$	2	0.4507	2513	0.4693	0.6731
11	T6w1.kd6	$Weight_1$	6	0.4727	2547	0.4883	0.7269
12	T6w2.kd6	$Weight_2$	6	0.5126	2577	0.5294	0.7500
13	T6w3.kd6	$Weight_3$	6	0.5104	2574	0.5278	0.7577
14	T6w4.kd6	$Weight_4$	6	0.5118	2580	0.5287	0.7538
15	T6w5.kd6	$Weight_5$	6	0.4595	2530	0.4795	0.6923
16	T6w1.kd8	$Weight_1$	8	0.4758	2517	0.4911	0.7423
17	T6w2.kd8	$Weight_2$	8	0.5174	2590	0.5288	0.7615
18	T6w3.kd8	$Weight_3$	8	0.5154	2588	0.5282	0.7615
19	T6w4.kd8	$Weight_4$	8	0.5162	2588	0.5277	0.7577
20	T6w5.kd8	$Weight_5$	8	0.4631	2532	0.4839	0.7000
21	T6w1.kd10	$Weight_1$	10	0.4789	2548	0.4923	0.7462
22	T6w2.kd10	$Weight_2$	10	0.5209	2593	0.5309	0.7692
23	T6w3.kd10	$Weight_3$	10	0.5192	2592	0.5296	0.7692
24	T6w4.kd10	$Weight_4$	10	0.5198	2595	0.5309	0.7731
25	T6w5.kd10	$Weight_5$	10	0.4662	2533	0.4868	0.7000
26	T6w1.kd15	$Weight_1$	15	0.4831	2548	0.4939	0.7462
27	T6w2.kd15	$Weight_2$	15	0.5267	2589	0.5331	0.7962
28	T6w3.kd15	$Weight_3$	15	0.5251	2589	0.5328	0.7962
29	T6w4.kd15	$Weight_4$	15	0.5253	2583	0.5308	0.8000
30	T6w5.kd15	$Weight_5$	15	0.4714	2533	0.4872	0.7192
31	T6w1.kd20	$Weight_1$	20	0.4862	2534	0.4924	0.7500
32	T6w2.kd20	$Weight_2$	20	0.5303	2575	0.5342	0.8038
33	T6w3.kd20	$Weight_3$	20	0.5276	2575	0.5320	0.8077
34	T6w4.kd20	$Weight_4$	20	0.5289	2575	0.5321	0.8038
35	T6w5.kd20	$Weight_5$	20	0.4748	2522	0.4905	0.7385
36	T6w1.kd50	$Weight_1$	50	0.4812	2429	0.4928	0.7769
37	T6w2.kd50	$Weight_2$	50	0.5024	2415	0.5162	0.8000
38	T6w3.kd50	$Weight_3$	50	0.5006	2408	0.5147	0.8038
39	T6w4.kd50	$Weight_4$	50	0.5013	2410	0.5133	0.8000
40	T6w5.kd50	$Weight_5$	50	0.4739	2427	0.4939	0.7692
41	T6w6.kd15.d2	$Weight_6(d = 2)$	15	0.5053	2566	0.5111	0.7769
43	T6w6.kd15.d10	$Weight_6(d = 10)$	15	0.5212	2584	0.5314	0.7808
44	T6w6.kd15.d20	$Weight_6(d = 20)$	15	0.5233	2586	0.5311	0.7962
45	T6w6.kd15.d50	$Weight_6(d = 50)$	15	0.5241	2587	0.5319	0.7962
46	T6w6.kd15.d100	$Weight_6(d = 100)$	15	0.5247	2588	0.5336	0.7962

Table 7.9: Results for TREC-6 Queries with the Word-based Approach

(BM25 and BM26 with different values of parameter k_d) when word-based document processing is used. A comparison of compound unit weighting methods for word-based document processing is illustrated in Figure 7.3, in which the last group of bars represents the results for the $Weight_6$ method with different values for parameter d ($d = 2, 10, 20, 50$, and 100 respectively). Figure 7.4 and Figure 7.5 illustrate the same comparisons for character-based document processing. Figure 7.6 shows in bar plots the comparison of word-based and character-based methods in terms of average precision over the 45 runs described in Tables 7.9 and 7.10. In the figure, darker bars represent the results for the character-based method and lighter bars for the word-based method.

In terms of single unit weighting, both the result from the word-based method (Figure 7.2) and the result from the character-based method (Figure 7.4) indicate that BM26 with $k_d > 0$ is better than BM25 (BM26 with $k_d = 0$). In terms of parameter setting for BM26, the results show that the best performance is

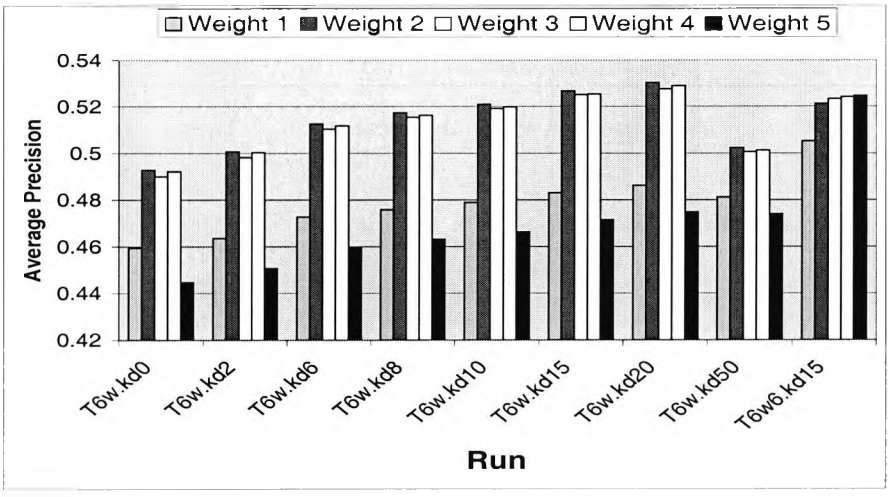


Figure 7.3: Comparison of Compound Unit Weighting Functions Using Word Methods

achieved when k_d is set to be 20 for word-based methods and when k_d is 20 or 15 for character-based methods. In terms of compound unit weighting, the results (see Figure 7.3 and Figure 7.5) confirm that $Weight_2$ is the best compound unit weighting formula for both word and character-based methods. This is more obvious in the results for character-based methods. We can say that character-based methods are more sensitive to the compound weighting functions. In addition, the results (see Figure 7.6) consistently show that the character-based method is better than the word-based method. Table 7.11 shows the improvements of BM26 over BM25 and character-based over word-based methods, in which $Weight_2$ is used for compound unit weighting.

7.2.3 Result Analyses and Discussions

Our experimental results indicate that the character-based method leads to better retrieval performance. Using the character approach for indexing, we obtained a better average precision than that of the word approach. One of the reasons possibly lies in the Chinese language itself: in Chinese, single characters (ideographs) may constitute a reasonably good representation of a text. This result clearly shows that the Chinese characters play an important role in making contribution to the good performance of Chinese retrieval systems. Similar results were also

	Run	Weighting Method	k_d	Average Precision	Total Rel Retrieved	R Precision	Precision at 10 docs
1	T6c1.kd0	$Weight_1$	0	0.4789	2550	0.4767	0.7077
2	T6c2.kd0	$Weight_2$	0	0.5341	2637	0.5416	0.7308
3	T6c3.kd0	$Weight_3$	0	0.4967	2537	0.5175	0.7115
4	T6c4.kd0	$Weight_4$	0	0.5113	2558	0.5244	0.7115
5	T6c5.kd0	$Weight_5$	0	0.4627	2493	0.4666	0.6692
6	T6c1.kd2	$Weight_1$	2	0.4833	2562	0.4813	0.7115
7	T6c2.kd2	$Weight_2$	2	0.5434	2653	0.5452	0.7462
8	T6c3.kd2	$Weight_3$	2	0.5096	2565	0.5279	0.7308
9	T6c4.kd2	$Weight_4$	2	0.5227	2568	0.5356	0.7308
10	T6c5.kd2	$Weight_5$	2	0.4693	2504	0.4735	0.6846
11	T6c1.kd6	$Weight_1$	6	0.4891	2566	0.4875	0.7231
12	T6c2.kd6	$Weight_2$	6	0.5551	2655	0.5505	0.7654
13	T6c3.kd6	$Weight_3$	6	0.5306	2575	0.5412	0.7615
14	T6c4.kd6	$Weight_4$	6	0.5389	2581	0.5504	0.7577
15	T6c5.kd6	$Weight_5$	6	0.4825	2516	0.4884	0.7000
16	T6c1.kd8	$Weight_1$	8	0.4921	2573	0.4864	0.7308
17	T6c2.kd8	$Weight_2$	8	0.5582	2647	0.5518	0.7769
18	T6c3.kd8	$Weight_3$	8	0.5348	2569	0.5404	0.7731
19	T6c4.kd8	$Weight_4$	8	0.5439	2577	0.5488	0.7615
20	T6c5.kd8	$Weight_5$	8	0.4880	2520	0.4965	0.7154
21	T6c1.kd10	$Weight_1$	10	0.4942	2574	0.4894	0.7308
22	T6c2.kd10	$Weight_2$	10	0.5603	2647	0.5544	0.7885
23	T6c3.kd10	$Weight_3$	10	0.5383	2560	0.5422	0.7731
24	T6c4.kd10	$Weight_4$	10	0.5476	2573	0.5526	0.7692
25	T6c5.kd10	$Weight_5$	10	0.4925	2515	0.5001	0.7231
26	T6c1.kd15	$Weight_1$	15	0.4981	2575	0.4956	0.7500
27	T6c2.kd15	$Weight_2$	15	0.5599	2621	0.5613	0.7962
28	T6c3.kd15	$Weight_3$	15	0.5417	2546	0.5494	0.7923
29	T6c4.kd15	$Weight_4$	15	0.5494	2566	0.5548	0.7885
30	T6c5.kd15	$Weight_5$	15	0.5007	2511	0.5106	0.7500
31	T6c1.kd20	$Weight_1$	20	0.5005	2570	0.4960	0.7615
32	T6c2.kd20	$Weight_2$	20	0.5545	2590	0.5540	0.7846
33	T6c3.kd20	$Weight_3$	20	0.5374	2518	0.5466	0.7962
34	T6c4.kd20	$Weight_4$	20	0.5450	2539	0.5488	0.8038
35	T6c5.kd20	$Weight_5$	20	0.5032	2491	0.5119	0.7538
36	T6c1.kd50	$Weight_1$	50	0.4892	2491	0.4985	0.7654
37	T6c2.kd50	$Weight_2$	50	0.4893	2325	0.5119	0.8038
38	T6c3.kd50	$Weight_3$	50	0.4677	2225	0.4995	0.7962
39	T6c4.kd50	$Weight_4$	50	0.4779	2256	0.5036	0.7923
40	T6c5.kd50	$Weight_5$	50	0.4805	2349	0.5037	0.7462
41	T6c6.kd15.d2	$Weight_6(d = 2)$	15	0.5144	2618	0.5102	0.7538
43	T6c6.kd15.d10	$Weight_6(d = 10)$	15	0.5421	2626	0.5387	0.7885
44	T6c6.kd15.d20	$Weight_6(d = 20)$	15	0.5471	2611	0.5460	0.7885
45	T6c6.kd15.d50	$Weight_6(d = 50)$	15	0.5484	2608	0.5474	0.7885
46	T6c6.kd15.d100	$Weight_6(d = 100)$	15	0.5488	2605	0.5493	0.7923

Table 7.10: Results for TREC-6 Queries with the Character-based Approach

obtained in [10] [60][73]. Another reason is in the use of compound unit weighting. Compound unit weighting functions work more effectively in character-based approaches than word-based approaches (see Figure 7.3 and Figure 7.5). This is because the number of single units that constitute a compound unit in character-based approaches is usually larger than that in word-based approaches, which results in more weights for the compound unit in character-based approaches.

Run	k_d	Indexing Method	Average Precision
T6w2.kd0 (BM25)	0	*word*	0.4927
T6c2.kd0 (BM25)	0	*character*	0.5341 (+8.40%)
T6w2.kd10 (BM26)	10	*word*	0.5209 (+5.72%)
T6c2.kd10 (BM26)	10	*character*	0.5603 (+13.72%)

Table 7.11: Results Comparison

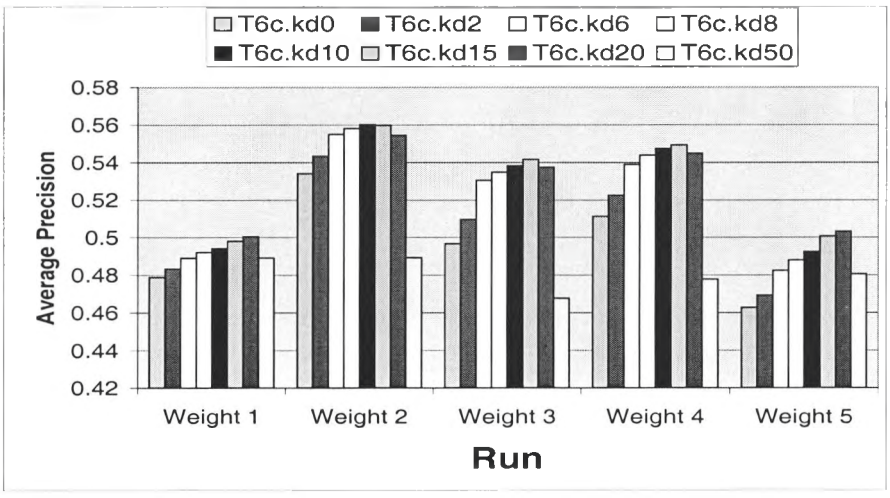


Figure 7.4: Comparison of Single Unit Weighting Functions Using Character Methods

While character-based approaches are more sensitive to compound unit weighting functions, word-based approaches seem to be more sensitive to single unit weighting functions (see Figures 7.2 and 7.4). One possible reason is that n in single unit weighting functions (BM25 and BM26) has smaller values in word-based approaches than in character-based approaches, which results in larger values in the left part of BM25 or BM26, which in turn leads to more influence of the variation of the value for k_d on the whole value of the function. Detailed analysis and discussions will be given in Chapter 8.

7.3 Comparisons with TREC participating systems

Among these submitted runs for TREC-5 and TREC-6 Chinese experiments, we can classify them into two categories. One uses automatically created queries for retrieval, while the other one uses manually modified queries. To see how our system performs, we only compare our Okapi Chinese results with the automatic run results from other systems participating in TREC-5 and TREC-6 experiments in this section. More analyses and discussions about this comparison will be given in Chapter 8.

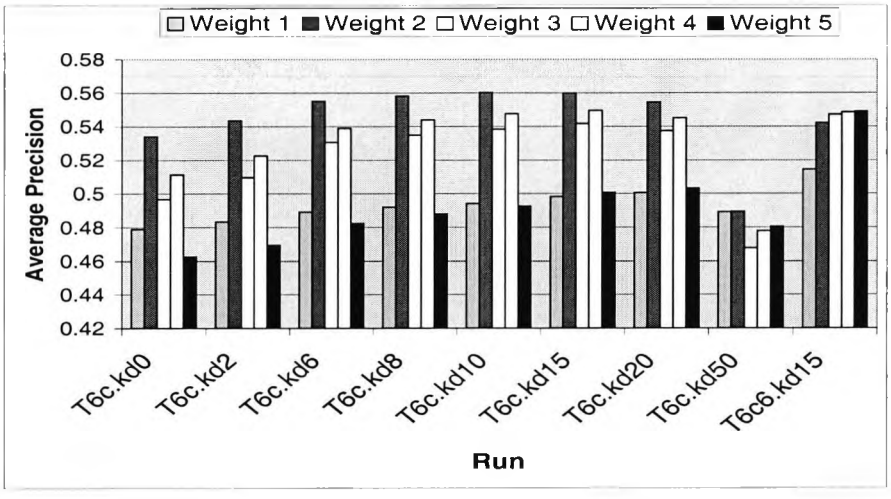


Figure 7.5: Comparison of Compound Unit Weighting Functions Using Character Methods

7.3.1 Comparison with TREC-5 participating systems

Nine different retrieval systems took part in TREC-5 Chinese ad-hoc experiments and 20 runs were submitted for evaluation. In total 15 runs were automatic and 5 runs were manually modified. Since we can only collect 19 TREC-5 queries' evaluation results of other participating systems, our comparison on the TREC-5 queries is based on these 19 queries[11]. All the 15 automatic runs' results for these 19 topics are presented in Table 7.12. In this table we compare two runs from our system with the runs from other systems. Figure 7.7 shows the precision-recall curves on TREC-5 experiments. In this figure we include the best automatic run from almost every participating institution in TREC-5. Those runs are compared with the run T5c2.BM25, which is one of our best runs at our TREC-5 experiments.

From these performance statistics, we can see that our results compare well with the best reported at TREC-5. In terms of average precision, the results at TREC-5 range from 0.027 to 0.38, with just two systems giving results better than 0.35 and our result from T5c2.BM25 is 0.3541.

[11]These 19 topics are: topic1-16 and topic21-23.

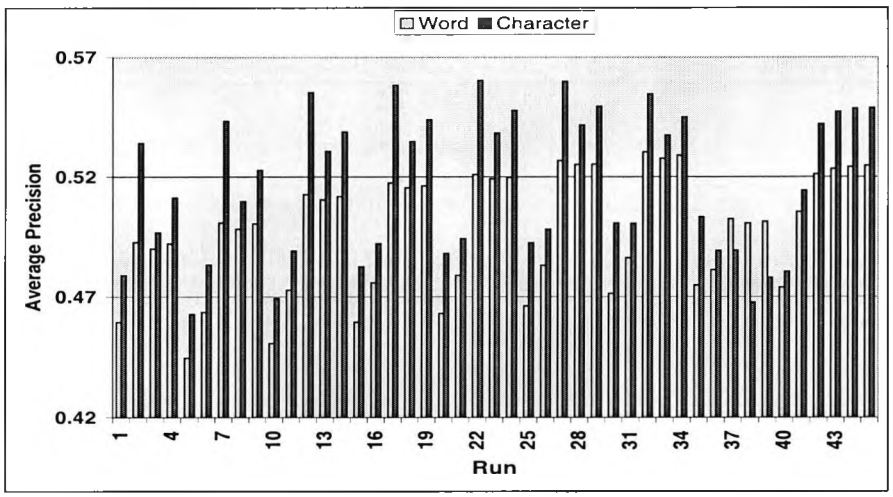


Figure 7.6: Comparison of Character and Word Methods

7.3.2 Comparison with TREC-6 participating systems

In total 12 groups participated in the TREC-6 Chinese ad-hoc experiment. Unlike TREC-5, the comparison on TREC-6 is based on all the TREC-6 queries. Table 7.13 shows the comparison on TREC-6. In this table two runs from our system are compared with the runs from other systems at TREC-6. Figure 7.8 shows the precision-recall curves on TREC-6 runs. In this figure we include all the best automatic runs from other Chinese systems at TREC-6. These best runs are selected to compare with one of our best runs $T6c2.kd10$ at TREC-6.

From these performance statistics, we can see that our results at TREC-6 are still pretty good even if the overall retrieval effectiveness was very high at TREC-6 and median performance was above 0.5 [12]. In terms of average precision, the average precision results at TREC-6 range from 0.34 to 0.62, with four systems giving results better than 0.55 and our result from T6c2.kd10 is 0.5603.

[12]It is not very clear why the results of other systems at TREC-6 are so high. We will analyse and discuss the possible reasons in chapter 8.

Run	Organization	Average Precision	Total Rel Retrieved	R Precision	Precision 100 docs
pircsCwc	*Queens College, CUNY*	0.3789	1313	0.3823	0.3253
pircsCw	*Queens College, CUNY*	0.3751	1297	0.3870	0.3211
Cor5c2ex	*Cornell University*	0.3598	1343	0.3829	0.3084
T5c1.BM25	*City University*	0.3541	1294	0.3662	0.2884
T5w1.BM25	*City University*	0.3456	1274	0.3510	0.2842
gmu96ca2	*George Mason University*	0.3274	1250	0.3571	0.2916
Cor5C1vt	*Cornell University*	0.3266	1286	0.3598	0.3026
BrklyCH1	*University of California, Berkeley*	0.3192	1246	0.3565	0.2853
gmu96ca1	*George Mason University*	0.2955	1202	0.3296	0.2595
CLCHNA	*CLARITECH Corporation*	0.2677	1182	0.2998	0.2653
iticn1	*Information Technology Institute*	0.1731	899	0.2289	0.1758
HIN300	*University of Massachusetts*	0.1519	542	0.2333	0.1805
HIN301	*University of Massachusetts*	0.1481	540	0.2275	0.1763
mds005	*Royal Melbourne Institute of Tech.*	0.0371	360	0.1045	0.0811
mds004	*Royal Melbourne Institute of Tech.*	0.0268	360	0.0867	0.0705

Table 7.12: Comparison with Other Retrieval Systems on TREC-5 Queries

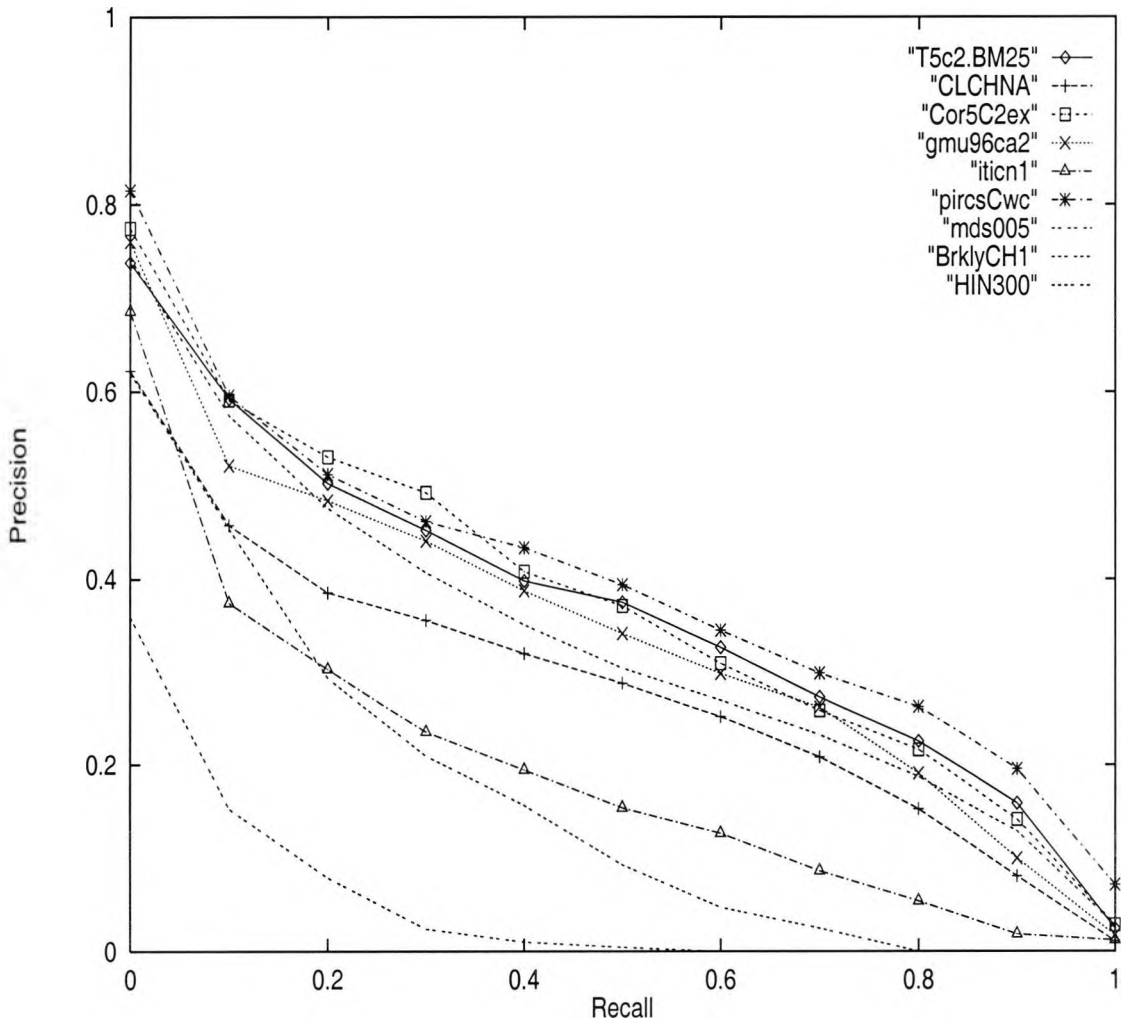


Figure 7.7: Precision-Recall Curves for Some Automatic Runs at TREC-5

Run	Organization	Average Precision	Total Rel Retrieved	R Precision	Precision 100 docs
pirc7Ca	*Queens College, CUNY*	0.6263	2795	0.5809	0.5542
ETHccA	*Swiss Federal Institute of Technology*	0.5733	2698	0.5598	0.5281
iss97CbD	*The Institute of System Science*	0.5646	2802	0.5515	0.5104
T6c2kd10	*City University*	0.5603	2647	0.5544	0.5208
T6c2kd15	*City University*	0.5599	2621	0.5613	0.5227
mds608	*Royal Melbourne Institute of Tech.*	0.5597	2665	0.5271	0.5165
BrklyCH4	*University of California, Berkeley*	0.5586	2573	0.5496	0.5427
Cor6CH2ns	*Cornell University*	0.5552	2763	0.5369	0.5185
Cor6CH1sc	*Cornell University*	0.5547	2765	0.5301	0.5162
CLARITcAS	*CLARITECH Corporation*	0.5494	2719	0.5357	0.4938
mds609	*Royal Melbourne Institute of Tech.*	0.5479	2665	0.5234	0.5131
mds607	*Royal Melbourne Institute of Tech.*	0.5436	2590	0.5236	0.5112
pirc7Cd	*Queens College, CUNY*	0.5423	2674	0.5175	0.5035
INQ4ch1	*University of Massachusetts*	0.5336	2662	0.5218	0.5096
BrklyCH3	*University of California, Berkeley*	0.5291	2551	0.5252	0.5296
INQ4ch2	*University of Massachusetts*	0.5223	2664	0.5137	0.4996
iss97CmD	*The Institute of System Science*	0.4903	2723	0.4941	0.4692
pirc7Ct	*Queens College, CUNY*	0.4755	2547	0.4630	0.4327
iss97CsD	*The Institute of System Science*	0.4709	2619	0.4689	0.4615
iticn1	*Information Technology Institute*	0.4541	2447	0.4745	0.4615
UdeMseg	*University of Montreal*	0.4524	2668	0.4748	0.4662
UdeMbi	*University of Montreal*	0.4467	2709	0.4655	0.4408
iticn2	*Information Technology Institute*	0.4145	2349	0.4452	0.4288
iticn3	*Information Technology Institute*	0.3427	2215	0.3881	0.3715

Table 7.13: Comparison with Other Retrieval Systems on TREC-6 Queries

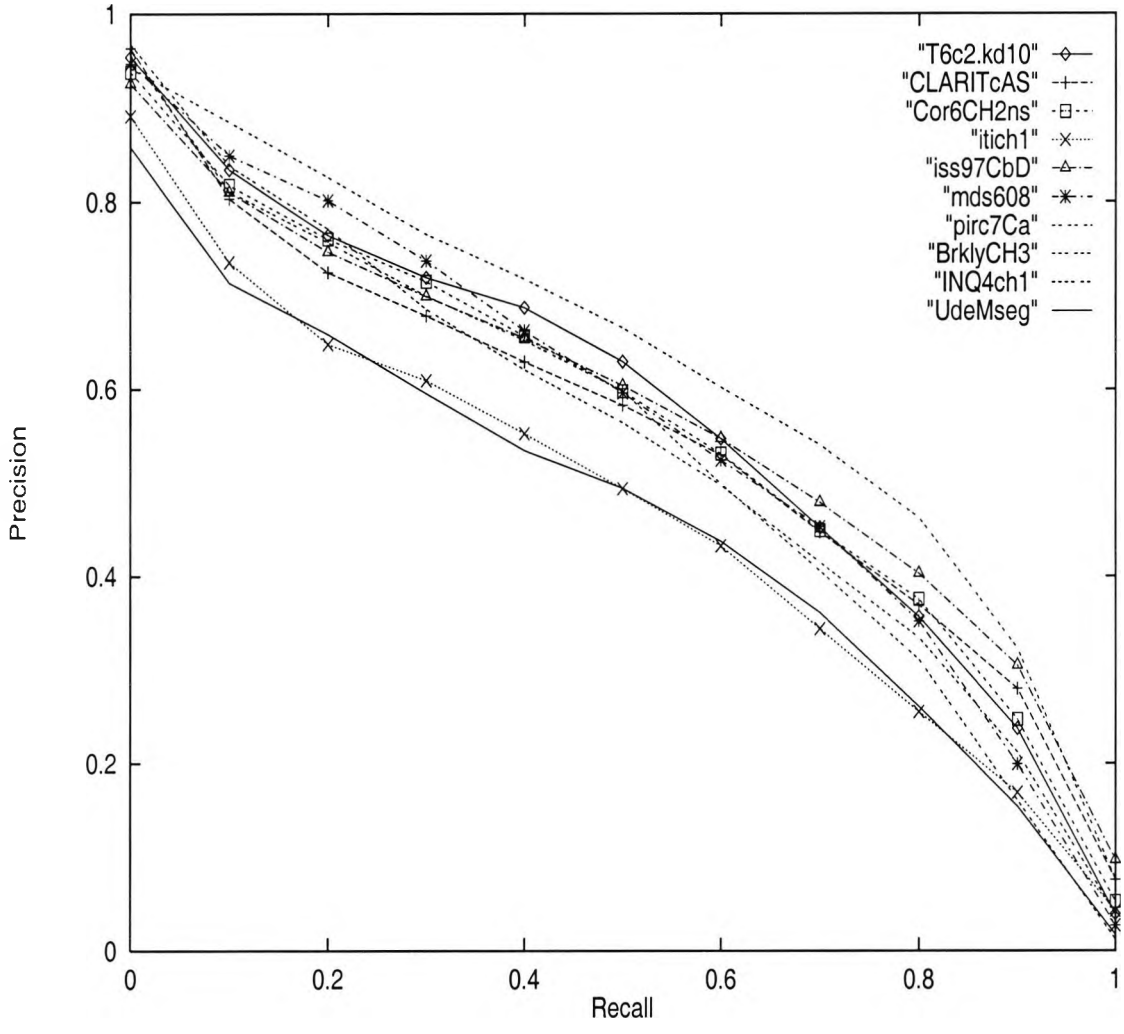


Figure 7.8: Precision-Recall Curves for Some Automatic Runs at TREC-6

Chapter 8

Analyses and Discussion

We have presented several empirical comparisons for Chinese text retrieval in Chapter 7. First, we compared a word-based segmentation method with a character-based segmentation method for document processing. Second, we compared six compound unit weighting methods. Third, we also compared three single unit weighting methods: BM11, BM25 and BM26. In this chapter, the above three issues will be analysed and discussed in detail. Finally, we will analyse and explain the comparison results of our Chinese Okapi retrieval system with other Chinese systems.

8.1 Word-based vs. Character-based Document Processing

8.1.1 Overview

The difference between Chinese IR and IR for most European languages lies in the fact that words are not separated in Chinese sentences. For example, the phrase "information retrieval system" is written as 信息检索系统. As we can see from this example, there is no space between each Chinese character. However, it is important to separate a sentence into smaller segments from the point of view of retrieval. Two types of segments may be used: N-grams or words.

An N-gram is a subsequent string of N Chinese characters. For example, the string of 信息检索系统 (information retrieval system) may be segmented into the

following unigrams (N=1) and bigrams (N=2)[1]:

N=1: 信 息 检 索 系 统

N=2: 信息 息检 检索 索系 系统

The word approach requires one to segment a Chinese sentence into words. This is not a trivial task because of the enormous amount of ambiguity. A sentence may often be segmented into several different sequences of legitimate words. For example, "上海滩" can be segmented into a single word "上海滩" or two words "上" and "海滩" or two words "上海" and "滩" [2]; "两成都市人" can be segmented into "两成 都市 人" which means "20 percent of municipal population" or "两成都 市 人" which means "two folks from Chengdu city"; "邓亚萍越战越勇" can be segmented into "邓亚萍 越 战 越 勇" or "邓亚萍 越战[3] 越 勇". From the above three examples, we can see that different segmentation results can lead to totally different meanings. The key problem for the word approach is to choose the correct segmentation from all the possible solutions. There are two basic segmentation approaches for Chinese: the approach based on a dictionary, and the approach based on statistics (see section 2.3.4 for more detailed discussions).

In the dictionary-based approach, one first finds all the legitimate words included in a sentence, then the longest matching algorithm may be applied to choose the sequence of words which covers the sentence with the longest words (or with the fewest words). A dictionary-based segmentation is usually augmented by a set of heuristic rules to recognize special sequences such as quantity-classifier sequences (e.g. one thousand and one).

As we know, there is no clear definition of words in Chinese. There are a number of long words/phrases that are composed of shorter words in many Chinese dictionaries. If a long word/phrase (such as "电脑网络") is encountered, the shorter words (such as "电脑" and "网络") contained in it are hidden, which causes the documents that contain only the shorter words to be missed. For example, if a document talks about 电脑网络 (computer network) and a query asks for "网络" (network), this document will not be retrieved. In order to avoid this problem, we

[1] It is possible to use longer N-grams for Chinese IR. However, it has been shown that bigrams are a good choice for Chinese IR [60].

[2] "上海滩" means the name of a film, "上 海滩" means "go to the beach" and "上海 滩" means "shanghai beach".

[3] "越战" means "Vietnam War".

can implement a segmentation algorithm which extracts all the possible compound words (composed of two characters or more) from a given character string. So for the sequence "电脑网络", three words will be extracted: "电脑网络", "电脑" (computer) and "网络". That is all the compound words included in a long word are also extracted.

The above approach may be further extended by also extracting all the single characters. So, for the string "电脑网络", we have the following segments extracted: "电", "脑", "网", "络". This indexing method is the same as the character indexing approach proposed and discussed in this thesis.

On the other hand, a statistical approach relies on statistical data to determine possible words and to select the best word sequence. Statistical data are usually obtained from a set of manually segmented training texts. According to the frequency of occurrences and co-occurrences, one may determine how probable a string (possibly within some context) may be a word.

