

1988

## A Hybrid Model for Document Retrieval Systems.

Zhen-bao Zou

*Louisiana State University and Agricultural & Mechanical College*

Follow this and additional works at: [https://digitalcommons.lsu.edu/gradschool\\_disstheses](https://digitalcommons.lsu.edu/gradschool_disstheses)

---

### Recommended Citation

Zou, Zhen-bao, "A Hybrid Model for Document Retrieval Systems." (1988). *LSU Historical Dissertations and Theses*. 4694.

[https://digitalcommons.lsu.edu/gradschool\\_disstheses/4694](https://digitalcommons.lsu.edu/gradschool_disstheses/4694)

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Historical Dissertations and Theses by an authorized administrator of LSU Digital Commons. For more information, please contact [gradetd@lsu.edu](mailto:gradetd@lsu.edu).

## INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book. These are also available as one exposure on a standard 35mm slide or as a 17" x 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.



University Microfilms International  
A Bell & Howell Information Company  
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA  
313/761-4700 800/521-0600



**Order Number 8917873**

**A hybrid model for document retrieval systems**

**Zou, Zhen-bao, Ph.D.**

**The Louisiana State University and Agricultural and Mechanical Col., 1988**

**U·M·I**

**300 N. Zeeb Rd.  
Ann Arbor, MI 48106**



# **A HYBRID MODEL FOR DOCUMENT RETRIEVAL SYSTEMS**

**A Dissertation**

**Submitted to the Graduate Faculty of the  
Louisiana State University and  
Agricultural and Mechanical College  
in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy**

**in**

**The Department of Computer Science**

**by**

**Zhen-bao Zou**

**B.S., Zhejiang University, China, 1977**

**M.S., University of Science and Technology of China, China, 1981**

**December 1988**

## ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my adviser, Dr. Donald H. Kraft, for his suggesting the research problem and guiding me throughout the doctoral work. His profound knowledge in the research area and unbiased attitude towards research work have given me tremendous benefits and unflagged encouragement. His unreserved support and enormous patience through some difficult times will always be remembered in my life.

My years in LSU were greatly enhanced by learning from my committee members, Dr. David Blouin, Dr. Sitharama Iyengar, Dr. Bush Jones, and Dr. Leslie Jones, through the many courses that have provided the foundation of my research. I would like to express my special gratitude to them for their devoted teaching, and for their valuable assistance and support to my doctoral research.

Finally, I would like to express my sincere thanks to all the people who have assisted and encouraged me in completing this work.

## Table of Contents

ACKNOWLEDGEMENTS .....	ii
TABLES OF CONTENTS .....	iii
ABSTRACT .....	v
1. Introduction .....	1
2. Composite Indexing Weighting Model .....	10
2.1 Literature Review .....	10
2.1.1 Statistical Index Term Weighting Models .....	10
2.1.2 Probabilistic Index Term Weighting Models .....	13
2.2 General Index Term Weighting Function .....	17
2.3 Composite Index Term Weighting Model .....	27
2.4 Strategy of Selecting Coefficients .....	32
3. Foundations of Composite Retrieval Model .....	43
3.1 Literature Review .....	43
3.1.1 Term Vector Space Model .....	43
3.1.2 Probabilistic Retrieval Model .....	45
3.1.3 Generalized Boolean Retrieval Model .....	48
3.2 Foundations of Composite Retrieval Model .....	53
4. Composite Retrieval Model .....	65



4.1 Description of Composite Retrieval Model .....	65
4.2 Composite Query Language .....	69
4.3 Indexing Model .....	75
4.3.1 General Description of the Indexing Model .....	75
4.3.2 Inverted File of Index Terms .....	83
4.3.3 Document Description File .....	84
4.3.4 User Classification File .....	93
4.4 Query Processing Model .....	99
4.5 Matching Model .....	105
4.6 Ranking Model .....	116
5. Conclusions: Towards an Expert System .....	127
BIBLIOGRAPHY .....	133
VITA .....	143

## ABSTRACT

A methodology for the design of document retrieval systems is presented. First, a composite index term weighting model is developed based on term frequency statistics, including document frequency, relative frequency within document and relative frequency within collection, which can be adjusted by selecting various coefficients to fit into different indexing environments. Then, a composite retrieval model is proposed to process a user's information request in a weighted Phrase-Oriented Fixed-Level Expression (POFLE), which may apply more than Boolean operators, through two phases. That is, we have a search for documents which are topically relevant to the information request by means of a descriptor matching mechanism, which incorporate a partial matching facility based on a structurally-restricted relationship imposed by indexing model, and is more general than matching functions of the traditional Boolean model and vector space model, and then we have a ranking of these topically relevant documents, by means of two types of heuristic-based selection rules and a knowledge-based evaluation function, in descending order of a preference score which predicts the combined effect of user preference for quality, recency, fitness and reachability of documents.

# CHAPTER 1

## INTRODUCTION

An information retrieval system is currently related to three different types of systems [1]. One is the well-known database management system, where the data or information about objects are well defined by means of a small number of attributes, and the retrieval process consists of identifying and providing the exact data which precisely fulfill the user's request in a pre-designed formal query language. As a contrast, a question-answering system represents the most sophisticated information retrieval system, where the facts are extracted from information objects combined with general world knowledge, and the retrieval process consists of identifying and providing the relevant fact(s) which most likely fulfill the user's question in natural language. The third one, a document or reference retrieval system, lies between the above two types of systems as far as complexity is concerned. In a typical document retrieval system, the objects may be documents, articles or other textual materials in natural language. It is impossible to identify their content exactly by means of a finite set of attributes. The retrieval process consists of identifying and providing the relevant documents which most likely fulfill the user's information request in terms of document descriptors. This paper is devoted to the third case so that the term "information retrieval system" will be used as an alternative to "document or reference retrieval system".

In the environment of document retrieval systems, there are two fundamental tasks: information analysis and information retrieval. Information analysis is known as

the indexing operation and is done at the time of system creation or new document entry. Information retrieval is a process of searching, finding and presenting relevant documents in response to a user's information request.

Indexing operations consist of two steps: selecting an appropriate set of document descriptors, known as index terms, and assigning to each individual document those index terms which are capable of representing the document contents and able to help distinguish the documents indexed by them from the others in the document collection.

Indexing operations have been carried out manually by a group of indexers who are experts in the subject area of the document contents and/or users in many document retrieval systems. However, with the aid of computing equipment, the indexing task can be done automatically. The advantages of cost reduction and improvement of retrieval effectiveness over the manual indexing systems make it promising [2].

In the automatic indexing environment, index terms are often extracted from a given collection of documents, hence one has an uncontrolled vocabulary in free text. Most of research work regarding indexing of free text starts with the observation that the frequency of individual word types in the natural language text has something to do with the significance of these words for the purpose of content representation. In fact, it is observed that the most frequent words in a text tend to be short function words such as "the", "of", or "and", while the least frequent words tend to be those rare words which have little effect on the content of the text. The well-known Zipf's law [3], which states that the product of the frequency of the use of the words and the rank order of the words(based on frequency) is approximately constant, has been

applied to various indexing experiments. The results show that words with medium frequency of occurrences are the most significant for indexing purpose [4,5]. Following Salton[4], an early proposal for index term extraction consists of six steps:

(1) For the given collection of  $N$  documents, calculate for each document the frequency of each unique word in that document;

(2) Calculate the total frequency of occurrences in the entire document collection for each word by summing up the frequency of each unique term across all  $N$  documents;

(3) Arrange the words in decreasing order according to their total frequency;

(4) Eliminate high frequency function words by selecting a threshold value and removing all words with a total frequency above this threshold;

(5) Similarly, eliminate the rare terms which do not affect the retrieval performance significantly by choosing another threshold value and removing all words with a total frequency below this threshold;

(6) Use the remaining medium-frequency words for assignment to the documents as index terms.

Although the above procedure is simple to implement, difficulties lie in the determination of both high and low threshold values. In an operational retrieval environment, the elimination of all high-frequency words might produce losses in recall, the proportion of the number of retrieved relevant documents to the total number of relevant documents in the collection with respect to user's query. Moreover, the elimination of low-frequency terms may produce losses in precision, the pro-

portion of the number of retrieved relevant documents to the total number of documents retrieved. The other defects are the large volume of index terms caused by redundant terms of the same type and the lack of consideration of word phrases as well as thesaurus classes (e.g., synonyms). Thus, an improved method of index term extraction was performed in SMART project [6], where a stop list was used to help eliminate function words of high frequency, word stems (removing prefixes and/or suffixes) instead of words were used as index terms, certain words of low frequency were incorporated into a thesaurus class and those of high frequency were used to construct phrases on the basis of co-occurrence frequency.

Other methods for extracting significant word types and phrases from natural language text involve uses of syntactic and/or semantic analysis. FASIT (Fully Automatic Syntactically based Indexing of Text) developed by Dillon and Grey [7] identifies content bearing words without a full parse and without using semantic criteria. In FASIT, the indexing procedure includes concept selection and concept grouping. The concept selection procedure selects syntactic categories for each word and solves adjective-noun and noun-verb ambiguities. The concept grouping procedure reduces the selected concepts to a canonical form to consolidate synonymous forms. Their experiment shows that significant terms in the text can be identified through syntactic patterns.

Braun and Schwind report their work in automatic semantics-based indexing of natural language texts for information retrieval systems [8]. The work is based on the hypothesis that single words within each phrase are related in a certain well-defined manner, i.e., the type of relations holding between concepts depends only on the

concepts themselves. Therefore, the relations can be stored in a semantic network. The authors claim success for phrase extraction from texts by this semantic method.

The next step in automatic indexing is the assignment of index terms to each individual document in the collection. In many commercial document retrieval systems, term assignment is a binary process; that is, an index term is either assigned or not assigned to a document according to some criteria specified by the system designers. A rule of thumb is to assign each index term to all those documents in which it explicitly appears. The pitfall of such a strategy is obvious; all the index terms assigned to a document are treated as being equally important. For many years, researchers have worked hard on a new methodology for index term assignment such that each assignment is attached a numerical value, called an index term weight, representing the relative importance of the index term with respect to the document indexed. Under this strategy, the previous method of index term assignment adopted in many commercial systems becomes a special case where only binary weights are allowed so that it is called a binary weighting scheme. That is, an implicit weight of one is assigned to all index terms that are present in the document; otherwise, the weight is zero. In the area of information retrieval, the binary weighting scheme is often described as unweighted, and many experimental document retrieval models have been characterized by their non-binary weighting schemes.

Two aspects have been emphasized for index term weighting. On one hand, a term must be a good representative of the information content of the document so as to render the document retrievable when it is wanted. On the other hand, an important term must be a good discriminator; that is, it must help distinguish the document in

which it plays an important role in content representation from the remainder of the collection in which it plays a minor role in order to prevent the indiscriminate retrieval of all documents.

There are two basic types of index term weighting methods, i.e., methods based on term frequency statistics and methods based on a probabilistic approach [2,9,10,11,12]. Generally, statistics-based term weighting models are document-oriented, which means that all frequency statistics can be obtained from documents through a simple text analyzer. On the other hand, probability-based term weighting models may assume that the behavior of index terms in a collection of documents follows a known probability distribution, or that the probability of relevance of a document bearing the term may be estimated through well selected samples.

In addition to information analysis(indexing), information retrieval is a routine operation for document retrieval systems. The retrieval process begins with the query formulation representing a user's information request. In the past, various mathematical models for document retrieval have been developed. These models are used to formally represent the basic characteristics, functional components, and retrieval processes of document retrieval systems. In the early stages of information retrieval, queries were represented by a single index term with regard to a subject. All the documents indexed by that term would be retrieved, which has been called the subject catalog model [13]. Later, a Boolean retrieval model was introduced, and has been the most popular one implemented on many existing commercial systems. In traditional Boolean retrieval systems [14], a query normally consists of index terms connected by the Boolean operators AND, OR and NOT; while documents are represented by



means of unweighted index terms, so that for each index term it is possible to say whether a document is either to be retrieved or not, depending on whether the terms are indexed or not according to the particular query.

The advantage of Boolean retrieval model is that it can be implemented as simply as the subject catalog model, and yet it provides a powerful structure to formulate the user request. However, one major problem in such a system is that the set of documents presented in response to a query is not ranked in any order of presumed importance to the user. Other problems involve the difficulties in properly constructing Boolean queries. A user must make a decision whether a term is to be chosen in his query or not. However, if more query terms are connected by AND operators, it might cause too few documents to be retrieved; while if more query terms are connected by OR operators, it might have the opposite effect. The problems observed suggest the necessity of developing more sophisticated models with a capability to support weighted index terms and weighted query terms to provide a ranked list of documents as the response to a user request. As a result of such efforts, the term vector space model has been developed and widely accepted in many experimental document retrieval systems [15,16]. Since 1960, when probabilistic indexing theory was introduced by Maron and Kuhns [17], a variety of probabilistic models have been developed [18,19,20,21,22]. In the 1970s, fuzzy subset theory was introduced for information retrieval, though with complaints [23]. Attempts have been made to extend the traditional Boolean retrieval model to a more general case, where both index term weights and query term weights were considered [24,25,26,27,28].

In this paper, an attempt has been made to develop a methodology for the design of a sophisticated document retrieval system which provides a variety of new features lacking in the current competing models. Chapter 2 presents a composite index term weighting model based on frequency statistics. This model has been shown to have the ability of combining various term significance indicators and accommodating itself to the different indexing environments.

Chapter 3 is concerned with the foundations of a composite retrieval model. It is suggested that a topical relevance measure be used to designate the relevance relationship between the documents and the query with respect to a user's information request and a preferable relevance measure be used to designate the relevance relationship between the documents and the query with respect to a user's information needs. Thus, a new design of a retrieval model includes a search for topically relevant documents by means of descriptor matching mechanism, and a ranking of these topically relevant documents in descending order of a preference score which predicts the combined effect of a user's preference for quality, recency, fitness and reachability of documents.

Chapter 4 provides a comprehensive design methodology for a composite retrieval system. This system includes a composite query language which mainly consists of a phrase-oriented fixed-level expression of unrestricted query descriptors with more than Boolean operators. An indexing model has been proposed, which incorporates a stem-based index term file, a phrase-based document description file, and a knowledge-based user classification file. A query preprocessor is designed to perform query reformulation and to function as a screen to produce an initial set of documents

in response to the query. A matching model is developed as a generalization of the weighted Boolean retrieval and the similarity retrieval in the vector space model. A new feature incorporated in our model is called partial matching based on a structural dependency among document descriptors. This feature, along with the use of three thesaurus classes, provides a strong means of improving the effectiveness of the retrieval. Finally, a ranking model we propose incorporates four user preference factors to rank the documents in terms of a preference score with respect to a user's information needs.

Chapter 5 concludes with a statement of the need for expertise in the implementation of a document retrieval system based on our hybrid model, and includes a discussion of possible future work. Included here are suggestions for exploring more expert knowledge to be incorporated into the document retrieval system, and an outline of some of the steps necessary in performing experiments and evaluations of the system.

## CHAPTER 2

### COMPOSITE INDEX TERM WEIGHTING MODEL

#### 2.1 Literature review

Two basic types of index term weighting methods, i.e., methods based on term frequency statistics and methods based on probabilistic approach, have been developed, tested, and evaluated in the past. A literature review is presented below to see the rationale underlying these methods.

##### 2.1.1 Statistical index term weighting models

###### (1) Simple Term Frequency

Among statistics-based index term weighting models, the simplest one is called the Simple Term Frequency(STF) scheme which sets  $WEIGHT_{ik}$ , the weight of index term  $TERM_k$  in document  $DOC_i$ , to  $FREQ_{ik}$ , the frequency of occurrence of term  $TERM_k$  in document  $DOC_i$ . That is,

$$WEIGHT_{ik} = FREQ_{ik}.$$

The underlying rationale of such a simple weighting system was best explained by Luhn [31], who said, "The justification of meaning of word significance of use-frequency is based on the fact that a writer normally repeats certain words as he advances or varies his arguments and as he elaborates on an aspect of a subject. This mean of emphasis is taken as an indicator of significance...". Thus, term significance is taken to be proportional to its occurrence frequency in each document. Experiments show that the STF scheme is of some value and by no means likely to degrade perfor-

mance [12].

## (2) Inverse Document Frequency

The Inverse Document Frequency (IDF) is one of the most widely accepted index term weighting schemes. The underlying principle of the IDF scheme is the fact that more specific terms may render a larger contribution towards discriminating between the relevant documents and the non-relevant documents with respect to a user request. Thus, term importance is set to be inversely proportional to the number of documents to which the term is assigned. The inverse document frequency method was originally advocated by Spark Jones [2] as a device for improving the retrieval performance of single unweighted terms, where she used a function  $f(n)=m$  such that  $2^{m-1} < n \leq 2^m$  to set the weight of a term with document frequency  $n$  to  $f(N)-f(n)+1$ . A more popular form of the inverse document frequency function that was developed later is given by

$$IDF_k = \log\left(\frac{N}{DOCFREQ_k}\right) + 1$$

where  $N$  is the number of documents in the collection and  $DOCFREQ_k$  is the document frequency of term  $TERM_k$  (i.e., the number of documents where  $FREQ_{ik} > 0$ ) [4].

## (3) Signal-Noise Ratio

Another index term weighting scheme was developed under a principle adopted from information science which states that the information contained in a symbol varies directly with the "surprise" value of the symbol, where "surprise" value can be measured as an inverse function of the probability of observing that symbol. By analogy to Shonnon's information measure [32], one can define the NOISE and SIGNAL of an index term  $TERM_k$  for a collection of  $N$  documents as

$$NOISE_k = - \sum_{i=1}^N P_i \log(P_i)$$

$$SIGNAL_k = \log(TOTFREQ_k) - NOISE_k,$$

where  $P_i = \frac{FREQ_{ik}}{TOTFREQ_k}$ . Note that  $NOISE_k$  is the average information conveyed by index term  $TERM_k$  over the collection of documents, and  $SIGNAL_k$  can be regarded as a measure of the deviation from the average of the information carried by an individual index term  $TERM_k$  [30].

#### (4) Term Discrimination Value

A more complicated index term weighting scheme is called the Term Discrimination Value(TDV), and is directly related to term vector space model. Here we have, given a pairwise similarity measure function between two documents  $DOC_i$  and  $DOC_j$ , an average value of similarity measure over a collection of documents reflecting the density of the document space [43]. That is,

$$SIM(DOC_i, DOC_j) = \frac{\langle DOC_i, DOC_j \rangle}{NF_1}$$

$$AVGSIM = \sum_{i=1}^N \sum_{\substack{j=1 \\ i \neq j}}^N \frac{SIM(DOC_i, DOC_j)}{NF_2}$$

Then,

$$TDV_k = AVGSIM_k - AVGSIM$$

where  $DOC_i$  and  $DOC_j$  are term vectors,  $\langle DOC_i, DOC_j \rangle$  denotes usual inner product of  $DOC_i$  and  $DOC_j$ ,  $NF_1$  and  $NF_2$  are normalization factors,  $AVGSIM$  is the average similarity value,  $AVGSIM_k$  is the average similarity value with term  $TERM_k$  removed, and  $TDV_k$  is called the term discrimination value of term  $TERM_k$ . When  $TDV_k$  is positive,  $TERM_k$

is identified as a good discriminator; otherwise, it is a poor discriminator.

### 2.1.2 Probabilistic index term weighting models

#### (1) Inductive weighting models

Probabilistic index term weighting methods have been developed and tested since the work reported by Maron and Kuhns [17]. They conjecture that in the case of a document indexed by a given term, it is only probable that the user of the system will find the document relevant to his query if he is interested in the field represented by the index terms. Thus, the weight of an index term in a document is defined by the probability of that event. In the experiments carried out by the authors such weights were intuitively estimated by indexers.

#### (2) Two-Poisson distribution model

Another well-known probabilistic index term weighting scheme is the 2-Poisson distribution model. The underlying assumption of the model is two-fold. First, over a collection of documents, the probability that a document will have  $n$  occurrences of a functional word (such as "the", or "for") follows a Poisson distribution, but that same distribution does not hold for content-bearing words which are of special interest for being chosen as index terms. Second, if the document collection can be divided into two portions such that the probability that a document will have  $n$  occurrences of a content-bearing word follows a Poisson distribution with respect to each portion, then the distribution can be described by a mixture of two Poisson distributions. That is,

$$f(n) = \frac{p_1 \lambda_1^n e^{-\lambda_1}}{n!} + \frac{p_2 \lambda_2^n e^{-\lambda_2}}{n!},$$

where  $p_i$  is the probability that the document is in portion  $i$  and  $\lambda_i$  is the mean occurrence of the word in portion  $i$ , for  $i=1,2$ ;  $n$  is the exact number of occurrence of the word in the document, and  $f(n)$  is the probability of  $n$  such occurrences. The ratio

$$\frac{p_i \lambda_i^n e^{-\lambda_i}}{\sum_{j=1}^2 p_j \lambda_j^n e^{-\lambda_j}}, \quad i=1,2,$$

which is the conditional probability that the particular document belongs to portion  $i$  is actually used to make the decision of weight assignment [11].