8.1.2 Statistics for Each Topic

In order to compare the word-based method with the character-based method for each TREC-5 and TREC-6 topic, we choose the best runs from these two different document processing methods in terms of average precision and total number of relevant documents retrieved. In terms of average precision, the best runs for TREC-5 and TREC-6 are $T5c2.kd0$ [4], $T5w2.kd0$, $T6c2.kd10$ and $T6w2.kd20$. In terms of total number of relevant documents retrieved, the best runs for TREC-5 and TREC-6 are $T5c2.kd0$, $T5w4.kd0$, $T6c2.kd6$ and $T6w4.kd10$. For the notation of the name of above runs, "T5" and "T6" represent the runs from TREC-5 and TREC-6 respectively. "c" and "w" represent the runs from character-based and word-based methods. Comparison of the word-based and character-based document processing in terms of average precision and number of relevant documents retrieved for TREC-5 are illustrated in Figure 8.1. Figure 8.2 illustrates the same comparisons for TREC-6. Table 8.1 and Table 8.2 compare the word and character approaches in terms of average precision and the number of relevant documents retrieved on the TREC-5 and TREC-6 datasets respectively. The numbers in these

[4]We only compare the results from BM25 at TREC-5

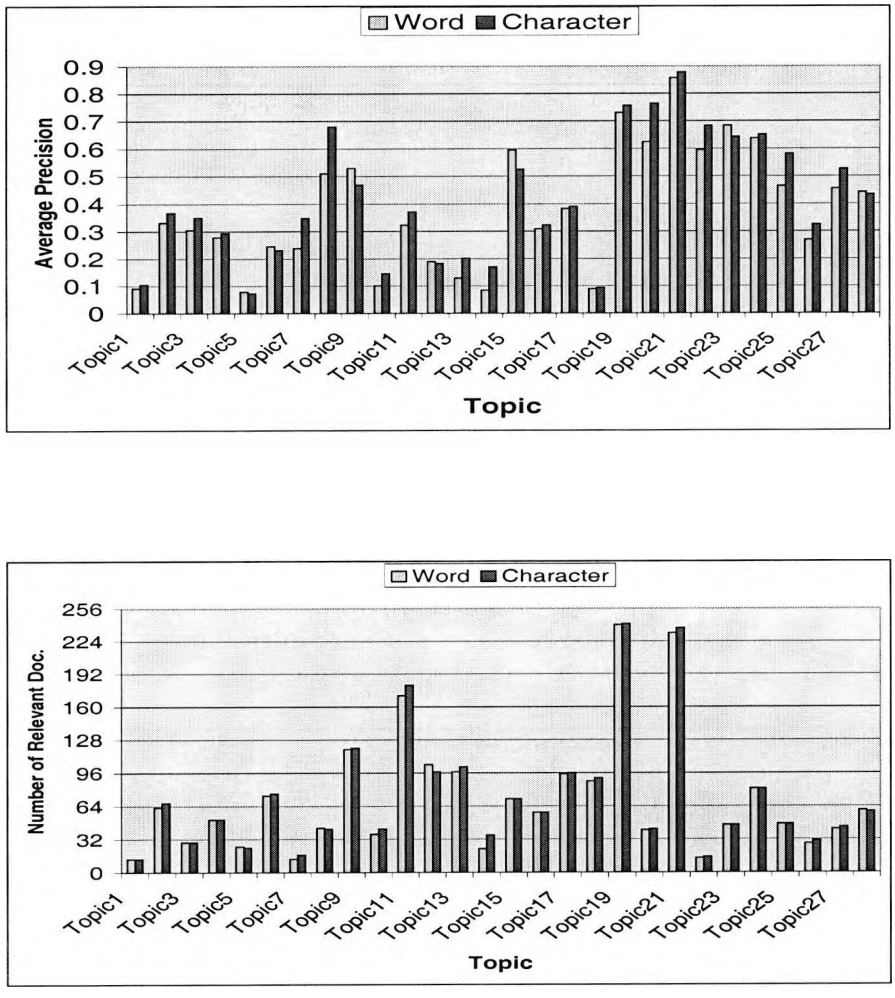
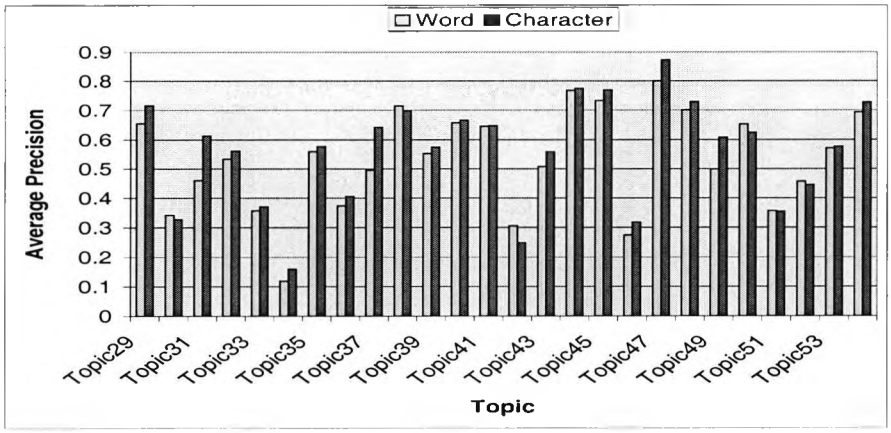


Figure 8.1: Comparison of Character and Word Methods for Each TREC-5 Topic in terms of Average Precision and Number of Relevant Documents Retrieved

two tables represent the number of topics on which the word approach is better than, equal to, or worse than the character approach. We can observe that, first, the character approach produces better results than the word approach in terms of average precision and, second, the character approach can find more relevant documents than the word approach for most of TREC-5 and TREC-6 topics.

We choose two best runs from TREC-5 and two best runs from TREC-6. In each of these two runs, one is from word-based approach and the other one is from character-based approach. Table 8.3 shows the precision for the two TREC-5 runs (T5w2.kd0 and T5c2.kd0) at different recall levels over all the TREC-5 topics. Table 8.4 shows the precision for the two TREC-6 runs (T6w2.kd20 and T6c2.kd10) at different recall levels over all the TREC-6 topics. For comparison



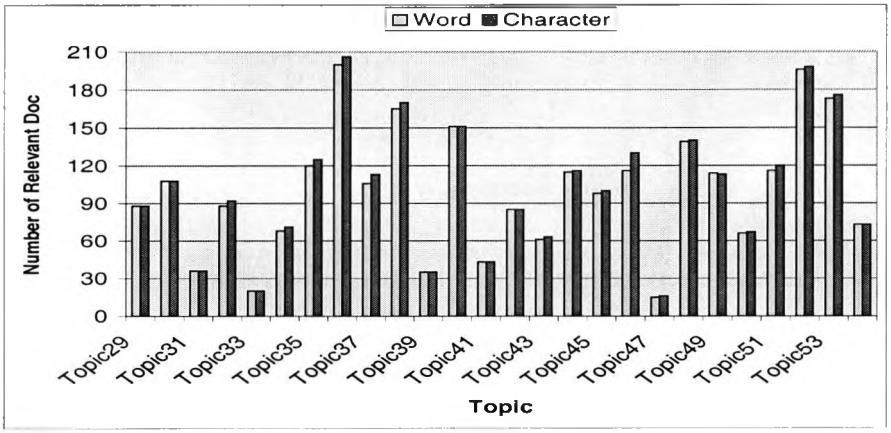


Figure 8.2: Comparison of Character and Word Methods for Each TREC-6 Topic in terms of Average Precision and Number of Relevant Documents Retrieved

purposes, the precision-recall curves for the two TREC-5 runs and the two TREC-6 runs are shown in Figure 8.3. The upper diagram of Figure 8.3 is for TREC-5 and the lower one is for TREC-6. We can observe from the precision-recall curves in Figure 8.3 that the precision values at initial recall levels (0.0 to 0.2) for the best character approach run are almost the same as the best word approach run (in fact the character approach is a little bit better than the word approach), while at later recall levels (0.3 to 1.0) the precision values for character approach run are greater than the word approach run. The precision-recall curves in Figure 8.3 show that the precision values at initial recall levels (0.0 to 0.1) for the best

Comparison	number of topics (on average precision)	number of topics (on relevant documents)
word > *character*	7	4
word = *character*	0	8
word < *character*	21	16

Table 8.1: Comparison of TREC-5 Results in terms of Average Precision and Number of Relevant Documents Retrieved

Comparison	number of topics (on average precision)	number of topics (on relevant documents)
word > *character*	6	1
word = *character*	0	9
word < *character*	20	16

Table 8.2: Comparison of TREC-6 Results in terms of Average Precision and Number of Relevant Documents Retrieved

character approach run are almost the same as the best word approach run [5] and the precision values for the character approach run at all the other recall levels (0.2 to 1.0) are greater than the word approach run.

8.1.3 Detailed Analysis of Some Examples

Experimental results presented in chapter 7 show that the character-based indexing method leads to better retrieval result than the word-based method in terms of average precision. The reasons, as we mentioned before, possibly lie first in the

[5] the value for the word approach is a little better than that for the character approach at 0.0 recall level and the value for the character approach is a little better that that of the word approach at 0.1 recall level

Recall	T5w2.kd0 (word)	T5c2.kd0 (character)
0.00	0.7738	0.7622
0.10	0.6427	0.6550
0.20	0.5385	0.5761
0.30	0.4813	0.5382
0.40	0.4447	0.4814
0.50	0.4004	0.4413
0.60	0.3563	0.3892
0.70	0.2865	0.3303
0.80	0.2285	0.2556
0.90	0.1287	0.1855
1.00	0.0257	0.0304
Average Precision	0.3775	0.4126
Relevant Retrieved	2003	2056

Table 8.3: Average Precision over All the TREC-5 Topics for the Best Runs from Word and Character Approaches

Recall	T6w2.kd20 (word)	T6c2.kd10 (character)
0.00	0.9496	0.9540
0.10	0.8304	0.8340
0.20	0.7620	0.7642
0.30	0.6977	0.7189
0.40	0.6540	0.6869
0.50	0.5866	0.6292
0.60	0.5065	0.5470
0.70	0.4071	0.4519
0.80	0.2922	0.3565
0.90	0.1801	0.2373
1.00	0.0065	0.0390
Average Precision	0.5303	0.5603
Relevant Retrieved	2575	2647

Table 8.4: Average Precision over All the TREC-6 Topics for the Best Runs from Word and Character Approaches

Chinese language itself: in Chinese, single characters (ideographs) may constitute a reasonably good representation of a text; second, partial matching works more effectively for character-based approach than word-based approach. For example, the character "牛" (cattle) can appear in many Chinese words which are all related to the word "牛" (cattle).

	Chinese	English		Chinese	English
1	牛	cattle	7	牛腩	tenderloin
2	公牛	ox, bull	8	牛奶	milk
3	母牛	cow	9	牛场	dairy
4	小牛	calf	10	牛排	beefsteak
5	牛肉	beef	11	牛皮	oxhide, brag
6	小牛肉	veal	12	牛尾	oxtail

Table 8.5: 12 Chinese Words Containing "Cattle"

This example is shown in Table 8.5 which contains 12 Chinese words.[6] As we can see, the Chinese character "牛" has a strong relationship with the other 11 Chinese words even if these 12 Chinese words have totally different English translations. By using a character-based indexing method and searching for the single character "牛", all the other 11 words can be found[7]. Obviously, the single Chinese character "牛" has a very good representation in meaning and thus partial matching is effective for the character-based approach.

In this section, we will have a closer look at some particular topics so that

[6]The single Chinese character "牛" is also a Chinese word.

[7]We can not find these 11 words by searching "牛" for word-based indexing method.

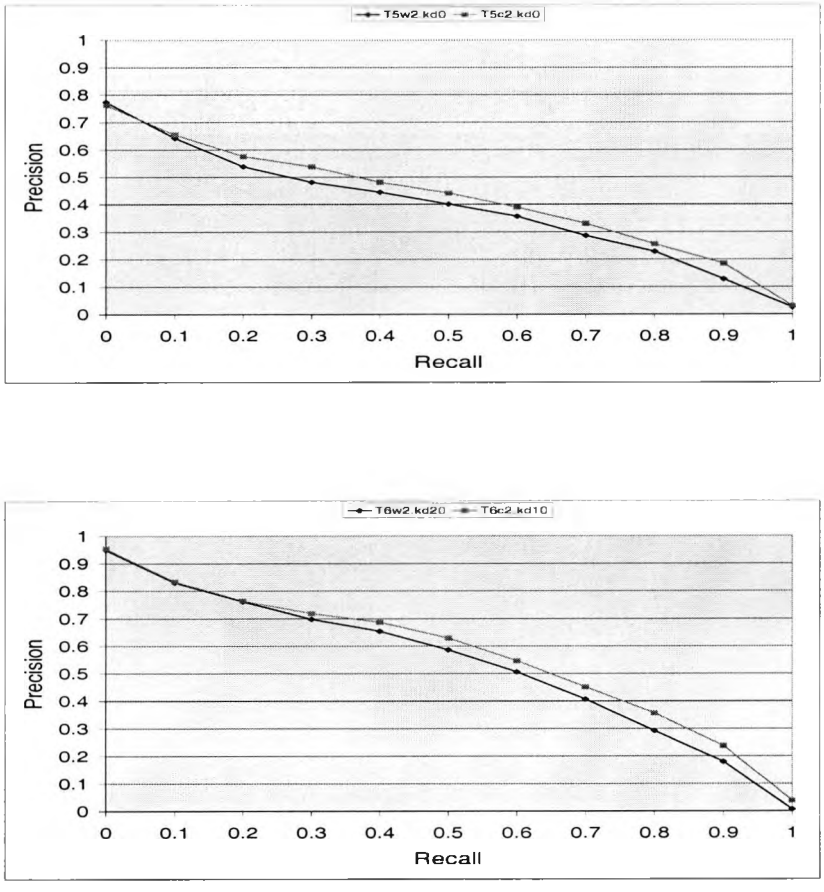


Figure 8.3: Precision-recall curves for the two best runs from word and character approaches at TREC-5 and TREC-6

we can discover the reasons to some extent why the character indexing approach performs better than the word indexing approach. In the following, we will choose two topics for our analysis and discussion. These two topics are topic 8 from TREC-5 and topic 37 from TREC-6.

Topic 8

For topic 8, we choose two best runs $T5c2.kd0$ (character) and $T5w2.kd0$ (word) for our comparison. Topic 8 is about the numeric indicators of earthquake severity in Japan. A relevant document of this topic should contain numeric indicators such as the magnitude of the earthquake, number of deaths or injuries, or property damage. In terms of average precision, the run $T5c2.kd0$ (0.6786) performed 33.16% better than the run $T5w2.kd0$ (0.5096). Table E.1 shows the precision for these

two best runs at different recall levels for the TREC-5 topic 8. The precision-recall curve for these two runs is shown in Figure $E.1$ (see Appendix E).

A close look at the set of retrieved documents revealed some factors responsible for the differences in average precision. For convenience, we only choose two typical documents of the top 1000 retrieved documents for the analysis of topic 8. One of the documents "CB033004-BFJ-346-170" shown in Figure 8.4 is relevant and the other one "pd9105-495" shown in Figure 8.5 is irrelevant. Table 8.6 shows the ranks of these two documents for the word approach run $T5w2.kd0$ and character approach run $T5c2.kd0$.

From the relevant document "CB033004-BFJ-346-170" we can find that there are many words which are synonymous with or closely related to the retrieval keyword "地震" (earthquake). These kind of word [8] contains a character "震" (quake) that appears in the retrieval keyword "地震". Therefore, these synonyms and closely related names can also make effective contribution to this document's relevance score in the character approach. But for the word approach these terms cannot make any contribution to the score of relevance even if they are synonymous with or closely related to the retrieval keyword. Many other examples similar to the above can be found for topic 8. We could say that the character-based method can find some closely related keywords from the documents. But for the word-based method, only the documents that exactly match with the indexing terms of documents can be found. It seems that the character method has minimized the synonym problem compared to the word method.

Table 8.6 shows that the non-relevant document ("pd9105-495" shown in Figure 8.5) obtains a much lower ranking position for the character approach (57) than the word approach (11). "pd9105-495" is about the earthquake in Peru. The reason is that the character approach usually ranks the relevant documents at higher positions than the word approach does. This also means that the character approach relatively moves the non-relevant documents to a lower rank positions compared to the word approach.

[8] such as "震灾" (earthquake disaster), "震后" (after earthquake) and "震灾地区" (earthquake area)

```
<DOC>
<DOCID> CB033004.BFJ (  346)     </DOCID>
<DOCNO> CB033004-BFJ-346-170 </DOCNO>
<DATE>     1995-02-04 22:33:59 (65) </DATE>
<TEXT>
<headline> 日本地震死者达５２５０人 </headline>
<p>
<s> 新华社东京２月４日电（记者张焕利）日本警察厅今天宣布，日本阪神大地震灾害波及１４个府
县，迄今已有５２５０人死亡。 </s>
</p>
<p>
<s> 据日本警察厅今天下午公布的调查统计，在阪神大地震中共死亡５２５０人，其中兵库县５２３
５人，大阪府１４人，京都府１人。 </s>
<s> 兵库县仍有６人下落不明。 </s>
<s> 这次地震造成的灾害波及１４个府县，共有２６８０４人受伤，１０７６１０栋建筑物遭破坏。
</s>
</p>
<p>
<s> 震后恢复工作虽然正在进行，但震灾对居民生活设施的破坏目前还难以全面恢复。 </s>
<s> 特别是神户市和大阪地区的煤气设施的破坏程度比当初估计的要严重得多，现在已修复的煤气管
道只有百分之十，仍有８０多万户居民得不到煤气供应。 </s>
</p>
<p>
<s> 目前，震灾地区有来自日本各地的８０００多人在抢修煤气供应设施，预计全面恢复通气还需要
１个多月的时间。 </s>
<s> （完） </s>
</p>
</TEXT>
</DOC>
```

Figure 8.4: A Relevant Document for Topic 8

Topic 37

For topic 37, we choose two best runs, $T6c2.kd10$ (character) and $T6w2.kd20$ (word), for our comparison. Topic 37 is about the collapse of the bubble economy in Japan. A relevant document should discuss the economic recession in Japan after the collapse of the bubble economy, especially in the areas of finance, real estate, and industry, and the Japanese government's policy to stimulate economy recovery. Discussions of predictions of Japanese economic growth are also relevant. In terms of average precision, the run $T6c2.kd10$ achieved 0.6424, which is 29.60%

Documents	Word (T5w2.kd0)	Character (T5c2.kd0)
CB033004-BFJ-346-170 (relevant)	25	8
pd9105-495 (irrelevant)	11	57

Table 8.6: Ranking Positions of Two Retrieved Documents for Topic 8

<DOC>

<DOCNO> pd9105-495 </DOCNO>

<HL>

日期： 06-MAY-91
星期： 星期一
版次： 7
版名:国际
标题： 秘鲁哥斯达黎加再次地震
正文:秘鲁哥斯达黎加再次地震

</HL>

<TEXT>

据秘鲁地震部门报道，秘鲁北部的圣马丁省境内4日早晨连续发生两次震级分别为里氏4．3级和5．1级的地震，受灾情况目前不详。

据哥斯达黎加地震观测部门4日报告，哥斯达黎加东南部地区3日晚再次发生较强地震，至少造成33人受伤。

</TEXT>

</DOC>

Figure 8.5: An Irrelevant Document for Topic 8

better than the run $T6w2.kd20$ whose average precision is 0.4957. Table E.2 shows the precision for these two best runs at different recall levels for the TREC-6 topic 37. The precision-recall curve for these two runs is shown in Figure $E.2$ (see Appendix E).

We choose two documents from the top 1000 retrieved documents for the analysis of topic 37. One document, "pd9207-5342", is relevant and the other, "pd9302-2926", is irrelevant. Table 8.7 shows the ranks of these two documents for the word approach run $T5w2.kd0$ and the character approach run $T5c2.kd0$. This table shows an example for topic 37 in which the character approach gives a higher rank for the relevant document and a lower rank for the irrelevant document than the word approach. The relevant document "pd9207-5342" is shown in Figure 8.6. We can observe that a word "气泡经济", which is synonym of the keyword "泡沫经济" (bubble economy) in topic 37, appears in this relevant document. The synonym word "气泡经济" contains three characters that appear in the retrieval keyword "泡沫经济" except the first character "气" (air). Obviously, the character approach can use both partial matching and exact matching for retrieval and the

word approach can only use partial matching for retrieval if the search terms are compound unit terms. Therefore, we can say that partial matching is more effective in the character approach than in the word approach. Partial matching is one of reasons why the character approach ranks the relevant document "pd9207-5342" at position 21, while the word approach ranks it at 57[9]. Some other reasons, such as the use of compound unit weighting, will be discussed later.

<DOC>

<DOCNO> pd9207-5342 </DOCNO>

<HL>

日本上市企业总收益下降

</HL>

<TEXT>

新华社东京７月１日电日本经济新闻社最近进行的调查表明，日本上市企业及其各有关企业（银行、证券、保险公司等除外）１９９２年３月决算的１９９１年度的总收益大幅度下降，纯利润比１９９０年度减少２８．４％。

这项调查是以８３２家公司、３２个行业为对象进行的。调查结果表明，如果把上市企业的各有关企业排除在外，仅上市企业单独决算，那么，其单独的纯利润就比１９９０年度减少１８．９％。

这说明，日本气泡经济的破灭，对上市企业各子公司的收益造成了直接的不良影响，而且在国外收买的企业和从事生产的子公司的萧条也拖了总收益的后腿。

此外，调查的３２个行业中，有２４个行业的收益比１９９０年度减少，特别是精密仪器行业，其纯利润减少了９６．９％，纸浆、机械、电机行业的纯利润也大幅度下降。

</TEXT>

</DOC>

Figure 8.6: A Relevant Document for Topic 37

Documents	Word (T6w2.kd20)	Character (T6c2.kd10)
pd9207-5342 (relevant)	57	21
pd9302-2926 (irrelevant)	52	156

Table 8.7: Ranking Positions of Two Retrieved Documents for Topic 37

[9]Presumably it is not always the case that the character approach boosts relevant documents and not the others. However, our example illustrates one reason why the character approach is better than the word approach on average.

Further Evidence

The above analysis on topic 8 and topic 37 indicates that the character approach can boost relevant documents and not the irrelevant documents. To determine whether this statement holds for other topics, we choose four more topics for further analysis. Two of these four topics, topic 5 and topic 14, are chosen from TREC-5 and the other two, topic 33 and topic 47, are chosen from TREC-6. For these four topics, the best runs in terms of average precision from the character-based and word-based indexing methods are chosen[10]. The improvement of the character-based approach over the word-based approach for these four topics is shown in Table 8.8.

topics	P(character)	P(word)	increase	topics	P(character)	P(word)	increase
5	0.0727	0.0808	-10.02%	33	0.3703	0.3569	3.75%
14	0.1696	0.0864	96.30%	47	0.8722	0.7994	9.11%
1. the increase of character approach over the word approach is calculated by the following function $\frac{P(character)-P(word)}{P(word)} \times 100\%$; 2. P(character) and P(word) are the average precision for character and word approach respectively.							

Table 8.8: Improvement of Character Approach over Word Approach for Topics 5, 14, 33 and 47 in terms of Average Precision

The ranking positions of all the retrieved relevant documents for topic 14 is listed in Table 8.9. Topic 14 is about the cases of AIDS in China. A relevant document should contain information on the areas in China that have the highest AIDS cases, how the AIDS virus is transmitted, and how the Chinese government combats AIDS.

The ranking positions of all the retrieved relevant documents for Topic 5, Topic 33 and Topic 47 are given in Appendix F. Comparisons of character and word approaches in terms of ranking positions for the retrieved relevant documents of the selected four topics are shown in Table 8.10, where the second (third or fourth) column stands for the number of retrieved relevant documents for which the character approach gives better (equal or worse) ranking position than the word approach. For example, for topic 14, the word approach gives better ranking position than the character approach on only 4 out of 36 retrieved relevant documents. The

[10]The best runs for character and word approaches at TREC-5 and TREC-6 are $T5c2.kd0$, $T5w2.kd0$, $T6c2.kd10$ and $T6w2.kd20$.

word approach gives the same ranking position as the character approach on one document. However, the character approach gives better ranking position than the word approach on 31 of the 36 retrieved relevant documents. 13 of these 31 relevant documents were retrieved only by the character approach, not by the word approach. The reason is that topic 14 uses a rarely-used translation for the word "AIDS". But the TREC documents use the official form of AIDS in Chinese for most of the cases in the collection. Consequently, the correct segmentation of topic 14 leads to no matching with many documents that use the official form for the word approach. However, when single characters are used, the second and third characters happen to be the same in both translations for AIDS and some matching between query and related relevant documents occurs.

index	relevant documents retrieved	character	word	index	relevant documents retrieved	character	word
1	CB001030-BFW-380-668	182	341	19	pd9112-898	50	>1000
2	CB015003-BFW-598-227	8	6	20	pd9201-1338	411	>1000
3	CB024004-BFW-1008-271	3	5	21	pd9206-2490	113	708
4	CB024007-BFW-271-443	12	12	22	pd9207-2143	221	>1000
5	CB026018-BFJ-726-591	18	22	23	pd9207-2506	216	225
6	CB027031-BFW-559-38	9	26	24	pd9207-5827	29	67
7	CB030031-BFW-541-16	527	>1000	25	pd9209-3980	191	>1000
8	CB034028-BFW-730-712	181	464	26	pd9209-3981	7	9
9	CB047021-BFW-574-186	79	>1000	27	pd9209-5871	262	892
10	CB048020-BFJ-460-94	141	452	28	pd9210-1684	77	130
11	CB049014-BFW-341-380	90	148	29	pd9211-5651	575	>1000
12	CB049020-BFW-302-207	127	20	30	pd9212-2047	24	40
13	pd9103-1150	138	>1000	31	pd9212-2048	167	894
14	pd9103-1326	750	>1000	32	pd9212-2053	65	>1000
15	pd9103-2932	5	4	33	pd9306-2625	4	49
16	pd9104-2229	15	14	34	pd9309-1755	193	>1000
17	pd9110-351	599	>1000	35	pd9312-1164	38	105
18	pd9112-2851	199	>1000	36	pd9312-706	71	109

Table 8.9: Ranking Positions of All the Retrieved Relevant Documents for Topic 14

topics	character>word	character=word	character<word
5	14	0	13
14	31	1	4
33	11	5	4
47	9	3	3

Table 8.10: Comparisons of Character and Word Approaches in terms of Ranking Positions for the Retrieved Relevant Documents

We can also see why use of characters can help achieve better performance in some circumstances in the following examples. One example is a word "死亡" from topic 22 meaning "dead" and can be expressed in related forms such as "死" or "死伤". "死" is the shared character of two similar words "死亡" and "死伤". Another example is "灭" meaning "extinct" in topic 25. "灭" is the shared character

of two similar words "灭绝" and "灭种". If perfect segmentation is performed but alternative forms are used in the query and a document, there will be no matching value between query and document even though they are about the same concept. However, if the Chinese words are broken up into single characters, some character(s) are shared and non-zero matching values result. Chinese characters do carry meaning though often imprecise; they probably lie between alphabets and words in this capacity.

8.2 Compound Unit Weighting

As we have discussed in the section 8.1, single Chinese characters and partial matching play an important role in making better performance for character-based approach and single characters (ideographs) constitute a reasonably good representation of a Chinese text. The use of compound unit weighting is another reason why the character-based approach performs better than the word-based approach.

From the experimental results in chapter 7, we can observe that compound unit weighting functions work more effectively in character-based approaches than word-based approaches. This is because the number of single units that constitute a compound unit in character-based approaches is usually larger than that in word-based approaches, which results in more weights for the compound unit in character-based approaches. On the contrary, the number of single units that constitute a compound unit in word-based approaches can only be 1 or 2 in our experiment. For this reason, we only analyse and discuss the results of compound unit weighting for the character approach.

8.2.1 Analyses for TREC-5 Dataset

On each TREC-5 topic, we have conducted empirical evaluation of five different compound unit weighting methods. The comparison results for the character approach in terms of which compound unit weighting methods produce the best average precision value are shown in Table 8.11 and Table 8.12. All these five compound weighting methods in our evaluation experiments use the same single

unit weighting function BM25. The results show that no compound unit weighting method produces the best results in terms of average precision on all the test topics. However, the compound unit weighting method $Weight_2$ outperforms all the other four compound unit weighting methods on most of the tested topics.

Topic	$Weight_1$	$Weight_2$	$Weight_3$	$Weight_4$	$Weight_5$
Topic 1		best			
Topic 2		best			
Topic 3				best	
Topic 4		best			
Topic 5	best				
Topic 6		best			
Topic 7					best
Topic 8		best			
Topic 9		best			
Topic 10		best			
Topic 11		best			
Topic 12	best				
Topic 13	best				
Topic 14		best			
Topic 15			best		
Topic 16		best			
Topic 17		best			
Topic 18	best				
Topic 19					best
Topic 20	best				
Topic 21	best				
Topic 22	best				
Topic 23		best			
Topic 24		best			
Topic 25		best			
Topic 26					best
Topic 27		best			
Topic 28					best

Table 8.11: Comparison of Different Compound Unit Weighting for TREC-5 Character Approach in terms of the Best Average Precision by Setting k_d to 0

k_d	Compound	Number of Topics
0	$Weight_1$	7
	$Weight_2$	15
	$Weight_3$	1
	$Weight_4$	1
	$Weight_5$	4

Table 8.12: Number of Topics for TREC-5 Character Approach in terms of the Best Average Precision

Although the formula $Weight_1$ or $Weight_5$ is not a good formula in general, it performs better than other formulae on some topics such as topic 5, 12, 13, 18, 20, 21 and 22 for $Weight_1$ and topic 7, 19, 26 and 28 for $Weight_5$. While $Weight_4$ produces the best result only on topic 3, its performance ranks the second in general. If we can find conditions under which each formula leads to best performance, we can select "right formulae" for different topics and can improve the general performance. Figure 8.7 shows the comparison of the character approach compound unit weighting methods for each TREC-5 topic in terms of average

precision.

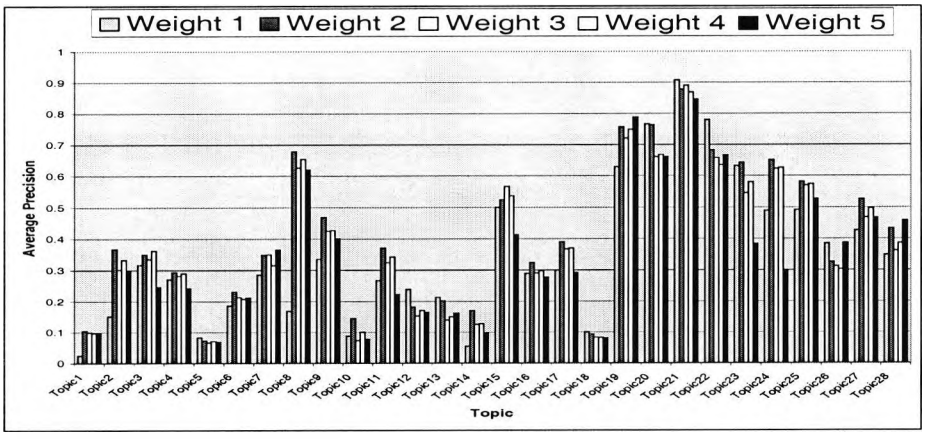


Figure 8.7: Comparison of the Character Approach Compound Unit Weighting Methods for Each TREC-5 Topic in terms of Average Precision

Table 8.13 shows the average precisions over all the TREC-5 topics for the runs using five compound unit weighting methods. These five compound unit weighting methods are $Weight_1$, $Weight_2$, $Weight_3$, $Weight_4$ and $Weight_5$ as defined in Table 4.5. As shown in Table 4.5, $Weight_3$ has no second boost weight. $Weight_1$ and $Weight_5$ have been assigned the biggest boost weights among these five methods, while $Weight_2$ and $Weight_4$ have been assigned moderate boost weights. The precision-recall curves for these five runs are shown in Figure 8.8. From these curves we can see that $Weight_2$ is obviously the best among all the other four methods at all the recall levels. $Weight_4$ ranks the second, but just a little bit better than $Weight_3$. $Weight_1$ and $Weight_5$ are not good at all compared to compound unit weighting methods $Weight_3$ and $Weight_4$ at the recall level (0.0 to 0.7). But the precision values at recall levels (0.7 to 1.0) for $Weight_1$ and $Weight_5$ are almost the same as those for $Weight_3$ and $Weight_4$.

8.2.2 Analyses for TREC-6 Dataset

On each TREC-6 topic, we also conducted an empirical evaluation of six different compound unit weighting methods. The comparison results for the character approach in terms of which compound unit weighting methods produce the best

Recall	$Weight_1$	$Weight_2$	$Weight_3$	$Weight_4$	$Weight_5$
0.00	0.7408	0.7622	0.7764	0.7566	0.7668
0.10	0.5588	0.6550	0.6243	0.6455	0.5811
0.20	0.4853	0.5761	0.5507	0.5564	0.5043
0.30	0.4362	0.5382	0.4987	0.5073	0.4436
0.40	0.4102	0.4814	0.4458	0.4656	0.4024
0.50	0.3558	0.4413	0.4247	0.4250	0.3809
0.60	0.3251	0.3892	0.3591	0.3602	0.3348
0.70	0.2833	0.3303	0.2711	0.2890	0.2754
0.80	0.2353	0.2556	0.2236	0.2204	0.2152
0.90	0.1483	0.1855	0.1408	0.1485	0.1337
1.00	0.0235	0.0304	0.0266	0.0299	0.0293
Average Precision	0.3475	0.4126	0.3795	0.3863	0.3507
Relevant Retrieved	2004	2056	1986	2011	1992

Table 8.13: Average Precision over All the TREC-5 Topics for the Runs from Five Compound Unit Weighting Methods

average precision value are shown in Table 8.14 and Table 8.15. All these six compound weighting methods in our evaluation experiments use the same single unit weighting function such as BM26 in Table 8.14 by setting k_d to 15. The results for TREC-6 also show that no compound unit weighting method produces the best results in terms of average precision on all the test 26 TREC-6 topics. However, the compound unit weighting method $Weight_2$ outperforms again all the other five compound unit weighting methods on most of the tested topics when k_d equals to 0, 2, 6, 8, 15 and 20. $Weight_1$ produces the best results on half of the TREC-6 topics when when k_d equals to 50 (see Table 8.15).

Topic	$Weight_1$	$Weight_2$	$Weight_3$	$Weight_4$	$Weight_5$	$Weight_6$
Topic 29	best					
Topic 30						best (K=20)
Topic 31	best					
Topic 32				best		
Topic 33					best	
Topic 34						best (K=100)
Topic 35	best					
Topic 36		best				
Topic 37				best		
Topic 38		best				
Topic 39					best	
Topic 40		best				
Topic 41		best				
Topic 42		best				
Topic 43				best		
Topic 44				best		
Topic 45		best				
Topic 46						best (K=100)
Topic 47	best					
Topic 48		best				
Topic 49		best				
Topic 50		best				
Topic 51		best				
Topic 52		best				
Topic 53		best				
Topic 54		best				

Table 8.14: Comparison of Different Compound Unit Weighting for TREC-6 Character Approach in terms of the Best Average Precision by Setting k_d to 15

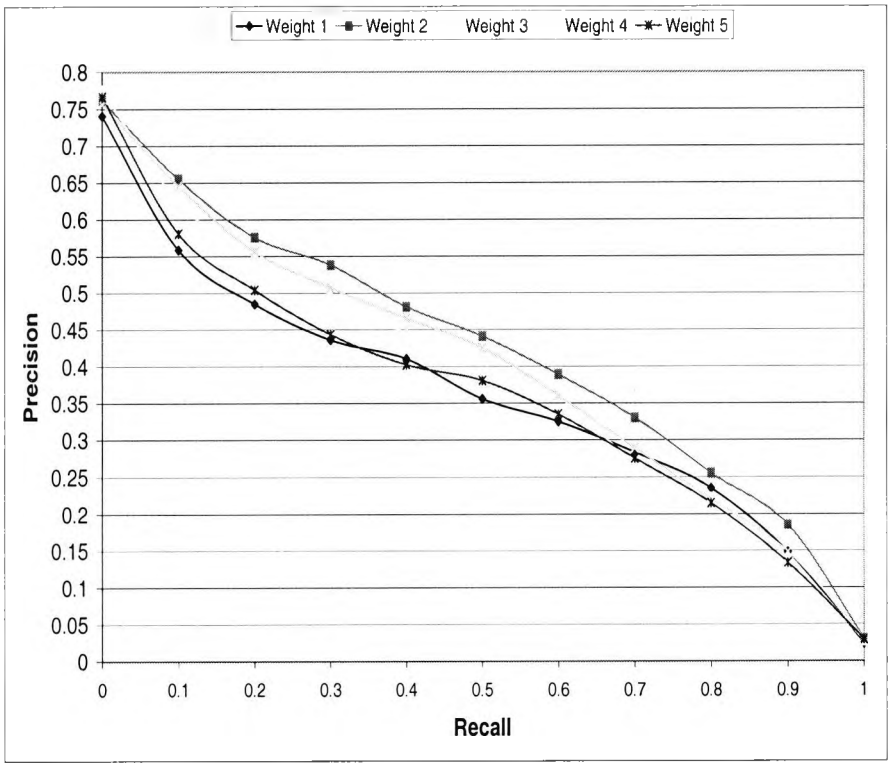


Figure 8.8: Precision-recall Curves for the Runs from Five Compound Unit Weighting Methods at TREC-5

All the compound unit weighting methods, except $Weight_3$, could produce the best results in terms of average precision by using the same single unit weighting function. Again, the formula $Weight_1$ or $Weight_5$ is not a good formula in general for TREC-6 topics. But both $Weight_1$ and $Weight_5$ can perform better than other formulae on some topics such as on seven topics for $Weight_1$ when k_d is set to 0 and three topics for $Weight_5$ when k_d is set to 15. The reason for the poor general performance of $Weight_1$ and $Weight_5$ is that these two compound unit weighting functions can produce best results on some of the topics, but they can also produce very poor results on some other topics. For example, $Weight_1$ produces the best results among all the methods on topic 29 and 47, but its results on topic 37, 39 and 49 are very poor comparing to other five methods; $Weight_5$ produces the best results among all the methods on topic 39, but produces very poor results on topic 29, 42 and 49. Although the method $Weight_4$ produces the best results on

k_d	Compound	Number of Topics	k_d	Compound	Number of Topics
0	$Weight_1$	7	10	$Weight_1$	4
	$Weight_2$	16		$Weight_2$	17
	$Weight_3$	0		$Weight_3$	0
	$Weight_4$	1		$Weight_4$	3
	$Weight_5$	2		$Weight_5$	2
2	$Weight_1$	5	15	$Weight_1$	4
	$Weight_2$	17		$Weight_2$	13
	$Weight_3$	0		$Weight_3$	0
	$Weight_4$	3		$Weight_4$	4
	$Weight_5$	1		$Weight_5$	2
				$Weight_6$	3
6	$Weight_1$	4	20	$Weight_1$	5
	$Weight_2$	19		$Weight_2$	15
	$Weight_3$	0		$Weight_3$	0
	$Weight_4$	2		$Weight_4$	4
	$Weight_5$	1		$Weight_5$	2
8	$Weight_1$	4	50	$Weight_1$	13
	$Weight_2$	18		$Weight_2$	8
	$Weight_3$	0		$Weight_3$	0
	$Weight_4$	2		$Weight_4$	0
	$Weight_5$	2		$Weight_5$	5

Table 8.15: Number of Topics for TREC-6 Character Approach in terms of the Best Average Precision

a smaller number of topics than $Weight_1$, its performance still ranks the second in general. The reason is that the results produced by $Weight_4$ are pretty stable. Even if $Weight_4$ does not produce best results on many tested topics, but always at position 2 or 3, it never generates very poor results.

Figure 8.9 shows the comparison of the character approach compound unit weighting methods for each TREC-6 topic in terms of average precision. Table 8.16 presents the average precisions over all the TREC-6 topics for the runs from different compound unit weighting methods by setting k_d to 0 and 15 respectively. The upper table is for the five runs by setting k_d to 0 and lower table is for the six runs by setting k_d to 15, in which the parameter d for the run using $Weight_6$ is set to 100. The precision-recall curves for the five runs using compound unit weighting methods $Weight_1$, $Weight_2$, ... and $Weight_5$ ($k_d = 0$) are shown in Figure 8.10 and the precision-recall curves for the five runs using compound unit weighting methods $Weight_1$, $Weight_2$, ... and $Weight_5$ ($k_d = 15$) are shown in Figure 8.11. From these curves we can see that $Weight_2$ is always the best among all the other methods at all the recall levels. $Weight_4$ ranks the second and $Weight_3$ ranks the third. The difference among $Weight_2$, $Weight_3$ and $Weight_4$ by setting k_d to 0 is greater than those of by setting k_d to 15. By setting k_d to 15, $Weight_4$ is just a little bit better than $Weight_3$. This means $Weight_2$ and $Weight_4$ make more improvement over $Weight_3$ by using BM25 than that of using BM26. Again for the TREC-6 topics, $Weight_1$ and $Weight_5$ are not good at all compared to

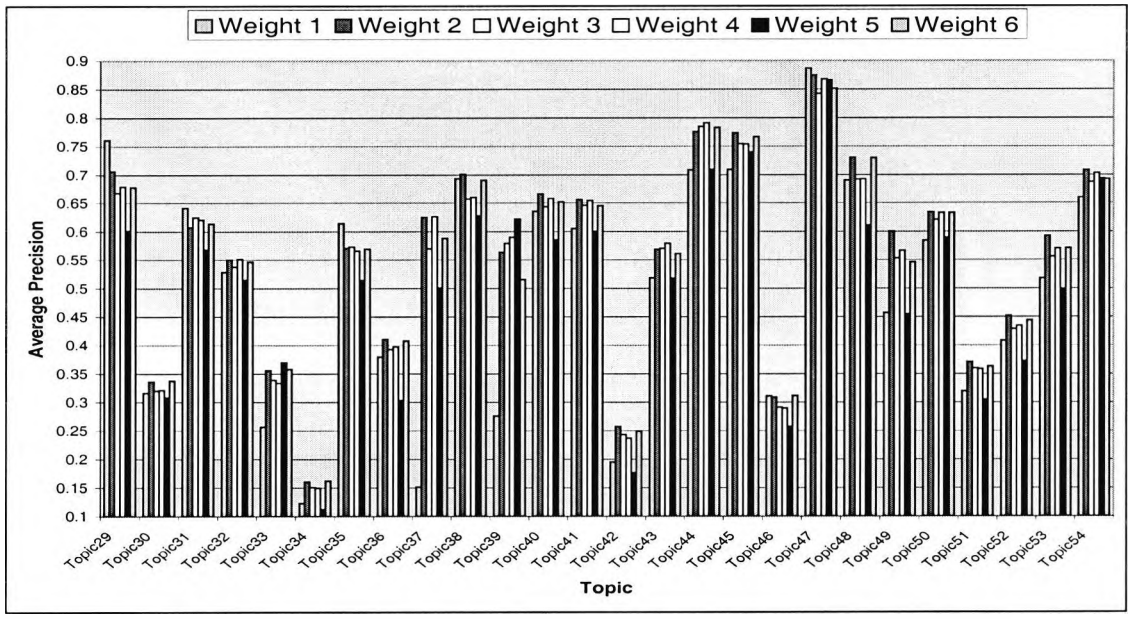


Figure 8.9: Comparison of the Character Approach Compound Unit Weighting Methods for Each TREC-6 Topic in terms of Average Precision

compound unit weighting methods $Weight_3$ and $Weight_4$ at the recall levels 0.0 to 0.7 ($k_d = 0$) and 0.0 to 0.8 ($k_d = 15$). But the precision values at recall levels 0.7 to 1.0 ($k_d = 0$) and 0.8 to 1.0 for the $Weight_1$ and $Weight_5$ are almost the same as those for the $Weight_3$ and $Weight_4$.

One interesting result is that $Weight_6$ [11] improves performance a lot over $Weight_1$ (see Figure 8.12). This discovery makes us understand that $Weight_1$ assigns too much weight to the compound unit and over-weighting for the compound units will downgrade the performance very much. Assigning no more extra weight to the compound unit such as $Weight_3$ is better than assigning too much extra weight such as $Weight_1$ and $Weight_5$.

[11] The motivation for designing $Weight_6$ is to lower down the weight assigned for $Weight_1$ (see Table 4.5).

Recall	$Weight_1$	$Weight_2$	$Weight_3$	$Weight_4$	$Weight_5$
0.00	0.9108	0.9150	0.9117	0.9160	0.8987
0.10	0.7593	0.8009	0.7663	0.7858	0.7467
0.20	0.6827	0.7442	0.7168	0.7239	0.6757
0.30	0.6119	0.6945	0.6646	0.6791	0.6326
0.40	0.5593	0.6594	0.6127	0.6436	0.5699
0.50	0.5016	0.5879	0.5552	0.5746	0.4893
0.60	0.4365	0.5201	0.4834	0.4992	0.4125
0.70	0.3787	0.4239	0.3889	0.4009	0.3366
0.80	0.3082	0.3326	0.3022	0.3084	0.2659
0.90	0.1870	0.2343	0.1743	0.1888	0.1797
1.00	0.0446	0.0399	0.0327	0.0294	0.0311
Average Precision	0.4789	0.5341	0.4967	0.5113	0.4627
Relevant Retrieved	2550	2637	2537	2558	2493

Recall	$Weight_1$	$Weight_2$	$Weight_3$	$Weight_4$	$Weight_5$	$Weight_6$
0.00	0.9158	0.9536	0.9427	0.9499	0.9005	0.9470
0.10	0.7835	0.8361	0.8307	0.8294	0.7953	0.8253
0.20	0.6943	0.7730	0.7651	0.7732	0.7211	0.7696
0.30	0.6428	0.7265	0.7116	0.7190	0.6703	0.7172
0.40	0.5866	0.6895	0.6740	0.6827	0.6210	0.6786
0.50	0.5239	0.6374	0.6154	0.6263	0.5648	0.6221
0.60	0.4528	0.5595	0.5259	0.5404	0.4635	0.5375
0.70	0.3833	0.4409	0.4217	0.4262	0.3530	0.4337
0.80	0.3201	0.3426	0.3091	0.3171	0.2755	0.3228
0.90	0.2075	0.2192	0.1957	0.2032	0.1882	0.2033
1.00	0.0336	0.0332	0.0202	0.0266	0.0272	0.0296
Average Precision	0.4981	0.5599	0.5417	0.5494	0.5007	0.5488
Relevant Retrieved	2575	2621	2546	2566	2511	2605

Table 8.16: Average Precisions over All the TREC-6 Topics for the Runs from Five (k_d=0) or Six (k_d=15) Compound Unit Weighting Methods

8.2.3 Detailed Analyses of Topic 42

Topic 42

For topic 42, we choose six typical character approach runs $T6c1.kd10$, $T6c2.kd10$, $T6c3.kd10$, $T6c4.kd10$, $T6c5.kd10$ and $T6c6.kd10$ [12] for our comparison. Topic 42 is about the flood prevention of dikes and reservoirs in the Seven Great Rivers in China. A relevant document for this topic should discuss specific dikes and reservoirs in the Seven Great Rivers region. Relevant documents should discuss the following information: construction projects, measures for flood and rescue work, reservoir water levels, or flood discharging. But documents discussing the Three Gorge Project are non-relevant.

Table 8.17 shows the comparison of six runs by using different compound unit weighting methods for topic 42 at all recall levels. The precision-recall curves for these six runs (k_d = 15) are shown in Figure 8.13. From these curves we

[12]The parameter k_d is set to 10

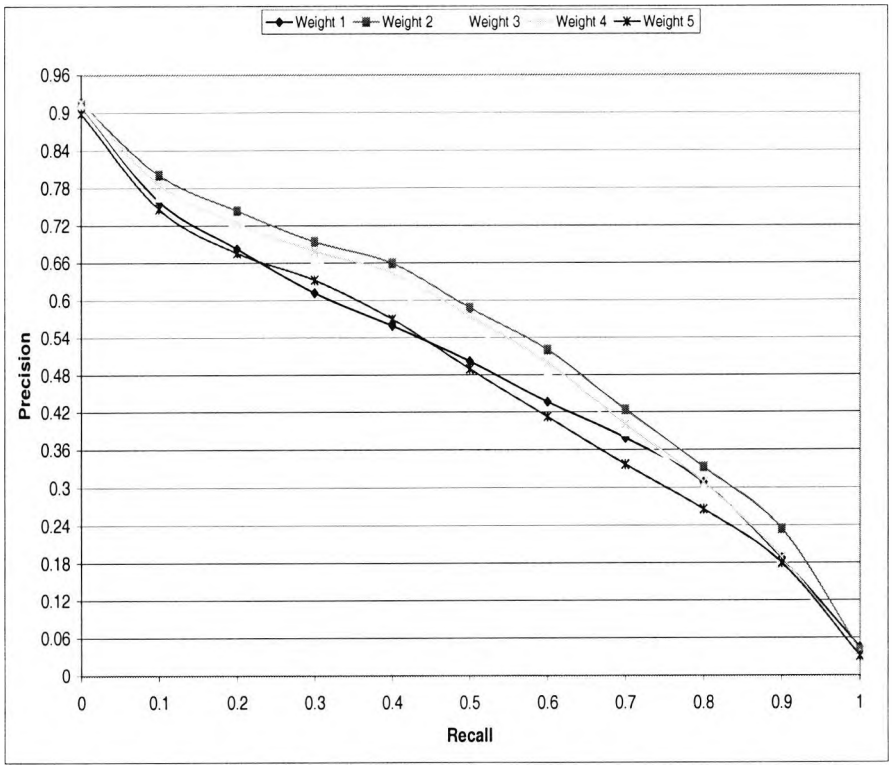


Figure 8.10: Precision-recall Curves for the Five Runs from $Weight_1$, $Weight_2$, $Weight_3$, $Weight_4$ and $Weight_5$ at TREC-6 (k_d=0)

can see that $Weight_2$ is always the best among the five compound unit weighting methods $Weight_1$, $Weight_2$, $Weight_3$, $Weight_4$ and $Weight_5$ at all recall levels. The performance of $Weight_3$ and $Weight_4$ is almost the same for topic 42. $Weight_1$ and $Weight_5$ are not good at all compared to the other three compound unit weighting methods $Weight_2$, $Weight_3$ and $Weight_4$. By lowering down the weight assigned by $Weight_1$, $Weight_6$ makes 19.53% improvement over $Weight_1$ in terms of average precision. In general, $Weight_6$ ranks second and produces the best results at the recall levels from 0.8 to 1.0 among all the compound unit weighting methods.

Based on our TREC-5 and TREC-6 experiments, first we can conclude that $Weight_2$ produces the best results in general among the six tested compound unit weighting methods. Second, by analysing the formulas and the results, we can conclude that the boost weight in $w(t_1\ adj\ t_2\ adj \cdots adj\ t_j)$ plays an important

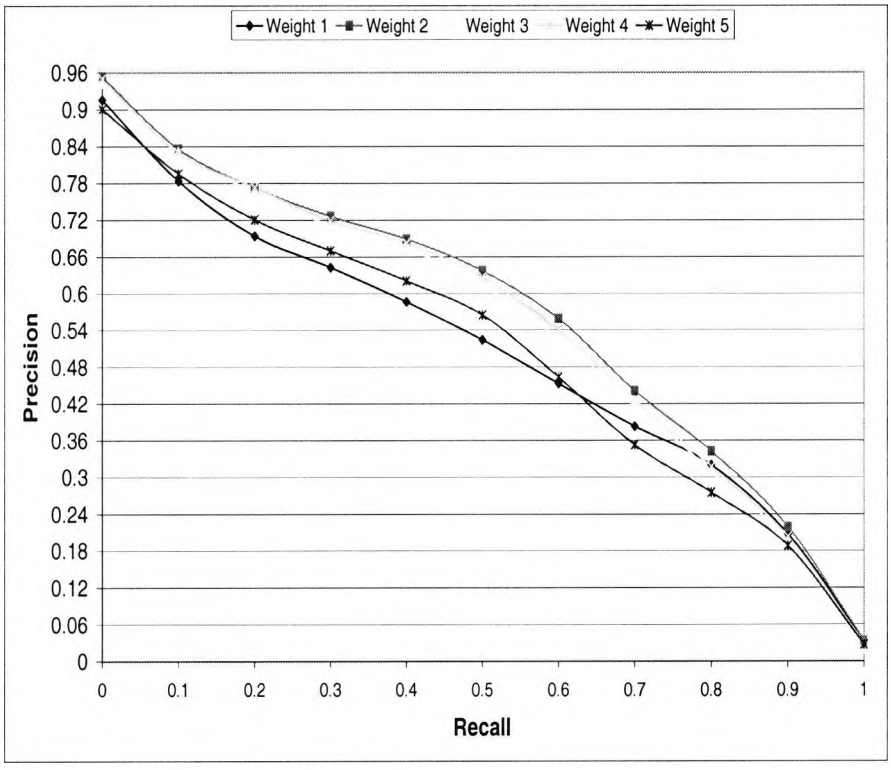


Figure 8.11: Precision-recall Curves for the Five Runs from $Weight_1$, $Weight_2$, $Weight_3$, $Weight_4$ and $Weight_5$ at TREC-6 (k_d=15)

role in the performance of the Chinese retrieval system. Big boost weights (like in $Weight_1$ and $Weight_5$) are not good. A too small boost weight (like in $Weight_3$) does not work well either, but better than a too big boost weight. Moderate boost weights (such as $Weight_2$ and $Weight_4$) produce the best results.

8.3 Single Unit Weighting

We have discussed two factors that can affect the performance of retrieval. These two factors are different document processing methods and different compound unit weighting methods. Another factor that can also affect the performance of retrieval is the single unit weighting. In the following, we will discuss why and how the single unit weighting function can make a difference for retrieval performance.

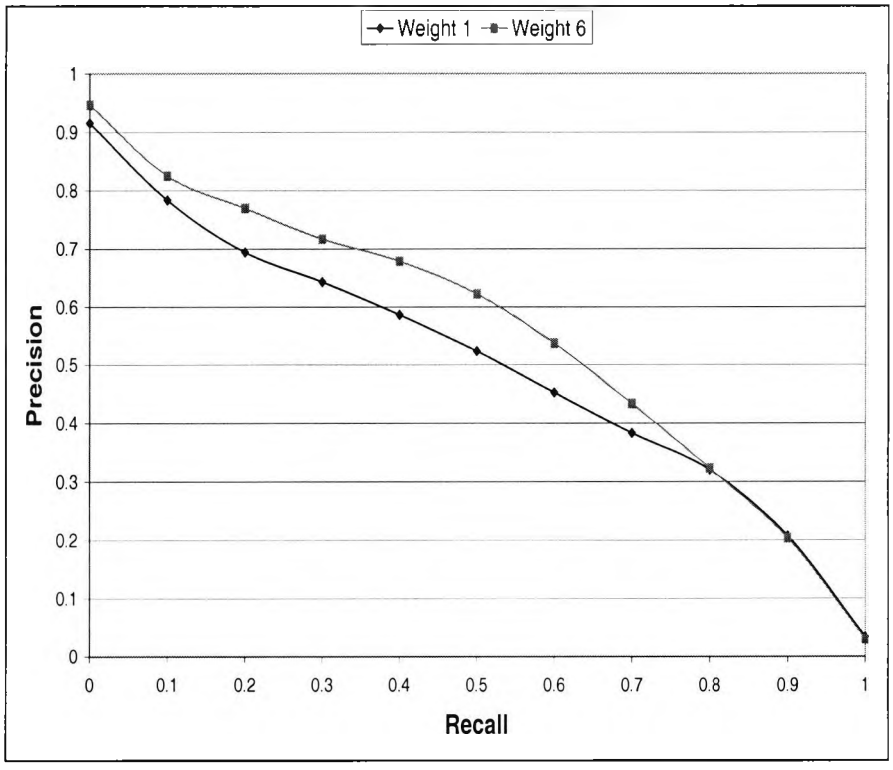


Figure 8.12: Precision-recall Curves for the Two Runs from $Weight_1$ and $Weight_6$ at TREC-6

While character-based approaches are more sensitive to compound unit weighting functions, word-based approaches seem to be more sensitive to single unit weighting functions (see Figures 7.3 and 7.4). One possible reason is that n (the number of documents containing a specific term) in single unit weighting functions (BM25 and BM26) has smaller values in word-based approaches than in character-based approaches, which results in larger values in the left part of BM25 or BM26, which in turn leads to more influence of the value for k_d on the whole value of the function.

8.3.1 Analyses for Character Approach on TREC-6 Dataset

On each TREC-6 topic, we have conducted an empirical evaluation of eight different single unit weighting methods for the five compound unit weighting methods

Recall	$Weight_1$	$Weight_2$	$Weight_3$	$Weight_4$	$Weight_5$	$Weight_6$
0.00	0.6667	1.0000	1.0000	1.0000	0.6667	1.0000
0.10	0.2917	0.3750	0.3143	0.3235	0.2857	0.3529
0.20	0.2857	0.3200	0.3101	0.3056	0.2857	0.3256
0.30	0.2762	0.3200	0.3101	0.2937	0.2762	0.3150
0.40	0.2267	0.3200	0.3101	0.2917	0.2375	0.3150
0.50	0.1858	0.3133	0.3019	0.2866	0.2115	0.3013
0.60	0.1791	0.2648	0.2661	0.2627	0.1602	0.2377
0.70	0.1507	0.2164	0.2006	0.1846	0.1320	0.2129
0.80	0.1048	0.1747	0.1210	0.1315	0.0000	0.1849
0.90	0.0000	0.0000	0.0000	0.0000	0.0000	0.0876
1.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Average Precision	0.1953	0.2573	0.2429	0.2371	0.1764	0.2491
Relevant Retrieved	82	84	82	82	70	85

Table 8.17: Average Precision of the Runs from Six Different Compound Unit Weighting Function for Topic 42 (k_d=15)

$Weight_1$, $Weight_2$, $Weight_3$, $Weight_1$ and $Weight_5$. The comparison results for the character approach in terms of which single unit weighting methods produce the best average precision value are shown in Table 8.18. All these eight single unit weighting functions in Table 8.18 use the same compound unit weighting function $Weight_2$. The results show that the single unit weighting function BM25 ($k_d = 0$) produces the best results in terms of average precision only on three [13] of the 26 TREC-6 topics. BM26 ($k_d > 0$) produces the best results on 23 of the 26 TREC-6 topics. Table 8.19 gives some more comparison results for BM25 and BM26 by using different compound unit weighting methods.

The empirical results showing how BM25 and BM26 can affect retrieval performance in terms of average precision by using different compound unit weighting formulae are presented in Appendix G. Table $G.1$ shows the detailed data of average precision over all the TREC-6 topics for the eight runs using $Weight_2$ compound unit weighting formula and BM26 with different k_d's values. The precision-recall curves for the two runs, $T6c2.kd0$ and $T6c2.kd10$ are depicted in Figure G.1. Table $G.2$ shows some more detailed data of average precision over all the TREC-6 topics for the eight runs using $Weight_1$, $Weight_3$, $Weight_4$ and $Weight_5$. The precision-recall curves for these eight runs are shown in Figures G.2, G.3, G.4 and G.5 in in Appendix G). From these curves we can see that BM26 works more effectively for $Weight_3$ and $Weight_5$ than the other compound unit weighting methods.

[13]The three topics are: Topic 31, 32 and 54

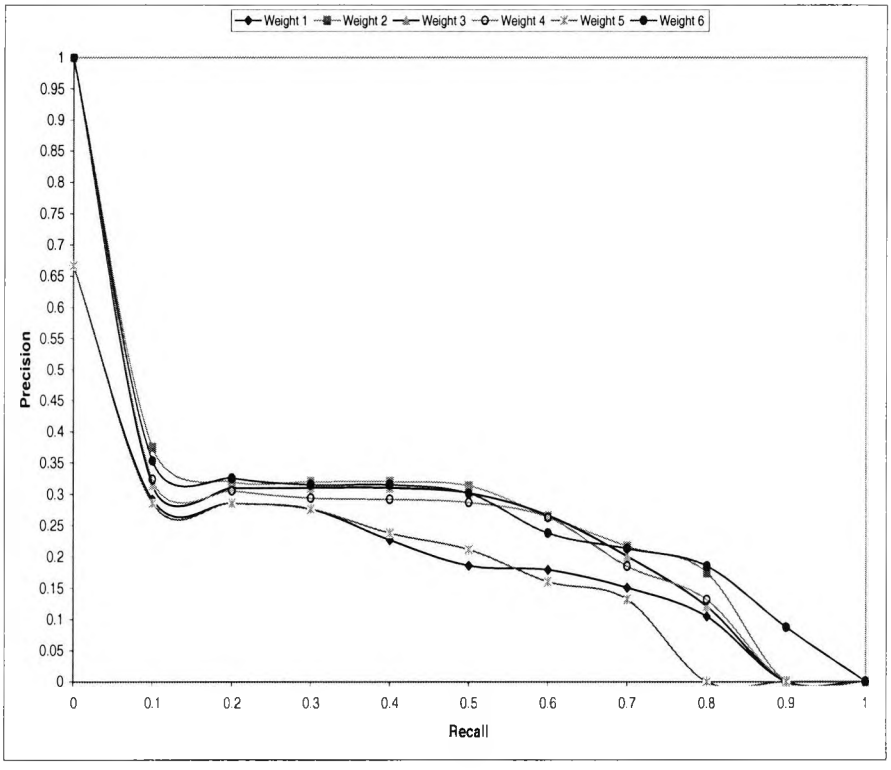


Figure 8.13: Precision-recall Curves of the Runs from Six Different Compound Unit Weighting Function for Topic 42 (k_d=15)

BM26 makes only a small improvement for $Weight_1$ and $Weight_2$. The improvement made by BM26 for $Weight_4$ is bigger than the improvement for $Weight_1$ and $Weight_2$.

From the TREC-6 experimental results on the character approach we can observe: first BM26 produces the better results than BM25 on most of tested topics no matter which compound unit weighting method is used; second, BM26 produces best results in terms of average precision on 14 of the 26 TREC-6 topics by setting k_d to 15 and 20; third, BM26 can make only a small improvement for $Weight_2$, but can make a lot of improvement for $Weight_3$ and $Weight_4$.

Topic	BM25 $k_d = 0$	BM26 $k_d = 2$	BM26 $k_d = 6$	BM26 $k_d = 8$	BM26 $k_d = 10$	BM26 $k_d = 15$	BM26 $k_d = 20$	BM26 $k_d = 50$
Topic 29		best						
Topic 30							best	
Topic 31	best							
Topic 32		best						
Topic 33	best							
Topic 34						best		
Topic 35					best			
Topic 36							best	
Topic 37			best					
Topic 38						best		
Topic 39					best			
Topic 40						best		
Topic 41							best	
Topic 42							best	
Topic 43								best
Topic 44						best		
Topic 45						best		
Topic 46			best					
Topic 47						best		
Topic 48						best		
Topic 49			best					
Topic 50							best	
Topic 51								best
Topic 52							best	
Topic 53							best	
Topic 54	best							

Table 8.18: Comparison of BM25 and BM26 for Character Approach by Using $Weight_2$ in terms of the Best Average Precision

Compound	Single	Number of Topics
$Weight_1$	BM25	4
	BM26	22
$Weight_2$	BM25	3
	BM26	23
$Weight_3$	BM25	2
	BM26	24
$Weight_4$	BM25	2
	BM26	24
$Weight_5$	BM25	2
	BM26	24

Table 8.19: Number of Topics for Character Approach in terms of the Best Average Precision

8.3.2 Analyses of Word Approach on the TREC-6 Dataset

By using $Weight_2$, the comparison results for the word approach in terms of which single unit weighting methods produce the best average precision value are shown in Table 8.20. The results show that single unit weighting function BM25 ($k_d = 0$) produces the best results only on topic 33. BM26 ($k_d > 0$) produces the best results on 25 of the 26 TREC-6 topics. Table 8.21 also give some more comparison results for BM25 and BM26 by using compound unit weighting methods $Weight_1$, $Weight_3$, $Weight_4$ and $Weight_5$.

From the above TREC-6 experimental results on the word approach we can observe: first, BM26 produces better results than BM25 on almost all the tested topics no matter which compound unit weighting method is used; second, BM26

Topic	BM25 $k_d = 0$	BM26 $k_d = 2$	BM26 $k_d = 6$	BM26 $k_d = 8$	BM26 $k_d = 10$	BM26 $k_d = 15$	BM26 $k_d = 20$	BM26 $k_d = 50$
Topic 29					best			
Topic 30							best	
Topic 31							best	
Topic 32				best				
Topic 33	best							
Topic 34							best	
Topic 35					best			
Topic 36							best	
Topic 37				best				
Topic 38							best	
Topic 39							best	
Topic 40							best	
Topic 41								best
Topic 42								best
Topic 43								best
Topic 44							best	
Topic 45							best	
Topic 46					best			
Topic 47								best
Topic 48							best	
Topic 49							best	
Topic 50								best
Topic 51								best
Topic 52							best	
Topic 53							best	
Topic 54		best						

Table 8.20: Comparison of BM25 and BM26 for Word Approach by Using $Weight_2$ in terms of the Best Average Precision

Compound	Single	Number of Topics
$Weight_1$	BM25	3
	BM26	23
$Weight_2$	BM25	1
	BM26	25
$Weight_3$	BM25	2
	BM26	24
$Weight_4$	BM25	2
	BM26	24
$Weight_5$	BM25	2
	BM26	24

Table 8.21: Number of Topics for Word Approach in terms of the Best Average Precision

produces best results in terms of average precision on 13 TREC-6 topics by setting k_d to 20 and on 6 TREC-6 topics by setting k_d to 50. We can conclude from this result that better results can be obtained for most of the TREC-6 topics by setting k_d to a bigger value such as 20 or 50. It seems that BM26 works more effectively for the word approach by setting k_d to a bigger value compared to the character approach. This can also explain why word-based approaches seem to be more sensitive to single unit weighting functions.

8.3.3 Detailed Analyses of Some Topics

In this section, we analyse six topics. For each of these six topics, we chose eight runs from the character approach and eight runs from the word approach to do a comparison. The eight runs from the character approach are $T6c2.kd0$, $T6c2.kd2$,

$T6c2.kd6$, $T6c2.kd8$, $T6c2.kd10$, $T6c2.kd15$, $T6c2.kd20$ and $T6c2.kd50$. The eight runs from the word approach are $T6w2.kd0$, $T6w2.kd2$, $T6w2.kd6$, $T6w2.kd8$, $T6w2.kd10$, $T6w2.kd15$, $T6w2.kd20$ and $T6w2.kd50$.

Topic 29

Topic 29 is about building the Information Super Highway. A relevant document should discuss building the Information Super Highway, including any technical problems, problems with the information infrastructure, or plans for use of the Internet by developed or developing countries. Table 8.22 presents the detailed data of average precisions over topic 29 for the character approach runs with eight single unit weighting methods using $Weight_2$. Table 8.23 presents some more detailed data of average precision over topic 29 for the eight word approach runs using $Weight_3$. In terms of average precision, the character approach run $T6c2.kd2$ (0.7292) performed 3.70% better than the run $T6c2.kd0$ (0.7032) and the word approach run $T6w2.kd10$ (0.6772) performed 10.28% better than the run $T6w2.kd0$ (0.6141). BM26 can make more improvement for word approach system than character approach system.