### (3) Weighting models with relevance information

The concept of index term weighting has been extended to weight the terms in a user information request, often referred to as query weights or relevance weights of search terms. Some models developed for this purpose are variations of the Inverse Document Frequency scheme [34]. Others involve the use of relevance information and are based on probabilistic decision theory [40,41,42,43]. That is, given parameter  $l_1$  representing the loss in value for retrieval of a non-relevant document and parameter  $l_2$  representing the loss in value for non-retrieval of a relevant document, a decision of retrieving a given document  $DOC$  is made if

$$l_2 p(\text{rel} | DOC) > l_1 p(\text{nonrel} | DOC),$$

where  $p(\text{rel} | DOC)$  and  $p(\text{nonrel} | DOC)$  are the probabilities of relevance and non-relevance for the given document,  $DOC$ , respectively.

Using Bayes' law, the above decision rule can be transformed into a more useful form in which the following statistic can be derived



$$TR_k = \frac{r / (R-r)}{(n-r) / (N-n-(R-r))},$$

where  $r$  is the number of relevant documents containing term  $TERM_k$ ,  $R$  is the total number of relevant documents with respect to the query,  $n$  is the total number of documents containing term  $TERM_k$ , and  $N$  is the total number of documents in the whole collection. This statistic is known as the term relevance of  $TERM_k$  [9], or, using the form  $\log(TR_k)$ , known as the relevance weight of  $TERM_k$  [42].

Similarly, if one specifies  $v_1$  as the gain in value for retrieval of a relevant document and  $v_2$  as the gain in value for non-retrieval of a non-relevant document, a term  $TERM_k$  can then be weighted by the utility weight [46]

$$UTILITY_k = r (v_1 + l_2) - (n - r) (v_2 + l_1).$$

Various interpretations of index term weights have been suggested. The probabilistic approach assumes that in the case of a document indexed by a given term, it is probable that the user of the system will find the document relevant to his query if he is interested in the field represented by the index term. Thus, the weight of an index term in a document is defined by the probability of that event [17]. Although such a probabilistic interpretation of term weights are convenient for mathematical treatments, it is nearly impossible to estimate them precisely. In the experiments carried out by Maron and Kuhns [17], such weights were intuitively estimated by indexers.

In the fuzzy set model of document retrieval systems [54], an index term weight has a continuum of values in the interval  $[0,1]$  and is interpreted as the membership function that measures the extent to which document is about the concept represented

by the index term. This interpretation of term weight is less restrictive in mathematical meaning, i.e., it only requires weights to be continuous in the range from zero to one. It is called a generalization of binary weights in the sense that when the membership function has the discrete range  $\{0,1\}$ , this system conforms to the binary weighting system.

In the vector space model [6], index term weights are referred to as the components of a document vector along a set of base terms. Term weights are usually required to have non-negative numerical values and are equal to zero if they are absent as descriptors from the document.

The term weighting schemes discussed above have been used in various experiments of information retrieval. Combinations of STF weights and the weights derived from some of the other methods have also been tested [4]. However, such efforts have been limited to simple multiplication of two types of weights. This paper tries to explore the intrinsic and extrinsic properties of some term weighting schemes in order to develop a more sophisticated term weighting scheme that will be a composition of various frequency statistics.

## 2.2 General index term weighting function

As far as the statistics-based index term weighting schemes are concerned, the statistical quantities characterizing the behavior of index terms for the document collection may be summarized as follows:

- (1)  $FREQ_{ik}$ , the frequency of occurrences of each term  $TERM_k$  in an individual document  $DOC_i$ ;
- (2)  $n_i$ , the number of different terms occurring in an individual document  $DOC_i$ ;
- (3)  $l_i$ , the total number of term occurrences (term postings) in an individual document  $DOC_i$ ;
- (4)  $N$ , the number of documents in the collection;
- (5)  $m$ , the total number of index terms for the given document collection;
- (6)  $DOCFREQ_k$ , the number of documents indexed by term  $TERM_k$ , called the document frequency or collection frequency;
- (7)  $TOTFREQ_k$ , the total frequency of term  $TERM_k$  over the document collection, which can be calculated by summing up  $FREQ_{ik}$  in (1) over the entire document collection;
- (8)  $LENGTH$ , the total number of term occurrences (term postings) in the document collection, calculated by summing up the  $l_i$  over the document collection.

It is observable that the statistics  $FREQ_{ik}$ ,  $n_i$ , and  $l_i$  have constant values for an individual document and are independent from other document statistics in the collection. These quantities can be viewed as an intrinsic property of an individual document. An index term weighting scheme based on an intrinsic property is said to be

document-oriented, and has the advantage of stability in the sense that weights need not be recalculated as the number of documents in the collection increases or decreases. On the other hand, the term weighting systems involving the other quantities are said to be collection-oriented, which means that they are subject to change when the number of documents in the collection increases or decreases.

There are different views on how the term weights should be treated. As in the SMART system [6], we shall consider each document to be represented by a term vector. That is, a particular document,  $DOC_i$ , is identified by a collection of term weights,  $W_{i1}, W_{i2}, \dots, W_{im}$ , where  $W_{ik}$  is assumed to be the weight of  $TERM_k$  as assigned to  $DOC_i$ . Figure 2.2.1 shows this as a term assignment matrix.

	$TERM_1$	$TERM_2$	...	$TERM_m$
$DOC_1$	$W_{11}$	$W_{12}$	...	$W_{1m}$
$DOC_2$	$W_{21}$	$W_{22}$	...	$W_{2m}$
$DOC_3$	$W_{31}$	$W_{32}$	...	$W_{3m}$
.				.
.				.
.				.
$DOC_n$	$W_{n1}$	$W_{n2}$	...	$W_{nm}$

Figure 2.2.1 Term assignment matrix

One view treats  $W_{ik}$  as a function of the relative frequency of the term occurrences in a document against the entire document collection, i.e.,  $W_{ik} = f(FREQ_{ik} / TOTFREQ_k)$ . The underlying principle here is adopted from the information theory approach, which states that the information of a symbol can be measured as an inverse function of the probability of receiving that symbol. That is to say, the higher the probability of occurrence of a word, the less information it conveys. In

other words, the information content of a term can be measured as  $-\log(p)$ , where  $p$  is the probability of occurrence of that term. By extension, when the document collection is characterized by  $m$  possible index terms, each occurring with a specified probability  $p_j$ , the expected information gained by using one of the terms is given by Shannon's formula [32]:

$$AVERAGE\ INFORMATION = -\sum p_j \log(p_j).$$

When all the  $p_j$  are equal, i.e.,  $p_j = 1/m$ , the above average information reaches its maximum value of  $\log(m)$ . This well-known Shannon's information measure is the underlying principle of the NOISE-SIGNAL term weighting scheme, in which the probability  $p_j$  is taken as the relative frequency of the term occurrences in an individual document over the entire document collection. Experiments show that such a scheme is unlikely to give an optimal result [4].

A second view treats  $W_{ik}$  as a function of the relative frequency of  $TERM_k$  against the total number of term occurrence in the document  $DOC_i$ , i.e.,  $W_{ik} = f(FREQ_{ik} / l_i)$ . When  $l_i$  is ignored, this function reduces to the STF scheme. It is clear that a term weighting scheme based on the relative frequency  $FREQ_{ik} / l_i$  is document-oriented. Experiments show that such a weighting scheme has approximately the same worth as the STF [12].

A third view treats  $W_{ik}$  as being inversely proportional to the document frequency  $DOCFREQ_k$ , i.e.,  $W_{ik} = f(1/DOCFREQ_k)$ . This view represents a term-document relationship characterized by the fact that each term has the same weight in all of the various documents to which it is assigned, i.e.,  $W_{ik} = W_{jk}$  for all documents indexed by  $TERM_k$ . This Inverse Document Frequency method has been successfully applied to various

document retrieval systems [4,12], which suggests that term specificity is an important measure of term importance.

Now let us consider some fundamental requirements of a general weighting function (GWF) on which our composite index term weighting model is developed. First, a good index term weighting function should be as independent as possible from the growth of the document collection. Justification of this requirement is obvious if we wish to maintain the currency of the weighting information at low cost. It would be a disaster to have to recalculate all weights when either  $N$ , the number of documents, or  $m$ , the number of terms, changes. It is observable that a document-oriented index term weighting function, either  $FREQ_{ik}$  or  $FREQ_{ik} / l_i$ , best fits into this requirement. The IDF scheme and TDV scheme are undesirable due to the involvement of the parameter  $N$ , while the SNR scheme lies in between as far as the cost of updating index term weights is concerned.

Second, a good index term weighting function should allow term weights to vary continuously in the interval  $[0,1]$ . This feature is especially significant in consideration of a generalized Boolean approach, since such a weighting function could be used as a membership function. Also, this feature is desirable for further mathematical development, as we want to determine a document ranking function to map the similarity relationship between a document and a query into a retrieval status value (RSV) indicating the degree to which the document will be found relevant to the user's information needs [13]. Obviously, all the above index term weighting functions could be normalized to fulfill this requirement except the TDV scheme. However, these functions often produce somewhat conflicting results with no flexibility to reflect any

possible relevance information from the indexing environment, which leads to the consideration of the next requirement.

Third, a good index term weighting function should be able to portray an ideal curve of term weights whenever possible relevance information is available. The so-called ideal weighting curve is based on two arguments. One is taken from Luhn's speculation [31] that the "resolving power" of the index terms extracted from document texts would peak in the middle-frequency range, where by "resolving power" we mean the ability of the index terms to identify relevant documents and to distinguish them from the non-relevant ones. The other argument is taken from Salton's optimal weighting theory and experiments [9]. This asserts that an optimal weighting system should be an increasing function of document frequency when document frequency changes from one to  $R$ , the number of relevant documents with respect to a user's query, and a decreasing function of document frequency when it changes from  $R$  to the possible maximum value, assuming a linear relationship between document frequency and the number of relevant documents containing the term.

**Definition 2.1:**

A General Weighting Function(GWF) is defined as

$$\frac{c}{X^a} (\ln(X) + b)$$

where  $X$  is a term frequency statistic satisfying  $X > 0$ , and  $a, b, c$  are positive constants to be interpreted below.

**Proposition 2.1:**

$GWF$  is an increasing function of  $X$  when  $X$  is in the range of  $(0, e^{1/a-b}]$ , and a decreasing function of  $X$  when  $X$  is in the range of  $[e^{1/a-b}, \infty)$ . Moreover,  $GWF$  reaches its maximum value at  $X = e^{1/a-b}$ .

Proof:

Take the first derivative of  $GWF$ , and set it to zero. This yields

$$GWF'(X) = c \left[ \frac{1}{X^{a+1}} \right] (1 - a \ln(x) - ab) \equiv 0$$

$$X = e^{1/a-b}$$

Since  $GWF''(e^{1/a-b}) < 0$ ,  $GWF$  reaches its maximum value at  $X = e^{1/a-b}$   $\square$

**Proposition 2.2:**

$GWF$ , as given in Definition 2.1, reaches its maximum value of 1 when the constant  $c = a e^{1-ab}$ .

Proof:

Set  $GWF(e^{1/a-b})$  to 1, we get

$$c = a e^{1-ab} \quad \square$$

Thus, we call  $c$  a normalization factor.

**Proposition 2.3:**

Given the  $GWF$  in Definition 2.1,  $X=u$  such that  $GWF(u)$  is maximum iff

$$a = \frac{1}{b + \ln(u)}$$



Proof:

Set  $u = e^{1/a-b}$ , yielding

$$a = \frac{1}{b+\ln(u)} \quad \square$$

Thus, we call  $a$  a modal factor.

**Proposition 2.4:**

Given the  $GWF$  in Definition 2.1,  $X=u$  such that  $GWF(u)$  is maximum iff

$$b = 1/a + \ln(u) \text{ and } b > -\ln(u)$$

Proof:

Set  $u = e^{1/a-b}$ , yielding

$$b = 1/a + \ln(u).$$

To guarantee  $GWF(u) > 0$ , we have

$$\frac{c}{u^a} (\ln(u) + b) > 0$$

or

$$b > -\ln(u) \quad \square$$

Thus, we call  $b$  a smoothing factor.

**Proposition 2.5:**

Given the  $GWF$  in Definition 2.1, and  $v$ , the lowest observed value of  $X$ ,

$GWF(X) \geq 0$  if

$$b \geq -\ln(v)$$

Proof:

Set  $\frac{c}{X^a} (\ln(X) + b) \geq 0$ , we have

$$b \geq -\ln(X).$$

Since  $-\ln(X)$  is a decreasing function in the

range  $(0, \infty)$ ,  $b \geq -\ln(v)$  ensures  $b \geq -\ln(X)$   $\square$

Proposition 2.5 implies a constraint on the selection of the smoothing factor  $b$  in order to ensure a positive value of the  $GWF$ .

It is clear that the normalization factor  $c$  will not affect the linear order of weight assignment and is easy to calculate once the modal factor  $a$  and the smoothing factor  $b$  are fixed. Under an operational environment, the range of statistics being investigated is already known; thus, it is more appropriate to choose a value  $u$  of  $X$  such that  $GWF(u)$  reaches its maximum as a first step. Then, the smoothing factor  $b$  may be chosen according to a distribution of the statistics  $X$  over its range. Finally, the modal factor  $a$  can be calculated by using Proposition 2.3.

As an example, we compare the weight assignment using General Weighting Function to the one using INDEXD system developed by Jones, et al. [78].

$WFN^2$	Freq	GWF	index term
54912	11	1.00000	fuzzy set theoretic model
52020	51	.78006	tirs model
38376	41	.84675	p-norm model
32041	179	.00000	model
29007	11	1.00000	set theoretic model
28116	33	.89873	document space
24768	8	.99396	vector space model
21316	146	.18459	space
19152	16	.99013	fuzzy set theoretic
14848	32	.90505	topological paradigm
13924	118	.37972	query
12540	19	.97811	vector space
10400	8	.99396	space model
9664	2	.87442	fuzzy set theory model
9604	98	.47302	set
8640	10	.99943	topological space
8580	11	1.00000	theoretic model
8424	18	.98245	fuzzy set
7992	3	.91999	model the fuzzy set
7740	9	.99753	boolean model
7424	2	.87442	n-dimensional topological space resulting
7392	1	.78722	vector space model query
7296	16	.99013	set theoretic
7101	3	.91999	space the fuzzy set
6966	3	.91999	tirs model retains

**Table 2.2.1 Weight assignment by INDEXD vs GWF**

In Table 2.2.1, the index terms are extracted by the INDEXD system from [76], the  $WFN^2$  denotes the weight assigned by the INDEXD system, the *Freq* denotes the frequency of occurrence of an index term within a document, and the *GWF* denotes the weight assigned by the General Weighting Function. To apply the General Weighting Function, we first look for the clues to set up a value of the frequency counts such that the GWF reaches the maximum; in this case,  $Freq=11$  is chosen so that the term with the largest weight is the same as the one given by the INDEXD system. Then, we choose a value for the smoothing factor  $b$ ; in this case, we chose  $b=-1.55667$  so that the

minimum weight calculated is zero. Finally, we choose  $a=0.358488$  and  $c=1.70266$  by applying Propositions 2.3 and 2.2, respectively.

$WFN^2$	Freq	GWF	index term
54912	11	1.00000	fuzzy set theoretic model
29007	11	1.00000	set theoretic model
8580	11	1.00000	theoretic model
8640	10	.99943	topological space
7740	9	.99753	boolean model
24768	8	.99396	vector space model
10400	8	.99396	space model
7296	16	.99013	set theoretic
19152	16	.99013	fuzzy set theoretic
8424	18	.98245	fuzzy set
12540	19	.97811	vector space
7992	3	.91999	model the fuzzy set
7101	3	.91999	space the fuzzy set
6966	3	.91999	tirs model retains
14848	32	.90505	topological paradigm
28116	33	.89873	document space
7424	2	.87442	n-dimensional topological space resulting
9664	2	.87442	fuzzy set theory model
38376	41	.84675	p-norm model
7392	1	.78722	vector space model query
52020	51	.78006	tirs model
9604	98	.47302	set
13924	118	.37972	query
21316	146	.18459	space
32041	179	.00000	model

**Table 2.2.2 Weight assignment sorted by GWF**

The Table 2.2.2, which is the same as the Table 2.2.1 but sorted in descending order of the *GWF*, provides a clearer picture to see that the very frequent terms such as "model", "space", and "query" are really insignificant ones; these terms will be eliminated from the document descriptor set in our indexing model, as proposed in Chapter 4.

### 2.3 Composite index term weighting model

The general index term weighting function stated above has been shown to be suitable for a given frequency statistic subject to the requirements of a basic term weighting curve. Now, we are able to define a composite index term weighting model which is a linear combination of three frequency statistics identified as term significance indicators expressed in terms of the general index term weighting function.

#### Definition 2.2:

A Composite Weighting Function(CWF) is defined as

$$\sum_{i=1}^3 \beta_i \left[ \frac{c_i}{X_i^{a_i}} (\ln(X_i) + b_i) \right]$$

where  $X_1 = 1/DOCFREQ_k$ ,  $X_2 = \frac{l_i}{FREQ_{ik}}$ ,  $X_3 = \frac{TOTFREQ_k}{FREQ_{ik}}$ ,

$\beta_i, i=1,2,3$ , are non-negative constants satisfying  $\beta_1 + \beta_2 + \beta_3 = 1$ , and

$a_i, b_i, c_i, i=1,2,3$ , are constants determined by the indexing environment.

#### Proposition 2.6:

The CWF given by Definition 2.2 reaches its maximum value at

$$X_i = e^{1/a_i - b_i}, \quad i=1,2,3$$

for appropriate constants  $a_i, b_i, c_i, \beta_i, i=1,2,3$ .

**Proof:**

Set the partial derivatives of CWF to zero, and the conclusion holds  $\square$

**Proposition 2.7:**

The CWF given in Definition 2.2 is a continuous function of  $X_1, X_2$  and  $X_3$ , bounded on  $[0,1]$  for appropriate constants  $a_i, b_i, c_i, \beta_i, i=1,2,3$ .

**Proof:**

The proof follows from Propositions 2.1, 2.2, 2.3, and the condition  $\sum_{i=1}^3 \beta_i = 1$   $\square$

**Proposition 2.8:**

Given the CWF in Definition 2.2,  $X_i = u_i, i=1,2,3$ , then the CWF( $u_1, u_2, u_3$ ) is maximum iff

$$a_i = \frac{1}{b_i + \ln(u_i)}, \quad i=1,2,3$$

**Proof:**

Set  $u_i = e^{1/a_i - b_i}$ , giving

$$a_i = \frac{1}{b_i + \ln(u_i)} \quad \square$$

**Proposition 2.9:**

Given the CWF in Definition 2.2,  $X_i = u_i, i=1,2,3$ , then the  $CWF(u_1, u_2, u_3)$  is maximum iff

$$b_i = \frac{1}{a_i} + \ln(u_i), \quad i=1,2,3$$

Proof:

Set  $u_i = e^{1/a_i - b_i}$ , giving

$$b_i = \frac{1}{a_i} + \ln(u_i) \quad \square$$

**Proposition 2.10:**

Given the CWF in Definition 2.2, and  $v_i$ , the lowest observed value of  $X_i, i=1,2,3$ ,  $CWF(X_1, X_2, X_3) \geq 0$  if  $b_i \geq -\ln(v_i)$

Proof:

From Proposition 2.5,  $b_i \geq -\ln(v_i), i=1,2,3$ , is a sufficient condition of  $CWF(X_1, X_2, X_3) \geq 0 \quad \square$

By Definition 2.2, the  $\beta_i$  can be viewed as weights for the three individual weighting functions. Moreover, Definition 2.2 may be changed to the form

$$\sum_{i=1}^3 \left[ \frac{\beta_i c_i (\ln(x_i) + b_i)}{X_i^{a_i}} \right].$$

We observe that the index term weights given by the above composite index term function CWF need to be recalculated when the number of documents in the col-

lection increases or decreases. However, the overhead of such work is relatively small, since the CWF involves only two collection-oriented statistics,  $DOCFREQ_k$  and  $TOTFREQ_k$ . For example, assume that the total number of index terms is equal to 5000, the average number of index terms in a document is 20, and the average document frequency is 250. Then only about 0.4% index term weights need to be recalculated after adding a document to the system. This also suggests that one may have good reason to ignore such a small effect, doing the recalculation of index term weights periodically at times of system reorganization.

The composite index term weighting function CWF given by Definition 2.2 provides great flexibility for the purpose of experiments. For example, as we know, a term with a document frequency of one is not of the interest, while the IDF scheme takes it as the most important. We can easily set the weights of all terms with document frequency  $R$  to the peak value by selecting  $a_1 = \frac{1}{\ln(R)+b_1}$ , where  $R$  would be an estimate of the number of relevant documents containing those index terms.

Although the application of the composite index term weighting model is simple and straightforward, two critical problems must be solved. One is how to define the objects, the so called index terms, and the other is how to specify the coefficients of the CWF. The latter issue will be discussed in detail in the next section. Let us now consider the concept of an index term to accommodate the composite index term weighting model.

It must be mentioned that the concept of an index term is rather vague and flexible in the literature. It may be taken as words and/or phrases, or may even include



some factual characteristics, such as citation strength, for a model [4]. More often, researchers in the area of information retrieval discuss their model without defining what an index term is, but rather take it as a given. However, we insist that the application of the composite index term weighting model may not be justified unless the concept of an index term has been properly defined.