Recall	BM25 T6c2.kd0	BM26 T6c2.kd2	BM26 T6c2.kd6	BM26 T6c2 kd8	BM26 T6c2.kd10	BM26 T6c2.kd15	BM26 T6c2.kd20	BM26 T6c2.kd50
0.00	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.10	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.20	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.30	0.9667	0.9655	0.9667	0.9655	0.9655	0.9667	1.0000	0.9730
0.40	0.9167	0.9250	0.9362	0.9388	0.9400	0.9400	0.9362	0.9487
0.50	0.9057	0.9038	0.9038	0.9388	0.9400	0.9400	0.9200	0.8596
0.60	0.7436	0.7534	0.7971	0.8116	0.8116	0.8358	0.8462	0.6250
0.70	0.4776	0.6882	0.5818	0.5614	0.5565	0.4672	0.4049	0.2896
0.80	0.3596	0.3967	0.3946	0.3744	0.3395	0.3160	0.2944	0.0928
0.90	0.1358	0.2097	0.2448	0.2857	0.2562	0.1436	0.0000	0.0000
1.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
AP	0.7032	0.7292	0.7283	0.7234	0.7169	0.7063	0.6972	0.6335

Table 8.22: Recall-Level Precision for Topic 29 Character Approach Using $Weight_2$ Method

We also choose a relevant document "CB032021-BCW-1502-401" which is shown in Figure 8.14 for our analysis. This document contains a very short but highly relevant passage. The document is about the treatments that a patient with a rare disease received, and contains a short relevant passage describing how the Internet was used to link different hospitals together. Table 8.24 presents the ranking positions of a relevant document "CB032021-BCW-1502-401" for character and word

Recall	BM25 T6w2.kd0	BM26 T6w2.kd2	BM26 T6w2.kd6	BM26 T6w2.kd8	BM26 T6w2.kd10	BM26 T6w2.kd15	BM26 T6w2.kd20	BM26 T6w2.kd50
0.00	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.10	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.20	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.30	0.9355	0.9355	0.9655	0.9688	1.0000	1.0000	1.0000	1.0000
0.40	0.9048	0.9091	0.9048	0.9024	0.9250	0.9487	0.9487	0.8605
0.50	0.8545	0.8070	0.7667	0.7231	0.7419	0.7273	0.7500	0.6216
0.60	0.3986	0.4783	0.6875	0.6395	0.5914	0.5446	0.5140	0.4508
0.70	0.3478	0.3575	0.4248	0.4672	0.4706	0.4507	0.4156	0.3459
0.80	0.2450	0.2580	0.3067	0.3274	0.3596	0.3544	0.3303	0.1587
0.90	0.0891	0.1488	0.1888	0.1916	0.1966	0.1790	0.1885	0.0000
1.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
AP	0.6141	0.6377	0.6701	0.6758	0.6772	0.6709	0.6598	0.5985

Table 8.23: Recall-Level Precision for Topic 29 Word Approach Using $Weight_2$ Method

approaches by using eight different single unit weighting functions.

k_d	character approach	word approach
$k_d = 0$	350	> 1000
$k_d = 2$	283	> 1000
$k_d = 6$	205	> 1000
$k_d = 8$	191	> 1000
$k_d = 10$	179	815
$k_d = 15$	165	718
$k_d = 20$	161	648
$k_d = 50$	170	571

Table 8.24: Ranking Positions of a Relevant Document Using Different Single Unit Weighting Methods for Topic 29

From Table 8.24 we can see that BM26 indeed makes positive contribution to topic 29. The ranking position of the relevant document "CB032021-BCW-1502-401" moves forward when more correction factor weight is added in the single unit weighting function by setting the parameter k_d to a high value. However, for the word approach, the influence of the parameter k_d can be viewable only at a higher value level such as $k_d \geq 10$.

The BM25 function includes some form of document length normalization when calculating the score of a document. Long documents that contain only a short relevant passage generally receive lower scores. For example, for topic 29, the relevant document "CB032021-BCW-1502-401" is assigned a high score by using BM26. However, this relevant document received a low score by using BM25 and was ranked at a position of 350 for the character approach and a position out of the top 1000 documents for the word approach since the rest of the document was not relevant. The ability to identify highly relevant sections inside a long document is

<DOC>
<DOCID> CB032021.BCW (1502) </DOCID>
<DOCNO> CB032021-BCW-1502-401 </DOCNO>
<DATE> 1995-07-21 18:10:49 (G) </DATE>
<TEXT>
<headline> 通讯：献爱心万众齐相助 克怪病晓渡获新生 </headline>

新华社北京七月二十一日电（记者杨国强、宗焕平）身患“怪病”引起广泛社会关注的山东农家姑娘杨晓渡今天兴高采烈地告诉记者，她已基本痊愈，再接受一段时间的康复治疗，就可在两、三个月后出院，重返山东老家上学。

“我好想回去上学啊！我的同学今年升初二了，昨天我还写了两封信去祝贺他们哩！”晓渡的语气中透着重获新生的欢快。这位甜美的山东小姑娘长着一双水灵的大眼睛和一张讨人喜欢的脸，个头比去年刚来时长高了三厘米，达到一点四九米了，体重也增加了六公斤。“没有那么多关心爱护我的爷爷、奶奶、叔叔、阿姨、哥哥、姐姐、弟弟妹妹们，我就没有今天，我长大后一定不叫他们失望。我现在正练习用左手写字，学好知识，将来为社会服务。”

今年十三岁的杨晓渡去年刚以优异成绩考上县重点中学，就发现患上了一种双肢溃烂不止的怪病，左右手肌肉迅速溃烂脱落，坏死组织不断蔓延、液化，形成出血、化脓、恶臭现象，严重地威胁她幼小的生命。父母亲跑遍了当地医院均无法确诊，抱着最后一线希望，去年十一月带她来到了北京军区总医院。

军区总医院立即成立了以副院长杨贵学为首的专家小组，首都十八家医院的几十位中西医专家都前来参加会诊。卫生部部长陈敏章来了，北京军区司令员李来柱上将、政委谷善庆上将也来了。他们为的是同一个目的：想尽一切办法把晓渡的病治好，还她一个充满希望的童真！全国成千上万的人们爱心涌动，有的捐款捐物，有的献出私家祖传秘方，他们中有老有少，有男有女，捐款从十元至五百元不等，总额达到八十六万元。合演了一出携手与病魔抗争的动人故事。陈敏章部长对这充满人间温馨的“杨晓渡现象”作了高度概括：医务人员们不计条件不计报酬地竭尽全力救治晓渡姑娘，这反映了卫生界救死扶伤、为人民服务的医德；而各界的人们纷纷伸出援助之手，又是今天人们真情汇成的暖流。

>
>国际医疗界也通过全球信息高速公路“ＩＮＴＥＲＮＥＴ”了解晓渡的病情发展，美国、英国、新加坡、香港等国家和地区的医
>学专家纷纷献计献策。
>

杨贵学告诉记者：“从医学上来讲，晓渡患的是一种复杂、少见、多种细菌混合感染引发的坏死性病变，这同她的免疫功能低下有关。目前，她的感染已完全得到控制，伤口也愈合了。虽然她的右前臂截掉了，左手两个拇指丧失功能，但她不久将装上美容性假肢，等长大后再装功能性假肢，基本能适应今后的学习和生活。现在看来，她的怪病复发的可能性极小。”

创伤外科主任医师刘树清介绍说：“针对晓渡的病，我们采取了多种有效治疗措施，包括寻找和控制病源菌，中西医结合提高病人的体质，使用先进的‘泰能’抗菌素、清创植皮等。实践证明这些措施的效果是不错的。目前，我们正对她进行最后的功能康复治疗。同时，我们正同流行病研究所合作，对她的病源进行最后鉴定，以利今后的临床治疗和科研。”

对于社会捐助的八十六万元钱，医院一直持十分严格、严肃的态度，目前医院对捐款一分未动。据了解，一个“杨晓渡救治金”管理委员会已经成立，由杨晓渡的监护人、她家乡当地政府代表、捐款单位和个人代表、北京和山东新闻界代表、北京军区总医院代表共同组成，全权负责救治金的管理和审批使用。记者访问了不少捐款人，他们都希望医院留取一部份医疗费，剩余款项用来救治其他有困难的病人。

晓渡的怪病终于被控制住了。当晓渡还在病床上时，一位探望者曾这样留言：“世界之所以如此灿烂，因为世间有爱，爱是人间最伟大的力量，你拥有了这人间最宝贵的财富——千千万万颗爱心！相信不久的将来，昔日的笑容会重浮你的脸上。”这位爱心使者的留言现在已被验证了，晓渡又笑了，她笑得很开心。（完）
</TEXT>
</DOC>

Figure 8.14: A Relevant Document for Topic 29

a strength of using BM26 as the single unit weighting function. This also suggests that passage retrieval would have a similar effect.

Further Evidence

The above analysis on topic 29 indicates that the ranking position of a relevant document “CB032021-BCW-1502-401” generally moves forward as the value for parameter k_d in the single unit weighting function increases. However, the example documents of Topic 29 were all way down the ranking. To determine whether this statement holds for documents much further up the ranking, we chose five more topics for further analysis. These five topics are topic 39, 40, 44, 45 and 50. Six relevant documents for these topics are also chosen, which are shown in Table 8.25.

Table 8.26 presents the ranking positions of these six relevant documents for the character and word approaches by setting different values for k_d. The table

	document number	topic
A	$CB016023 - BFJ - 372 - 225$	$topic39$
B	$CB057031 - BFW - 1291 - 88$	$topic40$
C	$CB002026 - BFW - 933 - 395$	$topic44$
D	$pd9108 - 3350$	$topic45$
E	$CB043030 - BFW - 622 - 134$	$topic50$
F	$pd9109 - 656$	$topic50$

Table 8.25: Six Relevant Documents for Topic 39, 40, 44, 45 and 50

shows that the BM26 function makes positive contribution to the results for these five topics. The ranking positions of all the selected relevant documents move forward by increasing the value for k_d, even though this is not the case for *all* relevant documents

	A (topic 39)		B (topic 40)		C (topic 44)		D (topic 45)		E (topic 50)		F (topic 50)	
k_d	char	word	char	word	char	word	char	word	char	word	char	word
0	17	25	12	24	26	32	47	62	32	49	38	37
2	15	25	8	15	26	32	46	58	30	49	35	36
6	15	21	5	11	21	30	45	53	29	46	33	35
8	13	20	5	10	13	28	44	53	27	44	31	33
10	13	18	5	9	12	28	43	51	25	43	30	31
15	8	16	4	8	12	20	36	48	21	41	23	26
20	7	12	4	6	11	17	33	41	19	35	20	25
50	2	6	6	6	7	10	15	22	12	25	11	18

Table 8.26: Ranking Positions of Relevant Documents Using Different Single Unit Weighting Methods for Topic 39, 40, 44, 45 and 50

8.4 Comparisons with Other Chinese Systems

The probabilistic view of information retrieval has inspired a number of very different approaches, models, methods and techniques. It is also true that many of the specific methods discussed in this thesis have been used in the context of other, non-probabilistic (or not explicitly probabilistic) approaches. Many comparisons could be made, at the level of theories, models, techniques, experimental results, or whatever, between the ideas discussed here and those reported by other researchers.

In this section, we make a selection of such comparisons, concentrating on some important issues Chinese text retrieval, and on ideas which may shed light on the foregoing discussions for Chinese text retrieval. The Chinese retrieval systems we

choose for our comparisons are the TREC-5 and TREC-6 Chinese retrieval systems. Since most of Chinese systems in TREC-6 are the same as the systems in TREC-5, we will only focus on the retrieval system developed at National Taiwan University [19] as an example of TREC-5 system. We only chose some of TREC-6 participating systems for comparisons because these systems may be closer to our reseach work. The Chinese systems who took part in TREC Chinese experiments generally explored the use of words vs. n-grams and methods of manually modifying queries. Some work was also done on retrieval methods particularly appropriate to Chinese retrieval.

8.4.1 TREC-5 Chinese Systems

National Taiwan University

A lot of experiments have been conducted on the TREC-5 dataset by the National Taiwan University. Some of the evaluation results are very interesting. The word segmentation methods used in the experiments are maximum matching (forward and backward)[14], minimum matching (forward and backward) and statistical methods (such as mutual information statistics). The maximum matching is to group the longest initial sequence of characters that matches a dictionary entry as a word and the minimum matching is to treat the shortest initial sequence of characters that matches a dictionary entry as a word. The dictionary used for the experiments contains 138,955 entries [15], including words, phrases, compounds, idioms, proper names, and so on. One group of queries is automatically constructed and the other group of queries is manually reformulated. The iterative process for constructing a manual query is as follows [19]: (i) Do a trial run using the current query; (ii) Examine the top-ranked document and manually select the terms that seem to be promising from the top documents; (iii) Add the chosen terms from the previous step to the current query and assign weights manually to the new terms to form a new query. The automatic process for constructing a query is not clear.

More information about the methods used in the experiments such as indexing

[14]forward starts from the beginning of the phrase and backward starts from the end of the phrase

[15]about 43% of the entries in this dictionary were manually selected from the TREC-5 Chinese document collection

methods and segmentation methods etc. is given in Table 8.27. The statistical indexing process is: (i) Collect occurrence frequencies in the collection for all Chinese characters occurring at least once in the collection; (ii) Collect occurrence frequencies in the collection for all Chinese bigrams occurring at least once in the collection; (iii) Compute the mutual information for all Chinese bigrams: $I(x,y) = log_2(p(x,y)/(p(x) * p(y))) = log_2((f(x,y) * N)/(f(x) * f(y)))$ [16]; (iv) If $I(x,y) >> 0$, x and y have strong relationship; if $I(x,y) \approx 0$, x and y have no relationship; if $I(x,y) << 0$, x and y have complementary relationship. (v) Apply Richard's algorithm to segment the text into words. The results for automatically constructed and manually constructed queries are presented in Table 8.28 and Table 8.29 respectively. In terms of average precision, the results from manually constructed queries are much better than the corresponding results from automatically constructed queries.

	index file	indexing terms	segmentation/index method	dictionary and stop-list used
1	unigram	unigram	unigram	none
2	bigram	bigram	bigram	stop-list only
3	trigram	trigrams	trigram	stop-list only
4	mi	bigrams, unigrams	statistical with mutual information	stop-list only
5	max (f)	word, phrase	maximum matching (forward)	both
6	max (b)	word, phrase	maximum matching (backward)	both
7	min (f)	word, phrase	minimum matching (forward)	both
8	min (b)	word, phrase	minimum matching (backward)	both

Table 8.27: Index Files Used by National Taiwan University

Recall	unigram	bigram	trigram	mi	max (f)	max (b)	min (f)	min (b)
0.00	0.7751	0.7504	0.6962	0.7696	0.8000	0.7966	0.7404	0.7265
0.10	0.5609	0.6241	0.5006	0.6500	0.6465	0.6414	0.5543	0.5611
0.20	0.4076	0.5243	0.3600	0.5355	0.5283	0.5028	0.4336	0.4432
0.30	0.3400	0.4773	0.2932	0.4705	0.4308	0.4518	0.3595	0.3734
0.40	0.2904	0.4375	0.2546	0.4324	0.3841	0.4085	0.3049	0.3245
0.50	0.2486	0.3864	0.2153	0.3872	0.3455	0.3671	0.2569	0.2903
0.60	0.2050	0.3295	0.1815	0.3346	0.2947	0.3131	0.2216	0.2351
0.70	0.1576	0.2749	0.1586	0.2843	0.2439	0.2678	0.1657	0.1912
0.80	0.0982	0.2173	0.1142	0.2353	0.1891	0.2017	0.1221	0.1217
0.90	0.0300	0.1241	0.0581	0.1378	0.1051	0.1105	0.0819	0.0778
1.00	0.0031	0.0108	0.0091	0.0208	0.0282	0.0341	0.0197	0.0118
Average Precision	0.2609	0.3677	0.2405	0.3744	0.3558	0.3465	0.2738	0.2862
	-26.67%	3.34%	-32.40%	5.23%	baseline	-2.61%	-23.04%	-19.56%
Relevant Retrieved	1614	2017	1735	1948	1910	1825	1731	1693

Table 8.28: Average Precision of Automatic Queries Using Different Segmentation Methods

From the experimental results, Chen et al [19] made some conclusions as follows: first, the average precision for the automatic unigram (0.2609) and the set

[16] See section 2.3.3 for details

Recall	unigram	bigram	trigram	mi	max (f)	max (b)	min (f)	min (b)
0.00	0.8624	0.8309	0.7008	0.8372	0.8551	0.8433	0.8154	0.7961
0.10	0.6880	0.6938	0.4720	0.6831	0.7304	0.7059	0.6590	0.6372
0.20	0.5757	0.6242	0.3464	0.6298	0.6429	0.6378	0.5679	0.5279
0.30	0.5286	0.5684	0.3005	0.5824	0.5787	0.5716	0.5093	0.4841
0.40	0.4756	0.5119	0.2652	0.5292	0.5105	0.5074	0.4570	0.4448
0.50	0.4263	0.4598	0.2349	0.4560	0.4583	0.4575	0.4060	0.4036
0.60	0.3829	0.4041	0.2082	0.4054	0.4146	0.4111	0.3544	0.3660
0.70	0.3404	0.3551	0.1528	0.3631	0.3514	0.3487	0.2873	0.3098
0.80	0.2809	0.3064	0.1326	0.3116	0.2894	0.2833	0.2253	0.2482
0.90	0.1859	0.2261	0.0868	0.2285	0.2314	0.2327	0.1544	0.1494
1.00	0.0356	0.0823	0.0122	0.0625	0.0499	0.0674	0.0340	0.0344
Average Precision	0.4203	0.4522	0.2397	0.4533	0.4519	0.4481	0.3937	0.3904
	-6.99%	0.06%	-46.95%	0.31%	baseline	-0.84%	-12.87%	-13.61%
Relevant Retrieved	2036	2088	1711	2064	2033	2020	2022	2008

Table 8.29: Average Precision of Manually Expanded Queries Using Different Segmentation Methods

of manually reformulated queries (0.4203) are good in comparison with the results of the dictionary-based maximum matching (0.3558); second, bigram indexing (0.3577 for automatic and 0.4522 for manual) and the mutual information segmentation (0.3477 for automatic and 0.4533 for manual) produce better performance in comparison to unigram indexing (0.2609 for automatic and 0.4203 for manual); bigram indexing and mutual information-based segmentation outperform the popular dictionary-based maximum matching. All these conclusions are consistent to what we have obtained with our Chinese Okapi system on the TREC-5 and TREC-6 datasets.

8.4.2 TREC-6 Chinese Systems

Cornell University

Cornell again in TREC-6 approached Chinese retrieval with no Chinese expertise but a very good retrieval system – the SMART system. They approached the task by using character based retrieval augmented with character bigrams [11]. For English, automatic query expansion using pseudo relevance feedback has traditionally been very useful in the ad-hoc task. In this approach, a set of documents is initially retrieved in response to a user query; the top ranked documents are assumed to be relevant (without any intervention from user); low-ranked documents are optionally assumed to be non-relevant; and these documents are then used in the Rocchio feedback method to expand the query.

Official Chinese results from Cornell are given as follows in Table 8.30. The first

Run	Index Method	Average Precision	Total Rel Retrieved	R Precision	Precision 100 docs
Cor6CH1sc	character only	0.5547	2765	0.5301	0.5162
Cor6CH2ns	character only	0.5552	2763	0.5369	0.5185

Table 8.30: Chinese TREC-6 Automatic Ad-hoc (Cornell)

official Chinese run Cor6CH1sc follows exactly the same procedure as the English run, except it was decided to treat the two-character phrases as being the base concepts of the SuperConcepts instead of the single terms as in the English term. The second official run, Cor6CH2ns, is exactly the same as Cor6CH1sc, except no expansion single characters were added. Instead of adding 15 single terms and 15 phrases to the original query from the top 20 initially retrieved documents, only 25 phrases were added. Both official runs from Cornell by using character indexing methods did very well compared to other TREC-6 Chinese results, even if the expansion on single characters no longer gave a performance gain.

Institute of Systems Science

The Institute of Systems Science carried out only automatic runs [17] by combining both bigram approaches and word based approaches at TREC-6. They looked at bigram based indexing vs. segmented based indexing for the Chinese ad-hoc task. They also investigated various methods for merging the different result sets to see the contributions of the two indexing methods. Words and phrases were discovered using a greedy and short segmentation algorithm. The merging method used was to merge based on the raw scores of the segmented based method and the bigram based method, removing the lower scoring duplicate documents. Table 8.31 summarizes the results obtained in the official submitted runs for TREC-6.

Run	Index Method	Average Precision	Total Rel Retrieved	R Precision	Precision 100 docs
iss97CbD	bigram	0.5646	2802	0.5515	0.5104
iss97CmD	merge	0.4903	2723	0.4941	0.4692
iss97CsD	word	0.4709	2619	0.4689	0.4615

Table 8.31: Chinese TREC-6 Automatic Ad-hoc (ISS)

[17] How the topic processing is performed automatically is not clear

The results in Table 8.31 show that the merged results are worse than the bigram results. However there is also consistency: the segmented approach was worst in all cases, the bigram was best in all cases, and the merged approach was between them. The merged numbers are closer to the segmented numbers, which may indicate that the merging algorithm favoured the segmented approach.

Queens College, CUNY

For TREC-6 Chinese ad-hoc experiments, Queens continue to use two-stage retrieval with pseudo-feedback from top-ranked unjudged documents and employ a combination of representation (character, bigram and short-word) strategy for indexing. Queens continue to use the short-word[18] segmenter developed in TREC-5 to segment Chinese texts. This procedure with a mixture of expert knowledge and statistical processing involves four steps: (i) lexicon[19] look-up using longest match to segment input texts into smaller chunks; (ii) simple language rules[20] to segment chunks into short-word candidates; (iii) discover new short-words based on frequency filtering; (iv) expand the initial lexicon with the new short-words and re-process collection. TREC-5's initial lexicon of 2K has been enlarged to 27K entries to provide better coverage of common short-words. After adding new words discovered from the collection, the final lexicon size for TREC-6 is about 43K. The words detected in the above 4-step procedure are used for document and query representation directly. The Chinese runs submitted for evaluation are automatic.

Queens conjecture that bigrams and short-word indexing with characters may complement each other, and they have used a combination strategy in these TREC-6 experiments. The collection was indexed in two ways: bigram and short-word with character representation. For each query, two separate retrievals were performed using the two representations, and the resultant document lists are combined using equal weights. The bigram retrieval composing this result has by itself an average precision of 0.5755 and relevant documents retrieved of 2735. Similarly, the short-word indexing with character alone has 0.6031 and 2791 values for

[18]Short-word means words of 2 to 3 characters long (with some proper names of 4 characters also)

[19]It is a manually created lexicon list of about 2,000 items. Each item is tagged as useful, useless (stopword), numeric, punctuation and a few other codes [61]

[20]These rules are also manually determined [61]

these measures. They combine to give values of 0.6263 and 2795 respectively, an improvement of about 4% in precision from the better one, and a few more in the relevant documents retrieved. The comparison results is shown in Table 8.32 and the official submitted Chinese results for Queens Chinese retrieval are tabulated in 8.33. In almost all cases of Queens Chinese experiments, two-stage retrieval improves over one stage only, ranging from 0% to nearly 32% in the case of titles using short-word with character representation.

Run	Index Method	Average Precision	Total Rel Retrieved
pirc7Cb	bi-gram	0.5755	2791
pirc7Cs	characters & short-word	0.6031	2735
pirc7Ca	combination	0.6263	2795

Table 8.32: Chinese TREC-6 Automatic Ad-hoc (Queens)

Run	Average Precision	Total Rel Retrieved	R Precision	Precision 100 docs
pirc7Ca	0.6263	2795	0.5809	0.5542
pirc7Cd	0.5423	2674	0.5175	0.5035
pirc7Ct	0.4755	2547	0.4630	0.4327

Table 8.33: Chinese TREC-6 Automatic Ad-hoc (Queens)

University of California, Berkeley

Berkeley believes that the coverage of the dictionary over the collection to index can have significant impact on the retrieval effectiveness of a Chinese text retrieval system that uses a dictionary to segment text. In TREC-5, Berkeley combined a dictionary found on the web and entries consisting of words and phrases extracted from the TREC-5 Chinese collections to create a dictionary of about 140,000 entries and used the dictionary to segment the Chinese collection [42]. This dictionary is the biggest in size of all the TREC-5 Chinese systems, yet the Berkeley team found that this dictionary still did not contain many important indexing terms, in particular names (such as personal names, transliterated foreign names, company names, university and college names, research institutions and so on). In the Chinese track of TREC-6, Berkeley focused on automatic and semi-automatic augmentation of the Chinese dictionary which they used to segment the

Chinese collection and further augmented their dictionary with 10,000 entries.

Berkeley submitted two runs, named BrklyCH3 and BrklyCH4 respectively, for the Chinese track. BrklyCH3 is the run using the original long queries with automatic query expansion and BrklyCH4 is the run based on the manually reformulated queries. For both runs, the collection was segmented using the dictionary-based maximum matching method. For the automatic run BrklyCH3, an initial retrieval run was carried out to produce a ranked list of documents, then 20 new terms were selected from the top 10 ranked documents for each query. The selected terms are those that occur most frequently in the top 10 documents in the initial ranked list. The chosen terms were added to the original long queries to form the expanded queries [35]. A final run was carried out using the automatically expanded queries to produce the results in BrklyCH3. Table 8.34 summarizes the results obtained in the official submitted runs for TREC-6. The process for manual query expansions is as follows [42]: (i) add new words; (ii) change weights (frequency) of words; (iii) add negative words. The process iterated several times before they obtained a final version. The retrieval result of manual queries improved 40% over the automatic run for TREC-6.

Run	Index Method	Average Precision	Total Rel Retrieved	R Precision	Precision 100 docs
BrklyCH3	word	0.5291	2551	0.5252	0.5296
BrklyCH4	word	0.5586	2573	0.5496	0.5427

Table 8.34: Chinese TREC-6 Automatic Ad-hoc (Berkeley)

University of Montreal

The Montreal effort concentrated on comparing word-based and bigram-based indexing for Chinese text retrieval. For the word-based approach, the Montreal system used dictionary-based word segmentation because no training text from the Chinese TREC collection was available. The word dictionary contains 87,600 entries and the maximum-matching algorithm is used for indexing [73]. In order to improve the retrieval performance for word-based approaches, three methods have been used:

First, as we know, there is no clear definition of words in Chinese. There are a number of long words/phrases that are composed of shorter words in many Chinese dictionaries. In order to avoid this problem, Montreal system's segmentation process extracts all the possible compound words (composed of two characters or more) from a given character string. So for the sequence "电脑网络", three words will be extracted: "电脑网络", "电脑" and "网络". In the official runs submitted from Montreal, this approach is used: that is all the compound words included in a long word are also extracted.

Second, in addition to a word dictionary of 87,600 entries, the Montreal system also dealt with some special character sequences which may be considered as words in Chinese. These sequences include: nominal pre-determiner and affix structure. A set of rules was set up for their recognition. For example, "２０００年" (year 2000).

Third, a normalization was performed when the segmentation method indexed numbers in the Chinese TREC collection. For example, "the year 2000" may be written in Arabic numbers which may be encoded in ASCII or in Chinese codes. It may also be written in Chinese numbers or as a mixture of Chinese and Arabic numbers

Comparison results from University of Montreal [73] are given as follows in Table 8.35. All these runs are automatic. However, it is not clear that how the topics were processed automatically. The star sign inside the table means the data is not available

Run	Index Method	Average Precision	Total Rel Retrieved	R Precision	Precision 100 docs
UdeMbi	bigram	0.4467	2709	0.4655	0.4408
UdeMseg	word	0.4524	2668	0.4748	0.4662
UdeMchar	single character	0.4615	*	*	*

Table 8.35: Chinese TREC-6 Automatic Ad-hoc (Montreal)

8.4.3 Discussion

We could classify the systems participating in TREC-6 experiments according to the performance of different indexing methods. Detailed information is given

in Table 8.36. The first one is the case that the performance of character and bi-gram indexing approaches are better than the performance of word indexing approaches (e.g. City University). The second one is the case that the performance of word indexing approaches are better than the performance of bi-gram indexing approach (e.g. Montreal). The third one is the case that the performance of short-word based with character indexing approaches are better than the performance of bi-gram and character indexing approaches (e.g. Queens College [21]). The topic processing methods for most automatic runs in TREC-5 and TREC-6 have been described in section 8.4.1 and 8.4.2. However, the topic processing methods for the remaining automatic runs are not clear.

The implication from Queens' best experiment results in Table 7.13 is that, if the document collection is indexed in short-word with character, the Chinese retrieval system will perform better in terms of average precision and R precision. This is based on the assumption that a small, good, manually selected lexicon has to be constructed first for the test collection and topics. If the retrieval system has to deal with a new test collection or new topics, this manually selected lexicon has to be modified.

Run	Index Method	Better Run	Combination
City	character & word	character better (5%)	No
Claritech	character only	character only	No
Cornell	character only	character only	No
ITI	character only	character only	No
ISS	character & word	character better (18%)	Combination not best
ETH	character only	character only	No
UMass	character & word	character better (2%)	No
Waterloo	character	user selected	Yes
Montreal	bi-grams & word	word better (1%)	No
MDS	bi-grams & word	word better (1%)	Combination best
Queens	bi-grams & short word	short word better (4%)	Combination best
Berkeley	word only	word only	No

Table 8.36: Comparing Character-based and Word-based Approaches

The Montreal team concentrated on improved word identification algorithms by using more sophisticated morphological analysis. The results of this approach was then compared to the bigram approach. The word based approach gave slightly

[21]The best experiment result comes from Queens College

better performance. The Montreal team also compared to the single character approach and found the single character approach produced best results among all these three indexing approaches [73].

As we can see, there are many factors that have to be taken into account in deciding which indexing method is best for Chinese text retrieval. But it is still the case that character or bi-gram approaches are comparable with any other individual technique and have the advantage of not requiring the difficult task of segmentation and constructing a small lexicon or large dictionary in order to employ word-based approaches (see Table 8.36). This is probably due to the greater semantic content of characters compared to any sub-word element in English and other European languages. So we still prefer to use single character or bi-gram indexing approaches for Chinese text retrieval.

8.5 Summary

In the past few years, most efforts in Chinese IR have been done on indexing. Chinese indexing approaches can generally be divided into character-based approaches (such as bigram and trigram indexing) and dictionary-based approaches (such as word-based approaches). Research conducted so far has found the following. First, n-grams (in particular bigrams) perform as well as, or even better than, words [74]. Kwok found that combining the results of short-word and character with bigram and character gave the best results [60]. Experimental results from the Berkeley group reported at the second NTCIR [76] workshop on Japanese and Chinese IR show that the bigram indexing outperforms the word-based indexing in Japanese retrieval. The bigram indexing is also reported to be highly effective in Chinese text retrieval [20]. Second, a better dictionary can increase IR effectiveness to some extent. However, the increase is very limited in comparison with the number of additional entries [75]. Increasing the size of the dictionary is not an effective way to increase Chinese IR performance. Third, the recognition of unknown words has a positive, but very small, impact on the IR performance [75, 146]. In this section, we will summarize our findings on the positive and negative factors that affect the performance of Chinese IR.

8.5.1 Positive Contribution Factors

We found in this research that three factors can provide positive contribution to the performance of retrieval. These three factors are different document processing methods (such as word vs character), different compound unit weighting methods and different single unit weighting (such as BM25 and BM26). The above three factors all contributed to improvement of the performance. Each factor can make a difference for the retrieval performance. In the following, we will discuss how much improvement is obtained by document processing, single unit weighting and compound unit weighting.

Recall	TREC-5 BM11 (word) city96c1	TREC-5 $Weight_3$ BM25 (word) T5w3.BM25	TREC-5 $Weight_3$ BM25 (char) T5c3.BM25	TREC-5 $Weight_2$ BM25 (char) T5c2.BM25	TREC-6 $Weight_3$ BM25 (word) T6w3.BM25	TREC-6 $Weight_3$ BM25 (char) T6c3.BM25	TREC-6 $Weight_2$ BM25 (char) T6c2.BM25	TREC-6 $Weight_2$ BM26 (char) T6c2.kd10
0.00	0.7547	0.7571	0.7764	0.7622	0.9557	0.9117	0.9150	0.9540
0.10	0.5823	0.6423	0.6243	0.6550	0.7902	0.7663	0.8009	0.8340
0.20	0.4868	0.5404	0.5507	0.5761	0.7093	0.7168	0.7442	0.7642
0.30	0.4245	0.4806	0.4987	0.5382	0.6594	0.6646	0.6945	0.7189
0.40	0.3739	0.4437	0.4458	0.4814	0.6099	0.6127	0.6594	0.6869
0.50	0.3416	0.4118	0.4247	0.4413	0.5503	0.5552	0.5879	0.6292
0.60	0.3017	0.3548	0.3591	0.3892	0.4587	0.4834	0.5201	0.5470
0.70	0.2320	0.2845	0.2711	0.3303	0.3698	0.3889	0.4239	0.4519
0.80	0.1687	0.2273	0.2236	0.2556	0.2670	0.3022	0.3326	0.3565
0.90	0.0915	0.1285	0.1408	0.1855	0.2567	0.1743	0.2343	0.2373
1.00	0.0076	0.0258	0.0266	0.0304	0.0073	0.0327	0.0399	0.0390
AP	0.3256	0.3762	0.3795	0.4126	0.4900	0.4967	0.5341	0.5603
	baseline	15.54%	16.55%	26.72%	baseline	1.37%	9.00%	14.35%
Rel	1891	2002	1986	2056	2542	2537	2637	2647

Table 8.37: Recall-Level Precision over the TREC-5 and TREC-6 Topics

The TREC-5 results in Table 8.37 show that the run $T5w3.BM25$ using BM25 performs 15.54% better than the run $city96c1$ using BM11. The character approach run $T5c3.BM25$ performs only 1.01% better than the word approach run $T5w3.BM25$. Both of these two runs $T5c3.BM25$ and $T5w3.BM25$ use $Weight_3$ as the compound unit weighting function and BM25 as the single unit weighting function. The character approach run using $Weight_2$ as compound unit weighting function performs 10.17% better than the character approach run using $Weight_3$. The experimental results on TREC-5 dataset clearly show that BM25 make a quite big improvement over the BM11 (15.54%) and the character approach does not make too much improvement over the word approach by using $Weight_3$ as compound unit weighting (only 1.01%). But using $Weight_2$ makes a pretty good improvement (10.17%) over the $Weight_3$ for the character approach.

The TREC-6 results in Table 8.37 also show that the character approach run

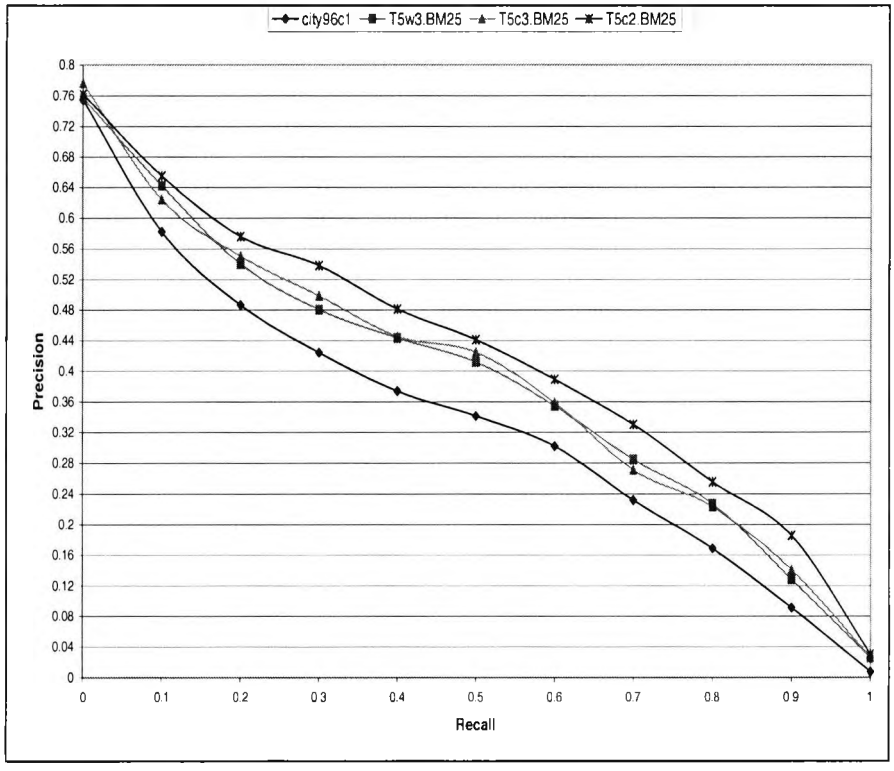


Figure 8.15: Precision-recall Curves for Four TREC-5 Runs city96c1, T5w3.BM25, T5c3.BM25 and T5c2.BM25

$T6c3.BM25$ performs only 1.37% better than the word approach run $T6w3.BM25$. Both of these two runs $T6c3.BM25$ and $T6w3.BM25$ use $Weight_3$ for compound unit weighting and BM25 for single unit weighting. The character approach run using $Weight_2$ as compound unit weighting function performs 7.63% better than the character approach run using $Weight_3$. The run $T6c2.kd10$ using BM26 performs 5.35% better than the run $T6c2.BM25$ using BM25. Both these two runs are character approach runs and use $Weight_2$ as compound unit weighting function. The experimental results on the TREC-6 dataset again confirm what we have found on the TREC-5 dataset that the character approach does not make too much improvement over the word approach by using $Weight_3$ as compound unit weighting (only 1.37%). Compound unit weighting function $Weight_2$ makes fairly good improvement over the $Weight_3$ on TREC-6 dataset (7.63%). For the character approach, the BM26 function makes 5.35% improvement over the BM25

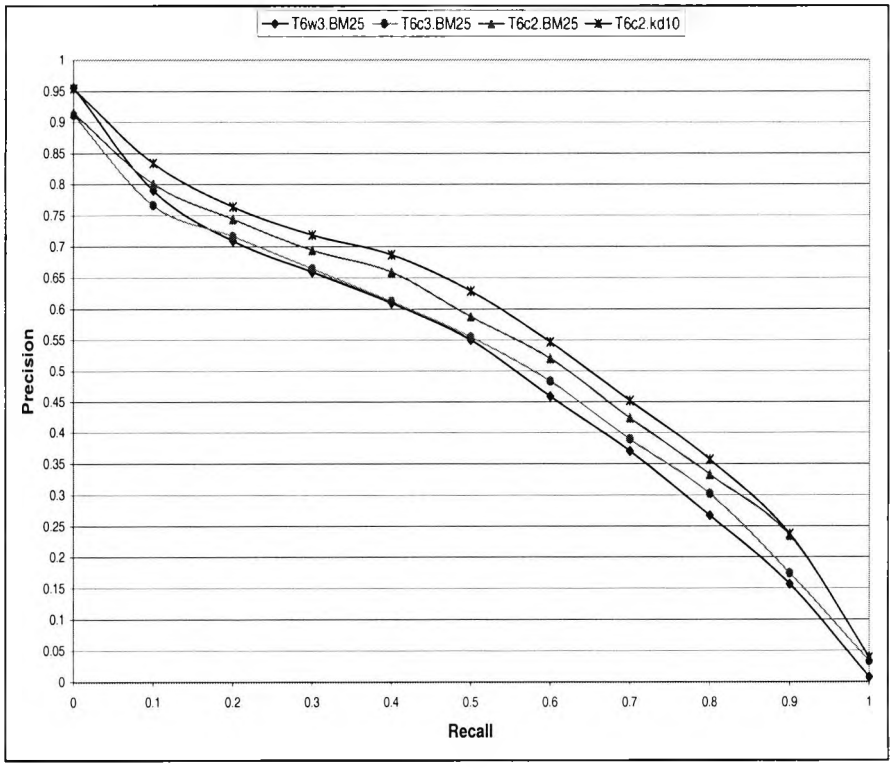


Figure 8.16: Precision-recall Curves for Four TREC-6 Runs T6w3.BM25, T6c3.BM25, T6c2.BM25 and T6c2.kd10

function.

For comparison purposes, the Precision-recall curves for the four TREC-5 runs and the four TREC-6 runs are shown in Figure 8.15 and Figure 8.16 respectively. The improvements on average precision for TREC-5 and TREC-6 datasets are shown in Figure 8.17. By using $Weight_3$ as compound unit weighting function, the indexing method factor does not affect the performance too much (see Table 8.37). However by using $Weight_2$ as compound unit weighting function, the character approach run $T5c2.BM25$ can make 9.30% improvement over the word approach run $T5w2.BM25$ on the TREC-5 dataset (see Table 8.38). For the TREC-6 dataset, the character approach run $T6c2.BM25$ using BM25 performs 8.40% better than the word approach run $T6w2.BM25$ using BM25 and the character approach run $T6c2.ld10$ using BM26 performs 7.56% better than the word approach run $T6w2.kd10$ using BM26 (see Table 8.38). The precision-recall curves

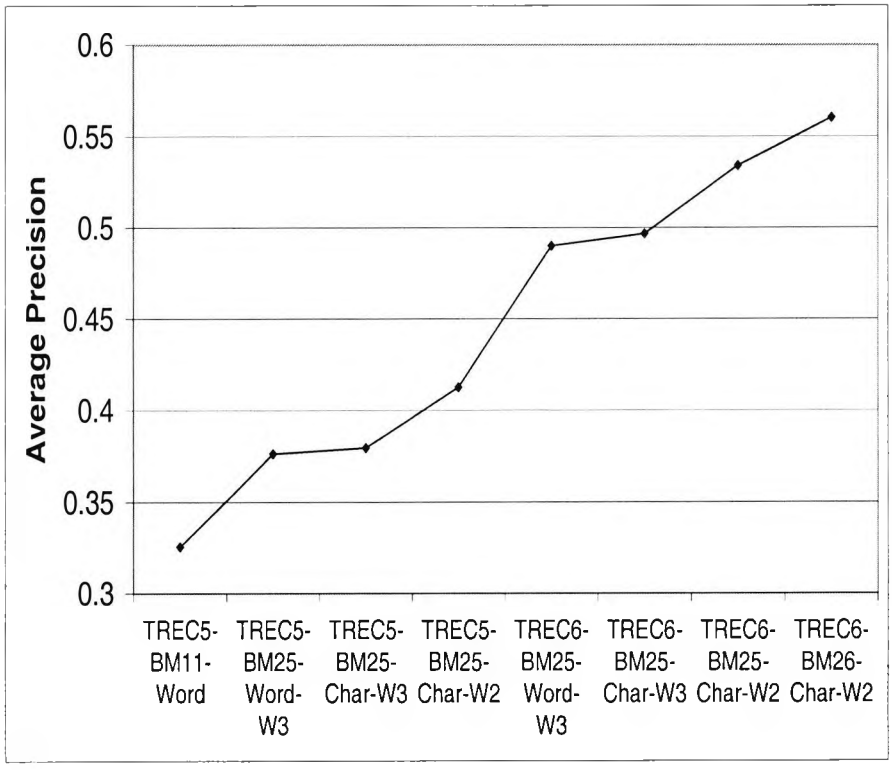


Figure 8.17: Improvements on Average Precision for TREC-5 and TREC-6 Datasets

for the runs T5w2.BM25 and T5c2.BM25 are shown in Figure 8.18. The improvements on average precision for character and word approaches by using $Weight_2$ are shown in Figure 8.19.

The above three factors that can affect the performance of retrieval all contributed to the good performance of our Chinese system. Only one factor does not change the performance too much. But all the three factors working together can make an obvious improvement. The word approach produces a higher value at the smallest recall level (i.e., 0.00) than that of the character approach. But the character approach produces higher values at the other recall levels compared to the word approach. BM26 can make a significant positive contribution to the quality of retrieval compared to BM25. No matter which compound unit weighting function is used, BM26 always produces better results than BM25.

Recall	TREC-5 $Weight_2$ BM25 (word)	TREC-5 $Weight_2$ BM25 (char)	TREC-6 $Weight_2$ BM25 (word)	TREC-6 $Weight_2$ BM25 (char)	TREC-6 $Weight_2$ BM26 (word)	TREC-6 $Weight_2$ BM26 (char)
	T5w2.BM25	T5c2.BM25	T6w2.BM25	T6c2.BM25	T6w2.kd10	T6c2.kd10
0.00	0.7738	0.7622	0.9557	0.9150	0.9641	0.9540
0.10	0.6427	0.6550	0.7920	0.8009	0.8121	0.8340
0.20	0.5385	0.5761	0.7124	0.7442	0.7414	0.7642
0.30	0.4813	0.5382	0.6636	0.6945	0.6904	0.7189
0.40	0.4447	0.4814	0.6186	0.6594	0.6460	0.6869
0.50	0.4004	0.4413	0.5553	0.5879	0.5724	0.6292
0.60	0.3563	0.3892	0.4616	0.5201	0.5094	0.5470
0.70	0.2865	0.3303	0.3715	0.4239	0.4001	0.4519
0.80	0.2285	0.2556	0.2685	0.3326	0.2926	0.3565
0.90	0.1287	0.1855	0.1656	0.2343	0.1879	0.2373
1.00	0.0257	0.0304	0.0072	0.0399	0.0063	0.0039
AP	0.3775	0.4126	0.4927	0.5341	0.5209	0.5603
	baseline	9.30%	baseline	8.40%	baseline	7.56%

Table 8.38: Recall-Level Precision over the TREC-5 and TREC-6 Topics

8.5.2 Negative Contribution Factors

We have found three factors that make negative contribution to our retrieval performance. First, the probabilistic retrieval methods used in our *C-Okapi* system did not use coordination level information and thus it is possible for documents containing only parts of the original query terms to be highly ranked. For example, topic 8 which is shown in Figure 8.20 included the query terms "Japan", "earthquake", "damage", "death", "injury" and "Richter scale". The relevant documents for topic 8 should be about "Earthquake in Japan", not "Earthquake", "damage", "death", "injury" that happened in somewhere else. Since the frequency of "Japan" in our TREC collections is much higher than that of "earthquake", "damage", "death", "injury" and "Richter scale", the probabilistic weight of "earthquake", "damage", "death", "injury" and "Richter scale" will be higher than that of "Japan". Without taking into account coordination level information, some of these documents containing "earthquake", "damage", "death" and/or "injury", but not "Japan" (these documents usually contain some other country's name, such as Mexico) were ranked highly by the *C-Okapi* retrieval methods. Some documents containing "flood" (not "earthquake"), "damage", "death" and/or "injury" were also ranked very high. This means that a lot of irrelevant documents may be retrieved for some topics by using the *C-Okapi* retrieval methods.

Second, the ambiguity of Chinese language can cause inaccurate retrieval results. For example, topic 20 which is shown in Figure 8.22 included query term "越战" meaning the Vietnam War which was collected in the Chinese word segmentation dictionary. The relevant documents for topic 20 should contain information

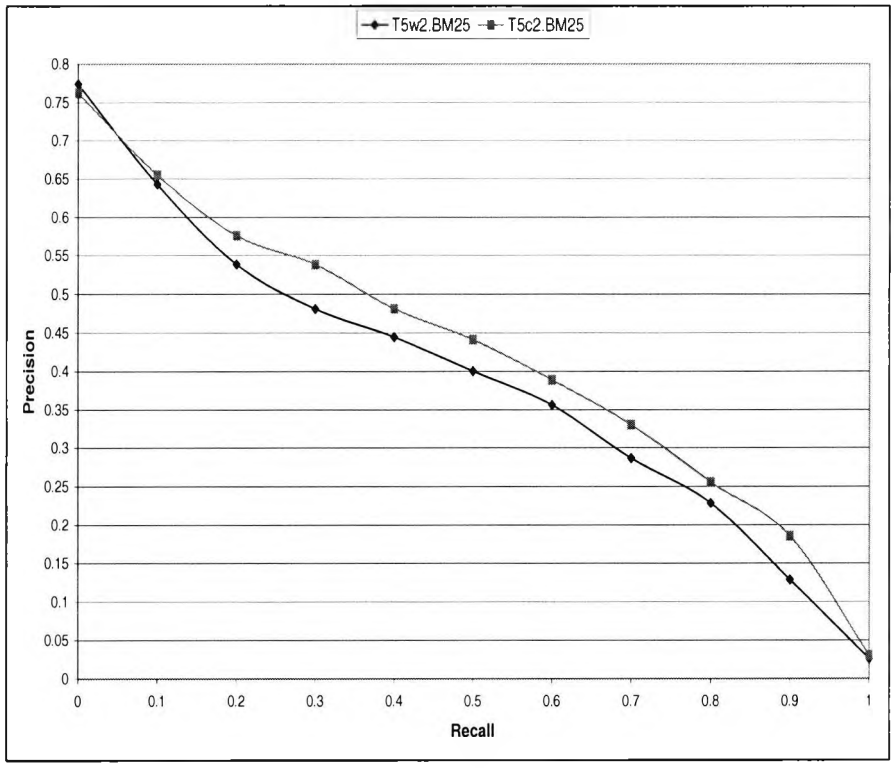


Figure 8.18: Precision-recall Curves for the Runs T5w2.BM25 and T5c2.BM25

related to the Vietnam War. But one document in the TREC collection contains a phrase "越战越勇" [22] which means "the more one fights, the more courage one has". This document is totally unrelated to the Vietnam War although the phrase "越战越勇" contains the word "越战". However, this document is ranked as a highly relevant document to topic 20 by both the word-based approach and character-based approach. Another example is topic 14 shown in Figure 8.21 which included the query term "爱滋病" (AIDS) [23]. The relevant documents for topic 14 should contain information related to AIDS. But one of the retrieved documents contains a Chinese character string "用爱滋润了病人的心" which means "comfort patients with love". This document has nothing to do with AIDS. However, this document is retrieved as a good relevant document by the character-based *C-Okapi* system

[22] "越战越勇" was not collected in the Chinese word segmentation dictionary

[23] "爱" means *love*, "滋" means *caused* and "病" means *disease* in English respectively

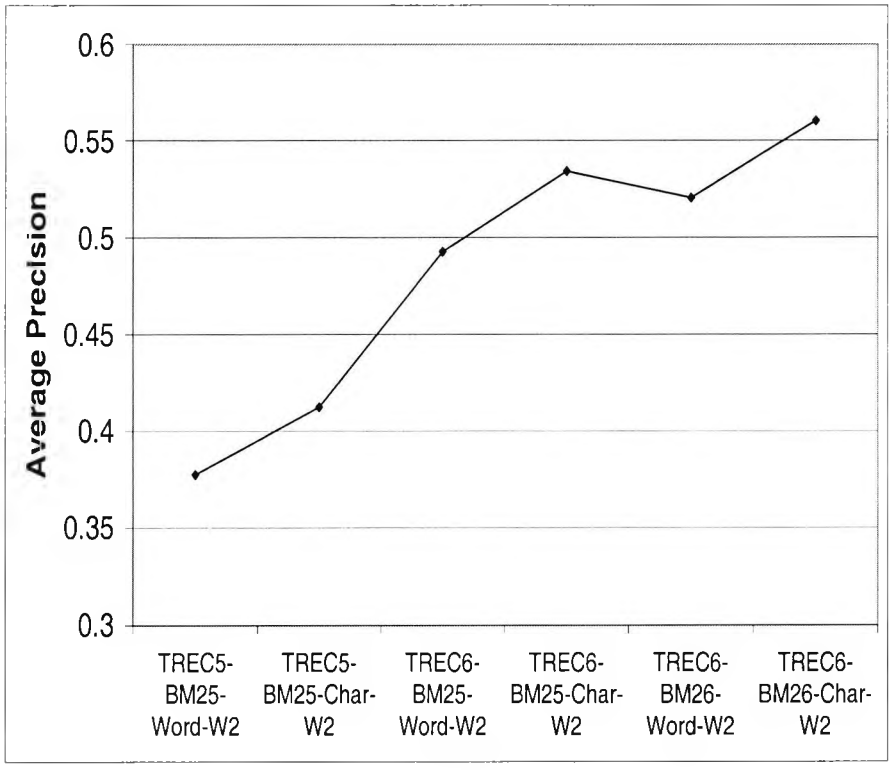


Figure 8.19: Improvements on Average Precision for Character and Word Approaches

because it contains the Chinese characters "爱" (love), "滋" (casued) and "病" (disease) which appeared in the same sentence but "爱滋" and "病" are separated by two other Chinese characters.

Third, sometimes the retrieval result is very poor if the TREC official Chinese topic used a rarely-used version for its query term translation in Chinese. For example, topic 14 is about AIDS cases in China. There are three translations for AIDS in Chinese so far. Translation 1 "艾滋病" (XBC) is to be considered formal, which is widely used in the our collection and translation 2 "爱滋病" (YBC) is informal, which is not used frequently in our collection but used in topic 14 as a query term. Translation 3 "爱之病" (YDC) is neither used frequently nor used in our TREC collection at all[24]. In a standard Chinese dictionary, only XBC

[24]X, Y, B, C and D are Chinese characters. XBC and YBC were selected as two words in our segmentation dictionary.

```
<top>

<num> Number: CH8

<E-title> Numeric Indicators of  Earthquake Severity in Japan
<C-title> 地震在日本造成的损害与伤亡数据

<E-desc> Description:
Japan, earthquake, damage, death, injury, Richter
scale

<E-narr> Narrative:
A relevant document should contain numeric
indicators such as the magnitude of the earthquake,
number of deaths or injuries, or property damage.

<C-desc> Description:
日本，地震，损失，死亡，级，受伤，芮氏地震仪

<C-narr> Narrative:
相关文件应包括地震的级数以及所造成的实际损害与伤亡数字,诸如
地震在芮氏地震仪上的级数,死亡与受伤人数,以及以金钱为单位的
财产损失数目.

</top>
```

Figure 8.20: Topic 8 from TREC-5

was selected as a word in the dictionary, not YBC and YDC. YBC and YDC are wrongly written Chinese words. In a non-Chinese speaking area, YBC is used quite a lot in Chinese newspapers. Actually, there are 1248 XBCs as a word in our Chinese TREC collection and there are only 36 YBCs as a word in our TREC collection. Also there are 158 XBCs as a part of a word in the TREC collection. If we just use YBC as a query term in our automatic runs, then the relevant documents retrieved will be very limited. The total number of relevant documents will be no more than 36. The problem is that AIDS in query 14 uses the informal translation 2 (YBC) while the TREC collection uses the formal translation 1 (XBC) in most cases. As we know, the spelling of some English words in standard English is different from these in American English, like "favo(u)rite". For topic 14, we could treat word XBC as a synonym of word YBC and put both translations (XBC and YBC) into the query as query terms. What we need do is to create a very simple synonym list in our program. However, according to strict TREC rules for the automatic runs, we should not really set up a synonym list (or any other kind of knowledge or linguistic structure) after seeing the topics. Therefore, our *C-Okapi*

```
<top>

<num> Number: CH14

<E-title> Cases of AIDS in China
<C-title> 中国的爱滋病例

<E-desc> Description:
China, Yunnan, AIDS, HIV, high risk group, syringe, virus

<E-narr> Narrative:
A relevant document should contain information on
the areas in China that have the highest AIDS
cases, how the AIDS virus was transmitted, and
how the Chinese government combats AIDS
problem.

<C-desc> Description:
中国,云南,爱滋病,HIV,高危险群患者,注射器,病毒

<C-narr> Narrative:
相关文件应当包括中国那些地区的爱滋病例最多,爱滋病毒在中国是如何传播
的,以及中国政府如何监测爱滋病并控制它的传染.

</top>
```

Figure 8.21: Topic 14 from TREC-5

retrieval system stays with the original query and only uses YBC as the query term AIDS translation for retrieval. This definitely caused very poor performance for all our automatic runs generated by the *C-Okapi* system.

One interesting thing for topic 14 is that the character-based approach performs much better than the word-based approach in terms of average precision. The improvement of the character-based approach over the word-based approach for topic 14 is shown in Table 8.39[25] (see Figure 8.1 and Figure 8.2 for more details). By observing this table, we can find that the topic 14's percentage increase is the biggest among all the other TREC-5 and TREC-6 Chinese topics. The reason for the better performance of character-based approach is that 艾滋病 (XBC) and 爱滋病 (YBC) have two characters in common. By searching the "YBC", there is still a good possibility to find relevant documents containing "XBC" for the character-based approach. However, it is impossible to find any documents

[25]In order to compare the word-based method with the character-based method for each TREC-5 and TREC-6 topic, we chose the best runs from these two different document processing methods in terms of average precision. The best runs in terms of average precision at TREC-5 and TREC-6 are $T5c2.kd0$, $T5w2.kd0$, $T6c2.kd10$ and $T6w2.kd20$.

```
<top>
<num> Number: CH20

<E-title> U.S. Military Personnel Missing in Action in Vietnam
<C-title> 越战失踪美军

<E-desc> Description:
Vietnam, MIA's

<E-narr> Narrative:
A relevant document presents any information on U.S. soldiers missing in action in Vietn
am. Document topics include missions to Vietnam, inter-government cooperation and discus
sions, effect on lifting the trade embargo, the Vietnamese Government's reaction to U.S.
statistics, MIA statistics, resolved cases, etc.
<C-desc> Description:
越南,失踪美军

<C-narr> Narrative:
相关文件:应包括任何有关美国军人在越南失踪的信息,包括美军在越南的任务,
美越政府间有关此问题的合作与讨论,以及美国停止对越南贸易制裁的影响.
此外,越南政府对美国有关在越战中失踪军人的统计数字的反应与已经解决的
案件等信息亦属相关文件.

</top>
```

Figure 8.22: Topic 20 from TREC-5

containing "XBC" for the word-based approach by using the query term "YBC". That is, the correct word segmentation of topic 14 leads to no matching with many documents that use the official form. However, when single characters are used, the second and third characters happen to be the same in both transliterations and some matching between query and relevant documents is restored. Again, this confirms that partial matching works more effectively for the character-based approach than the word-based approach.

topics	P(character)	P(word)	increase	topics	P(character)	P(word)	increase
1	0.1042	0.0908	14.76%	28	0.4327	0.4528	-4.44%
2	0.3675	0.3347	9.80%	29	0.7169	0.6547	9.50%
3	0.3501	0.3053	14.67%	30	0.3272	0.3424	-4.44%
4	0.2930	0.2788	5.09%	31	0.6129	0.4604	33.12%
5	0.0727	0.0808	-10.02%	32	0.5618	0.5339	5.23%
6	0.2304	0.2484	-7.25%	33	0.3703	0.3569	3.75%
7	0.3485	0.2381	46.37%	34	0.1600	0.1187	34.79%
8	0.6786	0.5096	33.16%	35	0.5767	0.5595	3.07%
9	0.4683	0.5366	-12.73%	36	0.4056	0.3740	8.45%
10	0.1451	0.0907	59.98%	37	0.6424	0.4957	29.59%
11	0.3714	0.3192	16.35%	38	0.6988	0.7152	-2.29%
12	0.1826	0.1903	-4.05%	39	0.5742	0.5526	3.91%
13	0.2022	0.1291	56.62%	40	0.6660	0.6571	1.35%
14	0.1696	0.0864	96.30%	41	0.6474	0.6445	0.45%
15	0.5246	0.5955	-11.91%	42	0.2491	0.3053	18.41%
16	0.3232	0.3089	4.63%	43	0.5589	0.5079	10.04%
17	0.3895	0.3800	2.50%	44	0.7749	0.7682	0.87%
18	0.0934	0.0896	4.24%	45	0.7695	0.7332	4.95%
19	0.7567	0.7385	2.46%	46	0.3183	0.2747	15.87%
20	0.7633	0.6266	21.82%	47	0.8722	0.7994	9.11%
21	0.8778	0.8564	2.50%	48	0.7292	0.7014	3.96%
22	0.6827	0.5950	14.74%	49	0.6077	0.4995	21.66%
23	0.6422	0.6654	-3.49%	50	0.6237	0.6530	-4.49%
24	0.6503	0.6321	2.88%	51	0.3540	0.3568	-0.79%
25	0.5814	0.4641	25.27%	52	0.4452	0.4569	2.56%
26	0.3254	0.2688	21.06%	53	0.5774	0.5711	1.10%
27	0.5261	0.4552	15.58%	54	0.7270	0.6936	4.82%

1. the increase of character approach over the word approach is calculated by the following function $\frac{P(character)-P(word)}{P(word)} \times 100\%$;
2. P(character) and P(word) are the average precision for character and word approach respectively.

Table 8.39: Improvement of Character Approach over Word Approach for TREC-5 and TREC-6 Topics in terms of Average Precision

Chapter 9

Concluding Remarks

9.1 Conclusions and Contributions

Text segmentation and term weighting are two important issues in Chinese text retrieval. We have presented two Chinese text segmentation methods and two types of term weighting methods. The methods include:

- a word-based text segmentation method
- a character-based text segmentation method
- a single unit weighting method
- several compound unit weighting methods

Our objective was to investigate the effectiveness of each of these methods on the performance of Chinese text retrieval. To achieve this objective, we conducted experiments that evaluated these methods on the Chinese TREC collections. Our investigation presented in this thesis obtained the following three conclusions, which constitute the main contributions of this thesis.

1. *Indexing.*

   Two document processing methods have been compared in our experiments: character-based indexing and word-based indexing. Generally speaking, our evaluation results demonstrated that the character-based document processing is better than word-based approaches for Chinese text retrieval using

probabilistic models in terms of average retrieval precision.[1] However, the significance of improvement of character-based over word-based approaches depends on what compound unit weighting method is used in the retrieval process. If formula $Weight_2$ is used for compound unit weighting, the improvement is significant. Otherwise, the improvement is negligible. Therefore, we can conclude that accurate word segmentation is not a pre-requisite for effective IR. We also believe that neither the word-based approach nor the character-based approach have reached their limits. Improvements are still possible.

2. *Probabilistic Weighting.*

   In terms of single unit weighting, the results indicate that using the BM26 weighting function makes a significant positive contribution to the quality of retrieval compared to using BM25. Concerning the use of compound unit weighting methods, we can draw a conclusion that the method of $Weight_2$ is the best among the 6 tested methods in terms of average precision. The results for compound unit weighting also indicate that the boost weight in the formulas for $w(t_1\ adj\ t_2\ adj \cdots adj\ t_j)$ plays an important role in the performance of the retrieval system. Neither a big boost weight as in $Weight_1$ and $Weight_5$ nor a too small boost weight as in $Weight_3$ leads to the best results. A moderate boost weight such as the boost weights in $Weight_2$ and $Weight_4$ produces better retrieval results. A number of experiments and analyses which have been conducted in Chinese TREC confirm our findings in this thesis.

3. *Probabilistic Model.*

   Our Chinese experiments show that the probabilistic model carries over to other languages. It would indeed be very surprising, given its generality, if it did not. Thus, for the TREC Chinese material discussed in this thesis and [117], our test results are useful as initial evidence that the probabilistic model interpretations not only carry over to other languages, but also to

[1] If we consider time and space, approaches based on words run faster and use less memory than character-based approaches.

ones where the nature of basic terms is totally different from that of English words. All in all, we suggest that the probabilistic model described so far for English is also both reasonably well-founded and of clear and substantial value in the design of Chinese information retrieval systems.

In summary, the experimental results obtained in this thesis show that (1) compound unit weighting is useful for improving system performance, especially for character based retrieval systems; (2) BM26 gives position contribution to the improvement of Chinese information retrieval; (3) generally, character-based indexing methods are better than word-based methods in terms of average precision. The experiments also demonstrate, for a large test collection, that the probabilistic model is effective and robust, and that it responds appropriately, with major improvements in performance, to key features of retrieval situations in Chinese text retrieval. The thesis also presents detailed analyses of empirical results on the TREC-5 and TREC-6 datasets. We believe that the data gathered and the analyses that have been done can shed a lot of light on the qualitative differences between different forms of indexing, word segmentation and term weighting for Chinese text retrieval.

In addition to the above findings, we can also say that the success of the compound-unit weighting method suggest a new approach to phrase weighting in English and other languages, particularly given that significant benefits from the use of phrases in English seem elusive. However, whether this new approach can bring benefits to English and other languages needs empirical evaluation.

9.2 Suggestions for Future Work

Our official results submitted to TREC-5 and TREC-6 are not the best. One reason is that we did not use any extra techniques in our runs to improve the effectiveness, as several other groups did (for example, most participated groups in TREC-5 and TREC-6 obtained some gain from expansion and manual intervention for topics). By using some techniques such as using the top ranked documents to do automatic feedback retrieval, the effectiveness may be increased. Our goal in

our experiments was to concentrate on comparing two different document processing methods, different single unit weighting methods and six different compound unit weighting methods. We may improve the retrieval performance by doing the following:

1. *Query Expansion.*

   A popular method for query expansion is to assume the top n (such as 10) retrieved documents are relevant, extract some new terms from these 10 documents and add them into the set of query terms.

2. *Use of Heuristic Rules in Word Segmentation.*

   For word approaches, the accuracy of segmentation may be improved by integrating more heuristic rules to recognize more special sequences, or by incorporating a thesaurus in the retrieval process.

3. *Passage Retrieval.*

   Modern full-text collections often contain long documents that have several topics, only one of which is relevant. Since these documents have only a small proportion of terms that match the query, low scores of relevant weight will be assigned to these documents even though they may contain extremely relevant parts. One solution to this problem is passage-based retrieval. In passage-based retrieval, full-text documents are broken down into passages and a document is retrieved only if it contains passages that are relevant. Using passage-based retrieval can help to identify relevant sections from long documents which may otherwise receive low similarity values. Passage-based techniques have been used successfully in English full-text retrieval. For our future work, it is interesting to show how these passage-based techniques can be used for Chinese text retrieval. Using the probabilistic methods to assign correct weights in passage-based retrieval is also an interesting topic for Chinese text retrieval.

For two years in a row Chinese retrieval in the TREC environment has shown much higher effectiveness (50% to 100% higher) than English for both long and short queries. It is not very clear if this is due to the data being much 'easier', or if

this is due to some intrinsic properties of the Chinese language. It is of interest to continue further experiments using more diverse collections and queries to throw some light onto this phenomenon.

We hope to continue the analysis started on the *TREC*-5 and *TREC*-6 data, and also intend to apply similar analysis between various n-grams to further understand the relationship between n-gram and segmented indexing. There is some anecdotal evidence that suggests that bigrams and short-words together with characters would give better precision [62], and this, among other hypotheses, will be tested in future work. A theoretical study of the relation between bigram-indexing and single-character-with-compound-unit-weighting might be interesting, as well as empirical studies. For example, we can calculate mutual information [19] of each bigram in a document collection and then segment the document and queries by sequentially removing the bigram with the current highest mutual information value. After this indexing process, we can apply our probabilistic methods for retrieval.

Bibliography

[1] Alta Vista Home Page, *http://altavista.digital.com/* (visited on 29 August 2001)

[2] An, A., Cercone, N., Chan, C., Huang, X. and Shan, N., "ELEM: A Method for Inducing Rules from Examples", *Proceedings of the 15th Annual Conference of the British Computer Society Specialist Group on Expert Systems*, pages 85–99, Cambridge, UK. December 1995.

[3] An, A., Cercone, N. and Chan, C. "Integrating Rule Induction and Case-based Reasoning to Enhance Problem Solving", *Proceedings of the Second International Conference on Case-Based Reasoning*, Rhode Island, USA, July 1997.

[4] An, A., Chan, C., Cercone, N. and Huang, X. "Water Demand Forecasting Using Case-Based Reasoning", *Proceedings of the IJCAI'97 Workshop on Practical Use of Case-Based Reasoning*, pages 99–110, Nagoya, Japan; 1997.

[5] Beaulieu, M.M., Gatford, M, Huang, X., Robertson, S.E., Walker, S. and Williams, P. "Okapi at TREC-5". In E.M. Voorhees and D.K. Harman, editors: *Proceedings of the Fifth Text REtrieval Conference (TREC-5).* NIST Special Publication 500-238, pages:143-166, Gaithersburg, MD, November 1997.

[6] Belkin, B. J. and Croft, W. B. "Information Filtering and Information Retrieval: Two sides of the same coin?" *Communication of the ACM*, 35(12):29-38, 1992.

[7] Bookstein, A. "Probability and Fuzzy-Set Applications to IR". *Annual Review of Information Science and Technology*, 20:117-151, 1985.

[8] Buckley, C., Salton, G., and Allan, J. "The Effect of Adding Relevance Information in a Relevance Feedback Environment". In W. B. Croft and C. Van Rijsbergen, editors. *SIGIR 94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 292–300, Dublin, Ireland. Springer-Verlag, 1994.

[9] Buckley, C., Salton, G., Allan, J. and Singhal, A. "Automatic Query Expansion Using SMART: TREC-3". In D. K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3)*, NIST Special Publication 500-226, pages 69–80, Gaithersburg, MD, November 1995.

[10] Buckley, C., Singhal, A. and Mitra, M. "Using Query Zoning and Correlation Within SMART: TREC-5". In E.M. Voorhees and D.K. Harman, editors: *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, NIST Special Publication 500-238, pages 105–118, Gaithersburg, MD, November 1997.

[11] Buckley, C., Walz, J., Mitra, M. and Cardie, C. "Using Clustering and Super-Concepts Within SMART: TREC-6", In E.M. Voorhees and D.K. Harman, editors: *Proceedings of the Sixth Text REtrival Conference (TREC-6)*, NIST Special Publication 500-240, pages 107-124, Gaithersburg, MD, November 1998.

[12] Callan, J. P., Croft, W. B. and Broglio, J. "TREC and Tipster Experiments with INQUERY", *Information Processing & Management.* 31(3): 327-343; 1995

[13] Chakravarthy, Anil S., Hnase, K. "Netserf: Using Semantic Knowledge to Find Internet Information Archives", *ACM SIGIR'95*, 1995.

[14] Chen, H. and Dhar, V. User Misconceptions of Online Information Retrieval Systems. *International Journal of Man-Machine Studies*, 32(6):673-692, June 1990.

[15] Chen, K. J. and Liu, S. H. "Word Identification for Mandarin Chinese Sentences". In *Proceedings of 5th International Conference on Computational Linguistics*, pages 101–107, August 1992.

[16] Chen, G. "On Single Chinese Character Retrieval System". *Journal of Information*, 11(1):11-18 (in Chinese), 1992

[17] Chen, H. "The Vocabulary Problem in Collaboration. *IEEE COMPUTER*, Special Issue on CSCW, 1994

[18] Chen, A., He, J., Xu, L., Gey, F. C. and Meggs, J. "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, Philadelphia, PA, July 1997.

[19] Chen, Hsin-Hsi. "Chinese Language Retrieval". Personal communication between Hsin-Hsi Chen and author, 1999.