According to the underlying principle of our composite index term weighting model, a straightforward way of defining an index term is to use word stems(or word types). Stem-based indexing significantly reduces the volume of the indexing vocabulary, as compared to word-based indexing, and makes it easy to collect the three frequency statistics necessary for calculating index term weights.

An extensive specification of index terms uses conceptual phrases, including single words which are strong enough to stand in its own right. In this case, we actually extend Zipf's law and information theory from words to conceptual phrases, while there seems no reason not to accept the validity of statistical interpretation of document frequency. One of the problems with phrase-based index terms lies in frequency counting. That is, should a word or phrase that in part repeats an index term be counted in the frequency count of that index term? The answer is not simple. In reality, there are at least three types of repetitions recognizable in the text:

- . a part of the index term,
- . partial repetition of the index term with a different modifier, and
- . partial repetition of the index term with a different ending.

We leave this problem, as well as selection of index terms, to be solved under the environment of system implementation.

## 2.4 Strategy of coefficient selection

In the previous discussion, we identified three factors as indicators of term significance in an automatic indexing environment. These three factors are document frequency, relative frequency within documents, and relative frequency within the collection. A non-linear function called the general index term weighting function was proposed, which is flexible to accommodate different requirements by properly choosing constants to set up the maximum value and slope of the weighting curve. Then, a linear combination of these three factors is used to quantitatively determine a unique value between zero and one representing the extent to which an index term is topically relevant to a document in the collection.

While our linear form to be chosen could be debated, its rationale can be seen as compared with the other possible function forms adopted in the literature [4]. There, a product form of a simple term weight and one of the other weights, such as IDF, or TDV, is used. No explanation is given as to why these two factors were combined and why the product form was chosen. There are at least two faults with the simple product form of the combined weighting scheme. First, two factors are treated as equal in the product form and it is hard to determine which factor is actually more critical to the weight since both weights were not normalized. Second, once a product function is chosen for automatic indexing, this function cannot be modified, even though the indexing environment may vary from one system to another.

The philosophy beneath the composite term weighting scheme is quite different. Although it suggests a uniform way to quantify the topical relevance of index terms

with respect to each document, which factor ought to be considered more important than another is left to be determined by the environment of the document retrieval system. For example, the characteristics of the document collection to be stored and retrieved by system users greatly influence that environment. These characteristics include the heterogeneity or homogeneity of the document collection. That is, the three factors could be treated as equally important when we don't know anything about the documents; but, if we have some knowledge about the document collection, we can use it by choosing different coefficients for the corresponding factors. The linear form of the composite index term weighting function also provides flexibility for the system to include more indicators of term significance by simply combining more items in the function.

In order to specify the coefficients of the three significance factors, we need to find some evidence or clues on which our decision is to be made. However, these must be from some other sources other than the statistical measures that have already been used in the composite index term weighting function.

Let us go back to where we generated our view of the term importance factors. Relative frequency within a document is considered as a significance indicator, because the repetition of certain words is assumed to be an indicator of emphasis, being divided by document length for normalization. Relative frequency within a collection is based on information theory, which states that rare words tend to convey more information than common words. However, our general index term weighting function is proposed in order to avoid an extreme case, where if a word is repeated too many times within a document or occurs too rarely within the entire collection, it is

not necessary to claim that they are extremely important. Considering the Zipfian curve of word frequency, we argue that the most important words are those of medium frequency of occurrences. This has been taken as a criterion for choosing those words as index terms. However, the simple product form of weighting function says that the higher the within-document frequency, the more important the term is once it has been chosen as an index term. This seems incorrect, since there exists a big leap from the role of useless common words to be eliminated and valuable content-bearing words to be used as surrogates of the document's content. So, we have to consider other aspects of term significance for the sake of retrieval effectiveness when we need to distinguish one document from another. Unlike the frequency of term occurrences, document frequency is thus taken with its statistical interpretation of term specificity. That is, a term with high document frequency would make it difficult to retrieve a relatively small group of documents as a response to a user's information request expressed in index terms. What we gain from the above discussion is that instead of developing a universal index term weighting model, we consider the role of term significance factors which is suitable for a given indexing environment. That is to say, an index term weighting function is an effective one if it works reasonably well under the operational environment of the system.

Our strategy is to construct an artificial environment under which we can derive a policy to differentiate the relative significance of three index term weighting factors. To accomplish this, we have to look for some knowledge about the document collection. Such knowledge would make it possible for us to differentiate among the documents in the entire collection for the purpose of information retrieval under an opera-

tional environment. That is to say, for a given information request, we must at least know that we can exclude some parts of the document collection and concentrate our search on the rest; otherwise, system efficiency could not be achieved. This is especially significant when we have a heterogeneous document base. Mathematically, for a given document collection  $D$ , we argue that according to the system designers' knowledge, it is always possible to define an equivalence relation  $R$  such that  $(D_1, D_2, \dots, D_r)$  is a partition of  $D$  induced by  $R$ , and for any particular information request under the operational environment the system response is a set of documents contained in one and only one equivalence class of  $D$ , denoted as  $D_i$ .

In fact, there are many possible equivalence relations we could define in the operational environment. For instance, we define  $d_1 R d_2$  iff  $d_1$  and  $d_2$  are written in the same language for any documents  $d_1$  and  $d_2$  in the document collection  $D$ . Thus, if there are  $r$  languages used in the entire document collection, we have  $r$  equivalence classes which is the quotient set of  $D$  induced by  $R$ , denoted as  $D/R$ . To respond to a particular information request, RETRIEVE ALL DOCUMENTS WRITTEN IN ENGLISH, the equivalence class in which all the documents are written in English would be presented, and the rest of the equivalence classes would be excluded.

Certainly, the partition of  $D$  induced by the above  $R$  is not all that attractive to us for the purpose of automatic indexing, though it might be useful to enforce such a partition mechanism in order to furnish the information requests which take language into consideration. Alternatively, we consider an equivalence relation  $R_0$ , called the subject-related relation, such that  $d_1 R_0 d_2$  iff  $d_1$  and  $d_2$  are of the same subject area for each  $d_1$  and  $d_2$  in document collection  $D$ . Now, the equivalence classes  $D/R_0$  receive a

simple interpretation in that all documents belonging to the same equivalence class are about the same subject. In an operational environment, such a partition could at least be implemented as a traditional catalogue, assuming that a patron's information request is within one and only one subject area.

If such a subject partition is obtainable, we then have some extra information for automatic indexing. Before the enforcement of a partition, our knowledge about the document collection may simply be limited to experts' opinion, or we may try to depict the document collection by comparing it to some other document collection on an existing information retrieval system by means of some simple variables, such as:

- diversity of the subject matter, measured by the proportion of the number of different subjects to the total number of documents in the collection;
- diversity of authors, measured by the proportion of the number of different authors of the document collection to the total number of documents in the collection;
- diversity of sources, measured by the proportion of the number of different sources from which the documents in the collection were published to the total number of documents in the collection;
- diversity of document types, measured by the proportion of the number of different types of documents in the collection to the total number of documents in the collection;
- diversity of indexing vocabulary, measured by the proportion of the number of index terms indexing the document collection to the total number of documents in the collection.

Generally, the higher these variables are, the more heterogeneous the document collection is. However, these measurements have not been used much at all in practice, since there are some reasons that make any judgement based on these measurements risky. One such reason is the difficulty in finding comparable document collection which would justify a comparison. Another is the difficulty in distinguishing the important variables from the others when the results are contradictory.

Now, with the partition we are able to specify more variables that characterize each equivalence class of the partition. These include:

- diversity of subject matters,
- diversity of authors,
- diversity of sources,
- diversity of document types, and
- diversity of indexing vocabulary

which can be measured as indicated above, and

- interaction of authorship, measured by the proportion of the number of authors who have their papers covering more than one equivalence class to the total number of authors of the document collection,
- interaction of sources, measured by the proportion of the number of sources that have published papers in more than one equivalence class to the total number of sources of the document collection,
- interactions of citations, measured by the number of the citations between different equivalence classes to the total number of citations in the document collection,
- interaction of vocabulary, measured by the proportion of the number of index

terms that have indexed the papers of more than one equivalence class to the total number of index terms for the collection.

Generally, the higher the values of these variables are, the higher the association between equivalence classes. In addition, the following data can be gathered:

- document frequency of each index term within the collection and within each equivalence class,
- relative frequency of each index terms within collection and within each equivalence class,
- distribution of document frequency for the entire document collection and for each equivalence class,
- distribution of relative frequency for the entire document collection and for each equivalence class.

In order to specify appropriate coefficients for the composite index term weighting function, we look for clues under an artificial indexing environment. Two strategies are suggested. One is based on general homogeneity/heterogeneity analysis and general statistical analysis. If the document collection is rather homogeneous, we shall place heavier weight on the factors which will promote distinguishing among documents. If document frequency and/or relative frequency within the collection are heavily concentrated on a narrow range, while the index terms in that interval cover more than one equivalence class, we shall place lighter weight on these collection related factors.

The other is based on a set of significant index terms selected as a benchmark. Then the statistics of these special index terms selected can be used to help specify the



coefficients. A simple rule is to take the arithmetic mean of frequency statistics as a peak value. For instance, if the average document frequency of these significant index terms is fifty, we may set  $X=50$  such that  $GWF(50)$  reaches the maximum.

The special list of significant index terms may be selected in various ways. If a set of index terms is available as a surrogate of an equivalence class, we may take them as candidates. Several hypotheses may also be used provided that their validity can be shown by prior experiments.

One technique is based on a single level partition of the document collection. According to information theory, which states that the information carried by a symbol is proportional to the probability of its appearance, we postulate that the most important index terms with respect to an equivalence class are those that have relatively low frequency of occurrences within the entire document collection and relatively high frequency of occurrences within the equivalence class. That is, if we assume that the entire document collection  $D$  is partitioned into  $(D_1, D_2, \dots, D_r)$  induced by a subject-related relation, then the following formula may be used as a criterion for selecting the significant index terms with respect to an equivalence class  $D_i$ :

$$\rho_i = \frac{f_i}{F},$$

where  $F$  is the term frequency within the collection  $D$  and  $f_i$  is the term frequency within a specific equivalence class  $D_i$ . Note that our interest here is only to select an elite set of index terms rather than to quantify the importance of the index terms by this measurement. Therefore, we might only choose those terms that have a maximum  $\rho$  value as the important terms for an equivalence class.

Another technique may be used under the environment of multi-level partitioning. We denote the set of all terms indexing the documents of an equivalence class  $D_i$  as  $TERM(D_i)$ ,  $i=1,2,\dots,r$ . We define an index term as a special term if and only if it belongs to one and only one  $TERM(D_i)$ , and an index term as a shared term if and only if it belongs to more than one  $TERM(D_i)$ . Then, it is not unreasonable to postulate that the special terms of  $TERM(D_i)$ , denoted as  $SP(D_i)$ , consist of some terminologies characterizing the subject, plus some rare words which are coincidentally used by authors. On the other hand, the shared terms, denoted as  $SH(D_i)$ , consist of some common terms (but not function words, such as articles or prepositions) and some terms which can be interpreted differently under different contexts. Generally, special terms are more important than shared terms in the sense that they are more useful in identifying the subject of a given document. Therefore, they can be used as clues for choosing the proper coefficients of the three factors in the composite index term weighting function. This is, within an equivalence class of the partition of document collection, we expect the average weight of special terms to be greater than that of shared terms, and we shall choose larger coefficient for the factors which reflects our expectation while the smaller coefficient for the factors which have negative effect.

However, one of the problem with which we have to deal is to exclude some rare terms in  $SP(D_i)$  and to include some useful terms in  $SH(D_i)$  such that the above decision would be more accurate. We denote the subset of  $TERM(D_i)$  which are considered more important as  $DES(D_i)$ , the descriptor set, and the subset of  $TERM(D_i)$  which are considered less important as  $ASC(D_i)$ , the ascriptor set. Initially,  $DES(D_i)=SP(D_i)$ , and  $ASC(D_i)=SH(D_i)$ . Then, we continue to seek more pre-

knowledge in addition to the frequency statistics. In a usual operational environment, information about the document collection would at least include sources (e.g., journals or monographs), authors, and bibliographies. A multi-level partition of the document collection might be enforced, rather than a single level subject-related partition. For instance, on each equivalence class  $D_i$  of  $D$  induced by a subject-related relation, we may define an equivalence relation  $R_1$ , denoted as citing/cited relation, such that  $d_1 R_1 d_2$  iff  $d_1$  is directly or indirectly citing/cited by  $d_2$  for any  $d_1$  and  $d_2$  in  $D_i$ . Thus,  $SP(D_i)$  is decomposed into two subsets  $SPSP(D_{ij})$  and  $SPSH(D_{ij})$  with respect to each equivalence class  $D_{ij}$  induced by  $R_1$  on  $D_i$ .  $SH(D_i)$  is also decomposed into two subsets  $SHSP(D_{ij})$  and  $SHSH(D_{ij})$  with respect to each equivalence class  $D_{ij}$  induced by  $R_1$  on  $D_i$ . Now, if we assume that the less important rare terms in  $SP(D_i)$  would remain in one of the  $SPSP(D_{ij})$ , and the less important common terms in  $SH(D_i)$  would remain in more than one  $SHSH(D_{ij})$ , i.e., rare terms remain rare and common terms remain common in the case of multiple-level partitions, then we have a strategy to modify the original descriptor set  $DES(D_i)$  and ascriptor set  $ASC(D_i)$ . That is, the terms that belong to both  $SP(D_i)$  and  $SPSP(D_{ij})$  are too exclusive to be important, and thus should be removed from  $DES(D_i)$  and added to  $ASC(D_i)$ . Similarly, the terms in both  $SH(D_i)$  and  $SHSP(D_{ij})$  are not common enough to be considered as unimportant terms, so we might add them to the  $DES(D_i)$  and remove them from  $ASC(D_i)$ . Hence, based on a two-level partition, we obtain a modified descriptor set and ascriptor set for each equivalence class  $D_i$ . We could use them as a benchmark for making the decision about the relevant importance of the three term significance indicators, i.e., for choosing appropriate values of  $\beta_i, i=1,2,3$ . A simple way is to calculate the three

individual weights in the CWF, with  $\beta_s$  being ignored, for the terms in  $DES(D_i)$  and in  $ASC(D_i)$ , respectively, as the first step. Then, the average term weights of the term sets  $DES(D_i)$  and  $ASC(D_i)$  could be determined. We expect that the average of the weights given by the CWF for the terms in the  $DEC(D_i)$  must be larger than the one for the terms in the  $ASC(D_i)$ . To accomplish this, we shall assign a big value to the  $\beta_i$  such that the factor weighted by  $\beta_i$  is the one that is most consistent with our expectation, and a small value to the one with the opposite effect. The final values of the assignment may be determined after several trials and comparisons.

## CHAPTER 3

### FOUNDATIONS OF COMPOSITE RETRIEVAL MODEL

#### 3.1 Literature review

In the past, various mathematical models for document retrieval systems have been developed. Among them, the term vector space model, probabilistic retrieval model and generalized Boolean retrieval model are representatives of the effort to support weighted indexing, weighted query formulation, and document ranking as a response to user's information request. A brief review of these three models is provided below.

##### 3.1.1 Term vector space model

The term vector space model has been extensively discussed by Salton, et al. [4,6,15,33]. In a system based on the vector space model, it is assumed that there exists a base of  $m$  terms,  $TERM_1, TERM_2, \dots, TERM_m$ , and a document  $DOC_i$  is represented as a vector  $\langle w_{i1}, w_{i2}, \dots, w_{im} \rangle$  of rank  $m$ , where  $w_{ik}$ ,  $k=1,2,\dots,m$ , is the  $k$ th component of document vector  $DOC_i$ , corresponding to the term  $TERM_k$ . This is often interpreted as the weight or importance of index term  $TERM_k$  assigned to document  $DOC_i$ . A particular query,  $Q_j$ , can be similarly identified as a vector  $\langle q_{j1}, q_{j2}, \dots, q_{jm} \rangle$ , where  $q_{jk}$  is interpreted as the weight or importance of term  $TERM_k$  assigned to query  $Q_j$ . The retrieval process involves computation of a similarity measure used to rank the documents to be presented to the user with respect to his information request. The similarity

measure function used in the SMART system [6] is the well-known cosine coefficient, defined as

$$\text{Cosine}(DOC_i, Q_j) = \frac{\langle DOC_i, Q_j \rangle}{|DOC_i| |Q_j|},$$

where  $\langle DOC_i, Q_j \rangle$  represents the dot product of the two vectors, and  $|DOC_i|$  and  $|Q_j|$  represent the magnitudes of the two vectors, respectively. This assumes that there exists  $m$  mutually orthogonal base vectors corresponding to  $TERM_1, TERM_2, \dots, TERM_m$  in a  $m$ -dimensional term vector space. Other popular similarity measures in the literature include Dice's coefficient, Jaccard's coefficient and the overlap coefficient, and have been found useful [4,5].

The vector space model has been widely used in experimental document retrieval systems. The justification of using vector similarity functions is given by Bookstein [16]. The major criticism against it is that Boolean logic has been totally abandoned. Vector similarity measures affect the matching process as if all query terms are ANDed together. In addition, although similarity measures, such as the cosine coefficient, work very well when the components of a document vector take either discrete value 0 or 1, it could exhibit an undesirable result when the components of a document vector take arbitrary values (even though positive) in a continuous interval. For instance, given a query  $Q=(q_1, q_2)$ , for all document vectors  $DOC_i=(\lambda w_{i1}, \lambda w_{i2})$ , where  $\lambda$  is a positive constant, the similarity measures calculated between  $DOC_i$  and  $Q$  would be identical, regardless of  $\lambda$ . The vector space model has been extended to include the Boolean operators AND and OR in the query with two different similarity

functions based on a p-norm distance measure [66]. Other efforts include imposing more algebraic structures on the vector space [68].

### 3.1.2 Probabilistic retrieval model

The first probabilistic model for document retrieval was proposed by Maron and Kuhns in 1960 [17]. In their model, each document is denoted by a set of properties, or simply a set of document descriptors, as well as a query. The function of a retrieval system is to compute for each document the probability that it will be judged relevant by a user with respect to a specific query. Documents could be ranked in decreasing order of this relevance probability, which is estimated as the number of times that the document is judged relevant by a user divided by the number of times that this type of query is submitted. When the properties or descriptors are considered as a set of index terms, this model provides an interpretation of index term weights as the probability that document  $DOC_i$  possesses index term  $TERM_k$  given that it is relevant to the query. In 1976, Roberson and Spark Jones proposed another probabilistic model for the document retrieval problem [42]. Again, each document is described as a set of properties or document descriptors. For retrieval, a user predicts the properties that a relevant document may have. Then, for any retrieved document possessing those properties, the retrieval system computes for each user the probability that he will judge a document having those properties relevant. This probability is estimated by the ratio of the number of documents having those properties and being relevant to the total number of documents having those properties. Moreover, it can be used to rank the documents. As a comparison, this model implies a theory of weighted query

formulation in which the weights assigned to query terms are interpreted as estimates of the relevance probability relative to a subset of document properties. The inherent property of these two models is that the relevance probability is estimated on an inductive basis. A unification of the above two models was proposed under a general conceptual frame [18].

A realization of the probability model has been studied using probabilistic decision theory paralleling the development of the weighting model with relevance information. That is, given parameter  $l_1$  representing the loss in value for retrieval of a non-relevant document and parameter  $l_2$  representing the loss in value for non-retrieval of a relevant document, a decision to retrieve a document DOC is made if

$$l_2 p(\text{rel} | \text{DOC}) > l_1 p(\text{nonrel} | \text{DOC})$$

where  $p(\text{rel} | \text{DOC})$  and  $p(\text{nonrel} | \text{DOC})$  are the probabilities of relevance and non-relevance for a given document, DOC, respectively.

Using Bayes' law, the above decision rule can be transformed into a more useful form,

$$\frac{p(\text{rel}) p(\text{DOC} | \text{rel})}{p(\text{nonrel}) p(\text{DOC} | \text{nonrel})} > \frac{l_1}{l_2}.$$

Assume that each document is described by a set of  $m$  properties represented by the binary-valued variables  $x_i$ ,  $i=1,2,\dots,m$ , which are conditionally independent on both relevant and non-relevant documents, and only the presence( $x_i=1$ ) or absence( $x_i=0$ ) of these properties are considered. Then the decision rule can be transformed into a linear function

$$g(\text{DOC}) = \sum_{i=1}^m x_i c_i + C$$



in which  $C$  is a constant once  $l_1, l_2$ ,  $p(\text{rel})$  and  $p(\text{nonrel})$  are specified. Moreover,

$$c_i = \log \frac{p(x_i=1 | \text{rel}) (1-p(x_i=1 | \text{nonrel}))}{p(x_i=1 | \text{nonrel}) (1-p(x_i=1 | \text{rel}))}$$

can be estimated by

$$\log \frac{r / (R-r)}{(n-r) / (N-n-(R-r))},$$

which is the term relevance weight if the properties are interpreted as index terms.

The parameters are defined as follows:  $r$  is the number of relevant documents containing  $TERM_k$ ,  $R$  is the total number of relevant documents with respect to a query,  $n$  is the number of documents containing  $TERM_k$ , and  $N$  is the total number of documents in the collection [4,5].

When document properties are not independent, a general form of dependence can be modeled as

$$p(DOC) = p(x_1) p(x_2 | x_1) \cdots p(x_n | x_1, x_2, \dots, x_{n-1})$$

which may be arbitrarily complex and nearly impossible to be evaluated in an operational environment. Two ways have been suggested for evaluation of  $p(DOC)$  in the literature. One is to apply pairwise dependency, instead of high order dependency, and to capture, as well as possible, the dependence relation [5,19,52]. The other is to apply the Bahadur-Lazarfeld expansion to calculate high order dependency when the number of properties is limited [51].

In early 1980s, Cooper and Huizinga proposed a new design for information retrieval based on the maximum entropy principle [21,22]. This design differs from the previous ones in that user is requested to estimate term relevance information which is used to form a joint distribution of maximum entropy. Then, for each

document, the system calculates the relevance probability which is used for ranking the response set of documents.

Unlike the previous probabilistic models, the one based on the maximum entropy principle needs no term independence assumption when the relevance probabilities are calculated. In fact, the relevance probability reflects term dependency in such a way that strongly dependent terms are treated as if they are ORed together and independent terms are treated as if they are ANDed together.

As theoretically rigorous as probabilistic retrieval models are, these models need to estimate full relevance information. A shortcoming for the model based on the maximum entropy principle may be efficiency, since it is very time-consuming to form a joint distribution from maximum entropy when the number of terms becomes large. Finally, statistical dependency is not necessarily consistent with the logical dependency of document descriptors, and it is more difficult to determine their operational effects than Boolean systems where explicit Boolean operators are used.

### 3.1.3 Generalized Boolean retrieval model

The research work on the generalization of Boolean retrieval model has been encouraged by the fact that, despite the flaws of the Boolean model, most commercial document retrieval systems are of the Boolean type. In an attempt to overcome some of the flaws of the Boolean retrieval model, fuzzy subset theory has been applied to enforce a partial-matching mechanism. This lets indexers and users to indicate the importance of index terms and query terms by attaching to them a numerical value between zero and one. It is called the generalized Boolean retrieval model in the sense

that the model conforms to regular Boolean logic when the numerical values specified are restricted to the values of zero and one.

A generalized Boolean retrieval model is described as follows [25]. There is a set of documents  $D$ , a set of index terms  $T$ , and a fuzzy membership function  $u: D \times Q \rightarrow [0,1]$  such that  $u(d_i, t_k)$  measures the extent to which document  $d_i$  is about the concepts represented by term  $t_k$ . Queries are made using a Boolean expression composed of index terms. For each single term  $t_k$ , there is a corresponding fuzzy subset  $M(t_k)$  of documents, called the meaning of term  $t_k$ , i.e.,

$$M(t_k) = \{ \langle d_i, u(d_i, t_k) \rangle \mid d_i \in D \wedge t_k \in T \}$$

For more complex queries, its meaning is constructed by standard set theory:

$$M(t_1 \text{ AND } t_2) = M(t_1) \cap M(t_2),$$

$$M(t_1 \text{ OR } t_2) = M(t_1) \cup M(t_2),$$

$$M(\text{NOT } t) = M(t)',$$

where the union, intersection and complement are fuzzy subset operations. Attempts at generalization strive to find a way to mathematically link the fuzzy membership function with the query weight to produce a retrieval status value which is essentially the fuzzy membership function of the document in the meaning of the query.

In the early literature, a membership function value of one was implicitly assumed for each query term, and thus was called fuzzy indexing with a Boolean query. The matching function provided was simply the application of standard fuzzy subset operators, such as MAX, MIN and one-minus, for OR, AND, and NOT, respectively. Thus, all the properties of the Boolean lattice were preserved except complementarity [24]. Further generalization leads to a matching function associated with

a general threshold via a "lambda-level meaning" such that the membership function value will remain the same if it is above the threshold, or drop to zero. In this system, it has been shown that some of the lattice properties still hold [25].

The most general case is called fuzzy indexing with a fuzzy query [26,27,28], where query terms are weighted as well as index terms. Problems immediately arise when one interprets query term weights as relevance weights. That is, a relevance weight suggests a monotonic function  $f$  to be used to bind the fuzzy membership function and relevance weight in order to maintain the structure of the Boolean lattice; however, the nature of the restrictiveness of the AND operator is just opposite to the nature of the expansiveness of the OR operator, and the function  $f$  will not satisfy both simultaneously.

Several solutions have been proposed. One is to use different functions for query terms connected by AND versus OR operators, as in Bookstein's model [26] and Yager's model [50]. In fact, Bookstein suggested

$$f(u(d,t),a) = a \cdot u(d,t),$$

unless the term is to be ANDed, else

$$f(u(d,t),a) = \text{MIN}(1, u(d,t)/a).$$

Here  $a$  is a relevance weight assigned to query term  $t$ . The retrieval status value thus calculated is consistent with the nature of Boolean logic, but violates a critical condition for maintaining lattice structure, the Waller-Kraft separability criterion [47], which says that a document is to be evaluated first along each term separately and then combined via the Boolean logic of the query. Yager suggested

$$f(u(d,t),a) = \text{MIN}(u(d,t),a)$$

unless the term is to be ANDed, else

$$f(u(d,t),a) = \text{MAX}(u(d,t), 1-a).$$

Again, the separability criterion is violated.

A solution suggested by Kantor [27] was to develop an alternative logic, the "logic of weighted queries," which preserves the commutativity, associativity, involution and deMorgan's law properties, but loses idempotence, distributivity, absorption, and, as in all fuzzy subset work, complementarity.

A third proposal [28,53,54] reinterprets the query weights as threshold values imposed by the user. Their suggested function is

$$f = \left(\frac{1+a}{4}\right) \left(\frac{u(d,t)}{a}\right)$$

when  $u < a$ ; otherwise,

$$f = \left(\frac{1+a}{4}\right) + \left(\frac{a}{2}\right) \left(\frac{u(d,t)-a}{1-a}\right)$$

This model produces a system mathematically consistent with the separability criterion.

As elegant as a fuzzy subset is, it has not been proven that an information retrieval system based on it would offer superior performance over other systems. A careful examination will immediately reveal that the functions suggested above could produce undesirable results in some special cases.

A great deal of effort has been made to unify the various models [18,76]. One of the most ambitious tries belongs Cater's Topological Information Retrieval System(TIRS) [76], based on the topological paradigm. He claims that the paradigm is a unifying model, in that all of the standard models, i.e., the Boolean, vector space,

fuzzy set theoretic, and probabilistic models, are instances of the paradigm. However, the TIRS type generalization aims at the level of conceptual representation of the documents; in this sense, the general mathematical model proposed in [13] is even more general. As far as the effectiveness of information retrieval is concerned, a critical problem lies in developing a mapping mechanism, which takes a document and the query to produce for the document a retrieval status value. In this aspect, the TIRS proposes a metric which is a distance-like measure and has not been shown to be as effective as the other models for information retrieval.

### 3.2 Foundations of composite retrieval model

It has been seen from the literature review that the primary concern of an information retrieval model is to describe the relationship between a user's information request and each document in a given collection. In the vector space model, such a relationship is designated in terms of a similarity measure between the query vector and the document vectors. The justification of using a vector similarity measure function was demonstrated by Bookstein [16]. He shows that for a variety of index term distributions, the vector space model precisely describes the optimal retrieval decision by the within-document frequency of occurrences when the projection of the query vector along the axis associated with a term measures the ability of the term to distinguish documents according to the probability of relevance to the request. What we gain from that discussion is that due to the complexity of measuring the query-document relationship as a probability of relevance, it is possible to use a much simpler vector similarity measure to provide a reasonable approximation that reflects an optimal retrieval rule. Nevertheless, this result is not surprising, since both models under discussion are based on the same underlying principles. That is, documents and queries are represented by index terms, and these index terms are assumed to be statistically independent.

It can be anticipated that the mathematics would become much more complicated for both models if we allowed interrelationship between index terms to exist [19,51]. Under these circumstances, we argue that the application of fuzzy subset theory may provide an alternative with the advantages of simplicity of mathematical

treatment and incorporation of term dependence. For example, fuzzy subset models as described in literature [26,27,28] directly support a retrieval system with the ability of weighted indexing, weighted queries, and ranked output presentation. That is, each index term of a document has attached a numerical weight representing the extent to which the document is about the the concepts represented by the term. Moreover, each query term can be assigned a relevance weight or threshold value, and the documents retrieved in response to a user's query are ranked according to the fuzzy membership function calculated for the query through standard fuzzy subset operations. In the fuzzy subset model, the notion of term dependence is directly built in; that is, when several terms are assigned to a document, their different membership function values actually reflect differences and relationships among terms. In retrieval, the logical relationships among query terms are also reflected by the logical operators connecting the query terms.

Critics of fuzzy subset model have two main arguments [23]. First, conceptually, fuzzy indexing, fuzzy queries, and ranking are not new to information retrieval. However, the most important contribution of putting these concepts all under the fuzzy subset framework is that we have removed the mathematical restrictions of these concepts, from a theoretical point of view. For example, in the interpretation of a fuzzy index, we are only concerned that the value of the membership function be in the interval  $[0,1]$ , but not about how the value is obtained. Therefore, it is possible for us to incorporate more factors of term significance instead of using only the within-document frequency of term occurrences, while there is not much choice in the probabilistic model. In the case of fuzzy queries, the probabilistic model may use complete



term relevance information to assign weights to search terms [42], or it may assign the precision value of each query term in order to construct a maximum entropy distribution [21]. These seem much more difficult and more restrictive than relevance weight assignment by the user, which might be chosen in the fuzzy subset model.

Second, the fuzzy subset model has been criticized for the use of the MAX evaluation mechanism for union operations and MIN for intersection operations, which some feel has been shown inappropriate for document retrieval [23]. These weaknesses are also observed in traditional Boolean retrieval systems. However, it should be noticed that choice of MAX and MIN are only implementation matters, rather than an intrinsic property of fuzzy subset model. It may be impossible, however, to invent any other functions which can fit perfectly under the fuzzy subset framework without a loss of mathematical properties [45]. Thus, it is the task of a new model to overcome such drawbacks, yet retain, as much as possible, the appealing features of document retrieval.

The fundamental principles of developing a composite retrieval model are (1) to reconcile the Boolean retrieval model with the vector space model while taking into consideration the logical dependence of query terms, and (2) to accommodate more relevance indicators to improve the effectiveness of retrieval.

Our main concern is not the mathematical aspects of generalizing the Boolean model such that the underlying lattice structure of the fuzzy subsets can be maintained. We are more concerned with the operational aspects which would allow a user to represent his information needs in a convenient and effective way. That is, a query language with predesigned levels could be provided to allow a user to portray the

ideal documents in his mind by assigning weights to query terms and designating the logical dependences among query terms. A query is interpreted by a new evaluation mechanism, which is consistent with the traditional Boolean model when weights take values from  $\{0,1\}$ , and which resembles the vector space model when only the Boolean operator AND is applied in a query expression.

Effectiveness of information retrieval, our second goal, is often evaluated in terms of some relevance measurement, one of the most important notions whose meaning has been discussed for decades in the field of information retrieval [36,37,38,39,62]. Some researchers advocate that the notion of pertinence should be separated from the notion of relevance, but that has never been fully accepted in current retrieval models. However, it is generally agreed that documents should be ranked and presented to the user in order to reflect the relevance relationship of documents with respect to a user's information needs rather than the user's information request [38].

As far as the notion of relevance is concerned, we feel that there are two aspects worth mentioning. First, the term "relevance" bears different meanings, as it can be interpreted from different points of view [36,37,38,39,62]. Second, the notion of a retrieval status value, as proposed by Bookstein [13], is a more appropriate phrase to be used in document ranking models, since then other standards, like relevance or pertinence, could be considered as a realization of the retrieval status value and be interpreted accordingly. Such a relationship between those two notions must be well understood so that the concept of relevance will not be abused. For example, it would hardly be correct to claim that a relevance measure in a ranking model reflects a great

degree of user satisfaction if only index terms are considered.

Various relevance indicators have been observed and tested in the past [29,35,58,59]. The rationale behind combining relevance indicators is the user's actual information seeking behavior. Mansur [63] states that from everyday experience it is known that users search for information by using index terms, plus authors' names, citations, and other attributes. Cleveland [35] tests the affinity relationship among four relevance indicators, index terms, journals, authors, and citations in his model. Other efforts have explored more relevance indicators, including incorporation of the age of documents into the retrieval process. Some have advocated using citation as a relevance measure that is different from index terms.

However, these have not yet been shown to be successful. For example, it has been suggested that for the vector space model, the coordinates of the vectors could be extended to cover other factors, such as factual identifiers and citation strength. But, problems can arise with the ranks of the document vectors, because the assumption of mutual orthogonality may not be valid; if not, the traditional vector similarity measure function would no longer work properly. In the model developed by Heine [64], the age of documents is incorporated into the probabilistic model of index terms; in that case, the computational complexity for evaluating the multi-variate distribution was still a problem for the operational environment. Moreover, the assumption of stochastic independence between various properties had to be made.

Our composite retrieval model differs from the others in that the retrieval process is decomposed into two phases. In the composite retrieval model, we define two different relevance measurements: a topical relevance score and a preferable relevance

score. The former is purely determined by an index term matching mechanism, and the latter is determined by the combined effects of several other factors concerned with user preference. That is, we differentiate relevance, or concept relatedness, by using the term "topical relevance" as a measure of subject relatedness of a document with respect to a user's query submitted to the system, and the term "preferable relevance" as a measure of the degree of satisfaction of a document with regard to a user's information needs. Thus, our notion of relevance, unlike the probabilistic model in which a document is considered either relevant or non-relevant, is a fuzzy one. In the composite retrieval model, retrieval proceeds by first searching for topically relevant documents by means of index terms only, and then ranking the topically relevant documents by means of some factors of user preference in order to achieve better satisfaction with respect to a user's information needs. We assume that topical relevance of a document with respect to a user's information request is a necessary condition of being preferably relevant with respect to the user's information needs. Here, our notion of preference factors is the same as the notion of relevance indicators [63], except that the factor of index terms is not included. Since a number of relevance indicators have been discussed in literature, the following work will explicate our basic ideas in order to organize selected preference factors in a natural way.

First, we want to describe the relationship between our composite retrieval model and the usual one-phase retrieval model. In our composite retrieval model, the first phase is a matching procedure in which index terms are exclusively used to determine a set of documents topically relevant to the user's information request. This is within the scope of Bookstein and Cooper's general mathematical model [13] in that the

result of the matching process is a weakly-ordered set of documents. This model would fail to function as a ranking model when the number of elements of a subset becomes large. That is to say, if we view the weakly-ordered set as being decomposed into a number of subsets in terms of a topical relevance score, all the elements belonging to a subset are indistinguishable. Another potential problem is that the ranks of subsets of the weakly-ordered retrievable document set are solely determined by the matching mechanism, which might lead to an undesirable result. For example, if document  $DOC_1$  is in the subset with relevance score 0.679 and document  $DOC_2$  is in the subset with relevance score 0.678, will it be always true that  $DOC_1$  is more topically relevant than  $DOC_2$  with respect to a user's information request? People have every reason to doubt it unless the accuracy of the matching mechanism is fully demonstrated. A third problem is that ranking documents in terms of topical relevance will not fulfill our fundamental goal of satisfying a user's information needs rather than his information request. Thus, the second phase in our composite retrieval model is specifically developed for the purpose of ranking documents in order to achieve better satisfaction with respect to a user's information needs.

Second, we want to defend the choice of index terms as topical relevance indicators used in the matching process. Index terms have been taken as the most important relevance indicator and been applied to most of the proposed experimental information retrieval models and all existing commercial information retrieval systems. The retrieval systems based on index terms of documents are relatively easy to implement and yet, as Salton reports, provide superior performance over some complicated syntax-oriented systems [4]. In our composite retrieval model, a user's information

request is presented in the form of a restricted Boolean expression of index terms, which is taken as a proper description, or mental prototype, of the ideal document that he is seeking. Under the circumstance that there is no other information provided to obtain an initial document set, we argue that the use of index terms as the topical relevance indicator is not only natural but also best among relevance indicators observed in the literature.

Let's consider citation first. As it has been discussed in the literature [59], each reference or citation between two documents represents a relationship indicator for the documents; however, a direct reference does not imply identity in the subject areas covered by the documents. Two stronger indicators discovered are bibliographic coupling and the co-citation link, where the coupling strength is defined as the number of references in common for both documents, and the strength of a link is defined as the number of documents that jointly cite the two documents. Then, a natural usage of citation retrieval is to find for a given set of documents some other ones which may be topically related to them. Hence, it will not help at the first stage of retrieval where we don't have an initial set of documents in response to a user's query. Furthermore, as Salton indicates [4], the occurrence of bibliographic citation is still a comparatively rare phenomenon. The results of an experiment showed that about 25 percent of all published papers were never cited at all. Of those that were cited in a particular year, 72 percent were cited once in that year, and 18 percent were cited twice. Thus only about 5 percent of the archive of citable papers were cited at least three times in a given year [55]. Some other relevance indicators, such as authorship or relationship between journals, share similar properties and thus may be used as a conditional

relevance indicator.

For the purpose of ranking documents according to their preferable relevance with respect to a user's information needs, we first consider why one document may be preferred to another. Our opinion is that a certain type of user prefers document  $DOC_1$  over document  $DOC_2$  simply because (1)  $DOC_1$  is more topically relevant to his description of the ideal documents he is seeking, (2)  $DOC_1$  possesses a better reputation for quality, (3)  $DOC_1$  represents more recent research work in the area of interest, (4)  $DOC_1$  is more suitable for him to read in terms of his background or research interest, and/or (5)  $DOC_1$  can be used to reach  $DOC_2$ , for example,  $DOC_2$  is in the reference list of  $DOC_1$  so that the user can access  $DOC_2$  after browsing  $DOC_1$ . In our composite retrieval model, the degree of topical relevance between a document and a user's information request is solely determined by the matching mechanism, which is based on the index structure. On the other hand, the degree of preferable relevance between a document and a user's information needs is determined by the combined effect of four preference factors: quality, recency, fitness, and reachability. The two-phase retrieval procedure in our model is somewhat like the one in SIRE system [77], which first retrieves a response set of documents by performing a traditional Boolean search, and then ranks the documents in the response set in descending order by the cosine correlation value computed between the retrieved documents and the query. However, our system is quite different from the SIRE in that our matching mechanism has included both the retrieval function and ranking function of SIRE, but our ranking facility works in terms of preference factors other than the index terms. The remaining question is how to precisely define and quantify these factors so that the order of

presentation of documents could be decided on the basis of preferable relevance with respect to a user's information needs.

By quality, we mean the external features of usefulness of a document which are valued from the view of a large population of information users. For example, the quality of a document might be determined by the author's reputation, or citation strength, or the reputation of sources. One may be convinced from daily experience that an author's reputation has an overwhelming effect on a user's information seeking behavior. If a person has a deep interest in the area of information retrieval, for example, he shouldn't miss Salton's papers. Thus, Salton's papers might be preferred by an information user over some other documents by authors whose names are not as well known. The remaining problem to be solved is how to quantify the relationship between quality and author's reputation, citation strength, and reputation of sources(e.g., journals). In our model, the quality of a document with respect to a given class of users is designated by a numerical value in  $[0,1]$ .

By recency, we mean the effect of time on a user's preference towards the documents to be retrieved. Generally, users prefer papers more recently published over papers published a long time ago. The value of a document has been modeled by Morse [65] as he describes book obsolescence as a Markovian model relating circulation in one year to that in the next. An example of the use of such a model in library planning is reported by Hindle [67]. However, this model of book obsolescence is not directly applicable to our ranking model, since the value of a specific document, as indicated by recency of publishment, might be insignificant with respect to a given user's preference. That is to say, people might not take a specific document published



in 1981 as being significantly more appealing than the one published in 1980. For one user, a three year gap since publication might significantly affect his choice, while for another user, it might be a gap of over five years. Thus, in our model, the contribution of recency of documents to the preference score is determined in the course of retrieval by a set of rules reflecting experts' opinion.

By fitness, we mean the match between a user's background and features of documents to be retrieved. It is the system designer's task to specify a number of features for the document collection. For example, we may specify type (e.g., surveys, articles, technical reports, or conference papers), level (e.g., theoretical or practical), and style (e.g., long or short) as three features. Then, for some users, a long theoretical article is preferred, while for another user, the opposite is desired.

By reachability, we mean the degree of easiness in obtaining a document. There are two aspects to consider. One is to trace the document by means of the citation network among the topically relevant documents. The other is to access the document by means of the document retrieval system or through a computer network. There are two cases of reachability which may affect the user's preference in terms of the citation network. First, we would give some priority to a document which is not in the reference lists of the other documents, because this document cannot be traced through the citation network and will be ignored if it is not directly browsed by the user. We call it as a principle of protecting referential losses. Second, we prefer retrieving document  $DOC_1$  over document  $DOC_2$  if  $DOC_2$  is cited by  $DOC_1$ , because we can easily reach  $DOC_2$  after we have browsed the document  $DOC_1$ . In terms of accessibility we mean that the harder it is to access the document, the more valuable the document

would appear to a user. Thus, system designers must carefully analyze both aspects and combine different views in order to set up rules for the assignment of the reachability score to each of the documents to be ranked.