[20] Chen, A., Gey, F. C., Jiang, H. "Berkeley at NTCIR-2: Chinese, Japanese, and English IR Experiments". In *Proceedings of the Second NTCIR Workshop Meeting on Evaluation of Chinese & Japanese Text Retrieval and Text Summarization*, pages 32–39, Tokyo, Japan, March 2001.

[21] Chien, Lee-Feng. "Fast and quasi-natural language search for gigabits of Chinese texts", In *Proceedings of the 18th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 95: 112-120; 1995.

[22] Chien, Lee-Feng. "Natural Language Information Retrieval with Speech Recognition Techniques for Network Chinese Resources Discovery", *Intenational Workshop on Information Retrieval with Oriental Languages*, Korea, 1996.

[23] Cleverdon, C.W., Mills, J. and Keen, E. M. "Factors determining the performance of indexing systems". In *Cranfield: College of Aeronautics*, 1966.

[24] Cooper, W., Chen, A and Gey, F. "Full Text Retrieval based on Probabilistic Equations with Coefficients fitted by Logistic Regression". In D.K. Harman, editor:*Proceedings of the Second Text REtrieval Conference (TREC-2)*. NIST Special Publication 500-215, pages:57-66, Gaithersburg, MD, November 1994.

[25] Crimmins, F., Smeaton, A.F., Dkaki. T. and Mothe., J. "TetraFusion: Information Discovery on the Internet". *IEEE Intelligent Systems and Their Application.* 55-62; July 1999.

[26] Croft, B. "The Use of Phrases and Structured Queries in Information Retrieval" *Proceedings of the 14th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 32-45, Chicago, IL. ACM Press., 1991.

[27] Ding, Du et al., "集韵", published in Song Dynasty (from 960 to 1269).

[28] Excite Home Page, "http://www.excite.com" (visited on 29 August 2001)

[29] Fidel, R. "User-Centered Indexing". *Journal of the American Society for Information Science*, 45:572-576, 1995.

[30] Frakes, W. and Baeza-Yates, R. *Information Retrieval: Algorithms and Data Structures.* Chapter 11. Prentice-Hall, 1992.

[31] Funio, M., et al., "Test Collection for Japanese Information Retrieval Systems from the Viewpoint of Evaluating System Functions", *Proceedings of the 1996 Workshop on Information Retrieval with Oriental Languages*, Taejon, Korea, page 42-47, 1996.

[32] Furnas, G. W., Landauer, T. K., Gomez, L. M. and Dumais, S. T. The Vocabulary Problem in Human-system Communication. *Communications of the ACM*, 30(11) :964-971, November 1987

[33] "信息处理用现代汉语分词规范－中华人民共和国国家标准GB13715". *Tsinghua University Publisher*, 1991.

[34] Gey, F., Chen, A., He, J., Xu, L. and Meggs, J. "Term Importance, Boolean Conjunct Trainning Negative Terms, and Foreign Language Retrieval: Probabilistic Algorithms at TREC-5". In E.M. Voorhees and D.K. Harman, editors: *Overview of the Fifth Text REtrieval Conference (TREC-5)*, NIST Special Publication 500-238, pages 181-190, Gaithersburg, MD, November 1997.

[35] Gey, F. and Chen, A. "Phrase Discovery for English and Cross-language Retrieval at TREC-6". In E.M. Voorhees and D.K. Harman, editors: *Overview of the Sixth Text REtrieval Conference (TREC-6)*, NIST Special Publication 500-240, pages 637-647, Gaithersburg, MD, November 1998.

[36] Glavitsch, U. and Schauble, P. "A System for Retrieving Speech Dcouments", *ACM SIGIR Conference on R&D in Information Retrieval*, pages 168-176, 1992.

[37] Google Home Page, "http://www.google.com" (visited on 29 August 2001)

[38] Harman, D. "Overview of the First Text REtrieval Conference", In D. K. Harman, editor, *The First Text REtrieval Conference (TREC-1), NIST Special Publication 500-207*, pages 1-20, Gaithersburg, MD, March 1993.

[39] Harman, D. K. "Overview of the Third Text REtrieval Conference (TREC-3)". In D. K. Harman, editor, *Information Technology: The Third Text REtrieval Conference (TREC-3)*, NIST Special Publication 500-225, pages 1–19, Gaithersburg, MD, November 1995.

[40] Harman, D. "Email from Donna Harman on 2 July 1996". Personal communication.

[41] Harter, S. P. "A Probabilistic Approach to Automatic Keyword Indexing", *Journal of the American Society for Information Science*, 26, pages:197-206 and 280-289, 1975.

[42] He, J., Xu, L., Chen, A., Meggs, J. and Gey, F. C. "Berkeley Chinese Information Retrieval at TREC-5: Technical Report". In E.M. Voorhees and D.K. Harman, editors: The Fifth Text REtrieval Conference (TREC-5), NIST Special Publication 500-238, pages 191–196, November 1997.

[43] Huang, X. "Chinese Full Text Retrieval based on Probabilistic Models", *Master thesis, Xidian University Publisher*, Xi'an, China, 1990.

[44] Huang, X. "Design and Implementation of an Optimal Retrieval Algorithm Based on Probabilistic Model". *Proceedings of the third National Conference of Youth on Computer*, in Chinese; Pages 226-260, August 1991.

[45] Huang, X. "Design and Analysis of a Probabilistic Full-Text Chinese Retrieval System and its Ranking Algorithm". *Journal of Wuhan University of Hydraulic and Electrical Engineering.* 25(3):1-26; June 1992

[46] Huang, X. "Implementation and Experiments of a Probabilistic Chinese Full-Text Retrieval System on a Middle Scale Dataset". *The 1993 IEEE International Conference on "Computers, Communications, Control and Power Engineering*, Beijing, China, October 1993.

[47] Huang, X, An, A., Robertson, S.E. and Tontiwachwuthikul, P. "A Knowledge Based Approach to Fuzzy Information Retrieval from Petroleum Databases", *Proceedings of the Six Saskatchewan Petroleum Conference*, sponsored by the Petroleum Society of CIM, Regina, Canada, 1995. 10 pages.

[48] Huang, X.; Robertson, S.E. "Application of Probabilistic Methods to Chinese Text Retrieval". *Journal of Documentation.* 53(1):74-79; 1997

[49] Huang, X.; Robertson, S.E. "Experiments on Large Test Collections with Probabilistic Approaches to Chinese Text Retrieval". *Proceedings of the 2nd International Workshop on Information Retrieval with Asian Languages*, IRAL'97, pages 25–41, Tsukuba-shi, Ibaraki-ken, Japan; 1997.

[50] Huang, X.; Robertson, S.E. "Okapi Chinese Text Retrieval Experiments at TREC-6". In E.M. Voorhees and D. K. Harman, editors, *The Sixth Text REtrieval Conference (TREC-6)*, pages 137–142, NIST Special Publication 500-240, Gaithersburg, MD, November 1998.

[51] Huang, X.; Robertson, S.E.; Cercone, N. and An, A. "Probability-Based Chinese Text Processing and Retrieval". *Proceedings of the Conference Pacific Association for Computational Linguistics*, PACLING'99, pages 223-235, Waterloo, Ontarion, Canada; August 1999.

[52] Huang, X. and Robertson, S.E. "A Probabilistic Approach to Chinese Information Retrieval: Theory and Experiments". In *Proceedings of the 22nd Annual BCS-IRSG Colloquium on Information Retrieval Research*, pages 178–193, Cambridge, England, April 2000.

[53] Huang, X. and Robertson, S.E. "Probability-Based Chinese Text Processing, Retrieval and Experiments". *Computational Intelligence: An International Journal.* 16(4):552-569; 2000

[54] Hull, D. A., Pedersen, J. O. and Schutze, H. "Method combination for document filtering". In H. P. Frei, D. K. Harman, P. Schauble, and R. Wilkinson, editors *SIGIR 96: Proceedings of the Nineteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 279–287, Zurich. ACM Press., 1996.

[55] Infoseek Home Page, "http://www.infoseek.com" (visited on 29 August 2001)

[56] Jacobs, P. and Rau, L. *"Innovations in Text Interpretation in Artificail Intelligence"*, North Holland, Amsterdam 1997

[57] Kahle, B. et al., "Wide Area Information Servers: An Executive Information System for Unstructured Files", *Electronic Networking: Research, Applications and Policy.* 2:59-68, Spring 1992.

[58] Kelledy, F. and Smeaton, A.F. "Automatic Phrase Recognition and Extraction from Text". In J. Furner and D.J. Harper, editors, *Proceedings of the 19th Annual BCS-IRSG Colloquium on Information Retrieval Research*, Aberdeen, Scotland, April 1997.

[59] Koster, M. "Robots in the web: Threat or Treat ?" Available on the Internet at *http://web.nexor.co.uk/mak/doc/robots.* (visited on 29 August 2001)

[60] Kwok, K. L., "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, Philadelphia, PA, July 1997.

[61] Kwok, K. L. and Grunfeld, L. "TREC-5 English and Chinese Retrieval Experiments using PIRCS". In E.M. Voorhees and D.K. Harman, editors: *The Fifth Text REtrieval Conference (TREC-5)*, National Institute of Standards and Technology Special Publication 500-238, Gaithersburg, MD, November 1997.

[62] Kwok, K. L., Grunfeld, L and Xu, J. H. "TREC-6 English and Chinese Retrieval Experiments using PIRCS". In Harman, D. K., editor, *The Sixth Text REtrieval Conference (TREC-6)*, National Institute of Standards and Technology Special Publication 500-240, Gaithersburg, MD, November 1998.

[63] Lee, Ahn and Lee, Shin, "An Effective Indexing Method for Korean Text Retrieval", *International Workshop on Information Retrieval with Oriental Languages*, Korea, 1996.

[64] Lewis, David D. and Karen Sparck Jones, "Natural Language Processing for Information Retrieval", *Communication of the ACM*, Vol. 39, No. 1, pages 92-101. January 1996

[65] Lexis/Nexis Home Page, *http://www.lexis-nexis.com/* (visited on 29 August 2001)

[66] Liang, T., Lee, S. and Yang, W. "Optimal Weight Assignment for a Chinese Signature File", *Information Processing and Management*, Vol 32, No. 2, pages. 227-237, 1996.

[67] Liu, Yuan et al., "信息处理用现代汉语分词规范及自动分词方法", ISBN 7-302-01430-2/TP-556, *Tsinghua University Publisher*, 1991.

[68] Luhn, H. P. "The Automatic Greation of Literature Abstracts". *IBM Journal of Research and Development*, 2:159-165, 1958

[69] Lycos Home Page, *http://www.lycos.com/* (visited on 29 August 2001)

[70] Maron, M.E. and Kuhns, J.L. "On Relevance, Probabilistic in Indexing and Information Retrieval". *Journal of the ACM*, 7:216-244, 1960.

[71] McKeown, K. D. "Generating Summaries of Multiple News Articles", In *Proceedings of the 18th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, July 1995.

[72] Nie, J. Y. and Ren, X. "On Chinese Text Retrieval". In *Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 225–234, Zurich, Switzerland, August 1996.

[73] Nie, J. Y. and Chevallet, J.P. "Between Terms and Words for European Language IR and Between Words and Bigrams for Chinese IR". In Harman, D. K., editor, *The Sixth Text REtrieval Conference (TREC-6)*, National Institute of Standards and Technology Special Publication 500-240, Gaithersburg, MD, November 1998.

[74] Nie, J.Y. and Ren, F. "Chinese Information Retrieval:Using Characters or Words". *Information Processing and Management.* 35(4), 443-462, 1999.

[75] Nie, J. Y., Gao, J, Zhang, J. and Zhou, M. "On the Use of Words and N-grams for Chinese Information Retrieval". *Proceedings of the 5th International Workshop on Information Retrieval with Asian Languages*, IRAL'2000, pages 78–86, Hong Kong, China; 2000.

[76] NTCIR Home Page, *http://research.nii.ac.jp/ ntcadm/index-en.html* (visited on 29 August 2001) (visited on 29 August 2001)

[77] National Institute of Informatics Home Page, *http://research.nii.ac.jp/* (visited on 29 August 2001) (visited on 29 August 2001)

[78] Ogawa, Y., "A New Character-based Indexing Organization Using Frequency Data for Japanese Documents", In *ACM SIGIR'95*, 1995.

[79] O'Kane, K. C., "World Wide Web-based Information Storage and Retrieval", *Online & CDROM Review*, Vol. 20, No.1, 1996.

[80] Open Text Home Page, "http://www.opentext.com" (visited on 29 August 2001)

[81] Pedersen, J., Politowski, G., Silverstein, C., and Vogt, C. "Verity TREC-6 Report". In E.M. Voorhees and D. K. Harman, editors, *The Sixth Text REtrieval Conference (TREC-6)*, National Institute of Standards and Technology Special Publication 500-240, Gaithersburg, MD, November 1998.

[82] Peoples' Daily Networked Chinese Full Text Information Retrieval System, *http://www.peopledaily.com.cn/query/* (visited on 29 August 2001)

[83] van Rijsbergen, C. J. "A Theoretical Basis for the Use of Co-occurrence Data in Information Retrieval". *Journal of Documentation*, 33:106-119, 1977.

[84] van Rijsbergen, C. J. *Information Retrieval.* Second Edition, London: Butterworths, 1979

[85] Robertson, S.E. and Sparck Jones, K. "Relevance Weighting of Search Terms". *Journal of the American Society for Information Science*, May-June, 129-146, 1976.

[86] Robertson, S.E. "A Theoretical Model of the Retrieval Characteristics of Information Retrieval Systems". *PhD Thesis*, University of London, 1976.

[87] Robertson, S.E. "The Probability Ranking Principle in IR". *Journal of Documentation*, Vol.33, No.4, 294-304, 1977.

[88] Robertson, S.E., van Rijsbergen C.J. and Porter M.F. "Probabilistic models of indexing and searching." In Oddy R.N. *et al.* (Eds.) *Information Retrieval Research*, pages 35-56. Butterworths London, 1981.

[89] Robertson, S.E. "The Methodology of Information Retrieval Experiment". In Sparck Jones, K., *ed. Information Retrieval Experiment*, pages 9-31. London: Butterworths, 1981.

[90] Robertson, S.E. "On Relevance Weight Estimation and Query Expansion". *Journal of Documentation.* 42, 182-188, 1986.

[91] Robertson, S.E. "On Term Selection for Query Expansion". *Journal of Documentation.* 46, 359-364, 1990.

[92] Robertson, S.E. and Hancock-Beaulieu, M. "On the Evaluation of IR Systems". *Information Processing and Management.* 28, 457-466, 1992.

[93] Robertson, S. E. and Walker, S. "Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval". In W.B. Croft and C.J. Van Rijsbergen, editors:*Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241, Dublin, Ireland, July 1994.

[94] Robertson, S.E. "Query-document Symmetry and Dual Models". *Journal of Documentation.* 50, 233-238, 1994.

[95] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M. and Gatford, M. "Okapi at TREC-3". In D. K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3)*, NIST Special Publication 500-226, pages 109–126, Gaithersburg, MD, November 1995.

[96] Robertson, S.E.; Walker, S.; Hancock-Beaulieu, M. "Large Test Collection Experiments on an Operational, Interactive System: OKAPI at TREC." *Information Processing & Management.* 31(3): 345-360; 1995

[97] Robertson, S.E. "Phrase Weighting – paper 4". *research notes*, 1996.

[98] Robertson, S.E. and Sparck Jones, K. "Simple, Proven Approaches to Text Retrieval". University of Cambridge: Computer Laboratory, Technical Report No.356, 1996. Available via http://www.cl.cam.ac.uk/ftp/papers/index.html. (visited on 29 August 2001)

[99] Robertson, S.E., Walker, S. and Beaulieu, M. "Laboratory Experiments with Okapi: Participation in the TREC Programme". *Journal of Documentation.* 53(1):20-34; 1997

[100] Robertson, S.E. and Walker, S. "On Relevance Weights with Little Relevance Information". In N.J. Belkin, A.D. Narasimhalu and P. Willett, editors: *Proceedings of SIGIR'97.* ACM, pages 16-24, Philadephia, 1997.

[101] Robertson, S.E. and Walker, S. "Threshold Setting in Adaptive Filtering". *Journal of Documentation.* 56:312-331; 2000.

[102] Robertson, S.E., Walker, S. and Beaulieu, M. "Experimentation as a Way of Life: Okapi at TREC". *Information Processing and Management.* 36:95-108; 2000.

[103] Salton, G., Yang, C. S. and Yu, C. T. "A Theory of Term Importance in Automatic Text Analysis". *Journal of American Society for Information Science*, 26(1): 33-44, January 1975.

[104] Salton, G., editor. *The SMART Retrieval System. Experiments in Automatic Document Processing.* Prentice-Hall, New Jersey, 1977.

[105] Salton, G., McGill, M. J. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, 1983.

[106] Salton, G., Fox, E. A. and Wu, H. "Extended Boolean Information Retrieval". *Communication of the ACM*, 26(12):1022-1036, 1983.

[107] Salton, G., Fox, E. A. and Voorhees, E. "Advanced Feedback Methods in Information Retrieval". *Journal of the American Society for Information Science*, 36:200–210, 1985.

[108] Salton, G. "Another Look at Automatic Text-Retrieval Systems". *Communications of the ACM*, 29(7):649-656, 1986.

[109] Salton, G. and Buckley, C. "Term-weighting Approaches in Automatic Text Processing". *Information Processing and Management*, 24(5):513-524, 1988.

[110] Salton, G. *Automatic Text Processing.* Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

[111] Salton, G. and Buckley, C.. "Improving Retrieval Performance by Relevance Feedback". *Journal of the American Society for Information Science*, 41:288-297, 1990.

[112] Salton, G. Allan, J. and Buckley, C. "Automatic Structuring and Retrieval of Large Text Files". *Communications of the ACM*, 37(2):97-108, February 1994.

[113] Schauble, P. "SPIDER Retrieval System at TREC-5". In E.M. Voorhees and D.K. Harman, editors: *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, NIST Special Publication 500-238, pages 217–228, Gaithersburg, MD, November 1997.

[114] Smeaton, A. F. "Retrieving Information from Hypertext: Issues and Problems". *European Journal of Information Systems.* 1:239-247; 1991.

[115] Smeaton, A.F."Information Retrieval and Hypertext: Competing Technologies or Complementary Access Methods". *Journal of Information Systems.* 2 :221-233; 1992.

[116] Smeaton, A.F. and Quigley, I. "Experiments on Using Semantic Distances Between Words in Image Caption retrieval". In H. P. Frei, D. K. Harman, P. Schauble, and R. Wilkinson, editors *SIGIR 96: Proceedings of the Nineteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 174–180, Zurich. ACM Press., 1996.

[117] Smeaton, A. and Wilkinson, R. "Spanish and Chinese Document Retrieval in TREC-5". In E.M. Voorhees and D.K. Harman, editors: *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, NIST Special Publication 500-238, Gaithersburg, MD, November 1997.

[118] Smeaton, A.F., Crimmins, F. "Relevance Feedback and Query Expansion for Searching the Web: A Model for Searching a Digital Library". In *Proceedings of the First European Conference on Digital Libraries*, Springer-Verlag, pages: 99–112, Berlin, Germany, 1997.

[119] Smeaton, A.F. and Kelledy, F. "User-Chosen Phrases in Interactive Query Formulation for Information Retrieval". In *Proceedings of the 20th Annual BCS-IRSG Colloquium on Information Retrieval Research*, Grenoble, France, April 1998.

[120] Smeaton, A.F. "TREC-6 Personal Highlights". *Information Processing and Management.* 36(1): 87-94; January 2000.

[121] Sohoo Chinese Search Engine Home Page, *http://search.sohoo.com/* (visited on 29 August 2001)

[122] Sparck Jones, K. and Van Rijsbergen, C. J. "Report on the Need for and Provision of an 'Ideal' Information Retrieval Test Collection". *British Library Research and Development Report 5266*, University of Cambridge: Computer Laboratory, 1975.

[123] Sparck Jones, K. "Search Relevances Weighting Given Little Relevance Information". *Journal of Documentation*, 35(1):30-48, 1979.

[124] Sparck Jones, K. "What Might be in a Summary"? In G. Knorz, J. Krause and C. Womser-Hacker, editors: *Information Retrieval*, pages: 9-26; September 1993.

[125] Sparck Jones, K. "Reflections on TREC". *Information Processing and Management.* 31(3): 291-314; 1995.

[126] Sparck Jones K., Jones, G.J.F. Foote, J.T. and Young, S.J. "Experiments in Spoken Document Retrieval". *Information Processing and Management.* 32:399-419; 1996.

[127] Sparck Jones, K. and Willett, P., editors. *Readings in Information Retrieval.* San Francisco:Morgan Kaufman, 1997.

[128] Sparck Jones, K., Walker, S. and Robertson, S.E. "A Probabilistic Model of Information Retrieval: Development and Status". University of Cambridge: Computer Laboratory, Technical Report No.446, August 1998. Available via http://www.ftp.cl.cam.ac.uk/ftp/papers/reports/#TR446. (visited on 29 August 2001)

[129] Sparck Jones, K. "Information Retrieval and Artificial Intelligence". *Artificial Intelligence.* 114(1-2): 257-281; 1999.

[130] Tenopir, C. "Full-Text Databases". *Annual Review of Information Science and Technology*, 19:215–246, 1984.

[131] Tenopir, C. and Ro, J. S. *Full Text Databases.* Greenwood Press, Westport, CT, 1990.

[132] Tseng, S.S., Yang, C.C. and Hsieh, C.C."On the Design of Chinese Textual Database", *Computer Processing of Chinese and Oriental Languages*, 4: pages 240-271. 1989.

[133] Voorhees, E.M. and Harman, D. K. "Overview of the Fifth Text REtrieval Conference (TREC-5)", In E.M. Voorhees and D. K. Harman, editors, *Proceedings of The Fifth Text REtrieval Conference (TREC-5), NIST Special Publication 500-238*, pages 1-28, Gaithersburg, MD. November 1997.

[134] Voorhees, E.M. and Harman, D. K. "Overview of the Sixth Text REtrieval Conference (TREC-6)", In E.M. Voorhees and D. K. Harman, editors, *Proceedings of the sixth Text REtrieval Conference (TREC-6). NIST Special Publication 500-240*, pages 1-24, Gaithersburg, MD. November 1998.

[135] Walker, S., Robertson, S.E., Boughanem, M., Jones, G.J.F., Sparck Jones, K. "Okapi at TREC-6 Automatic Ad hoc, VLC, routing, filtering and QSDR". In E.M. Voorhees and D. K. Harman, editors, *Proceedings of the sixth Text REtrieval Conference (TREC-6). NIST Special Publication 500-240*, pages 125-136, Gaithersburg, MD. November 1998.

[136] Wang, X., Wang, K., Li, Z. "Minimal Word Segmentation and its Algorithm", *Journal of Science.* 13: 1030-1032; 1989 (in Chinese)

[137] Webcrawler Home Page, "http://www.webcrawler.com" (visited on 29 August 2001)

[138] Wilkinson, R. "Chinese Document Retrieval at TREC-6". In E.M. Voorhees and D. K. Harman, editors, *The Sixth Textn REtrieval Conference (TREC-6)*, National Institute of Standards and Technology Special Publication 500-240, Gaithersburg, MD, November 1998.

[139] Willett, P. "Document retrieval experiments using indexing vocabularies of varying size. II. Hashing, truncation, digram and trigram encoding of index terms", *Journal of Documentation*, 35(4): 296-305; 1979

[140] Wong, S. K. M. and Yao, Y. Y. "A Probabilistic Method for Computing Term-by-Term Relationships", *Journal of the American Society for Information Science*, 44(8):431-339, 1993.

[141] Wong, S., Cai, Y., and Yao, Y. Computation of term associations by a neural network. In Korfhage, R., Rasmussen, E., and Willett, P., editors, *SIGIR 93:*

Proceedings of the Sixteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh. ACM Press. 1993.

[142] Wu, Z. and Tseng, G. "Chinese Text Segmentation for Text Retrieval: Achievements and Problems". *Journal of the American Society for Information Science*, 44(9):532–542, 1993.

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

[144] Yahoo Home Page *http://www.yahoo.com/* (visited on 29 August 2001)

[145] Yu, Ling et al., "辞海", *ShangHai CiShu Publisher*, 1989.

[146] Zhang, J., Gao, J. and Zhou, M. "Extraction of Chinese Compound Words - An Experimental Study on a Very Large Corpus", *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics Workshop on Chinese Language Processing*, pages 99–110, Hong Kong, October 2000.

[147] Zhang, Yushu et al., "康熙字典", published in Qing Dynasty (from 1644 to 1911).

[148] Zeng, Minzu, *Database Development and Chinese Information Needs*, London, Aslib, 1990.

Appendix A

TREC-5 and TREC-6 Chinese Track Topics

<top>

<num> Number: CH1

<E-title> U.S. to separate the most-favored-nation status from human rights issue in China.
<C-title> 美国决定将中国大陆的人权状况与其是否给予中共最惠国待遇分离

<E-desc> Description:
most-favored nation status, human rights in China, economic sanctions, separate, untie

<E-narr> Narrative:
A relevant document should describe why the U.S. separates most-favored nation status from human rights. A relevant document should also mention why China opposes U.S. attempts to tie human rights to most-favored-nation status.

<C-desc> Description:
最惠国待遇，中国，人权，经济制裁，分离，脱钩

<C-narr> Narrative:
相关文件必须提到美国为何将最惠国待遇与人权分离；相关文件也必须提到中共为什么反对美国将人权与最惠国待遇相提并论.

</top>

<top>

<num> Number: CH2

<E-title> Communist China's position on reunification
<C-title> 中共对于中国统一的立场

<E-desc> Description:
China, one-nation-two-systems, Taiwan, peaceful reunification, economic and trade cooperation, cross-strait relationship, science and technology exchanges

<E-narr> Narrative:
A relevant document should describe how China wishes to reach reunification through the implementation of "one-nation-two-systems." If a document merely states a foreign nation's support of China's sovereignty over Taiwan or discusses trade cooperation as well as cultural and technical exchanges between China and a country other than Taiwan, then the document is irrelevant.

<C-desc> Description:
中国，一国两制，台湾，和平统一，经贸合作，两岸关系，科技、文化交流

<C-narr> Narrative:
相关文件必须提到中共如何经由实现一国两制来达到台湾与大陆统一的目的.如果文件只是外国政府重申支持中共对台湾拥有主权或提到中共与其他国家之经贸、科技、文化交流，则为不相关文件.

</top>

<top>

<num> Number: CH3

<E-title> The operational condition of nuclear power plants in China.
<C-title> 中共核电站之营运情况

<E-desc> Description:
nuclear power plant, Daya Bay (nuclear power plant), Qinshan (nuclear power plant), safety

<E-narr> Narrative:
A relevant document should contain information on the current safety practices in China's nuclear power plants. Any article on safety regulations, accident reports and safety practices are relevant.

<C-desc> Description:
核电站，大亚湾，秦山，安全

<C-narr> Narrative:

相关文件必须提到中国目前投产的核电站的安全营运情况.
任何有关安全之规则或法令，安全措施之执行，意外事故
报告之文件皆属相关文件.

</top>

<top>

<num> Number: CH4

<E-title> The newly discovered oil fields in China.
<C-title> 中国大陆新发现的油田

<E-desc> Description:
oil field, natural gas, oil and gas, oil reserves, oil quality

<E-narr> Narrative:
A relevant document should contain information
on the oil reserves in the newly discovered oil fields
in Mainland China, any concrete description of
specific oil fields, or China's plan to develop these
fields.

<C-desc> Description:
油田，天然气，油气，储量，油质

<C-narr> Narrative:
相关文件必须提到中国大陆近几发现的油田的储量，各油田的
特点，以及中国开发油田的计划.

</top>

<top>

<num> Number: CH5

<E-title> Regulations and Enforcement of Intellectual Property Rights in China
<C-title> 中国有关知识产权的立法与政策以及执法情况

<E-desc> Description:
intellectual property rights, trade mark, copyright,
patent.

<E-narr> Narrative:
A relevant document should describe laws established in China to protect intellectual property rights. If a document contains information such as: China's violation of intellectual property rights as the basis for imposing trade sanctions against China; or, China taking up intellectual property rights as part of its economic reform, then the document is irrelevant.

<C-desc> Description:
知识产权法，商标法，著作权法，专利法

<C-narr> Narrative:
相关文件必须提到中国有关保护知识产权的法律.
非相关文件包括将中国违反知识产权作为对中国贸易制裁之依据或中国以知识产权作为经济改革的项目.

</top>

<top>

<num> Number: CH6

<E-title> International Support of China's Membership in the WTO
<C-title> 国际社会对中共加入世界贸易组织所给予之支持

<E-desc> Description:
World Trade Organization (WTO), GATT, market access, world trade structure, multilateral trade, member nation

<E-narr> Narrative:
A relevant document should indicate support given by specific nation(s) for China's membership in WTO.

<C-desc> Description:
世界贸易组织，关贸总协，市场准入，世界贸易体系，多边贸易，成员（国）

<C-narr> Narrative:
相关文件必须提到某一国家或某些国家对中国加入世界

贸易组织所给予之支持.

</top>

<top>

<num> Number: CH7

<E-title> Claims made by both PRC and Taiwan over islands in the South China Sea
<C-title> 中国大陆与台湾对南海诸岛的立场

<E-desc> Description:
The Spratly Islands, the Dongsha Islands, the Xisha Islands, China, Taiwan, sovereignty

<E-narr> Narrative:
A relevant document should include the following information: (1) why the Spratly Islands became the disputed area among China, the Philippines, Vietnam, and Indonesia; or (2) what are the natural resources found in the South China Sea; or/and (3) what are the sovereign rights claimed by the PRC and Taiwan; or/and (4) what are the suggestions proposed by the ASEAN to solve the territorial dispute over the Spratly Islands and South China Sea.

<C-desc> Description:
南沙（群岛），东沙（群岛），西沙（群岛），中国，台湾，主权

<C-narr> Narrative:
相关文件应包括下列信息：(1)为何南沙群岛成为中国、菲律宾、越南、印尼等国冲突的所在地； (2)南海有那些天然资源：(3)中国大陆与台湾对南海诸岛之主权立场为何；以及 (4)东盟国家对解决南沙群岛与南海争端有什么建议.

</top>

<top>

<num> Number: CH8

<E-title> Numeric Indicators of Earthquake Severity in Japan
<C-title> 地震在日本造成的损害与伤亡数据

<E-desc> Description:
Japan, earthquake, damage, death, injury, Richter scale

<E-narr> Narrative:
A relevant document should contain numeric indicators such as the magnitude of the earthquake, number of deaths or injuries, or property damage.

<C-desc> Description:
日本，地震，损失，死亡，级，受伤，芮氏地震仪

<C-narr> Narrative:
相关文件应包括地震的级数以及所造成的实际损害与伤亡数字，诸如地震在芮氏地震仪上的级数，死亡与受伤人数，以及以金钱为单位的财产损失数目.

</top>

<top>

<num> Number: CH9

<E-title> Drug Problems in China
<C-title> 中国毒品问题

<E-desc> Description:
narcotics, cocaine, heroin, marijuana, ton(s), kilogram(s), drugs use, drugs sale

<E-narr> Narrative:
A relevant document should contain information on drug problems in China, how the government cracks down on illegal drug activities, what types of drug rehabilitation program exist in China, and how the Chinese government cooperates with international organizations to stop the spread of drug trafficking.

<C-desc> Description:

毒品，可卡因，大麻，海洛因，吨，公斤，吸食毒品，
毒品买卖

<C-narr> Narrative:
相关文件应包括目前毒品在中国所造成的危害，中国打击非法买卖毒品的措施，有否有戒毒设施，以及中国是否与国际执法组织合作来遏制国际毒贩的走私活动．

</top>

<top>

<num> Number: CH10

<E-title> Border Trade in Xinjiang
<C-title> 新疆的边境贸易

<E-desc> Description:
Xinjiang, Uigur, border trade, market,

<E-narr> Narrative:
A relevant document should contain information on the trading relationship between Xinjiang, China and its neighboring nations, including treaties signed by China and former Soviet Republics that are bordering China and foreign investment. If a document contains information on how China develops Xinjiang, it is not relevant.

<C-desc> Description:
新疆，维吾尔，边境贸易，边贸，市场

<C-narr> Narrative:
相关文件必须包括中国新疆与其邻近国家的贸易关系，此关系包括中国与前苏联共和国之间所签署的贸易条约以及彼此间的外贸投资．如果文件只论及中国如何建设发展新疆，则属非相关文件．

</top>

<top>

<num> Number: CH11

<E-title> UN Peace-keeping Force in Bosnia
<C-title> 联合国驻波斯尼亚维和部队

<E-desc> Description:
Bosnia, Former Yugoslavia, Balkan, U.N.,
NATO, Muslim, weapon sanction, peace-keeping

<E-narr> Narrative:
A relevant document should contain information on
how UN peace-keeping troops carry out their mission
in the war-torn Bosnia.

<C-desc> Description:
波斯尼亚，前南斯拉夫,巴尔干,联合国,北约,武器禁运,维和,维持和平

<C-narr> Narrative:
相关文件必须包括联合国和平部队如何在战火蹂躏的波斯尼亚进行维持和平
的任务.

</top>

<top>

<num> Number: CH12

<E-title> World Conference on Women
<C-title> 世界妇女大会

<E-desc> Description:
UN, world, women's conference, women's issues,
women's status

<E-narr> Narrative:
A relevant document should contain information on
the 4th World Conference on Women, especially
on ways to improve women's social status and economic
situations through education and legislation.

<C-desc> Description:
联合国,世界,妇女大会,妇女问题,妇女地位

<C-narr> Narrative:
相关文件必须是关于第四届世界妇女大会中讨论的妇女问题,特别是经由教育

和立法来改进妇女的社会地位和经济情况的措施.

</top>

<top>

<num> Number: CH13

<E-title> China Bids for 2000 Olympic Games
<C-title> 中国争取举办西元2000年奥运

<E-desc> Description:
China, economic strength, Olympic games,
preparatory work

<E-narr> Narrative:
A relevant document should contain information on
how China bids for the 2000 Olympic Games,
China's reasons for sponsoring the 2000 Olympic games.

<C-desc> Description:
中国,经济实力,奥运,世界运动大会,奥林匹克,筹备工作

<C-narr> Narrative:
相关文件必须包括中国如何争取举办西元2000年奥运,中国所持的理由为何.
中国选手在奥运会中的表现属于不相关文件.

</top>

<top>

<num> Number: CH14

<E-title> Cases of AIDS in China
<C-title> 中国的爱滋病例

<E-desc> Description:
China, Yunnan, AIDS, HIV, high risk group, syringe, virus

<E-narr> Narrative:
A relevant document should contain information on
the areas in China that have the highest AIDS
cases, how the AIDS virus was transmitted, and

how the Chinese government combats AIDS
problem.

<C-desc> Description:
中国,云南,爱滋病,HIV,高危险群患者,注射器,病毒

<C-narr> Narrative:
相关文件应当包括中国那些地区的爱滋病例最多,爱滋病毒在中国是如何传播的,以及中国政府如何监测爱滋病并控制它的传染.

</top>

<top>

<num> Number: CH15

<E-title> The UN peace-keeping troops help Haiti return to democracy
<C-title> 联合国维和部队如何帮助海地恢复民主制度

<E-desc> Description:
Haiti, UN, U.S., multination-troops, peace-keeping troops,
democracy

<E-narr> Narrative:
A relevant document should contain information on the U.S. efforts to help Haiti resume its democracy, UN resolutions on Haiti, and the Latin-American nations reactions to the UN resolutions.

<C-desc> Description:
海地,联合国,美国,多国部队,维和部队,民主

<C-narr> Narrative:
相关文件必须提到美国如何帮助海地民主政府重建海地;联合国安理会对海地问题之决议,以及拉美国家对联合国决议之反应.
不相关文件则为海地仅为新闻或电视广播提要,或新闻分析中提及海地但新闻主题不在海地.

</top>

<top>

<num> Number: CH16

<E-title> The Debate of UN Sanctions Against Iraq
<C-title> 联合国对伊拉克经济制裁的辩论

<E-desc> Description:
UN, Iraq, economic sanction

<E-narr> Narrative:
A relevant document should contain information on why the UN carries out economic sanctions against Iraq; the impact of the economic sanctions on Iraq; the UN debate on when to lift the sanctions; Iraq's reaction to the sanctions. An irrelevant document is such that it only mentions the UN sanctions against Iraq but does not give any details on the discussions, impact, and Iraq's reaction about the sanctions. Non-relevant documents include summaries without any details like the French government's setting up a representative office in Iraq thus reducing its economic sanctions toward Iraq, Iran's criticizing of the UN sanctions when seeking diplomatic relations with Iraq, or UN sanctions against Iraq.

<C-desc> Description:
联合国,伊拉克,经济制裁

<C-narr> Narrative:
相关文件应提到联合国为何对伊拉克实施经济制裁; 经济制裁 对伊拉克的影响;联合国对何时解除此经济制裁的辩论;以及 伊拉克对经济制裁的反应.不相关文件为法国为了在伊拉克设代表处而减少其对伊之经济 制裁;
中国对联合国在中东维和行动的评论;伊拉克与伊朗关系正常化中批评联合国之制裁;
或联合国对伊拉克之经济制裁仅为新闻提要而未详细报道.

</top>

<top>

<num> Number: CH17

<E-title> China's Expectations about APEC
<C-title> 中国对亚太经济合作组织的期望

<E-desc> Description:
APEC, China, GATT, WTO

<E-narr> Narrative:
A relevant document should contain information on China's economic growth; the importance of China in the development of economics and trade in the Asian-Pacific region; and China's efforts in resuming its status as a signatory state of GATT and a member nation of the WTO. An irrelevant document only mentions APEC when discussing bilateral trade relations with other nations but does not give details on why China wants to be a member of WTO.

<C-desc> Description:
亚太经济合作组织,中国,关贸总协,世界贸易组织

<C-narr> Narrative:
相关文件应提到中国之经济成长;中国在亚太地区经济贸易发展的地位;中国为恢复关贸总协缔约国地位以及成为世界贸易组织成员国所做的努力.不相关文件为中国与外国代表讨论双边经贸关系提及亚太经济合作组织,但未谈具体方案者.

</top>

<top>

<num> Number: CH18

<E-title> The Mid-East Peace Talks
<C-title> 中东和平会议

<E-desc> Description:
Israel, Palestine, the Mid-East, peace talks

<E-narr> Narrative:
A relevant document should contain information on what the United States hopes to achieve in the Mid-East peace talks; how many countries participate in the peace talks; what is the agenda to be discussed; Arab nations positions toward Israel; and the Chinese view on the

peace talks.
A non-relevant document mentions the support of leaders of the Western nations and China for the Mid-East peace talks, but it does not contain information on the crux of the problem and how to solve it.

<C-desc> Description:
以色列,巴勒斯坦,中东,和平会议

<C-narr> Narrative:
相关文件:应包括美国对中东和平会议的期望,哪些国家出席中东和平会议,主要讨论的议题为何,阿拉伯国家对以阿冲突的态度,以及中国对整个中东问题的看法.不相关文件:如果文件只是西方首脑或中国领导人表示支持召开中东和平会议,但是未提到中东问题的症结和解决中东和平问题的的建议,则属不相关文件.

</top>

<top>

<num> Number: CH19

<E-title> Project "Hope"
<C-title> 希望工程

<E-desc> Description:
China, Project Hope, educational level, education

<E-narr> Narrative:
A relevant document should contain information on Project Hope's objectives and its results. Any document containing information on raising teachers pay, improving remote areas' education, educational reform laws, or the amount of private contributions to Project Hope is relevant.
An irrelevant document mentions Project Hope but does not provide any concrete data on the success of the project such as how each area carries out the Project and how many people have benefited from it.

Documents such as letters to the editor asking where to donate money for the Project are irrelevant. Documents that mention educational reform but do not give concrete measures are also irrelevant.

<C-desc> Description:
中国，希望工程，文化程度，教育

<C-narr> Narrative:
相关文件应提到希望工程是什么，它的目标为何，实施成果如何.有关改进教师待遇，文化扶贫工作与捐款等文件亦属相关文件.不相关文件包括听众信箱之问题，或文件提到教育法但未提具体法案内容，或仅提希望工程之名但没有具体数据以及推行办法者.

</top>

<top>

<num> Number: CH20

<E-title> U.S. Military Personnel Missing in Action in Vietnam
<C-title> 越战失踪美军

<E-desc> Description:
Vietnam, MIA's

<E-narr> Narrative:
A relevant document presents any information on U.S. soldiers missing in action in Vietnam. Document topics include missions to Vietnam, inter-government cooperation and discussions, effect on lifting the trade embargo, the Vietnamese Government's reaction to U.S. statistics, MIA statistics, resolved cases, etc.

<C-desc> Description:
越南，失踪美军

<C-narr> Narrative:
相关文件：应包括任何有关美国军人在越南失踪的信息，包括美军在越南的任务，美越政府间有关此问题的合作与讨论，以及美国停止对越南贸易

制裁的影响.此外,越南政府对美国有关在越
战中失踪军人的统计数字的反应与已经解决的
案件等信息亦属相关文件.

</top>

<top>

<num> Number: CH21

<E-title> The Role of the Governor of Hong Kong
in the Reunification with the PRC
<C-title> 香港总督彭定康在香港回归中国一事上所扮演的角色

<E-desc> Description:
Hong Kong issue, special administrative zone, Peng DingKang, plan, proposal
<E-narr> Narrative:
A relevant document presents information on the role
of the Governor of Hong Kong, Peng DingKang, in the
reunification of China. Issues include any of the
Governor's announcements, his official visits to China
and meetings with Chinese officials, PRC criticism of
Peng's legislative plans or proposals, etc. Non-relevant
documents discuss any reactions to the Governor's
actions or his politics in Hong Kong reunification from
sources other than Hong Kong, UK, or the PRC.

<C-desc> Description:
香港问题, 特别行政区, 彭定康, 计划, 建议

<C-narr> Narrative:
相关文件:应包括香港总督彭定康在香港问题上所
扮演的角色,包括所有彭定康发表过的声明,彭定康
到中国访问与中国政府官员的谈话,以及中国政府对
彭定康提出的有关香港立法改革的批评等.
不相关文件:任何非来自香港,英国,或中国的有关
彭定康的评论皆属非相关文件.

</top>

<top>

<num> Number: CH22

<E-title> The Spread of Malaria Infection in Various Parts of the World
<C-title> 世界各地感染疟疾的情况

<E-desc> Description:
malaria, number of deaths, number of infections

<E-narr> Narrative:
A relevant document presents numeric information about malaria infection or death rate at a national or international level. Non-relevant documents discuss health policies related to communicable diseases or vaccination against malaria without numeric information.

<C-desc> Description:
疟疾，死亡人数，感染病例

<C-narr> Narrative:
相关文件应包括有关世界各地感染疟疾的情况，包括病例统计与死亡人数．凡属讨论与传染性疾病有关的卫生政策或预防疟疾之疫苗接种而未提及感染或死亡人数的资料则为非相关文件．

</top>

<top>

<num> Number: CH23

<E-title> Soviet Union's Mediation Role in the Gulf War
<C-title> 苏联在海湾战争中如何担任调停的角色

<E-desc> Description:
Soviet Union, Gulf War, peace proposal, Iraq

<E-narr> Narrative:
A relevant document discusses the Soviet Union's mediation in the Gulf War, including communication with Iraq, cease-fire resolution to the UN Security Council and their peace proposal for withdrawal of multi-national troops, etc.

<C-desc> Description:
苏联，海湾战争，和平建议，伊拉克

<C-narr> Narrative:
相关文件应提及苏联在海湾战争中如何担任调停
的角色，包括与伊拉克之间的沟通，苏联在联合国
安理会中提出的停火协议
以及要求多国部队从伊拉克撤出的和平建议.

</top>

<num> Number: CH24

<E-title> Reaction to Lifting the Arms Embargo for Bosnian Muslims
<C-title> 对取消向波黑穆斯林武器禁运的反应

<E-desc> Description:
Bosnia-Herzogovenia, Muslims, arms embargo, United Nation's Security Council
<E-narr> Narrative:
A relevant document discuss international reaction to lifting
the international arms embargo against the Former Yugoslavia.
Document topics include statements in support or opposition
by Government officials or officials of international
organizations, pressure from U.S. legislative initiatives, etc.

<C-desc> Description:
波黑，波斯尼亚-黑塞哥维那，穆斯林，武器禁运，
安理会，联合国安理会

<C-narr> Narrative:
相关文件应提及国际社会对取消向前南斯拉夫
武器禁运的反应.文件内容应包括各国政府或
国际组织官员对武器禁运所持的正反意见，
以及美国国会反对武器禁运而对联合国施加压力等.

</top>

<top>

<num> Number: CH25

<E-title> China's Protection of Pandas
<C-title> 中国对熊猫的保护

<E-desc> Description:
Ecoprotection, panda, nature preserve, endangered species

<E-narr> Narrative:
A relevant document discusses China's
protection of pandas, such as how the
Government sets up nature preserves for
pandas, existing nature preserves,
the nature preserve environment, the total
number of pandas in China, or increases in
the panda population. An irrelevant document
covers panda sighting, without any details
about protective measures, like how the
Government is helping pandas to reproduce.

<C-desc> Description:
生态保护，熊猫，保护区，濒临灭绝

<C-narr> Narrative:
相关文件应提到中国对熊猫的保护，
比如中国政府如何设立熊猫的保护区，
目前熊猫的保护区包括那些地区；
熊猫的生态环境如何；目前中国的熊猫
总数大约有多少；以及受到保护后
熊猫数量的增长.不相关文件则包括新闻中
只提到在某个地区看到熊猫,但是
没有提出具体的保护方法,诸如政府如何
设立保护区来帮助熊猫的繁殖.

</top>

<top>

<num> Number: CH26

<E-title> Measures to Prevent Forest Fires in China
<C-title> 中国森林火灾的防范措施

<E-desc> Description:
Mongolia, Manchuria (Northeast China) forest, fire, raging fires,

<E-narr> Narrative:

A relevant document presents causes for forest fires in China, the area affected, acreage damaged, number injured and dead, or preventive measures adopted by the Chinese Government. Any document without the abovementioned information is not relevant.

<C-desc> Description:
蒙古，东北，森林，火灾，大火

<C-narr> Narrative:
相关文件应提及中国森林火灾发生的原因，
发生地区，受害面积，受伤与死亡人数，
以及政府采取什么样的防范措施．
如果没有上述的信息则属不相关文件．

</top>

<top>

<num> Number: CH27

<E-title> Robotics Research in China
<C-title> 中国在机器人方面的研制

<E-desc> Description:
robotics, automation

<E-narr> Narrative:
A relevant document should have the following information: the functions of manufactured robots in China, the universities and institutes that are involved in robotic research, or direction of the research.

<C-desc> Description:
机器人，自动控制

<C-narr> Narrative:
相关文件应提供下列的信息：
中国研制成功的机器人主要有什么功用，
有那些大学与研究 机构参与机器人的研究设计，
研究的方向为何．

</top>

<top>

<num> Number: CH28

<E-title> The Spread of Cellular Phones in China
<C-title> 移动电话在中国的成长

<E-desc> Description:
digital, cellular, cellular phone, net, automatic roaming

<E-narr> Narrative:
A relevant document contains the following kinds of information: the number of cellular phone users, area coverage, or how PSDN is implemented for national cellular communication. A non-relevant document includes reports on commercial manufacturers or brand name cellular phones.

<C-desc> Description:
数字,蜂窝式,移动电话,网络,自动漫游

<C-narr> Narrative:

相关文件应包括下列信息：中国移动电话用户数，覆盖地区，中国如何以数据分组交换网覆盖全国移动电话的通讯．不相关文件则包括 有关制造移动电话厂商的报道，以及移动电话的厂牌．

</top>

<top>
<num> Number: CH29

<C-title> 信息高速公路的建设

<E-title> Building the Information Super Highway

<C-desc> Description:
信息高速公路，建设

<E-desc> Description:
Information Super Highway, building

<C-narr> Narrative:
相关文件应提到信息高速公路的建设，包括任何技术上的，或与信息基础设施有关的问题，以及为已开发国家或开发中国家， 甚至国际网络的应用的计划.

<E-desc> Description:
A relevant document should discuss building the Information Super Highway, including any technical problems, problems with the information infrastructure, or plans for use of the Internet by developed or developing countries.

</top>

<top>
<num> Number: CH30

<C-title> 中国旅游业的发展，

<E-title> The Development of the Tourist Industry in China 1983-1993

<C-desc> Description:
旅行社， 旅游业，旅游客，收入，外汇收入，

<E-desc> Description:
tourist agency, tourist industry, tourist, revenue, foreign exchange revenue

<C-narr> Narrative:
相关文件应提及1983-1993中国旅游业的成长．包括国内外旅客的人数．营业收入，并且就1983-1993海外旅客与外汇收入作比较．有关新旅馆的建设以及中国旅游服务的改进，如利用电脑网络订位以及提供旅游信息的文件亦属相关文件．

<E-narr> Narrative:
A relevant document should discuss the growth of the tourist industry in China and quantify that trade in terms of the total number of domestic and foreign tourists and revenue in any year betweeen 1983-1993. Moreover, it should compare the amount of foreign exchange revenue generated by foreign tourists in any year between 1983-1993. Any discussion of the construction of new hotels and service improvement in the Chinese tourist industry, such as providing information and making reservations through the computer network makes the document a relevant one.

</top>

<top>
<num> Number: CH31

<C-title> 美国政府对古巴难民的新政策

<E-title> New U.S. Government policy concerning Cuban Refugees

<C-desc> Description:
古巴，美国，非法移民，移民政策，难民

<E-desc> Description:
Cuba, U.S., illegal immigrant, immigration policy, refugee

<C-narr> Narrative:
相关文件应提及克林顿政府针对大量古巴难民涌入美国所制定的新难民政策以及卡斯特罗对此政策之批评．有关在美的合法与非法古巴移民人数之统计，60年代与90年代古巴难民潮产生之背景，以及任何外国政府对克林顿政府新古巴难民政策批评的文件亦属相关文件．

<E-narr> Narrative:
A relevant document should discuss the new official U.S. Government policy toward Cuban immigration and Castro's reaction to the policy. Any document that contains statistics o f legal and illegal Cuban immigrants in the United States, the differences between the 1 960's and 1990's Cuban refugee waves, as well as foreign Government criticism of the Cli nton administration's policy on the Cuban refugees is also relevant.

</top>

<top>
<num> Number: CH32

<C-title> 拉丁美洲的贩毒集团

<E-title> Drug Traffikers in Latin America

<C-desc> Description:
贩毒集团，卡利贩毒集团，麦德林贩毒集团，拉丁美洲，走私，贩毒，毒品市场，洗钱

<E-desc> Description:
Drug traffikers, Cali Cartel, Medina Cartel, Latin America, smuggling, drug selling, dr

ug market, money laundering

<C-narr> Narrative:
相关文件应提及贩毒集团在中南美洲（拉丁美洲）的贩毒活动，特别是在哥伦比亚．巴拿马，墨西哥．讨论贩毒集团走私武器与颠覆政府之活动亦属相关文件．

<E-narr> Narrative:
A relevant document describe activities related to drug traffikers in Latin America, especially in Colombia, Panama, and Mexico. A document that discusses drug traffiers' activities of arms smuggling and overturning Governments is also relevant.

</top>

<top>
<num> Number: CH33

<C-title> 两岸劫机

<E-title> Hijackings between Taiwan and the Mainland

<C-desc> Description:
两岸，劫机，劫机犯，海基会，海协会

<E-desc> Description:
cross-strait hijackings, hijackers, Strait Exchange Foundation, Association for Relations Across the Strait

<C-narr> Narrative:
相关文件必须提到劫机者与被劫持之飞机以及旅客，诸如劫机动机，劫机过程中有无伤亡，及对劫机者之判刑．若文件只提及海基会与海协会对两岸劫机犯遣返问题之协商而非针对某一特定劫机事件之处理则属非相关文件．

<E-narr> Narrative:
A relevant document should describe some aspect of a specific airline hijacking from the Mainland to Taiwan, such as the hijackers motive, casualty or deaths during the hijacking, the sentencing of the hijackers. Discussions about the return of hijackers in the context of Taiwan-Mainland talks are not relevant unless a specific hijack event is described.

</top>

<top>
<num> Number: CH34

<C-title> 旱灾在中国造成的影响

<E-title> The Impact of Droughts in China

<C-desc> Description:
旱灾，干旱地区，救灾款，粮食总产，面积，雨量，中国

<E-desc> Description:
drought, arrid region, relief assistance, food production, area, rainfall, China

<C-narr> Narrative:
相关文件应提到旱灾在中国造成的影响，包括受灾地区，受害人数，受灾农地之面积，以及干旱对农作物与畜牧业所造成的损失．讨论政府帮助农牧民救灾的措施亦属相关文件．

<E-narr> Narrative:
A relevant document should discuss the impact of droughts in China. Concrete indicators of impact include areas, number of people and acerage affected as well as total loss of crops and livestock. Any documents that discuss the Chinese Governmental relief assistance and measures of combating droughts are also relevant.

</top>

<top>
<num> Number: CH35

<C-title> 一九九四年南非总统大选前之事件

<E-title> Acts of Violence in South Africa Prior to the April 27 Presidential Election

<C-desc> Description:
暴力事件，暴力冲突，种族隔离，南非，四月二十七日，全民大选，暴乱地区，屠杀，曼德拉，

<E-desc> Description:
violent events, violent conflict, apartheid, South Africa, April 27, General Election, riot area, massacre, Mandela

<C-narr> Narrative:
相关文件应提到1994年４月２７日南非总统大选之前各地所发生的暴力事件以及暴力事件发生的原因与地区．提及参与暴力事件的团体及南非政府对平息暴力所作的努力的文件亦为相关文件．

<E-narr> Narrative:
A relevant document should discuss the violence, the causes and areas affected in Sout h Africa prior to the April 27 South African Presidential election. Any document that di scusses groups participating in the riots and South African Government's efforts to quel l riots is also relevant.

</top>

<top>
<num> Number: CH36

<C-title> 中国对外贸易的成长

<E-title> The Growth of China's Foreign Trade

<C-desc> Description:
对外经贸，出口，进口，进出总额，外汇，外贸，国际市场，出口商品，进口商品，中国

<E-desc> Description:
foreign trade, exports, imports, total amount of trade, foreign exchange, foreign funds, international market, export product, import product, China

<C-narr> Narrative:
相关文件应提到中国外贸政策，进出口总额，出口何种商品，进口何种商品，中国在国际市场上的竞争力，与台湾香港的贸易关系，对外贸易成长的百分比，以及主要输出国与输入国.

<E-narr> Narrative:
A relevant document should discuss: (1) China's foreign trade policy, (2) total amount o f trade, (3) export products, (4) import products, (5) China's competitiveness in the in ternational markets, (6) China's trade relationship with Taiwan and Hong Kong, (7) growt h of China's foreign trade in percentage, or (8) major export and import countries.

</top>

<top>
<num> Number: CH37

<C-title> 日本泡沫经济的破灭

<E-title> The Collapse of the Bubble Economy in Japan

<C-desc> Description:
泡沫经济，破灭，不景气，经济衰退，经济复苏

<E-desc> Description:
bubble economy, collapse, recession, economic downturn, economic recovery

<C-narr> Narrative:
相关文件应提到自泡沫经济破灭後，日本所经历的经济复苏,特别是金融，房地产业与企业的萧条,以及日本政府为刺激经济复苏所采取的政策．对日本经济成长的预测亦属相关文件．

<E-narr> Narrative:
A relevant document should discuss the economic recession in Japan after the collapse of the bubble economy, especially in the areas of finance, real estate, and industry, and the Japanese government's policy to stimulate economy recovery. Discussions of the predictions of Japanese economic growth are also relevant.

</top>

<top>
<num> Number: CH38

<C-title> 中国野生动物保护形势

<E-title> Protection of Wildlife in China

<C-desc> Description:
野生动物保护，《野生动物保护法》，野生动物保护协会，珍稀动物，濒危动物

<E-desc> Description:
Protection of Wildlife, Legislation Protecting Wildlife, Associations for the Protection of Wildlife, rare and precious animals, endangered species

<C-narr> Narrative:
相关文件应提到中国野生动物保护形势．相关文件包括下列信息：(一)《野生动物保护法》,(二）珍稀动物，(三）捕猎和销售野生动物，(四）采取措施抢救珍稀动物，(五）市场管理工作，或（六)建设濒危动物饲养繁殖研究基地．

<E-narr> Narrative:
A relevant document should discuss protection of endangered species in China. Relevant documents include the following information: (1) "Legislation protecting endangered species", (2) rare and precious animals, (3) hunting and selling wild animals, (4) adopting m

easures to rescue rare animals, (5) market surveliance work, or (6) establishing preser vation grounds for endangered species.

</top>

<top>
<num> Number: CH39

<C-title> 在阿尔及利亚发生恐怖活动

<E-title> Terrorism in Algeria

<C-desc> Description:
阿尔及利亚，恐怖主义，宵禁，暗杀，反对党，紧急状态

<E-desc> Description:
Algeria, terrorism, curfew, assasination, opposition party, state of emergency

<C-narr> Narrative:
相关文件必须提到在阿尔及利亚发生恐怖活动，此活动包括阿尔及利亚当局对恐怖主义采取的措施，与反对党领袖举行会谈，或者恐怖暴力活动.

<E-narr> Narrative:
A relevant document should discuss terrorist activity in Algeria, including the Algerian authorities measures against terrorism, discussions with the opposition party, or viole nt terrorist activity.

</top>

<top>
<num> Number: CH40

<C-title> 省采取有效措施减轻农民负担

<E-title> Provincial effective measures to Lighten the Burden for Peasants

<C-desc> Description:
农民负担，三乱，收费，减轻，省

<E-desc> Description:
the burden for peasants, three turmoils, fee collection, lighten, province

<C-narr> Narrative:
相关文件必须提到中央在某些省落实减轻农民负担的切实有效措施，诸如税收政策的改进减轻向农民税收的情形或监督减轻非法收费的执行．如果文件只论及农民负担的问题或增加农民负担的原因，则属非相关文件．

<E-narr> Narrative:
A relevant document should discuss effective policies to lighten the burdent for peasants in specific provinces. Relevant documents include discussion of changes in taxation policy, indications of reduction of tax burden, or supervision to reduce indiscriminate taxing. If a document merely describes the problem, or reasons for the increased burden, it is not relevant.

</top>

<top>
<num> Number: CH41

<C-title> 京九铁路的桥梁隧道工程

<E-title> Bridge and Tunnel Construction for the Beijing-Kowloon Railroad

<C-desc> Description:
京九铁路，桥梁，隧道，贯通，特大桥，

<E-desc> Description:
Beijing-Kowloon Railroad, bridge, tunnel, connection, very large bridge

<C-narr> Narrative:
相关文件必须提到京九铁路的桥梁隧道工程，包括地点、施工阶段、长度．

<E-narr> Narrative:
A relevant document discusses bridge and tunnel construction for the Beijing-Kowloon Railroad, including location, construction status, span or length.

</top>

<top>
<num> Number: CH42

<C-title> 七大江河的防洪水库大堤

<E-title> Dikes and Reservoirs in Flood Prevention in the Seven Great Rivers

<C-desc> Description:
长江、黄河、淮河、海河、珠江、辽河、松花江等七大江河的防洪,防汛，水库，堤，坝，

<E-desc> Description:
Flood prevention on the Seven Great Rivers, Yangtze River, Yellow River, Huaihe River, Haihe River, Pearl River, Liaohe River, Songhua River, flood control, reservoir, dike, embankment

<C-narr> Narrative:
相关文件必须提到提高七大江河地区的某一些水库与防堤．相关文件应包括下列信息：（一）建设项目；（二）抗洪抢险（三）水库水位以及（四）开闸泄洪．凡三峡工程则属非相关文件．

<E-narr> Narrative:
A relevant document should discuss specific dikes and reservoirs in the Seven Great Rivers region. Relevant documents discuss the following information: (1) construction projects, (2)measures for flood and rescue work, (3) reservoir water levels, or (4) flood discharging. Documents discussing the Three Gorge Project are non-relevant.

</top>

<top>
<num> Number: CH43

<C-title> 十四世达赖喇嘛

<E-title> The Fourteenth Dali Lama

<C-desc> Description:
十四世达赖喇嘛，西藏

<E-desc> Description:
The fourteenth Dali Lama, Tibet

<C-narr> Narrative:
相关文件必须提到十四世达赖喇嘛的生活或活动以及其对西藏独立的立场．任何有关中央政府对达赖喇嘛的立场之文件皆属相关文件．讨论到其他政府对达赖喇嘛的立场的文件亦属相关文件．

<E-narr> Narrative:
A relevant document should discuss the life the Dali Lama, his activities, or his stand

on Tibetan indepedence. Articles on the position of the Chinese Government toward the Da li Lama are relevant. Documents discussing the position of other Governments toward the Dali Lama are also relevant.

</top>

<top>
<num> Number: CH44

<C-title> 三峡工程与移民

<E-title> The Three Gorges Project and Resettlement

<C-desc> Description:
三峡工程，移民，

<E-desc> Description:
The Three Gorges Project, resettlement

<C-narr> Narrative:
相关文件必须提到移民政策，如何执行及移民反应．任何环境及文化的负面影响则属非相关文件．

<E-narr> Narrative:
A relevant document should discuss the resettlement plan, the implementation, and reaction of the resettlement population. Non-relevant documents discuss the environmental and cultural impact.

</top>

<top>
<num> Number: CH45

<C-title> 中国红十字会

<E-title> China Red Cross

<C-desc> Description:
中国红十字会，救济物资，援助，捐款，赈济，

<E-desc> Description:
China Red Cross, providing relief goods and materials, aiding, donating, relieving (disa

ster victims)

<C-narr> Narrative:
相关文件必须提到中国红十字会的各种活动如援助及被援助种类，作为中间人的活动包括所扮演的角色及其功能，以及中国红十字会各种活动的贡献.

<E-narr> Narrative:
A relevant document should discuss the activities of the China Red Cross, including the type of aid and the recipient. For relevant documents in which the China Red Cross is an intermediary, the document should describe the role or function that the China Red Cross is performing and the beneficiary of the activity.

</top>

<top>
<num> Number: CH46

<C-title> 中越两国关系的新进展

<E-title> New advances in the Relationship between China and Vietnam

<C-desc> Description:
中越关系，越南，正常化，经济合作，民间互市，交流，协议

<E-desc> Description:
Sino-Vietnamese relations, Vietnam, normalization, economic cooperation, nongovernmental border and port trade, exchanges, agreements

<C-narr> Narrative:
相关文件应提到中越两国关系正常化后的新进展．文件须提到边民的互市，原则性协议的达成，经贸、科技、文教等领域的交流与合作，以及柬埔寨问题的解决.

<E-narr> Narrative:
A relevant document should discuss new advances in the Sino-Vietnamese relationship after normalization. Relevant documents should identify border trade; basic agreements reached between the two countries; exchanges and cooperation regarding economy and trade, science and technology, or culture and education; or resolution of the Campuchea problem.

</top>

<top>

<num> Number: CH47

<C-title> 1 9 9 1 年菲律宾皮纳图博火山爆发造成的后果

<E-title> The Impact of the 1991 Mount Pinatubo Volcano

<C-desc> Description:
菲律宾，皮纳图博火山，火山灰，岩浆，爆发

<E-desc> Description:
Philippines, Mount Minatubo, volcanic ash, magma, eruption

<C-narr> Narrative:
相关文件应提到以下信息:北半球的气侯，火山周围居民的撤离，火山爆发造成的伤亡和损失，美国苏比克海军基地与克拉克空军基地的损害和臭氧层的破坏.

<E-narr> Narrative:
A relevant document should discuss the following kinds of information: weather in the Northern Hemisphere; evacuation of citizens; casualties, deaths, and losses resulting fr om the eruption; damage to U.S. Subic Bay Naval base and Clark Air Force base; or dama ge to the ozone layer.

</top>

<top>
<num> Number: CH48

<C-title> 海湾战争之后的科威特石油业

<E-title> Kuwaiti Oil Industry after the Gulf War

<C-desc> Description:
科威特，海湾战争，油井，石油生产，石油业，

<E-desc> Description:
Kuwait, Gulf War, oil well, oil production, oil industry

<C-narr> Narrative:
相关文件应提到海湾战争对科威特石油业所造成的经济损失与科威特如何恢复石油生产两大方面．经济损失方面包括燃烧油井的数量，灭火工作的进行以及中国参加灭火工作的情形．恢复石油生产方面须提及战后重建工作包括各种建设工程，与各国签订合同等.

<E-desc> Description:
A relevant document should discuss the economic losses and recovery of the Kuwaiti oil i
ndustry after the Gulf War. Economic losses include the number of burning oil fields, t
he efforts to extinguish fires, and the Chinese firefighters work. The recovery of oil p
roduction includes post-war rebuilding such as construction and contracts with various c
ountries.

</top>

<top>
<num> Number: CH49

<C-title>

<E-title> 中国对核裁军立场

<C-desc> Description:
中国，核试验，核裁军，销毁核武器，《不扩散核武器条约》，《削减战略武器条约》

<E-desc> Description:
China, nuclear tests, nuclear disarmament, destruction of nuclear weapons, Non-Prolifera
tion Treaty, START treaty

<C-narr> Narrative:
相关文件应提到中国对核裁军的立场，包括中国如何履行不扩散核武器的义务；中国如何发展核武器与地下核试验以及中国如何不帮助无核国家发展核武器而促进国际核能的和平利用．若文件提及《不扩散核武器条约》的延长问题或中国对赞成别国加入《不扩散核武器条约》亦属相关文件．提及《削减战略武器条约》的文件则属非相关文件．

<E-desc> Description:
A relevant document should discuss China's position on nuclear disarmament, including ho
w China fulfills its commitment to non-proliferation, how China is developing its own n
uclear program and underground nuclear tests, or how China is not helping non-nuclear co
untries to develop nuclear weapons but promotes international peaceful use of nuclear
power. If a document discusses the extension of the non-proliferation treaty or China's
approval of a country's becoming a treaty member, it is relevant. Non-relevant documents
discuss the START treaty.

</top>

<top>

<num> Number: CH50

<C-title> 关于中国与英国政府在香港新机场问题上所达成的谅解

<E-title> China and Britain Reach an Understanding regarding the New Airport in Hong Kong

<C-desc> Description:
中国，英国，新机场，建设

<E-desc> Description:
China, Britain, new airport, construction

<C-narr> Narrative:
相关文件应提到新机场问题产生的背景，中方为何反对香港新机场的建设，以及中英《关于香港新机场建设及有关问题的谅解备忘录》之内容为何.

<E-desc> Description:
A relevant document should discuss the setting in which the problem of the airport arose, why the Chinese opposed the construction of the the new airport, or the contents of the memorandum of agreement regarding issues related to the construction of the new airport in Hong Kong.

</top>

<top>
<num> Number: CH51

<C-title> 中国对保护环境的政策

<E-title> China's Policy of Protecting the Environment

<C-desc> Description:
中国，环境，保护，酸雨，大气污染，水污染，空气污染，经济

<E-desc> Description:
China, environment, protection, acid rain, air pollution, water pollution, air pollution, economy

<C-narr> Narrative:
相关文件应提到中国对环境保护的政策. 相关文件应包括下列信息：（一）造成环境污染的因素及其对环境危害的程度，（二）经济成长与环境污染之间的相关性，或（三）中国政府对环境所制定的立

法与政策．若文件提及世界性的环境问题或中国以外的环境污染问题则属非相关文件．

<E-desc> Description:
A relevant document should discuss China's policy toward environmental protectionism. Relevant documents include the following information: (1) the reasons for and extent of the pollution, (2) the relationship between economic growth and environmental pollution, or (3) the legislation and policies formaulated by the Chinese Government. Non-relevant documents discuss global environmental problems or problems with environmental pollution in other countries.

</top>

<top>
<num> Number: CH52

<C-title> 中国房地产业的改革与发展

<E-title> Reform and Growth in China's Real Estate Industry

<C-desc> Description:
中国，房地产业，投资，规模，交易，转让，炒卖，暴利，

<E-desc> Description:
China, real estate industry, investment, scale, trade, transferring possession, wild selling, huge profits

<C-narr> Narrative:
相关文件应提到中国房地产业所面临的问题以及政府采取何种措施来促进房地产业的建康发展．房地产业的问题包括炒买炒卖地产，开发规模过大，投资者获取超额的利润，国有土地出让过多过滥，交易行为不规矩，交易价格混乱等．政府对房地产业所采取宏观管理则包括开征土地增值税，实施土地使用管制，颁布城市房地产业管理法等来促进房地产业建康发展之措施．有关中国住房制度改革之文件亦属相关文件．

<E-desc> Description:
A relevant document should discuss problems faced in the real estate industry and the various measures adopted by the Government to promote healthy growth for the industry. Problems in the real estate industry include wild buying and selling of real estate, developing at an excessive scale, investors obtaining excessive profits, excessive and indiscriminate selling of public lands, unrestricted trade practices, trade price speculation, etc. Growth policies being adopted by the Government for macro-management of the industry include measures to promote the healthy growth in the industry by collecting a value added tax on land, implementing controls for land use, promulgating the urban contr

ol of land use law, etc. Documents about the reform of the housing system are also relev ant.