The ranking submodel of our composite retrieval model includes use of a user classification file, conceptually similar to the personality file discussed by Kemp [39], when he proposed the idea. Our user classification file would be created by system designers, based on general world knowledge and their previous experiences with on-line information retrieval. The user classification file provides an operational basis for the ranking submodel such that all of the users that fall into the same category would be treated as though they share a common view on preference factors such as document quality, recency, fitness and reachability, as described above. However, users belonging to different groups may have quite different tastes, which are represented in term of a fuzzy membership function determined by a set of rules specified by the system designers. The logical relationships among preference factors is also presumed to be determined on the basis of expert knowledge such that the preferable relevance score will be produced in a similar way to that of the evaluation of a logical expression of index terms. Thus, the ranking model functions as a reasoning mechanism, and the documents finally presented to a user are ordered by means of the retrieval status value, which is implemented as a preferable relevance score.

## **CHAPTER 4**

### **COMPOSITE RETRIEVAL MODEL**

#### **4.1 Description of composite retrieval system**

The composite document retrieval system proposed here consists of a query language, a number of databases, and six functional components. The main databases include the document collection, document profiles, document description records, and knowledge bases (including a user classification file and inferential rules). The functional components are an indexing component, a query processing component, a matching component, a ranking component, a physical access component, and a control component. These components might be considered as subsystems or modules to form an integrated document retrieval system.

There are two main features of our composite retrieval system. The first is the use of a ranking model, in addition to a matching model, to rank the response set of documents by means of a preferable relevance score which reflects the combined effects of four preference factors discussed in the previous Chapter. The second is the use of a composite indexing model which advocates separating index terms and document descriptors. Index terms are based on word types and are organized as an inverted file to be used by the query processing module to locate all topically related documents as quickly as possible. Descriptors are based on conceptual phrases and are organized as a document description file to be used by the matching module in order to differentiate between the documents in the response set by means of a topical

relevance measure. In addition, we assume that the entire document set is partitioned into a number of subsets to form an artificial environment under which several strategies can be applied to help select coefficients for the Composite Weighting Function(CWF).

The retrieval process is as follows. A user submits an information request in the composite query language, which is a retrieval statement called the raw query. The raw query is processed by the query processing module to validate the syntax and then to decompose the query into an exact retrieval part and a relevant retrieval part. An appropriate subset of documents, denoted as  $D_0$ , is first selected from the entire partitioned document collection, which is sufficient as an initial response set with respect to the user's raw query. Next, the exact retrieval part is manipulated as in a traditional database system, resulting in  $D_1$ , a reduced set of  $D_0$ . Then, the relevant retrieval part is manipulated by means of the indexing structure, resulting in  $D_2$ , a reduced set of  $D_1$ . Manipulation of the relevant retrieval part is generally a repetitive process in which heuristic rules and user feedback are incorporated to revise the response set  $D_2$  so that the specified performance criteria are fulfilled as well as possible. Once the response set  $D_2$  is fixed, the retrieval procedure proceeds by invoking the matching module to obtain a response set  $D_3$ , which is the same as  $D_2$  except that the documents in  $D_3$  are weakly ordered in terms of topical relevance scores. A topical relevance score is a numerical value in  $[0,1]$  to reflect the extent to which the document being assigned is topically related to the query. Finally, the ranking module, which is a knowledge-based reasoning mechanism, is invoked to produce a response set  $D_4$  which is the same as  $D_3$  except that documents in  $D_4$  are ranked in terms of preferable relevance

scores. A preferable relevance score is a numerical value in  $[0,1]$  indicating the degree of user satisfaction with respect to the information needs of the user from the system's point of view. The ranking model works in such a way that only a subset of documents is to be ranked each time according to the topical relevance score and related selection policy. As each subset of documents is submitted to the user, a simple evaluation is obtained from him. The ranking module can then modify the parameters of the ranking algorithm according to the user's feedback in order to improve the degree of consistency between the views from the system and from the user. The preferable relevance score is treated as a retrieval status value. Each subset of documents, ranked in decreasing order of preferable relevance scores, are presented to the user through the access module, which has logical access to the databases. The whole retrieval process is accomplished under the control of the control module. The architecture of the composite retrieve system is illustrated in Figure 4.1.1.

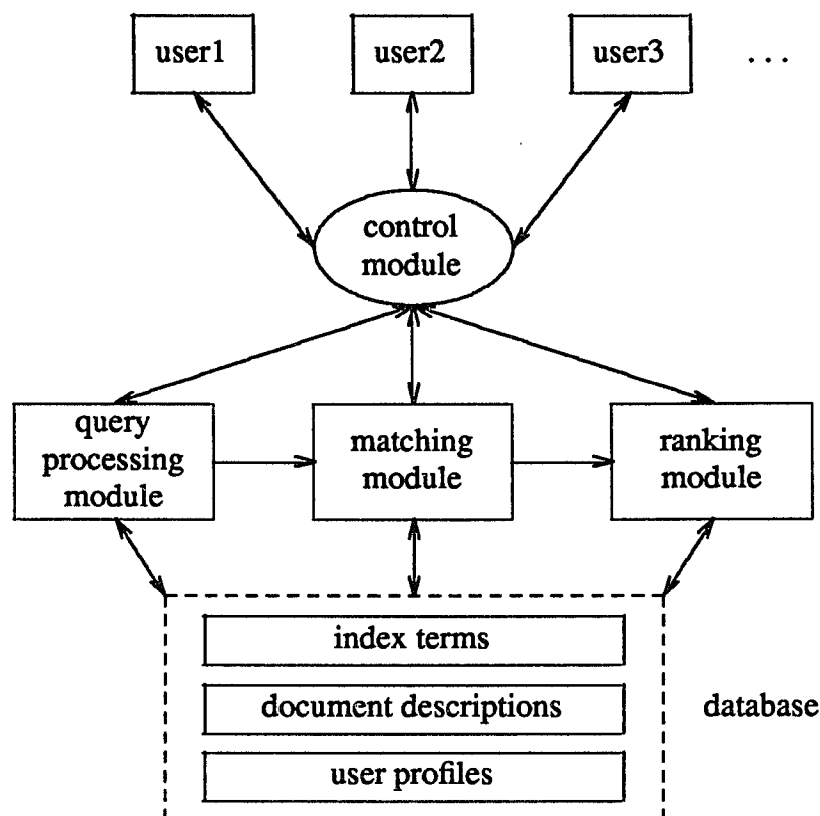


Figure 4.1.1 Architecture of the composite retrieval system

Mathematical models for the indexing subsystem, the query processing subsystem, the matching subsystem, and the ranking subsystem, as well as the composite query language are described in subsequent sections. Since this paper is devoted to information retrieval, data retrieval is purposely ignored. Users interested in that area are encouraged to refer to [48,49]. Also, for notational simplicity, we shall use the term information query to denote the relevant retrieval expression in the user's request in subsequent discussions.

## 4.2 Composite Query Language

Various query languages have been discussed for the purpose of information retrieval in the literature [14,60,61,73]. As a result, three types of expressions have been adopted or proposed under either an experimental or commercial environment.

TYPE 1: retrieval by pre-specified attributes, for example, retrieval by author's name or by title.

TYPE 2: retrieval by index terms (also referred to as keywords).

TYPE 3: retrieval by statements in natural language.

For a TYPE 1 expression, there are three typical cases: (1) using a single attribute, (2) using multiple attributes connected by Boolean operators, and (3) using Boolean expression of attributes with qualifications. An example of case (3) is  
 RETRIEVAL DOCUMENTS WITH AUTHOR\_NAME='JONES' AND  
 MAGAZINE\_NAME='JASIS' AND 1980 < PUBLISH\_DATE < 1983.

For a TYPE 2 query expression, there are four typical cases: (1) using single keywords, (2) using multiple keywords connected by Boolean operators, (3) using weighted key words connected by Boolean operators, and (4) using single or multiple keywords with a thesaurus. We consider a term vector query as a specific case of (3) in which only the Boolean operator AND is implicitly imposed.

For a TYPE 3 expression, the actual effect is not as 'natural' as the name 'natural language' appears to be, because a natural language query has to be converted into a proper form of TYPE 1 and/or TYPE 2 query before the process of search actually proceeds. The natural language query has been implemented in some systems with

restrictions on vocabulary and/or structures, while rejected by some researchers as being inappropriate for the purpose of information retrieval.

The composite query language proposed here is a combination of TYPE 1 and TYPE 2 expressions, while query terms are no longer limited to keywords. That is, a user can request information by specifying predesigned attributes connected by Boolean operators and/or by describing what the information seeking is about in his own words or phrases connected by logical operators. The simplified syntax of retrieval statements of the proposed query language is

RETRIEVE WHERE <TYPE1-expression>  
ABOUT <TYPE2-expression>.

Generally, a TYPE1-expression is designated as data retrieval and can be implemented by means of a relational query language like SEQUEL [48]. For example, a query RETRIEVAL WHERE AUTHOR='KRAFT' is equivalent to the query

SELECT \*  
FROM DOCUMENT  
WHERE AUTHOR='KRAFT'

in SEQUEL, assuming DOCUMENT is a relation with AUTHOR as one of its attributes. Our discussion focuses on the TYPE2-expression, which is often referred to as information retrieval in the sense that it is an imprecise representation of original documents and response seeking is only relevant to the issue presented by the user. Our TYPE2 expression differs from traditional weighted Boolean expression in three aspects: (1) the query structure is limited to three levels (to be defined below), (2) query weights bear more meanings to be interpreted, and (3) query terms are based on



conceptual phrases which are supported by a powerful matching model with an expert's knowledge. Thus, we refer to our TYPE2 expression as a Phrase-Oriented Fixed-Level Expression, or POFLE.

The highest level of POFLE is designated as a clause, a variation of a weighted Boolean expression. A clause may be a single facet or a number of facets connected by OR operators. Each facet in a clause has attached to it a numerical value, known as a facet weight, in the interval [0,1] to reflect the relative importance of that facet. Where there is only one facet, a full weight of one is assumed.

The facet is one of the most important concepts in POFEL. A facet describes a profile of the ideal documents for which a user is seeking. The OR operator connecting the facets differs from the conventional interpretation of an OR in the sense that the facets being connected by it represent relatively independent coverage instead of highly correlated synonyms. A facet may consist of an element or a number of elements connected by AND operators. Each element in a facet has attached to it a numerical value, known as an element weight, in the interval [0,1]. When there is only one element, a full weight of one is assumed.

While facet weights differentiate between the relative importance of facets connected by OR operators, element weights differentiate between the relative importance of elements connected by AND operators. However, differences between the logical operators OR and AND cause element weights in POFLE to carry more meaning than facet weights. That is to say, for a given query  $A_a \text{ OR } B_b$ , where the subscripts  $a$  and  $b$  are the weights of the query expressions  $A$  and  $B$ , respectively, we mean that documents about  $A$  and documents about  $B$  are equally significant when  $a=b$ , and the

former would be preferred to the latter when  $a > b$ . On the other hand, for a given query  $A_a \text{ AND } B_b$ , we mean that the desired documents must cover both  $A$  and  $B$  when  $a = b$ , and the documents will become more desirable if they cover  $B$  in addition to  $A$  when  $a > b$ . Thus, we refer to the AND operator of POFLE as the 'loose' AND operator for a weighted query.

We define two types of elements. One is denoted as simple element, which consists of a simple word or a simple phrase, with or without the negation operator NOT. The meaning of a single word is obvious. A simple phrase is a phrase in the usual sense, i.e., a meaningful composition of single words conforming to various syntactic and semantic rules. We also refer to a simple element as a query descriptor. The other is denoted as compound element, which may consist of a list of alternative simple elements, called a selective element, or a list of co-existent simple elements, called a joint element, or an element followed by another element representing the relationship that the former is significant if and only if the second is present, called a conditional element. The NOT operator is only allowed to be used in front of a simple element.

A selective element designates the synonymity relationship between certain words or phrases such that either one could be used as an alternative representative in the context of a query. A joint element designates the co-existence relationship among certain words or phrases that must be covered simultaneously by the document being sought. A conditional element designates the existence-dependent relationship to mean that an element represents meaningful coverage only when some other element is present as a context for information seeking.

Let us consider a few examples to demonstrate the usefulness of the above structure of POFLE. Suppose a user is seeking for documents about the application of fuzzy subset theory in the area of information retrieval. He might formulate his query as a joint element (FUZZY SUBSET & INFORMATION RETRIEVAL). There may be a problem in that since INFORMATION RETRIEVAL is a very popular term in the documents related to the subject, it carries a much lighter weight than the term FUZZY SUBSET; however, the use of the joint element, i.e., the conventional AND operator, will make it a dominant one over the descriptor FUZZY SUBSET, which is contrary to the user's true intention. So, a more appropriate way for user is to formulate his query as a conditional element (FUZZY SUBSET | INFORMATION RETRIEVAL). As a result, the matching model will first look for the documents in the context of INFORMATION RETRIEVAL, i.e., documents either indexed by INFORMATION RETRIEVAL or documents being members of a document subset which is related to the subject of INFORMATION RETRIEVAL. Then, from the above documents, the system picks up those that are topically related to FUZZY SUBSET, with topical relevance scores being determined solely by the descriptor FUZZY SUBSET.

Another example is the use of the selective element. Suppose that a user is looking for documents about DOCUMENT RETRIEVAL SYSTEM, or, synonymously, INFORMATION RETRIEVAL SYSTEM. He will risk missing some desired documents if he merely puts down one of these two descriptors. However, if he decides to put them together as a clause, he would have to assign facet weights to the two facets. The system would then try to distinguish them if he does so. He could formulate his

query as a selective element (INFORMATION RETRIEVAL SYSTEM, DOCUMENT RETRIEVAL SYSTEM) so that the matching model would treat them as two alternative choices. Since our matching model is designed to consider two facets connected by the OR operator as two relatively independent or partially related topics, and descriptors involved in a selective element as synonymous terminologies, the user's appropriate choice of query structure provides a valuable source for system learning in order to improve the indexing facility. For instance, the statistics of use of selective elements, joint elements and conditional elements could be gathered and applied to enlarge or modify the synonym dictionary and thesaurus dictionary to be defined in subsequent section.

### 4.3 Indexing model

#### 4.3.1 General description of the indexing model

The indexing subsystem of the composite document retrieval system is a software tool for system designers to create and maintain a variety of databases to be used for the purpose of information retrieval on the basis of the source databases and predesigned algorithms.

Formally, our indexing model is defined by the triple

$$I = \langle D', D, g \rangle$$

where  $D'$  is a set of source databases,  $D$  is the set of object databases and  $g$  is a set of algorithms. The source databases are those to be initially loaded as input to the indexing subsystem, while the object databases are those to be created and maintained by the indexing subsystem for use by the information retrieval routines throughout the life of the system. The algorithms are a set of procedures which creates and reorganizes the object databases based on the source databases and integrity rules. The main algorithm is known as the indexing model.

There are two types of source databases. One is a general purpose database which can be adopted by a document retrieval systems under one of several different environments, while the other is a special purpose database which would be available only under the environment of a concrete system being developed.

General databases include:

(1) Dictionary of non-informative words. We define non-informative words, as opposed to informative words, as those that are used for the purpose of satisfying syn-

tactic or rhetoric needs and those whose meaning is trivial with respect to the information conveyed by the documents. They are also called "stopping words" in the sense that all the words that appear in this dictionary would be excluded from the set of index terms. The non-informative words usually include articles, prepositions, pronouns, verbs with trivial meaning, and so on. Use of this dictionary of non-informative words effectively removes hopeless candidates from the indexing vocabulary during the indexing procedure. Table 4.3.1 gives a sample of the dictionary of non-informative words, in which the numerical number following a word represents an entry of possible actions regarding the word for the text analysis.

a	(article)	01001
about	(preposition)	02001
about	(adverb)	03001
almost	(adverb)	03002
are	(verb)	04001
be	(verb)	04002
before	(preposition)	02002
but	(conjunctive)	05001
can	(auxiliary)	06001

**Table 4.3.1 Sample: Dictionary of non-informative words**

(2) Dictionary of linguistic synonyms. We define linguistic synonyms as those single informative words that are of the same part of speech and have the same morphological features. Such linguistic synonyms can be regarded as synonymous thesaurus. In many cases, a word can be replaced by its linguistic synonym without distorting the information conveyed by the information unit in which the original word plays a role. For the purpose of document retrieval, we suggest that the linguistic synonyms in the dictionary be limited to the nouns and adjectives. Table 4.3.2 gives an excerpt of a sample dictionary of the linguistic synonyms.

abdomen	:	belly	stomach	paunch
aberration	:	derangement	alienation	
ability	:	capacity	capability	
able	:	capable	competent	qualified
abnormal	:	atypical	aberrant	
abortion	:	miscarriage		
abstracted	:	preoccupied	absent	distraught
accident	:	casualty	mishap	
accidental	:	casual	fortuitous	contingent
acquirement	:	acquisition	attainment	accomplishment
actor	:	player	performer	thespian
acute	:	critical	crucial	

**Table 4.3.2 Sample: Dictionary of linguistic synonyms**

(3) List of suffixes. Suffix stripping has been shown as an effective technique to obtain a set of word types instead of words that can be taken as index terms. Thus, "work", "working", "works", and "worked" all become the stem "work". Use of suffix stripping effectively reduces the size of the indexing vocabulary in the indexing system. Table 4.3.3 gives an excerpt of the list of suffixes.

able	ed	ial	tion
age	en	ian	tious
al	ence	ible	tress
ally	ency	ic	trix
an	ent	ical	tude
ant	er	ing	ure
ard	ery	ise	ward
ary	es	ish	ways
ate	ess	ism	wise
ation	est	ist	y

**Table 4.3.3 Sample: List of suffixes**

The special databases include

(1) The collection of documents available to all system users. Documents might be full text or some representative parts, such as bibliographic information (e.g.,

authors and title), depending on system resources and usage.

(2) The collection of document profiles, each corresponding to an individual document in the document collection. The attributes of a document profile are specified by the system designers and are subject to change. Table 4.3.4 gives a possible list of attributes for a document profile.

**Table 4.3.4 Attributes of document profile**

- . document identification(id) number
- . title
- . author(s)
- . authors' address(es)
- . date of publication
- . publisher
- . volume, number, pages
- . type of paper(journal, technical report,...)
- . source (journal title)
- . references
- . keyword

(3) A dictionary of the subject catalog, prepared either manually or automatically for the given document collection. Each subject in the dictionary has an entry to describe it in terms of descriptors, called subject description record. Note that, if system designers intend to make use of the subject descriptions to find out a specific subject with respect to a user query, then the degree of orthogonality between the subject descriptions must be considered; otherwise, a subject description is simply a pool of contextual thesaurus under its subject name and can be used to enhance a user query. The subjects are specified in such a way that the entire document collection could be partitioned into equivalence classes with each one being sufficient to provide a response set of documents with respect to a given query. An example of subject cata-



log for a document retrieval system in the area of computer science would include operating systems, information systems, programming languages, computing theory, and so on. Table 4.3.5 gives an example of the dictionary of the subject catalog in the area of computer science.

0001	artificial intelligence
0002	compiler/assembler
0003	computer architecture
0004	computer graphics
0005	computer simulation and modeling
0006	computing theory
0007	database system
0008	information retrieval
0009	operating systems
0010	pattern recognition
0011	programming languages
0012	software engineering

**Table 4.3.5 Sample: Catalog dictionary for computer science**

(4) A dictionary of synonymous terminologies related to each subject. The synonymous terminologies differs from the linguistic synonyms in that they are not limited to single words and are synonymous with respect to a subject. An initial set of synonymous terminologies may be specified by human experts in the subject area. Then, the dictionary will be enlarged or modified by the system through heuristic rules and usage statistics. This dictionary provides a means to increase the chances of matching between a query and relevant documents. For example, synonymous terminologies may be used to enlarge a selective element in the query when matching effort fails or is unsatisfactory. Table 4.3.6 gives an excerpt of a sample dictionary of synonymous terminologies.

auxiliary storage	:	secondary storage	
Boolean algebra	:	switching logic	
compilation	:	translation	interpretation
data bank	:	database	
document retrieval	:	information retrieval	reference retrieval
Polish notation	:	parenthesis-free notation	
privileged instruction	:	supervisor call	
structured programming	:	modular programming	top-down design

**Table 4.3.6 Sample: Dictionary of synonymous terminologies**

(4) A dictionary of contextual thesaurus. We define the contextual thesaurus as those phrases (single word or multiple words) that are related in terms of co-existence dependency or conditional dependency with respect to users' information requests. Thus, unlike the key-word-in-context index which is typically produced by removing non-informative words from titles or text portions and including in the index an entry for each of the remaining text words, our contextual thesaurus is to be generated by means of joint elements and conditional elements appeared in users' query expressions. A method of selecting the contextual thesaurus is discussed in the query processing model. Table 4.3.7 gives a sample of the dictionary of contextual thesaurus.