</top>

<top>
<num> Number: CH53

<C-title> 中国汽车工业的发展与市场

<E-title> The Development of the Chinese Auto Industry, and the Chinese Auto Market

<C-desc> Description:
中国，生产，制造，汽车，汽车工业，汽车市场

<E-desc> Description:
China, production, manufacture, auto, auto industry, auto market

<C-narr> Narrative:
相关文件应提到中国政府对发展汽车工业之计划，包括如何吸引外资与技术，计划生产车辆类型与年产量以及中国内市场对汽车需求．有关中国政府为保护本国汽车工业发展所制定的政策亦属相关文件．

<E-desc> Description:
A relevant document should discuss the Chinese Government's plan to develop the auto in dustry, including how to attract foreign investment and technology, or how to plan for production of vehicle types and annual output as well as for the demand in the domestic auto market. Documents which discuss policies formulated to protect the Chinese auto ind ustry are also relevant.

</top>

<top>
<num> Number: CH54

<C-title> 中国关于美国政府向台湾出售Ｆ－１６战斗机的反应

<E-title> China's Reaction to U.S. Sale of F-16 Fighters to Taiwan

<C-desc> Description:
中国，美国，台湾，Ｆ－１６战斗机，出售

<E-desc> Description:
China, U.S., Taiwan, F-16 fighter, sale

<C-narr> Narrative:
相关文件应提到中美“八·一七”联合公报中对美国向台湾出售武器之决定，以及为何中央认为布什总统决定售予台湾Ｆ－１６战斗机是违反中美“八·一七”联合公报之精神并损害中美关系.

<E-desc> Description:
A relevant document should discuss the resolution concerning U.S. weapon sales to Taiwan in the Sino-American "8-17" Joint Communique and why the Chinese consider President Bush 's decision to sell F-16 fighters to Taiwan to be in violation of the spirit of the Sino-American "8-17" Joint Communique and to be damaging to Sino-American relations.

</top>

Appendix B

English Translation of the Sample Chinese Text

So sung:

O so vast, O so mighty, The Great River rolls to sea, Flowers do waves thrash, Heroes do sands smash, When all the dreams drain, Same are lose and gain.

Green mountains remain, As sunsets ingrain, Hoary fishers and woodcutters, And some small rafts and calm waters, In autumn moon, in spring winds, By the wine jars, by porcelains, Discuss talk and tale, Only laugh and gale.

Three Heroes Swear Brotherhood In The Peach Garden; One Victory Shatters The Rebels In Battlegrounds.

Domains under heaven, after a long period of division, tends to unite; after a long period of union, tends to divide. This has been the rule since antiquity. When the rule of the Zhou Dynasty weakened, seven contending kingdoms sprang up, warring one with another until the kingdom of Qin prevailed and possessed the empire. But when Qin's destiny had been fulfilled, arose two opposing kingdoms, Chu and Han, to fight for the mastery. And Han was the victor.

The rise of the fortunes of Han began when Rucker-Lewis the Supreme Ancestor slew a white serpent to raise the banners of uprising, which only ended when the whole empire belonged to Han (BC 202). This magnificent heritage was handed down in successive Han emperors for two hundred years, till the rebellion of Frederick-Gorman caused a disruption. But soon Winkler-Lewis the Latter Han Founder restored the empire, and Han emperors continued their rule for another two hundred years till the days of Emperor Sprague, which were doomed to see the beginning of the empire's division into three parts, known to history as The Three Kingdoms.

But the descent into misrule hastened in the reigns of the two predecessors of Emperor Sprague—Emperors Henson and Bonner—who sat in the Dragon Throne about the middle of the second century.

Emperor Henson paid no heed to the good men of his court, but gave his confidence to the palace eunuchs. He lived and died, leaving the scepter to Emperor Bonner, whose advisers were the Regent Marshal Hood-Dickson and the Imperial Guardian Derrick-Kane. Hood-Dickson and Derrick-Kane, disgusted with the abuses of the eunuchs in the affairs of the state, plotted the destruction for the power-abusing eunuchs. But the Chief Eunuch Harding-Saito was not to be disposed of easily. The plot leaked out and the honest Hood-Dickson and Derrick-Kane were put to death, leaving the eunuchs stronger than before.

Appendix C

Discretized Document Length

discretized document length	frequency
1	1752
2	19311
3	12629
4	7082
5	8194
6	8017
7	6468
8	5036
9	4399
10	4306
11	4309
12	4122
13	4251
14	4219
15	3942
16	3778
17	3597
18	3509
19	3243
20	3012
21	2898
22	2731
23	2596
24	2385
25	2346
26	2142
27	1992
28	1859
29	1710
30	1612
31-40	11349
41-50	6289
51-60	3479
61-70	2012
71-80	1152
81-90	759
91-100	479
101-200	1350
201-300	261
301-400	116
402-944	59
945-5882	4

Table C.1: Discretized Document Length for TREC Chinese Dataset

Appendix D

Results for TREC-5 Queries

	Run	Weighting Method	k_d	Average Precision	Total Rel Retrieved	R Precision	Precision at 100 docs
1	T5w1.kd0	$Weight_1$	0	0.3691	1995	0.3873	0.3164
2	T5w2.kd0	$Weight_2$	0	0.3775	2003	0.3865	0.3189
3	T5w3.kd0	$Weight_3$	0	0.3762	2002	0.3860	0.3204
4	T5w4.kd0	$Weight_4$	0	0.3773	2005	0.3864	0.3193
5	T5w5.kd0	$Weight_5$	0	0.3657	1992	0.3812	0.3164
6	T5w1.kd2	$Weight_1$	2	0.3783	2006	0.3925	0.3189
7	T5w2.kd2	$Weight_2$	2	0.3888	2017	0.3934	0.3246
8	T5w3.kd2	$Weight_3$	2	0.3872	2016	0.3927	0.3250
9	T5w4.kd2	$Weight_4$	2	0.3878	2016	0.3939	0.3250
10	T5w5.kd2	$Weight_5$	2	0.3747	2001	0.3849	0.3171
11	T5w1.kd6	$Weight_1$	6	0.3901	2016	0.3985	0.3243
12	T5w2.kd6	$Weight_2$	6	0.4022	2021	0.4128	0.3318
13	T5w3.kd6	$Weight_3$	6	0.4011	2020	0.4101	0.3314
14	T5w4.kd6	$Weight_4$	6	0.4018	2020	0.4117	0.3321
15	T5w5.kd6	$Weight_5$	6	0.3870	2008	0.3967	0.3221
16	T5w1.kd8	$Weight_1$	8	0.3926	2012	0.3968	0.3239
17	T5w2.kd8	$Weight_2$	8	0.4051	2019	0.4082	0.3329
18	T5w3.kd8	$Weight_3$	8	0.4036	2018	0.4078	0.3329
19	T5w4.kd8	$Weight_4$	8	0.4051	2020	0.4090	0.3325
20	T5w5.kd8	$Weight_5$	8	0.3908	2010	0.3978	0.3229
21	T5w1.kd10	$Weight_1$	10	0.3947	2006	0.4012	0.3246
22	T5w2.kd10	$Weight_2$	10	0.4067	2015	0.4082	0.3357
23	T5w3.kd10	$Weight_3$	10	0.4058	2014	0.4081	0.3357
24	T5w4.kd10	$Weight_4$	10	0.4065	2015	0.4075	0.3357
25	T5w5.kd10	$Weight_5$	10	0.3933	2006	0.4022	0.3236
26	T5w1.kd15	$Weight_1$	15	0.3918	1977	0.4012	0.3261
27	T5w2.kd15	$Weight_2$	15	0.4024	1975	0.4090	0.3336
28	T5w3.kd15	$Weight_3$	15	0.4015	1970	0.4073	0.3339
29	T5w4.kd15	$Weight_4$	15	0.4023	1978	0.4083	0.3346
30	T5w5.kd15	$Weight_5$	15	0.3925	1974	0.4012	0.3225
31	T5w1.kd20	$Weight_1$	20	0.3839	1937	0.3942	0.3279
32	T5w2.kd20	$Weight_2$	20	0.3935	1928	0.4044	0.3354
33	T5w3.kd20	$Weight_3$	20	0.3928	1920	0.4040	0.3357
34	T5w4.kd20	$Weight_4$	20	0.3933	1928	0.4037	0.3350
35	T5w5.kd20	$Weight_5$	20	0.3876	1946	0.3994	0.3268
36	T5w1.kd50	$Weight_1$	50	0.3212	1659	0.3709	0.3039
37	T5w2.kd50	$Weight_2$	50	0.3224	1634	0.3693	0.3071
38	T5w3.kd50	$Weight_3$	50	0.3216	1627	0.3701	0.3079
39	T5w4.kd50	$Weight_4$	50	0.3224	1634	0.3693	0.3089
40	T5w5.kd50	$Weight_5$	50	0.3289	1691	0.3707	0.3057
41	T5w6.kd15.d2	$Weight_6(d = 2)$	15	0.3988	1982	0.4055	0.3321
43	T5w6.kd15.d10	$Weight_6(d = 10)$	15	0.4018	1975	0.4074	0.3346
44	T5w6.kd15.d20	$Weight_6(d = 20)$	15	0.4019	1973	0.4066	0.3339
45	T5w6.kd15.d50	$Weight_6(d = 50)$	15	0.4018	1971	0.4068	0.3343
46	T5w6.kd15.d100	$Weight_6(d = 100)$	15	0.4018	1971	0.4073	0.3336

Table D.1: Results for TREC-5 Queries with the Word-based Approach

	Run	Weighting Method	k_d	Average Precision	Total Rel Retrieved	R Precision	Precision at 100 docs
1	T5c1.kd0	$Weight_1$	0	0.3475	2004	0.3611	0.2918
2	T5c2.kd0	$Weight_2$	0	0.4126	2056	0.4251	0.3368
3	T5c3.kd0	$Weight_3$	0	0.3795	1986	0.3963	0.3189
4	T5c4.kd0	$Weight_4$	0	0.3863	2011	0.4017	0.3275
5	T5c5.kd0	$Weight_5$	0	0.3507	1992	0.3619	0.2968
6	T5c1.kd2	$Weight_1$	2	0.3523	2013	0.3690	0.2946
7	T5c2.kd2	$Weight_2$	2	0.4233	2057	0.4318	0.3418
8	T5c3.kd2	$Weight_3$	2	0.3994	2009	0.4070	0.3289
9	T5c4.kd2	$Weight_4$	2	0.4028	2027	0.4083	0.3325
10	T5c5.kd2	$Weight_5$	2	0.3641	2006	0.3667	0.2989
11	T5c1.kd6	$Weight_1$	6	0.3591	2022	0.3804	0.3007
12	T5c2.kd6	$Weight_2$	6	0.4275	2056	0.4283	0.3475
13	T5c3.kd6	$Weight_3$	6	0.4137	2019	0.4166	0.3375
14	T5c4.kd6	$Weight_4$	6	0.4135	2030	0.4184	0.3371
15	T5c5.kd6	$Weight_5$	6	0.3805	2014	0.3840	0.3079
16	T5c1.kd8	$Weight_1$	8	0.3606	2025	0.3807	0.2996
17	T5c2.kd8	$Weight_2$	8	0.4214	2028	0.4278	0.3436
18	T5c3.kd8	$Weight_3$	8	0.4116	1997	0.4155	0.3357
19	T5c4.kd8	$Weight_4$	8	0.4086	2014	0.4198	0.3364
20	T5c5.kd8	$Weight_5$	8	0.3846	2012	0.3922	0.3086
21	T5c1.kd10	$Weight_1$	10	0.3603	2022	0.3809	0.2986
22	T5c2.kd10	$Weight_2$	10	0.4155	1999	0.4258	0.3429
23	T5c3.kd10	$Weight_3$	10	0.4043	1957	0.4137	0.3364
24	T5c4.kd10	$Weight_4$	10	0.4027	1977	0.4147	0.3350
25	T5c5.kd10	$Weight_5$	10	0.3836	2004	0.3913	0.3139
26	T5c1.kd15	$Weight_1$	15	0.3562	1982	0.3789	0.2996
27	T5c2.kd15	$Weight_2$	15	0.3975	1922	0.4156	0.3436
28	T5c3.kd15	$Weight_3$	15	0.3862	1861	0.4084	0.3314
29	T5c4.kd15	$Weight_4$	15	0.3856	1895	0.4120	0.3350
30	T5c5.kd15	$Weight_5$	15	0.3745	1962	0.3895	0.3125
31	T5c1.kd20	$Weight_1$	20	0.3503	1951	0.3773	0.3000
32	T5c2.kd20	$Weight_2$	20	0.3776	1838	0.4057	0.3339
33	T5c3.kd20	$Weight_3$	20	0.3628	1781	0.3958	0.3257
34	T5c4.kd20	$Weight_4$	20	0.3648	1817	0.3964	0.3279
35	T5c5.kd20	$Weight_5$	20	0.3562	1882	0.3791	0.3093
36	T5c1.kd50	$Weight_1$	50	0.3017	1692	0.3437	0.2807
37	T5c2.kd50	$Weight_2$	50	0.2704	1484	0.3326	0.2779
38	T5c3.kd50	$Weight_3$	50	0.2504	1366	0.3131	0.2668
39	T5c4.kd50	$Weight_4$	50	0.2592	1432	0.3206	0.2704
40	T5c5.kd50	$Weight_5$	50	0.2689	1547	0.3341	0.2779
41	T5c6.kd15.d2	$Weight_6(d = 2)$	15	0.3824	1975	0.3996	0.3243
43	T5c6.kd15.d10	$Weight_6(d = 10)$	15	0.3997	1917	0.4200	0.3386
44	T5c6.kd15.d20	$Weight_6(d = 20)$	15	0.3985	1909	0.4226	0.3418
45	T5c6.kd15.d50	$Weight_6(d = 50)$	15	0.3977	1985	0.4261	0.3425
46	T5c6.kd15.d100	$Weight_6(d = 100)$	15	0.3971	1887	0.4260	0.3418

Table D.2: Results for TREC-5 Queries with the Character-based Approach

Appendix E

Comparisons for Topic 8 and Topic 37

Recall	T5w2.kd0 (word)	T5c2.kd0 (character)
0.00	1.0000	1.0000
0.10	0.9091	1.0000
0.20	0.9091	1.0000
0.30	0.8125	1.0000
0.40	0.6552	0.8571
0.50	0.5000	0.6750
0.60	0.3766	0.6750
0.70	0.2246	0.4493
0.80	0.1538	0.3977
0.90	0.1204	0.2097
1.00	0.1135	0.0000
Average Precision	0.5096	0.6786
Relevant Retrieved	43	42

Table E.1: Average Precision of the Two Best Runs from Word and Character Approaches for Topic 8

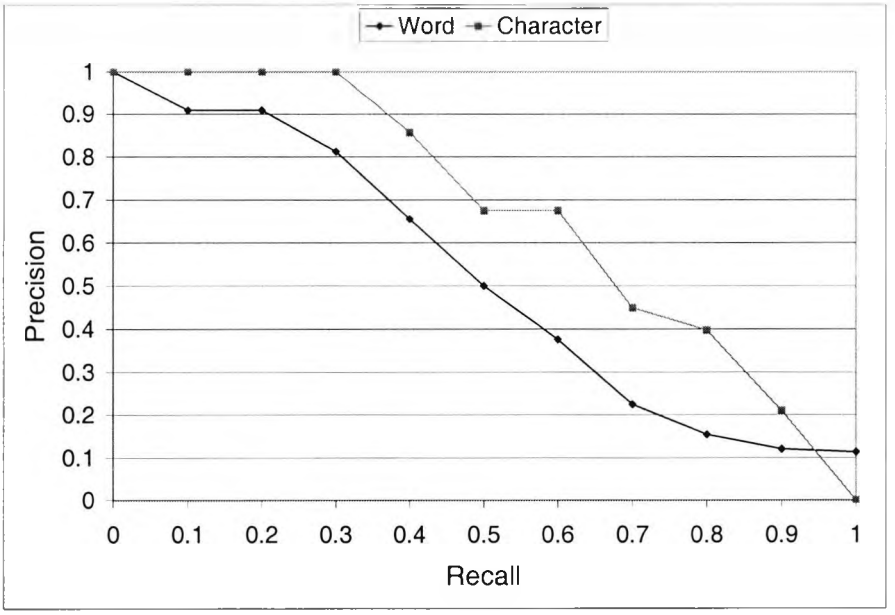


Figure E.1: Precision-recall Curves of the Two Best Runs from Word and Character Approaches for Topic 8

Recall	T6w2.kd20 (word)	T6c2.kd10 (character)
0.00	1.0000	1.0000
0.10	0.9412	0.9444
0.20	0.8000	0.8462
0.30	0.6923	0.8462
0.40	0.6234	0.8276
0.50	0.5259	0.7284
0.60	0.4675	0.6698
0.70	0.3097	0.5764
0.80	0.2043	0.4095
0.90	0.0000	0.2004
1.00	0.0000	0.0000
Average Precision	0.4957	0.6424
Relevant Retrieved	104	114

Table E.2: Average Precision of the Two Best Runs from Word and Character Approaches for Topic 37

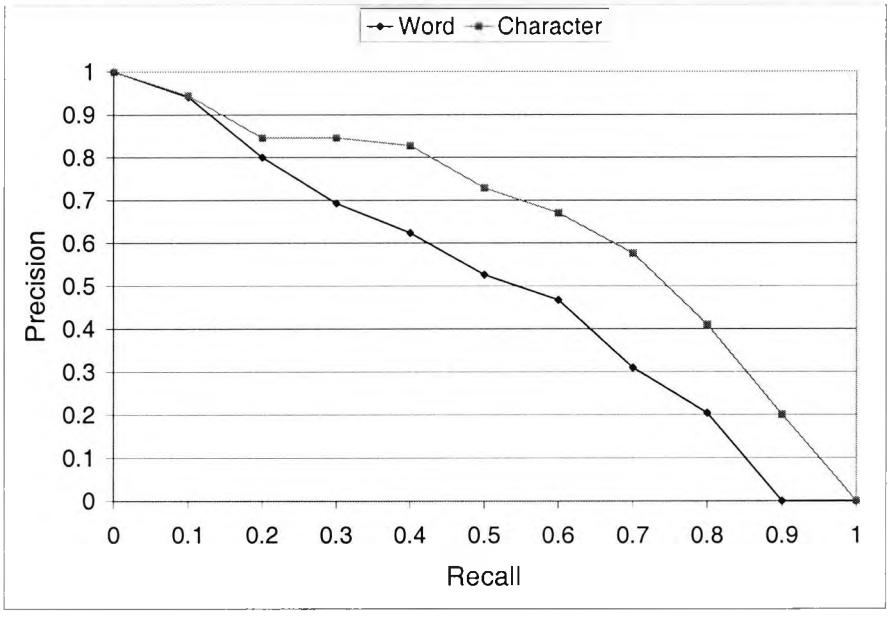


Figure E.2: Precision-recall Curves of the Two Best Runs from Word and Character Approaches for Topic 37

Appendix F

Ranking Positions for Topic 5, Topic 33 and Topic47

index	relevant documents retrieved	character	word
1	CB015016-BFW-517-421	>1000	263
2	CB021020-BFW-816-94	178	240
3	CB027020-BFW-1463-664	25	31
4	CB027021-BFW-1186-696	43	52
5	CB028010-BFW-787-84	238	192
6	CB028021-BFW-821-259	204	>1000
7	CB029005-BFW-686-380	97	46
8	CB029028-BFW-500-609	205	183
9	CB034025-BFW-4082-358	132	147
10	CB034025-BFW-4082-405	131	146
11	CB034216-BCW-3640-431	19	35
12	CB038027-BFW-632-548	152	121
13	CB041005-BFW-3904-458	>1000	180
14	pd9104-1399	133	75
15	pd9105-2840	219	150
16	pd9106-1060	47	235
17	pd9106-1158	169	81
18	pd9110-3026	>1000	344
19	pd9111-815	220	>1000
20	pd9201-3681	72	80
21	pd9208-3997	166	246
22	pd9209-2176	172	229
23	pd9209-3181	103	143
24	pd9302-2057	108	141
25	pd9308-891	167	84
26	pd9308-892	168	108
27	pd9308-894	159	107

Table F.1: Ranking Positions of All the Retrieved Relevant Documents for Topic 5

index	relevant documents retrieved	character	word
1	CB010007-BBW-164-399	64	75
2	CB039021-BFW-105-239	10	10
3	CB040006-BBW-177-416	90	99
4	CB047010-BFW-103-282	13	13
5	pd9304-1414	34	38
6	pd9304-1477	35	41
7	pd9304-1599	62	61
8	pd9304-1843	104	108
9	pd9306-1870	45	54
10	pd9306-950	33	34
11	pd9306-951	77	93
12	pd9307-1373	7	6
13	pd9308-144	14	15
14	pd9308-1493	3	3
15	pd9308-1592	69	69
16	pd9310-369	6	5
17	pd9310-372	61	64
18	pd9311-2566	2	2
19	pd9311-2618	26	23
20	pd9311-2844	20	12

Table F.2: Ranking Positions of All the Retrieved Relevant Documents for Topic 33

index	relevant documents retrieved	character	word
1	CB019026-BFJ-406-358	2	4
2	pd9106-1986	1	1
3	pd9106-2372	5	6
4	pd9106-2498	3	2
5	pd9107-1291	15	15
6	pd9107-3016	7	5
7	pd9107-3102	4	7
8	pd9108-1451	20	25
9	pd9112-1224	17	19
10	pd9201-5398	14	12
11	pd9202-4617	13	14
12	pd9204-2636	11	10
13	pd9209-6113	8	11
14	pd9211-5322	9	9
15	pd9212-3431	42	>1000
16	pd9212-3432	10	103

Table F.3: Ranking Positions of All the Retrieved Relevant Documents for Topic 47

Appendix G

Effect of BM26 on $Weight_1$, ..., $Weight_5$

Recall	BM25 T6c2.kd0	BM26 T6c2.kd2	BM26 T6c2.kd6	BM26 T6c2.kd8	BM26 T6c2.kd10	BM26 T6c2.kd15	BM26 T6c2.kd20	BM26 T6c2.kd50
0.00	0.9150	0.9252	0.9684	0.9505	0.9540	0.9536	0.9609	0.9692
0.10	0.8009	0.8082	0.8289	0.8330	0.8340	0.8361	0.8298	0.8299
0.20	0.7442	0.7464	0.7514	0.7636	0.7642	0.7730	0.7777	0.7606
0.30	0.6945	0.7009	0.7108	0.7122	0.7189	0.7265	0.7336	0.7002
0.40	0.6594	0.6673	0.6815	0.6862	0.6869	0.6895	0.6875	0.6423
0.50	0.5879	0.6037	0.6126	0.6192	0.6292	0.6374	0.6333	0.5641
0.60	0.5201	0.5258	0.5395	0.5436	0.5470	0.5595	0.5557	0.4367
0.70	0.4239	0.4420	0.4515	0.4516	0.4519	0.4409	0.4253	0.3074
0.80	0.3326	0.3393	0.3535	0.3546	0.3565	0.3426	0.3225	0.1999
0.90	0.2343	0.2413	0.2454	0.2409	0.2373	0.2192	0.1958	0.0638
1.00	0.0399	0.0426	0.0389	0.0381	0.0390	0.0332	0.0165	0.0011
AP	0.5341	0.5434	0.5551	0.5582	0.5603	0.5599	0.5545	0.4893

Table G.1: Recall-Level Precision for Character Approach Using $Weight_2$ Method

Recall	BM25 T6c1.kd0	BM26 T6c1.kd20	BM26 T6c3.kd0	BM26 T6c3.kd15	BM26 T6c4.kd0	BM26 T6c4.kd15	BM26 T6c5.kd0	BM26 T6c5.kd20
0.00	0.9108	0.9189	0.9117	0.9427	0.9160	0.9499	0.8987	0.9085
0.10	0.7593	0.7885	0.7663	0.8307	0.7858	0.8294	0.7467	0.8107
0.20	0.6827	0.7025	0.7168	0.7651	0.7239	0.7732	0.6757	0.7250
0.30	0.6119	0.6496	0.6646	0.7116	0.6791	0.7190	0.6326	0.6763
0.40	0.5593	0.5882	0.6127	0.6740	0.6436	0.6827	0.5699	0.6264
0.50	0.5016	0.5286	0.5552	0.6154	0.5746	0.6263	0.4893	0.5710
0.60	0.4365	0.4592	0.4834	0.5259	0.4992	0.5404	0.4125	0.4709
0.70	0.3787	0.3844	0.3889	0.4217	0.4009	0.4262	0.3366	0.3627
0.80	0.3082	0.3189	0.3022	0.3091	0.3084	0.3171	0.2659	0.2765
0.90	0.1870	0.2031	0.1743	0.1957	0.1888	0.2032	0.1797	0.1828
1.00	0.0446	0.0353	0.0327	0.0202	0.0294	0.0266	0.0311	0.0200
AP	0.4789	0.5005	0.4967	0.5417	0.5113	0.5494	0.4627	0.5032

Table G.2: Recall-Level Precision for Character Approach Using $Weight_1$, $Weight_3$, $Weight_4$ and $Weight_5$ Methods

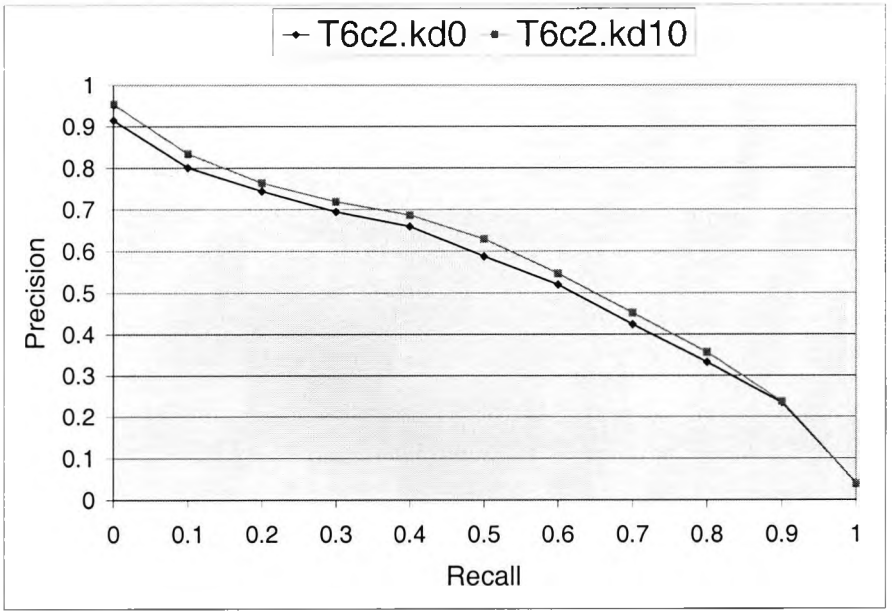


Figure G.1: Precision-recall Curves of the Two Runs T6c2.kd0 and T6c2.kd10

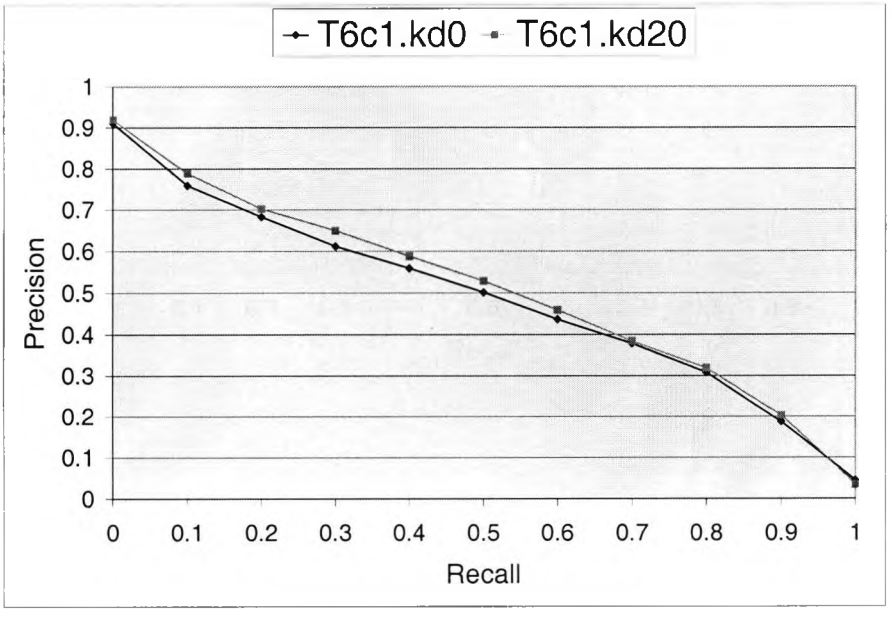


Figure G.2: Precision-recall Curves of the Two Runs T6c1.kd0 and T6c1.kd20

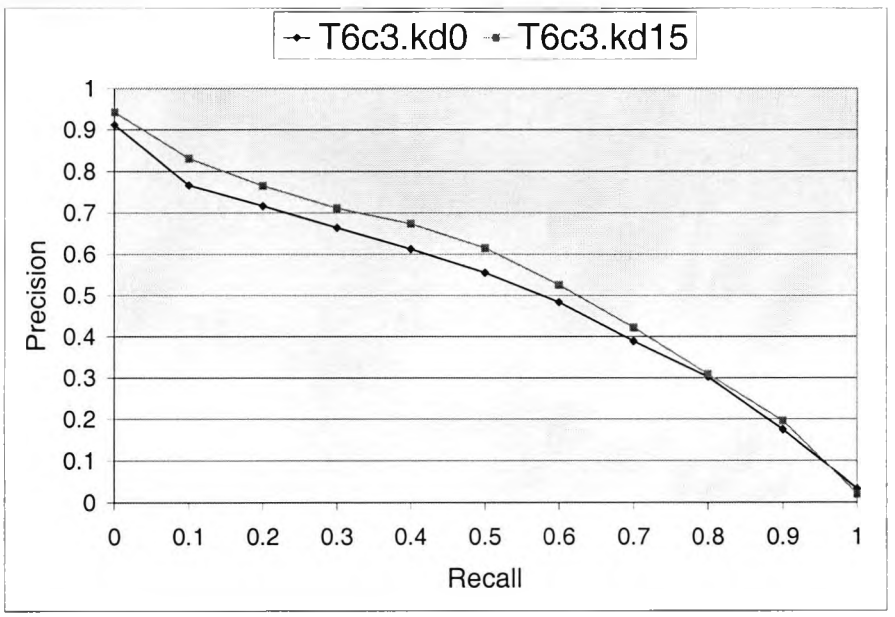


Figure G.3: Precision-recall Curves of the Two Runs T6c3.kd0 and T6c3.kd15

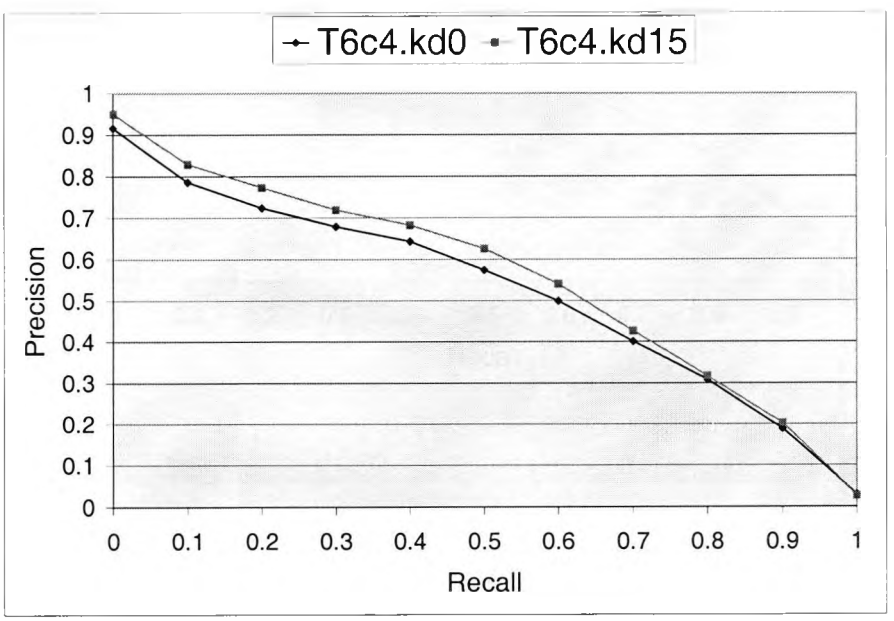


Figure G.4: Precision-recall Curves of the Two Runs T6c4.kd0 and T6c4.kd15

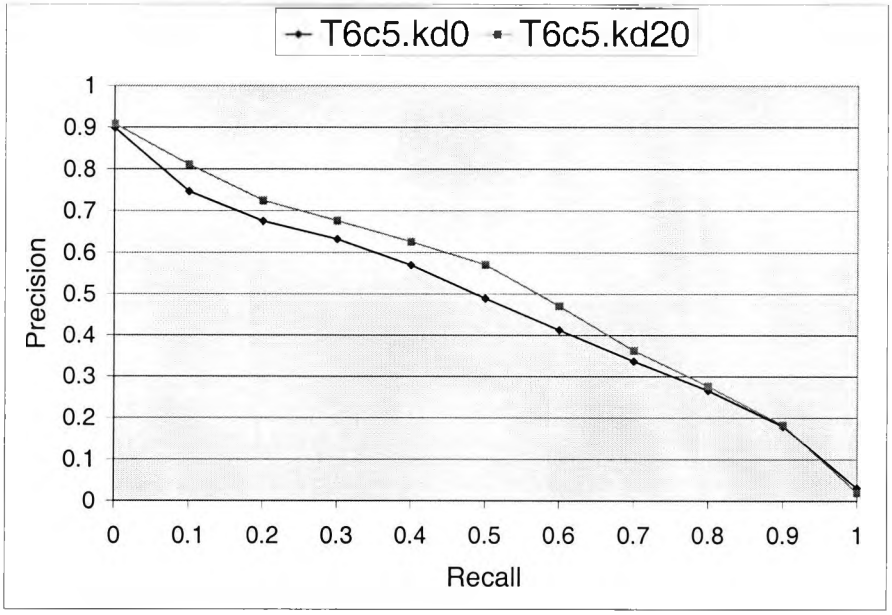


Figure G.5: Precision-recall Curves of the Two Runs T6c5.kd0 and T6c5.kd20