TERMS IN CONTEXT		
fuzzy set theoretical model	document retrieval	vector space model
bibliographic coupling	document retrieval	citation link
relevance	document retrieval	pertinence
retrieval status value	document retrieval	preference score
automatic indexing	document retrieval	automatic ranking

**Table 4.3.7 Sample: Dictionary of contextual thesaurus**

The object databases are of two types, primary and auxiliary. The auxiliary databases include those source databases that are modified and organized as reusable

resources for reorganization of the indexing subsystem, such as the list of suffixes, and the dictionary of synonyms. The primary databases are those newly created ones that are to be maintained by the indexing subsystem for the routine use of information retrieval, including:

(1) The inverted file of stem-based index terms, with each index term linked to its topically related documents with various weights assigned.

(2) The document description file consisting of document description records, with each document description record being a representative or surrogate of a corresponding document. The document description record contains a number of document descriptors, which are selected using both linguistic and statistical methods, with each having a numerical weight attached to indicate the significance of its role in the document surrogate. The document description records are partitioned into a number of classes corresponding to the document classes, which partition the entire document collection by means of subject cataloging. The document description records of the same class are also linked in terms of citations.

(3) The user classification file, which is created on the basis of user properties specified by the system designers. Each record contains a set of inferential rules consisting of a condition and a decision. The condition is a set of specified values reflecting a user's features, while the decision is a set of specified ranges of the specified values that are expected to affect a user's preference.

The above object databases characterize our indexing model. The design differs from traditional ones in the use of document description file to support a delicate matching (either exact or partial) mechanism between a user's query and documents

and in the use of user classification file for the purpose of ranking documents in response set.

The separation of document descriptors from index terms provides a number of advantages. Stem-based indexing is known to give a significant reduction of the indexing vocabulary and quick response for locating topically related documents. However, stem-based coordination matching tends to suffer from some semantic problems. For example, the terms 'VENETIAN BLINDS' and 'BLIND VENETIANS' would lead to the same retrieval result under stemming. Although a more sophisticated model might include locational factors to solve those problems, the use of terms 'WATER PLANT' and 'WATERING PLANT' would still cause some trouble in retrieval. Separation of the document descriptors and index terms not only avoids many semantic problems, but also provides flexibility for system designers to furnish a more sophisticated matching mechanism in order to improve the effectiveness of retrieval. One way to achieve such improvement is to organize the document description file as a knowledge representation base after an extensive syntactic and semantic analysis of the documents with the help of dictionaries created manually. Here, we avoid building knowledge-based representatives for documents, while providing extensive features for matching a user's query to the document description records in terms of phrase-based descriptors.

Under this design, index terms are used to locate topically related documents as quickly as possible. Logical relationships among index terms are not of concern. However, numerical weights attached to index terms provide information that can be used in an analysis of the constitution of phrase-based document descriptors and query

descriptors. For example, given a query descriptor INFORMATION RETRIEVAL SYSTEM, we look up the weights assigned to each of three individual terms and decide that the query descriptor INFORMATION RETRIEVAL SYSTEM legitimately matches the document descriptor INFORMATION RETRIEVAL, for the term SYSTEM carries a small weight so that its appearance could be ignored.

#### 4.3.2 Inverted file of index terms

An inverted file of stem-based index terms has been one of the most popular tools in information retrieval. The algorithm to create the inverted file with weight assignments is outlined below.

##### **Algorithm 4.3.2.1** Creating inverted file of index terms

(1) Scan each document in the given collection to obtain a list of words with complete frequency statistics gathered. These statistics include total frequency of word occurrences with respect to the entire document collection and each document class, document frequency with respect to the entire document collection and each document class, and frequency of word occurrences and postings within each document.

(2) Remove non-informative words from the above list by means of the dictionary of non-informative words, resulting in a reduced set of potential index terms.

(3) Use a stemming technique to obtain a reduced list of stem-based index terms with revised frequency statistics by means of the dictionary of word suffixes.

(4) Calculate the index term weights  $W(t_k, C_j)$ ,  $j=1,2,\dots,l$ , and  $W(t_k, d_{ij})$  by means of the Composite Weighting Function, where  $W(t_k, C_j)$  denotes the weight of index term  $t_k$  with respect to document class  $C_j$ , and  $W(t_k, d_{ij})$  denotes the weight of index term  $t_k$  with respect to document  $d_{ij}$  in class  $C_j$ . Here, the document classes refer to the partition of the entire document set induced by a subject-related relation according to the dictionary of the subject catalog.

(5) Organize the weighted index terms into an inverted file in which each index term record will contain all  $\langle C_j, W(t_k, C_j) \rangle$ ,  $j=1,2,\dots,l$ , and  $\langle d_{ij}, W(t_k, d_{ij}) \rangle$ ,  $i=1,2,\dots,n_j$ , where  $n_j$  is the number of documents in document class  $C_j$ .

Two problems in Algorithm 4.3.2.1 need to be mentioned. In step (3), some rules must be set up for revising the frequency statistics and solving semantic ambiguities among certain words. For example, the simple addition of frequency counts of all words with the same stem may tremendously change the degree of significance of that stem-based index term as we apply the Composite Weighting Function to calculate index term weights. In step (4), it is critical to make appropriate policies in order to apply the Composite Weighting Function. Such policies will involve a series of decisions on what frequency statistics are to be chosen, i.e., to choose the collection-oriented or the class-oriented statistics, and how to specify the coefficients. Some possible heuristic strategies have been discussed in Chapter 2.

#### 4.3.3 Document description file

The document description file plays an essential role in the matching module.

Each document description record consists of a limited number of document descriptors, which are conceptual phrases of limited length extracted from the document. Extraction of document descriptors may be done by means of syntactic and/or semantic methods, as briefly mentioned in Chapter 1. However, we shall propose a text scanner to extract conceptual phrases on an individual document basis.

We first define text delimiters as those non-informative words in the dictionary along with a number of punctuation marks, such as the comma, period, colon, semicolon, and question mark. Text delimiters can then be organized in a delimiter dictionary such that each delimiter is linked to a subroutine which provides corresponding interpretations and actions. Since the number of delimiters are quite limited, we assume that the delimiter dictionary can be prepared manually to fulfill the requirement of simple text analysis performed by a text scanner. Further, we suggest that some linguistic methods proposed in the literature [7,71,72] can be used to build the delimiter dictionary as well as the rules for suffix analysis. For example, an article indicates the beginning of a conceptual phrase; the preposition 'of' may help detect the end of a conceptual phrase; the word ending 's' may be used to recognize a plural noun or third person singular present tense verb; the word ending 'ed' may be used to help detect a verb in the past tense or a past participle; the word ending 'ing' may help eliminate an extra verb in the progressive form from a phrase. In addition to the delimiter dictionary, the text scanner makes use of an input buffer to store the word read in, a phrase register to store the words in a phrase, and two delimiter registers to save the precedent and succedent delimiters of a phrase. The sequence of adjacent words between any two delimiters is examined. This includes eliminating the ending word

that is a present participle or a past participle to obtain a raw phrase of one of five types: a noun phrase  $N^*$  consisting of a single noun or a composition of nouns, an adjective phrase  $A^*$  consisting of a single adjective or two adjectives connected by the word 'and', an adverbial phrase  $AA^*$  consisting of a composition of an adverb and an adjective phrase, an attributive phrase  $A^*N^*$  consisting of a composition of an adjective phrase and a noun phrase or of an adverbial phrase and a noun phrase, and a prepositional phrase  $NPN$  consisting of two nouns connected by the preposition 'of'. Let  $N$  denote the set of nouns,  $A$  the set of adjectives,  $A_d$  the set of adverbs, and  $P$  the set of prepositions. The above types of phrases are summarized formally in Table 4.3.8.

Table 4.3.8 Definition of phrase types

- (1)  $\langle \text{RAW PHRASE} \rangle ::= \langle N^* \rangle \mid \langle A^* \rangle \mid \langle AA^* \rangle \mid \langle NPN \rangle \mid \langle A^*N^* \rangle$
- (2)  $\langle N^* \rangle ::= \langle N \rangle$
- (3)  $\langle N^* \rangle ::= \langle N \rangle \langle N^* \rangle$
- (4)  $\langle A^* \rangle ::= \langle A \rangle$
- (5)  $\langle A^* \rangle ::= \langle A \rangle \text{ and } \langle A \rangle$
- (6)  $\langle AA^* \rangle ::= \langle A_d \rangle \langle A \rangle$
- (7)  $\langle AA^* \rangle ::= \langle A_d \rangle \langle A \rangle \text{ and } \langle A \rangle$
- (8)  $\langle NPN \rangle ::= \langle N \rangle \text{ of } \langle N \rangle$
- (9)  $\langle A^*N^* \rangle ::= \langle A \rangle \langle N^* \rangle$
- (10)  $\langle A^*N^* \rangle ::= \langle A \rangle \text{ and } \langle A \rangle \langle N^* \rangle$
- (11)  $\langle A^*N^* \rangle ::= \langle A_d \rangle \langle A \rangle \langle N^* \rangle$
- (12)  $\langle A^*N^* \rangle ::= \langle A_d \rangle \langle A \rangle \text{ and } \langle A \rangle \langle N^* \rangle$

Note that the prepositional phrase is identified only when there are no extra words between two nouns other than 'of'. For example, 'PART OF SPEECH' is identified as a prepositional phrase, but 'PART OF HIS SPEECH' is treated as two noun phrases: 'PART' and 'SPEECH'. As the result of the scanning process, the number of raw phrases for a document must exceed a specified value to avoid shallow indexing; otherwise, we shall scan an additional part of the document besides the title



and abstract, even an extensive part of the document such as the bibliography. A general description of extracting conceptual phrases from a document is given in Algorithm 4.3.3.1.

**Algorithm 4.3.3.1** Extracting conceptual phrases

- (1) Set the initial status of phrase register and delimiter register to empty;
- (2) Scan each sentence (the title is treated as a single sentence) of the given document word by word and push each word into the phrase register until a delimiter is encountered;
- (3) Determine the validity of the phrase in the phrase register by means of the delimiter registers, the delimiter dictionary and rules of simple semantic analysis based on suffix hints;
- (4) Add the conceptual phrase validated in step (3) to the phrase list of the given document;
- (5) Save the current delimiter in the precedent delimiter register, clear the succedent delimiter register and phrase register, and go to step (2) until all sentences have been processed.

The conceptual phrases we obtain are called raw phrases and need to be refined. The refinement of the raw phrases includes two phases: a decomposition phase and a selection phase. The purpose of the decomposition is to remove the lengthy phrases from the set of document descriptors in such a way that these removed phrases are still in effect by means of a partial matching mechanism, which provides an important

feature for our composite retrieval model to tolerate the inconsistencies between query descriptors and document descriptors. The selection phase is a natural step following the decomposition to remove unworthy phrases from the set of document descriptors. In our indexing model, we shall assume that the maximum length of a noun phrase is limited to three (i.e., at most three adjacent nouns), for the sake of simplicity; however, it would be a trivial task to extend our model to deal with longer noun phrases. Using the numerical labels in Table 4.3.8 to denote the corresponding phrase types, we set up two groups of rules for the decomposition and the selection, respectively.

#### **Decomposition rules:**

- (1) All noun phrases of length two are saved;
- (2) For each noun phrase of length three, which is in the form  $n_1n_2n_3$ , decompose it into two phrases  $n_1n_2$  and  $n_2n_3$ , and the phrases  $n_1n_2$  and  $n_2n_3$  as well as the original phrase are saved;
- (3) All the noun phrases of length one are saved except those that are partial repetitions of a saved noun phrase;
- (4) Adjective phrases of types (4) and (5) are dropped;
- (5) Adverbial phrases of type (6) are saved;
- (6) For each adverbial phrase of type (7) in the form  $a_d a_1$  and  $a_2$ , decompose it into phrases  $a_d a_1$  and  $a_d a_2$ , which are saved, while the original phrase is dropped;
- (7) prepositional phrases of type (8) are saved;
- (8) For each attributive phrase of type (9), if it is in the form  $an$ , then it is saved; if it is in the form  $an_1n_2$ , then it is saved as well as decomposed phrases  $an_1$  and  $n_1n_2$ ; if

it is in the form  $an_1n_2n_3$ , then it is decomposed into  $an_1n_2$ ,  $n_1n_2n_3$ ,  $n_1n_2$ ,  $n_2n_3$ , which are saved, while the original phrase is dropped;

(9) For each attributive phrase of type (10) in the form  $a_1 \text{ and } a_2 N^*$ , it is decomposed into phrases  $a_1 N^*$  and  $a_2 N^*$  and further processed according to rule (8);

(10) For each attributive phrase of type (11), if it is in the form  $a_d an$ , then it is decomposed into  $a_d a$  and  $an$ , and saved; if it is in the form  $a_d an_1n_2$  then phrases  $a_d a$ ,  $an_1$ ,  $an_1n_2$ ,  $n_1n_2$  are saved while the original one is dropped; if it is in the form  $a_d an_1n_2n_3$ , then phrases  $a_d a$ ,  $an_1n_2$ ,  $n_1n_2n_3$ ,  $n_1n_2$ ,  $n_2n_3$  are saved, while the original one is dropped;

(11) For each attributive phrase of type (12) in the form  $a_d a_1 \text{ and } a_2 N^*$ , it is decomposed into phrases  $a_d a_1 N^*$  and  $a_d a_2 N^*$ , and further processed according to rule (10).

The decompositions enforced by decomposition rules (2),(8) and (10) are denoted as hierarchical decompositions. The decomposition phase results in phrases with a maximum length of three. The types of these phrases are:

- (1)  $N$
- (2)  $NN$
- (3)  $A_d A$
- (4)  $AN$
- (5)  $NNN$
- (6)  $ANN$

where any  $N$  type phrase cannot be a partial repetition of any  $NN$  type phrase.

The selection phase proceeds by obtaining a non-redundant list of phrases along with their within-document frequencies of occurrences. Each phrase in the list has a

maximum length of three. As with stem-based index terms, the frequency of occurrences and the document frequency for each phrase with respect to each document class and the entire collection can be calculated. We shall assume that the justification of the significance of words being based on frequency statistics remains valid in the case of conceptual phrases. Then, it is possible to apply the Composite Weighting Function to calculate the numerical weights for the phrases in each document, provided that appropriate constants in the formula are specified. For the selection of the final set of phrase-based descriptors for each document, we shall define three types of roles that a word may play in a phrase, and a criterion called the binding strength to characterize the structure of a phrase acquired from the decomposition phase.

**Definition 4.3.3.1:**

Given a phrase  $t = t_1 t_2 \dots t_l$  containing the word  $t_k$ ,

(1) word  $t_k$  is critical to the phrase  $t$  iff

$$(\forall t_i \in t)(i \neq k \rightarrow w(t_k) - w(t_i) \geq \delta),$$

where  $w(t_i)$  is the weight assigned to word (index term)  $t_i$ , and  $\delta$  is a non-negative constant.

(2) word  $t_k$  is said to be minor to the phrase  $t$  iff  $t_k$  is not critical and there exists a critical word other than  $t_k$  in the phrase  $t$ .

(3) word  $t_k$  is said to be general to the phrase  $t$  iff  $t_k$  is not critical and there does not exist a critical word in the phrase  $t$ .

According to Definition 4.3.3.1, a phrase may consist of all general words, or the combination of a critical word and minor word(s). These concepts are built up relative to the value of constant  $\delta$ . We regard  $\delta$  as a critical value. The higher the critical value specified, the less phrases become a combinative structure. The classification can be achieved by means of stem-based index term weights.

**Definition 4.3.3.2:**

Given a phrase  $t=t_1t_2...t_l$ , the binding of  $t$  is said to be strong iff

$$w(t) > \sum w(t'),$$

where  $t'$  denotes a child phrase of  $t$  from a hierarchical decomposition and  $w(.)$  denotes the phrase weight.

According to Definition 4.3.3.2, the concept of strong binding is based on phrase weights. An interpretation of strong binding is that there has been an extra gain in information after two or three phrases were bound together.

For the final selection of document descriptors, we specify the following rules on the basis of each individual document.

**Selection rules:**

- (1) If the phrase is of  $N$  type,  $NN$  type, or  $A_dA$  type, it is selected;
- (2) For each phrase of  $AN$  type, it is selected if it was obtained not only from the decomposition of a phrase in the form  $an_1n_2$ ; otherwise, it is selected only when word  $n_1$  is critical to the phrases  $an_1n_2$ ;

(3) For each phrase of type  $NNN$ , it is selected if its binding is strong;

(4) For each phrase of type  $ANN$ , it is selected if it is strongly bound, or the  $AN$  phrase from the decomposition of  $ANN$  is not selected.

Selection rule (2) checks the validity of the decomposition of phrase  $an_1n_2$  into  $an_1$  and  $n_1n_2$ , based on the concept of a critical word. Selection rules (3) and (4) provide criteria to help decide if phrases of length three are kept or not. This strategy guarantees that the information conveyed by a phrase will not be reduced due to decomposition, which favors lengthy phrases. The phrases selected are denoted as document descriptors, and all the document descriptors, along with their numerical weights, of a document are organized as a document description record in which each descriptor may stand on its own right, or as related to the other descriptors in terms of the structurally hierarchical relationship, imposed by the hierarchical decomposition rules.

The remaining task is to organize the document description records into a linked network. The link or relationship is built up in terms of citations. Unlike previous work, our citation network does not account for strength of association between documents but provide information so that some documents can be traced from others by means of a citation link.

We organize the document description records on levels based on the date of publication. Documents at a higher level may be cited by the ones at a lower level, but not the reverse. No citation links exist between documents at the same level. We call such a citation relationship a navigational one, which will be used in the ranking

model. We shall summarize the processes of creating the document description file in Algorithm 4.3.3.2.

**Algorithm 4.3.3.2** Creating the document description file

(1) For each document in the set, obtain a list of raw phrases by using Algorithm 4.3.3.1;

(2) Refine the raw phrases into a list of document descriptors with weights for each document by means of the decomposition rules, selection rules and the Composite Weighting Function;

(3) For each document in the set, create a document description record in the form of

$$(<t_1, w_d(t_1)>, <t_2, w_d(t_2)>, \dots, <t_k, w_d(t_k)>)$$

where  $t_i, i=1, \dots, k$ , is a phrase-based descriptor of document  $d$  and  $w_d(t_i)$  represents the \*weight of descriptor  $t_i$  with respect to document  $d$ ;

(4) Organize the document description records into a file in which the records are linked hierarchically by means of citation relationships.

**4.3.4 User classification file**

Let  $A_1, A_2, \dots, A_l$  be a set of properties of users which characterize the main factors impacting upon a user's information seeking behavior.

**Definition 4.3.4.1:**

The user classification function is defined as a mapping

$$\rho: U \rightarrow 2^R$$

where  $U$  is the Cartesian product  $A_1 \times A_2 \times \cdots \times A_l$ , and  $R$  is a set of rules specifying the logical implication between a user's features and their impact on the preferences about the requested information.

Each  $l$ -tuple  $\langle a_{i_1}, a_{i_2}, \dots, a_{i_l} \rangle \in U$  is designated as a user profile, where  $a_{i_j}$  is a specified value of the property  $A_{i_j}$ . Each user profile represents a group of corresponding users such that their preferences about the requested information are indistinguishable in terms of the specified properties. The user classification file consists of a set of user classification records, each containing a user profile and the corresponding value of the user classification function. Obviously, if property  $A_{i_j}$  has  $m_{i_j}$  values, the complete user classification file contains  $\prod_{j=1}^l m_{i_j}$   $l$ -tuples or profiles, and in turn  $\prod_{j=1}^l m_{i_j}$  classification records.

As an example, attributes  $A_{i_j}, i=1,2,\dots,l$ , can be specified as the users' educational background, pre-knowledge about the subject in question, and the objectives of the information seeking activities. Then, the educational background may be classified as high, medium and low; the pre-knowledge may be classified as very much, much, medium, little, and very little; and the objectives may be classified as general research, dissertation/thesis research, survey, and general learning.

Before continuing our discussion of the specification of rules, we shall briefly review some work on identifying important authors, important articles, and important



journals in the literature. Virgo reports on identification of important articles by using both citation frequency and expert judges. She suggests that citation frequency and ranking using judges produce sets of important articles that are virtually identical [56]. Hurt examines the problem of identification of important authors in the area of quantum mechanics by using both a bibliometric approach and a historical approach. A gamma test of association results in a significant association between the ranks of authors [57]. Wiberly investigates journal ranking through citation studies [69]. We suggest that their methodologies be applied to rank the authors by a numerical measure, called the author's rank, rank the documents by a citation measure, called the citation rank, rank the sources by a numerical measure, called the source rank (if all the sources are journals, then it becomes the journal rank). Thus, we can incorporate into the document profile the measures of an author's rank, citation rank and source rank, in addition to a measure of the time factor, which is simply the date of publication.

The first type of rules specified by the user classification function is represented by a pair of numbers. The first indicates a policy to be applied to a specified factor, called a sensitivity measure, which is consistent with the measurement used for ranking the corresponding factor. The second indicates the relative weight assigned to the factor. Four rules of this type are described as follows:

(1)  $\langle v_a, a \rangle$  where  $v_a$  indicates a sensitivity measure of an author's rank with respect to a user's preference, and  $a$  is a numerical weight assigned to the factor of author's rank;

(2)  $\langle v_c, c \rangle$  where  $v_c$  indicates a sensitivity measure of the citation rank with respect to the user's preference, and  $c$  is a numerical weight assigned to the factor of citation rank;

(3)  $\langle v_s, s \rangle$  where  $v_s$  indicates a sensitivity measure of source's rank with respect to the user's preference, and  $s$  is a numerical weight assigned to the factor of source's rank;

(4)  $\langle v_t, t \rangle$  where  $v_t$  indicates a sensitivity measure of the time factor with respect to a user's preference, and  $t$  is a numerical weight assigned to the factor of time.

Taking  $\langle v_t, t \rangle$  as an example, if a college student looks for some papers for the purpose of general learning, the system may specify  $v_t=5$ , so that documents published within last five years are not distinguishable in terms of this criterion. The system may assign  $v_t=2$  for a doctoral student doing searching as part of his dissertation research, which implies that recency is more critical to him.

The second type of rules specified by the user classification function is represented in the form of

$$\langle v_1, v_2, \dots, v_n \rangle$$

where  $n$  is the number of the specified document features such as the type, level and style, and  $v_i, i=1, 2, \dots, n$ , designates the value of the  $i$ th feature that is preferred by that particular type of user. When a  $v_i$  is missing, we mean that the  $i$ th feature is not applicable in that case.

The author's rank, the citation rank and the source rank are taken as factors determining the quality of a document, and accordingly, the weights  $a$ ,  $c$  and  $s$  may be

specified as a comparable group during the time of indexing and subject to change later. The factor of quality, together with the time factor, the fitness factor and the reachability factor will form another comparable group in our ranking model, and their numerical weights,  $q, t, f, r$ , respectively, will be assigned accordingly. Since the measure of the fitness and the measure of the reachability will be determined at the time of on-line information request, the weights  $q, t, f, r$  are kept pending until the ranking process is in progress. These coefficients, as well as the relationship between the various factors, will be further defined in the ranking model.

The user classification function is viewed as a rule-based inferential mechanism. We need an expert's opinion to classify future users into different categories and to assign a measure for each class of users with regard to each of the four factors. In addition to the problem of initialization, the user classification file must be organized in such a way that experience drawn from retrieval activities can be used for dynamic modification.

In summary, the main tasks for the indexing subsystem include selection of stem-based index terms, weight assignment to index terms by means of the Composite Weighting Function, extraction of phrase-based descriptors, weight assignment to document descriptors by means of the Composite Weighting Function, and creation of the user classification file. It is obvious that some tasks may be combined and processed simultaneously. Among all object databases, the document description file is of the most important, and is to be used frequently for the matching process in order to produce a response set of documents ranked by topical relevance score with respect to

a user's information request. The user classification file provides an expert's knowledge in order to rank topically relevant documents by means of a preferable relevance score with respect to a user's information needs, which will be further discussed in the ranking model.

#### 4.4 Query processing model

The query processing component of the composite retrieval system is designed to accomplish two basic tasks: validate the syntax of a user's information query and prepare an unordered response set of documents which will be further processed by the matching component. In fact, validation of syntax is not a difficult task, so we shall describe the portion of query processing module that is concerned with the second task.

Formally, our query processing model is defined by a tuple

$$\rho = \langle q, D, D_0, D_1, D_2, g \rangle.$$

Here,  $q$  refers to an information query,  $D$  refers to a number of source databases,  $D_0$  refers to a particular document set of the partitioned document collection,  $D_1$  refers to the reduced set of  $D_0$  after processing a TYPE1 expression in query  $q$ ,  $D_2$  refers to the reduced set of  $D_1$  after processing a TYPE2 expression(POFLE), and  $g = \{g_1, g_2, g_3, g_4\}$  refers to a set of algorithms that accomplish the transition from  $D$  to  $D_2$  with respect to the query  $q$ . The transition procedure is illustrated in Fig 3.6.1.

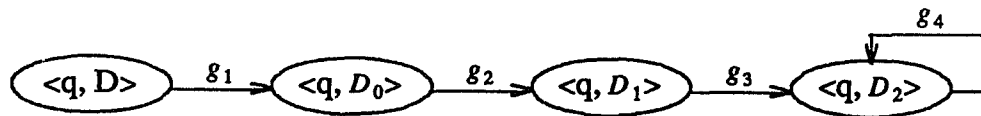


Figure 4.4.1 Transition diagram of query processing model

For a given query, the system automatically looks up the subject catalog dictionary from  $D$ . Algorithm  $g_1$  determines an appropriate subject with respect to the query and selects a specific set of documents related to the subject, denoted as  $D_0$ . The sys-

tem may then display its choice along with the subject catalog for the user to verify. Moreover, an implementation of the algorithm  $g_1$  could be simply posting the subject catalog and letting the user make a choice. Note that the subject-related relationship is defined as an equivalence relation by the system designers, and the user's feedback indicating that the documents in different classes are overlapped may lead to modification of the the subject catalog.

Algorithm  $g_2$  performs data retrieval. The TYPE1 expression is analyzed and manipulated, resulting in a reduced set of  $D_0$ , denoted as  $D_1$ . If no TYPE1 expression is present in query  $q$ , then  $D_1$  is the same as  $D_0$ .

Algorithm  $g_3$  initiates retrieval expressed in TYPE2 expression(POFLE). It functions as a screen test that a document of  $D_1$  that fails to pass it would be removed from  $D_1$ , resulting in a reduced set denoted as  $D_2$ .

Algorithm  $g_3$  does not perform actual matching between query  $q$  and documents in  $D_1$ , but rather plays a role as a screen to prevent those unqualified candidates from being further processed by the matching model. Algorithm  $g_3$  takes each facet in a clause as an information 'chunk', to use Miller's term [70]. An element in a facet is then considered as an information cell, a smaller information unit than a chunk. Corresponding to a simple element, selective element, joint element and conditional element, we have a simple cell, selective cell, joint cell, and conditional cell, respectively. Algorithm  $g_3$  works based on the following assumptions:

- (1) If a document contains any chunk of the query, it passes the screen test;
- (2) If a document contains a cell of any type, it contains the chunk that contains

that cell;

(3) A selective cell is said to be contained by a document if and only if there exists at least one simple cell that is contained in the document;

(4) A joint cell is said to be contained by a document if and only if all of its simple cells are contained in the document;

(5) A conditional cell is said to be contained in a document if and only if its conclusive part is contained in the document;

(6) A simple cell is said to be contained in a document if and only if the term set that indexed the document contains the whole cell or its nucleus. Here, we regard a simple cell as a phrase, and identify a specific part of it as its nucleus. When a whole cell is contained, we say it is a complete implication, and otherwise, a partial implication.

The above assumptions provide a basis for a screen test. Neither complete implication nor partial implication would guarantee logical implication in terms of semantics, but rather suggests that one item contain some information, delivered by the word used, of the other item. In order to detect partial implication, we have to develop some rules on which the cell nucleus can be identified. In the following heuristic rules, we shall classify a query descriptors in terms of its length, i.e., the number of words it contains, and identify the cell nucleus corresponding to the indexing model.

(1) For a query descriptor of length two, the critical word by Definition 4.3.3.1 is taken as its nucleus;

(2) If a query descriptor is of length more than two, either of its two adjacent

words can be taken as the nucleus.

System designers may develop other selection rules to identify the nucleus of a query descriptor based on some other knowledge. In general, the more restrictive the rules that are specified, the tighter the screen will be. The main idea here is to find an initial response set without hurting the level of recall. The screen should not be too tight, since user's query is only an approximate expression of his information needs (his needs may not be clear), a document surrogate is only an approximate expression of the document, and any similarity measure between them is an approximate one in the sense that the assumptions for any rigorous mathematical model may not hold. Thus, our query processing model only functions as a screen to exclude those seemingly non-related documents from  $D_1$  in order to have a reduced initial set, denoted as  $D_2$ , to be processed by the matching model.

Algorithm  $g_4$  checks the set  $D_2$ .  $D_2$  has to be enlarged if it is too small. This may be caused by one of several reasons: (1) there are too many query descriptors in a joint element, (2) the query descriptors used are rare words or phrases, or (3) there are too few query descriptors in the query. An excellent work on query modification can be found in [4]. In our case, since a joint element is treated like traditional ANDed terms, it will lower the chance of a document passing the screen test. Algorithm  $g_4$  will detect the joint element that eliminates the most documents and modify it by breaking it down into two joint elements. If a query descriptor is a rare phrase, there will be two opposite effects. When a rare descriptor appears in a joint element, it will screen out an excessive number of documents. When a rare descriptor appears in a



selective element, it is of very little use. In the first case, it can be separated from the rest to form an additional joint element. In the second case, it can be enhanced by adding synonymous terms or by replacing some of its words by their linguistic synonyms. If there are too few query descriptors, the processing performance will inevitably degrade. Once algorithm  $g_4$  judges the query to have too few descriptors, it will modify the query by adding synonymous terminologies in the selective element and adding contextual thesaurus as a new facet to enlarge the query. Algorithm  $g_4$  works to obtain a new set  $D_2$  of proper size which is specified by the user or determined by the system at default. Then, the modified query will be posted to draw feedback from the user. The process of modification will repeat until it is acknowledged by the user as being satisfactory. The final reduced set denoted as  $D_2$  is thus confirmed.

The strategy that lets a user be the authority clears up the situation; otherwise, the descriptors added by system have to be justified in terms of statistical criteria. This strategy is closer to the work performed by a human mediator. On the other hand, the query processing module is designed to learn from the interactive procedure. In addition to the modification of the subject catalog dictionary, the query processing module extracts the contextual thesaurus from the joint elements and conditional elements. One way suggested here is to add a statistical measure, such as Pearson's correlation coefficient, to guarantee that the contextual thesauri are those with high frequency of co-occurrence in the users' query expressions. We shall leave this problem to be solved by the system designers at the time of system implementation.

In summary, the query processing model is characterized by a partial implication mechanism and an interactive modification facility in order to provide an initial response set for the subsequent stages. The fact that the query processing module does not perform an actual matching between the documents and the user's query implies that it can be done simply by means of the inverted file of the stem-based index terms. Thus, the time complexity for locating the initial response set will be the same as required by a regular inverted file system. Furthermore, since the number of documents in the response set has been significantly reduced, the extra time needed for the subsequent processing is expected to be acceptable for on-line information retrieval.

## 4.5 Matching model

The matching module is to further process the user's query to produce a response set ordered in terms of the topical relevance score. For notational simplicity, we shall refer to the POFLE part of the retrieval statement as the user's information query. A set of information queries expressed in POFLE is denoted by  $Q$ , and an initial set of documents provided by the query processing subsystem is denoted by  $D$ . For each  $d \in D$ ,  $d$  has the form

$$(<t_1, w_d(t_1)>, <t_2, w_d(t_2)>, \dots, <t_k, w_d(t_k)>)$$

where  $t_i, i=1,2,\dots,k$ , is a descriptor of document  $d$  and  $w_d(t_i)$  represents the weight of descriptor  $t_i$  with respect to document  $d$ , as described in the indexing model. We shall use  $w(t_i)$  as an equivalent form of  $w_d(t_i)$ .

Formally, the matching model is defined by a triple

$$M = \langle Q, D, u \rangle,$$

where  $u$  is a matching function defined by Definition 4.5.5 below. The matching model provides two features, a partial matching facility and a generalized evaluation mechanism. We shall first define a binary relation on a phrase-based descriptor set, show the properties of the indexing model, and then describe the matching function  $u$  which characterizes the matching model.

### Definition 4.5.1:

Given a phrase-based descriptor set  $T$ , a binary relation  $R$  on  $T$  is defined as

$$(\forall t, t' \in T) ((t, t') \in R \leftrightarrow t = Mt'N)$$

where  $M$  is a single word or a phrase that modifies  $t'$ ,  $N$  is a single noun or noun

phrase that is modified by  $t'$ , and  $M$  and  $N$  are not simultaneously empty.

Example:

$t = \text{ON-LINE INFORMATION RETRIEVAL SYSTEM}$

$t' = \text{INFORMATION RETRIEVAL SYSTEM}$

$t'' = \text{INFORMATION RETRIEVAL}$

We claim  $t R t'$ ,  $t' R t''$ , and  $t R t''$ , according to Definition 4.5.1.

When  $t$  and  $t'$  are in relation  $R$ , i.e.,  $t R t'$ , we say that  $t$  is structurally more restricted than  $t'$ . A binary relation is called an ordering relation if and only if it is irreflexive, antisymmetric and transitive. We shall show that relation  $R$  is an ordering relation.

#### Theorem 4.5.1:

Relation  $R$  on descriptor set  $T$  is an ordering relation.

Proof:

irreflexiveness:  $(\forall t \in T) ((t, t) \notin R)$

antisymmetry:  $(\forall t, t' \in T) ((t, t') \in R \rightarrow (t', t) \notin R)$

transitivity: For any  $t, t', t'' \in T$ , if  $t R t'$  and  $t' R t''$ , then  $t = M t' N$ ,  $t' = M' t'' N'$ . Thus,  $t = M M' t'' N' N = M'' t'' N''$ , where  $M'' = M M'$  modifies  $t''$  and  $N'' = N' N$  is modified by  $t''$ .

That is,  $(\forall t, t', t'' \in T) ((t, t') \in R \wedge (t', t'') \in R \rightarrow (t, t'') \in R) \quad \square$

We shall show that the indexing model presented in 4.3 possesses some important properties in terms of relation  $R$ .

**Theorem 4.5.2:**

Hierarchical decomposition in the indexing model satisfies relation  $R$ .

Proof:

The hierarchical decomposition in the indexing model is governed by three rules.

For decomposition rule(2), we have  $n_1n_2n_3Rn_1n_2$  and  $n_1n_2n_3Rn_2n_3$ ;

For decomposition rule(8), we have  $an_1n_2Ran_1$ ,  $an_1n_2Rn_1n_2$ ,  $an_1n_2n_3Ran_1n_2$ ,  $an_1n_2n_3Rn_1n_2n_3$ ,  $n_1n_2n_3Rn_1n_2$ , and  $n_1n_2n_3Rn_2n_3$ ;

For decomposition rule(10), we have  $a_danRa_da$ ,  $a_danRan$ ,  $a_dan_1n_2Ra_da$ ,  $a_dan_1n_2Ran_1n_2$ ,  $an_1n_2Ran_1$ ,  $an_1n_2Rn_1n_2$ ,  $a_dn_1n_2n_3Ra_da$ ,  $a_dn_1n_2n_3Ran_1n_2$ ,  $a_dn_1n_2n_3Rn_1n_2n_3$ ,  $n_1n_2n_3Rn_1n_2$  and  $n_1n_2n_3Rn_2n_3$ .  $\square$

Let  $t$  and  $t'$  be two document descriptors produced by the indexing model described in section 4.3, and let  $S(t)$  and  $S(t')$  be the two sets of documents described by  $t$  and  $t'$ , respectively. We now show the property of inclusiveness by the following, which gives an interpretation of relation  $R$ .

**Theorem 4.5.3:**

$$t R t' \rightarrow S(t) \subseteq S(t')$$

Proof:

For each document  $d \in S(t)$ , we have  $d \in S(t')$  by the selection rules developed in the indexing model.  $\square$

Let  $t$  and  $t'$  be two document descriptors, with weights  $w(t)$  and  $w(t')$ , respectively, of an individual document as a result of indexing. We shall show the property of weights with the following, which provides a principle of partial weight assignment.

**Theorem 4.5.4:**

$$t R t' \rightarrow w(t) > w(t')$$

Proof:

According to the hierarchical decomposition rules developed in the indexing model, if  $t R t'$  then there exists a descriptor  $t''$  such that  $t R t''$ .

According to the selection rules developed in the indexing model,  $t$  will be selected if and only if  $w(t) > w(t') + w(t'') > w(t')$ .  $\square$

We now propose a methodology for partial weight assignment with the following definitions.

**Definition 4.5.2:**

Given a query descriptor  $x=x_1x_2\cdots x_j$ , a document descriptor  $y=y_1y_2\cdots y_l$  is said to be exactly matched  $x$  iff

(1)  $l = j$  and

(2)  $x_1=y_1, x_2=y_2, \dots, x_{j-1}=y_{l-1}$ , and  $x_j$  matches  $y_l$  in terms of stem matching.

**Definition 4.5.3:**

Given a query descriptor  $x=x_1x_2\cdots x_j$ , a document descriptor  $y=y_1y_2\cdots y_l$  is said to be partially matched  $x$  iff

- (1)  $l < j$  and
- (2) there exists an integer  $i$  such that  $x_i=y_1, x_{i+1}=y_2, \dots, x_{i+l-2}=y_{l-1}$ , and  $x_{i+l-1}$  matches  $y_l$  in terms of stem matching.

**Definition 4.5.4:**

Given a query descriptor  $x$  and the set  $X$  of document descriptors of a document  $d$  that partially match  $x$ , we define

- (1)  $X'$  as a subset of  $X$  such that for all  $t, t' \in X$ , if  $t R t'$ , then  $t' \notin X'$ ,
- (2) a partial weight assigned to  $x$  with respect to  $d$  as

$$w^p(x) = \text{MAX} (1, \sum_{t \in X'} w(t_i))$$

Now we are able to describe the matching function  $u$  for the model.

**Definition 4.5.5:**

The matching function  $u$  is a mapping

$$u: Q \times D \rightarrow [0,1]$$

and for each  $q \in Q$  and  $d \in D$ ,  $u$  is defined in the following cases:

Case 1:

$q \in Q$  is a simple element in the form of  $(A)_a$  where  $A$  is a simple element and  $a$  is the weight of  $A$ .

(1) if there is a document descriptor  $t$  with weight  $w(t)$  that exactly matches  $A$ ,  
then

$$\begin{aligned} u(q) = u(A) &= a * w(A) && A \text{ is not negated} \\ &= 1 - a * w(A) && \text{otherwise} \end{aligned}$$

(2) else if there is one or more document descriptors that partially matches  $A$ ,  
then

$$\begin{aligned} u(q) = u(A) &= a * w^p(A) && A \text{ is not negated} \\ &= 1 - a * w^p(A) && \text{otherwise} \end{aligned}$$

(3) else

$$u(q) = 0$$

Case 2:

$q \in Q$  is a compound element

(1) if  $q$  is in the form  $(A_1, A_2, \dots, A_l)_a$ , i.e., a selective element, where  $A_i$ ,  $i=1, 2, \dots, l$ , are simple elements, and "," is used as a selective operator, then

$$u(q) = a * \text{MAX}(u(A_i))$$

(2) else  $q$  is in the form  $(A_1 \& A_2 \& \dots \& A_l)_a$ , i.e., a joint element, where  $A_i$ ,  $i=1, 2, \dots, l$ , are simple elements, and "&" is used as a joint operator, then

$$u(q) = a * \text{MIN}_i(u(A_i))$$

(3) else if  $q$  is in the form  $(C_1 | C_2)_a$ , i.e., a conditional element, where  $C_1$  and  $C_2$  are either selective elements or joint elements, and "|" is used as a conditional operator, then

$$u(q) = a * u(C_1) \quad \text{if } u(C_2) \neq 0$$



$$= 0 \quad \text{otherwise}$$

Case 3:

$q \in Q$  is a facet in the form of  $(X_1)_s \text{ AND } (X_2)_s \text{ AND } \cdots \text{ AND } (X_m)_s$ , where  $X_i$ ,  $i=1,2,\dots,m$ , are elements,  $s \in \{a_1, a_2, \dots, a_l\}$  is a set of element weights such that  $l \leq m$ , then

$$u(q) = \frac{\sum_{i=1}^l a_i \min_{s=a_i} (u(X_j))}{\sum_{i=1}^l a_i}$$

Case 4:

$q \in Q$  is a clause in the form of

$$(F_1)_{a_1} \text{ OR } (F_2)_{a_2},$$

where  $(F_1)_{a_1}$  and  $(F_2)_{a_2}$  are two facets, then the first step is to calculate

$$u'(F_i) = a_i * u(F_i), \quad i=1,2,$$

and the second step is to calculate

$$u(q) = u'(F_1) + u'(F_2) - u'(F_1) * u'(F_2)$$

This procedure can be applied to the more general case when multiple facets are connected by OR operators by setting

$$u(q) = u(C) + u'(F) - u(C) * u'(F),$$

where  $C$  itself is a clause.

To describe the properties of this model, we define some concepts as follows.

**Definition 4.5.6:**

A response to query  $q$  from the matching model is the fuzzy subset

$$r(q) = \{ \langle d, u_d(q) \rangle \mid d \in D \}.$$

In  $r(q)$ , documents are weakly ordered in terms of  $u_d(q)$ . We denote  $u_d(q)$  as the topical relevance score of  $d$  with respect to query  $q$ .

**Definition 4.5.7:**

Query  $q_1$  is said to be broader than query  $q_2$  iff

$$r(q_1) \subset r(q_2)$$

where  $\subset$  is defined for fuzzy sets.

**Definition 4.5.8:**

Query  $q_1$  is said to be narrower than query  $q_2$  iff

$$r(q_1) \supset r(q_2)$$

The retrieval model has the following properties:

**Theorem 4.5.5:**

The measure of topical relevance score,  $u_d(q)$ , satisfies the condition

$$0 \leq u_d(q) \leq 1$$

**Proof:**

Let  $q = (F_1)_{f_1} \text{ OR } \cdots \text{ OR } (F_m)_{f_m}$ .

First, consider the case  $m=1$ , so,  $q=(F_1)_{f_1}=((X_1)_s \text{ AND } (X_2)_s \text{ AND } \cdots \text{ AND } (X_m)_s)_{f_1}$ ,  
 $s \in \{a_1, a_2, \dots, a_l\}$ .

$$\text{Since } u(q) = \frac{\sum_{i=1}^l a_i \text{ MIN}(u(X_j))}{\sum_{i=1}^l a_i} \leq \frac{\sum_{i=1}^l a_i}{\sum_{i=1}^l a_i} = 1, \text{ all } a_i \text{ s are non-negative, and } 0 \leq u(X_j) \leq 1,$$

$$0 \leq u_d(q) \leq 1.$$

Assuming that the conclusion holds for  $m=k$ , we consider the case  $m=k+1$ . Since

$$\begin{aligned} u(q) &= u((F_1)_{f_1} \text{ OR } \cdots \text{ OR } (F_k)_{f_k} \text{ OR } (F_{k+1})_{f_{k+1}}) \\ &= v + u((F_{k+1})_{f_{k+1}}) * (1-v) \leq 1, \end{aligned}$$

and both  $v = u((F_1)_{f_1} \text{ OR } \cdots \text{ OR } (F_k)_{f_k})$  and  $u((F_{k+1})_{f_{k+1}})$  are non-negative, we have

$$0 \leq u_d(q) \leq 1 \quad \square$$

#### Theorem 4.5.6:

Given queries  $q$  and  $q'$  with respect to an initial set  $D$ , if query  $q'$  is composed by adding a facet to query  $q$ , then query  $q'$  is broader than query  $q$ .

Proof:

For any  $\langle d, u(d, q') \rangle \in r(q')$ ,

$$\begin{aligned} u(q') &= u(q \text{ OR } (F)_f) \\ &= u(q) + u((F)_f) * (1 - u(q)) \\ &\geq u(q) \quad \square \end{aligned}$$

**Theorem 4.5.7:**

Given queries  $q$  and  $q'$  with respect to an initial set  $D$ , if query  $q'$  is composed by adding an element to an existing facet in query  $q$ , then  $q'$  is narrower than query  $q$ .

Proof:

Proof follows directly from Definition 3.7.5 that the topical relevance score of the facet with a new added element will not increase due to the MIN function used.

□

The matching model described above can be viewed as a generalized Boolean retrieval model with a restricted query expression. This can be seen easily from the fact that for the Boolean query with discrete weights, if it consists of the ANDed query descriptors, then it will be treated as a facet in our model, and the MIN function is in effect, as described in case 3, which will lead to the same result as in the traditional Boolean model; if it consists of the ORed query descriptors, then it will be treated as a clause and the function described in case 4 will lead to the same result as in the traditional Boolean model. For the Boolean query with fuzzy weights, our evaluation function works in such a way that the AND operators in a facet become "loose", i.e., each query descriptor with a non-zero weight may have its contribution to the topical relevance score, which will avoid a highly restrictive retrieval caused by the traditional AND operator; The OR operator in a clause will increase the topical relevance score of a document if it matches both query descriptors instead of one, while the traditional Boolean model will make no difference in this case. In addition, our model supports the conditional element for the retrieval by context, the selective

element for the retrieval by synonyms, and the joint element for the retrieval by co-existence relationship, which are beyond traditional Boolean logic.

The matching model described above is also more general than the term vector space model. In fact, a query expression in the traditional vector space model can be viewed as a facet in the POFLE. Then, our evaluation function for a facet, described in case 3, resembles the similarity measure function in the traditional vector space model when the weights of the query descriptors are different; otherwise, the actual effect of the evaluation is a combined result of the MIN function value and the similarity measure.

## 4.6 Ranking Model

Given a weakly ordered document set produced by the matching model, the ranking model proposed in our composite retrieval model will further decompose the document set according to the preferable relevance score by means of a rule-based reasoning mechanism. The ranking model works on the basis of the following assumptions:

(1) A document being preferably relevant to a user's information needs implies that it is topically relevant to user's information request, but the reverse is not necessarily true. That is to say, only those documents whose topical relevance scores are greater than zero would be taken into consideration in the ranking model.

(2) The preferable relevance score determined by the ranking model overrides the topical relevance score obtained from the matching model for any individual document. That is to say, the final order of presentation is fully determined by the preferable relevance score, which is the result of the logical consequences of combining all preference factors.

(3) In our ranking model, the only preference factors considered are quality, recency, fitness and reachability, in which topical relevance is fully determined by document descriptors or index terms, and quality of a paper is affected by author's rank, citation strength and source reputation.

The knowledge bases used in the ranking model include the document profile attached by a group of measurements indicating the author's rank, citation strength and source rank; the user classification file; and a set of rules specified by the system for reasoning. The ranking procedure is described by the following algorithm:

- Step 1: Using the TYPE1 selection rule, select an initial set of documents from the weakly ordered document set produced by the matching procedure;
- Step 2: Rank the documents in the set by the evaluation function of the ranking model;
- Step 3: Present the ranked documents set and draw feedback from the user;
- Step 4: Go to Step 5 until all documents with non-zero topical relevance score have been ranked or the process is terminated by the user;
- Step 5: Using the TYPE2 selection rule, select the next set of documents from the remaining elements of the weakly ordered document set produced by the matching procedure, and go to step 2.

Assume that  $d_1, d_2, \dots, d_n$  is a sequence of documents produced by the matching procedure such that  $r(d_i) > 0$  and  $r(d_i) \geq r(d_{i+1})$  for  $i=1, 2, \dots, n$ , where  $r(d_i)$  denotes the topical relevance score of document  $d_i$  in the sequence. That is,  $d_1, d_2, \dots, d_n$  is a weakly ordered set induced by the topical relevance score. Then, there exists a sequence of subsets of documents  $S_1, S_2, \dots, S_m$  such that for any  $d_i, d_j$ , if  $d_i \in S_k$  and  $d_j \in S_k$ , then  $r(d_i) = r(d_j)$ , and for any  $d_i, d_j$ , if  $d_i \in S_k$ ,  $d_j \in S_l$  and  $k < l$ , then  $r(d_i) > r(d_j)$ , i.e.,  $r(d_{s_1}) > r(d_{s_2}) > \dots > r(d_{s_m})$ , in which  $r(d_{s_k})$  denotes the topical relevance score of document subset  $S_k$ .

TYPE1 selection rules are used to select a subset of documents,  $S$ , for the purpose of reordering the documents in  $S$  by means of the preferable relevance score. Initially, TYPE1 selection rules work on the initial sequence of subset of documents,  $S_1, S_2, \dots, S_m$ .

**TYPE 1 Selection rules:**

$$(1) \forall d (d \in S_1 \rightarrow d \in S)$$

$$(2) \forall d (d \in S_i \wedge |r(S_1) - r(S_i)| \leq \delta_1 \rightarrow d \in S)$$

$$(3) \forall d (d \in S_i \wedge |r(S_1) - r(S_i)| \leq \delta_2 \wedge d \notin REF(S) \rightarrow d \in S)$$

where  $\delta_1$  is called the correction value of matching accuracy,  $\delta_2$  is called the protection value of referential losses, and  $REF(S)$  denotes the set of all documents cited by the documents of document set  $S$  being selected.

The values  $\delta_1$  and  $\delta_2$  are specified by the system in an expert system mode according to the analysis of the sequence of documents  $S_1, S_2, \dots, S_m$ . For example, the system may specify a non-zero  $\delta_1$  in two cases: (1)  $S_1$  and  $S_i$  contains very few documents, or (2) the topical relevance score of  $S_i$  is very close to the topical relevance score of  $S_1$ , that is,  $|r(S_1) - r(S_i)|$  is very small. In the first case, documents  $S_1$  and  $S_i$  are mingled together before being submitted to the ranking model, since it would not hurt if we mix up a few documents at the top of the presentation, provided that the user would at least review these documents. In the second case, documents in  $S_1$  and  $S_i$  are mingled together before being submitted to the ranking model, since a small gap between  $r(S_1)$  and  $r(S_i)$  may exist due to a lack of matching accuracy and thus should be ignored to protect the documents in  $S_i$  from unjust evaluation. The protection value of the referential losses  $\delta_2$  is based on the philosophy that some documents in  $S_i$  must be treated as special since they can not be traced from the bibliographies of the documents being selected and yet they have a reasonably high score of topical relevance. That is to say, some documents must be protected from referential losses



in case that only those documents at the top rank are to be browsed by users. Here,  $\delta_2 > \delta_1$  since the condition in rule (2) is contained in rule (3).

After initial selection, the original weakly ordered set  $d_1, d_2, \dots, d_n$  is reduced to  $d'_1, d'_2, \dots, d'_l$ ,  $l < n$ , and the original sequence of subset  $S_1, S_2, \dots, S_m$  is accordingly constructed as  $S'_1, S'_2, \dots, S'_h$ ,  $h < m$ , to which selection rules are applied. For the sake of notational simplicity, we relabel the remaining subset of documents after each selection so that the following TYPE2 selection rules can be described using the same notation of the original weakly ordered set and the original subset sequence as well.

#### TYPE 2 Selection rules:

$$(1) \forall d (d \in S_1 \rightarrow d \in S)$$

$$(2) \forall d (d \in S_i \wedge |r(S_1) - r(S_i)| \leq \delta_1 \rightarrow d \in S)$$

$$(3) \forall d (d \in S_i \wedge |r(S_1) - r(S_i)| \leq \delta_3 \wedge AS(\{d\}, R) > AS(S_1, R))$$

where  $\delta_1$  is the correction value of matching accuracy as in TYPE1 selection rules,  $\delta_3$  is called the boundary value of association gain, and AS is the function used to calculate the average association strength between two document sets.

The meanings of rule(1) and rule(2) are interpreted the same as in TYPE1 selection rules. Rule(3) is enforced to combine feedback information from the previous result of the ranking model, in which R is a set of documents either judged preferred by the user or judged as being most topically relevant by the system. The system picks up a document d in  $S_i$  with  $r(S_i)$  being close enough to  $r(S_1)$ , calculates the average association strength between d and document set R, and adds d to document set  $S_1$  when this value exceeds the average association strength between document set  $S_1$  and

R. The function AS could be implemented in three ways.

The first implementation of the function AS is the well-known similarity function under Salton's vector space model [4], where the similarity function is applied to two cases: (1) between the document and the query in order to produce a numerical value representing the degree of similarity; and (2) between documents in order to form document clusters.

Second, AS can be implemented as a co-citation strength, where co-citation strength is defined as the number of documents that are jointly cited by two documents.

Third, AS can be implemented as a bibliographic coupling, where the coupling strength is defined as the number of references in common for both documents.

For the first implementation, there must be a set of linearly independent term vectors in which documents are represented; this is not seen in our model. However, a similarity measure function can be applied to produce an approximate value in an operational environment. For the second and third implementations, our model has provided all necessary information, such as reference list of each document in order to construct a citation network among a collection of documents. Since the calculation of co-citation link or bibliographic coupling can be carried out within a small set of documents, the expected time of calculation can be acceptable.

Now, let us consider the evaluation function to be used in our ranking model. Our problem is, for given a set of documents, to form a logical expression in which the significance of four preference factors are reasonably accounted and to define a ranking mechanism by which the logical expression can be evaluated to produce a

unique value of the preferable relevance score for each document. In order to accomplish that, we have to specify the logical relationship of the four preference factors.

First, we argue that an ideal document with respect to a user's information needs shall possess all four preference factors to some degree. That is, a document to be presented at the top of the list ought to be of high quality, relatively new, well fitted to the user's background, and capable of reaching other potential useful documents via citations. Here, quality is the most important factor. Once it is established that a document is of high quality, recency may immediately come to mind as a result of a user's information seeking behavior. In other words, high quality plus recency will greatly increase the chances that the document is preferred. Then, it is reasonable to check whether the document fits the user's background. If it does, we are almost certain that the document will be pertinent. Finally, if there are a number of qualified documents satisfying the above set of conditions, we might further differentiate among them by adding a venial score, designating the degree of reachability via citations, to complete the ranking process.

Second, we argue that a document can be considered as of relatively high quality if one of the three quality factors is observed. That is, if a document was written by highly ranked author(s), shows strong citation strength, or is published in a source that enjoys a good reputation, there is very little doubt about its quality. Such a strategy may cause concern but we would like to claim that our reasoning is based on a reasonable model of human information seeking behavior. We summarize our discussion by presenting the ranking model below.

**Definition 4.6.1:**

The ranking model is defined as a weighted logical expression

$$(A_a \text{ OR } C_c \text{ OR } S_s)_q \text{ AND } T_t \text{ AND } F_f \text{ AND } R_r,$$

Where  $A, C$  and  $S$  represent the factors of an author's rank, citation strength, and source rank, respectively,  $T, F$  and  $R$  represent the time factor(recency), the fitness factor and the reachability factor, respectively. The lower case letters  $a, c, s, q, t, f$  and  $r$  are the corresponding numerical weights in  $[0,1]$ , indicating the relative importance of each factor when calculating the preference score.

Note that the weights  $a, c, s, q$  are comparable in terms of the quality factor, and the weights  $q, t, f$  and  $r$  form another comparable group with respect to the roles of quality, recency, fitness and reachability. Let  $\psi(\cdot)$  denote a numerical score in  $[0,1]$ , indicating a user's preference with a specific factor. Then  $\psi(A), \psi(C), \psi(S)$  and  $\psi(T)$  can be obtained by directly applying the appropriate user's classification record to the document profiles of the given subset of the documents to be ranked. Taking  $\psi(T)$  as an example with the corresponding rule  $\langle v_t, t \rangle$ , if the most recent date of publication is the year  $x$  and the oldest is the year  $y$ , then there will be  $\frac{x-y}{v_t}$  ranks in terms of recency. We may assign  $\psi(T)=1.0-i\frac{v_t}{x-y}$  to the documents published between the year  $x-v_t, i$  and the year  $x-v_t(i+1)$ ,  $i=0,1,2,\dots,\frac{x-y}{v_t}-1$ . For  $\psi(F)$ , we may simply take the counts of the matches between the preferred features designated by the type2 rules in the user classification record and the specified features of a document in its profile, and normalize the counts into a numerical score in  $[0,1]$ . To calculate  $\psi(R)$ , we first

arrange the documents in the given subset to be ranked into a citation network. A document  $d$  is said to be at the level one if it is not cited by any other documents; a document  $d$  is said to be at the level  $l(d) = \min(l(d_1), l(d_2), \dots, l(d_k)) + 1$ , where  $l(d_i), i=1, 2, \dots, k$  denotes the level number of the document  $d_i$  which cites the document  $d$ . We then convert the level number of a document into  $\psi(R)$  in a similar way as for the recency factor. Once the  $\psi(\cdot)$ s are figured out, each factor should be assigned a numerical weight to indicate the relative importance of its role. The preferable relevance score of each documents in the subset to be ranked is evaluated by the following evaluation function.

**Definition 4.6.2:**

Given a subset of documents selected by means of selection rules, the evaluation function for ranking model is defined as follows:

- (1) The weighted scores of factors  $T$ ,  $F$ , and  $R$  are evaluated as

$$u(T) = t \psi(T)$$

$$u(F) = f \psi(F)$$

$$u(R) = r \psi(R)$$

- (2) The weighted score of the quality factor is calculated using the following quality evaluation function:

$$u(Q) = q * \text{MAX}(a \psi(A), c \psi(C), s \psi(S))$$

- (3) The preferable relevance score  $\rho$  is calculated using the following preference evaluation function:

$$\rho = \text{MAX}(\rho_1, \rho_2, \rho_3, \rho_4),$$

where  $\rho_1 = \text{MIN}(1, \alpha_0 u(Q))$ ,

$$\rho_2 = \text{MIN}(1, \rho_1 + \alpha_1 u(T)),$$

$$\rho_3 = \text{MIN}(1, \rho_2 + \alpha_2 u(F)),$$

$$\rho_4 = \text{MIN}(1, \rho_3 + \alpha_3 u(R)),$$

and  $\alpha_i, i=0,1,2,3$ , are appropriate non-negative constants specified by the ranking subsystem.

The properties of the ranking function given by Definition 4.6.2 is described in the theorems below.

**Theorem 4.6.1:**

The measure of preferable relevance,  $\rho(Q, T, F, R)$  satisfies

$$0 \leq \rho(Q, T, F, R) \leq 1$$

**Proof:**

Since the specified scores  $u(A)$ ,  $u(C)$ , and  $u(S)$ , and their corresponding weights  $a$ ,  $c$ , and  $s$  are numerical values between zero and one, we have

$$0 \leq u(A) \leq 1$$

$$0 \leq u(C) \leq 1$$

$$0 \leq u(S) \leq 1$$

Thus,  $0 \leq u(Q) = \text{MAX}(a * u(A), c * u(C), s * u(S)) \leq 1$ .

Further, since  $\rho_1, \rho_2, \rho_3$ , and  $\rho_4$  are bounded by 1, and constants  $\alpha_i \geq 0, i=1,2,3$ , we have

$$0 \leq \rho(Q, T, F, R) \leq 1 \quad \square$$

**Theorem 4.6.2:**

Given two documents expressed in terms of their quality scores, time scores, fitness scores, and reachability scores, and two documents

$$d_1 = \langle u(Q_1), u(T_1), u(F_1), u(R_1) \rangle \quad \text{and} \quad d_2 = \langle u(Q_2), u(T_2), u(F_2), u(R_2) \rangle,$$

then,  $\rho(d_1) \geq \rho(d_2)$  if

$$u(Q_1) \geq u(Q_2)$$

$$\alpha_1 \leq \rho_1(d_1) - \rho_1(d_2),$$

$$\alpha_2 \leq \rho_2(d_1) - \rho_2(d_2),$$

and  $\alpha_3 \leq \rho_3(d_1) - \rho_3(d_2).$

**Proof:**

Since  $u(Q_1) \geq u(Q_2)$ , we have

$$\rho_1(d_1) \geq \rho_1(d_2)$$

$$\rho_2(d_1) = \min(1, \rho_1(d_1) + \alpha_1 * u_{d_1}(T))$$

$$\geq \min(1, \rho_1(d_2) + \alpha_1 * u_{d_1}(T))$$

$$\geq \rho_2(d_2)$$

$$\rho_3(d_1) = \min(1, \rho_2(d_1) + \alpha_2 * u_{d_1}(F))$$

$$\geq \min(1, \rho_2(d_2) + \alpha_2 * u_{d_1}(F))$$

$$\geq \rho_3(d_2)$$

$$\rho_4(d_1) = \min(1, \rho_3(d_1) + \alpha_3 * u_{d_1}(R))$$

$$\geq \min(1, \rho_3(d_2) + \alpha_3 * u_{d_1}(R))$$

$$\geq \rho_4(d_2)$$

Thus,  $\rho(d_1) \geq \rho(d_2)$   $\square$

Theorem 4.6.2 provides a useful methodology for the selection of constants  $\alpha$ . During the ranking process, we shall first calculate the quality scores for each document in the set to be ranked. The system then multiplies the quality scores by a weight  $\alpha_0$  such that the differences among them would be increased or decreased to form several score 'cliques', i.e., two scores within the same clique are relatively close, and those in different cliques are significantly different. In fact, we have ranked the documents in terms of quality scores. This ranking result can be altered as we continue the ranking process by combining more factors. However, Theorem 4.6.2 provides a method for us to control the later alteration. For example, if we want to allow the documents with their scores falling in a clique to alter their position of rank, but keep the general ranks of documents in different cliques after combining the time factor, we can accomplish this by setting

$$\alpha_1 = \rho_1(d) - \rho(d')$$

where  $\rho_1(d)$  and  $\rho_1(d')$  are in two adjacent cliques  $C_1$  and  $C_2$  such that any  $\rho_1$  in  $C_1$  is greater than  $\rho_1$  in  $C_2$ , and  $\rho_1(d) = \min_{C_1}(\rho_1)$ ,  $\rho_1(d') = \max_{C_2}(\rho_1)$ . Similarly, we can control the ranking procedure by setting appropriate values of  $\alpha_2$  and  $\alpha_3$  when the fitness and reachability factors are combined.

The ranking model provides great flexibility for a variety of different situations. The logical relationships among four preference factors are subject to change as the different selection of constants alters the interpretation of the logical expression. More heuristic rules could be built into the ranking model.



## **CHAPTER 5**

### **CONCLUSIONS: TOWARDS AN EXPERT SYSTEM**

We have presented a hybrid model for document retrieval systems. There are two main topics studied in this work. The first topic was to develop a composite index term weighting model which incorporated three views of term significance, supported by previous research and experimental results, into a uniform way to calculate the index term weights. The composite weighting function developed here was a linear combination of three factors of term significance with each of them represented by a general weighting function which is able to portray an ideal weighting curve and to accommodate itself to different views by choosing appropriate constants.

Two points were observed about the composite weighting model. One is its adjustability to various indexing environments, mainly characteristics of the heterogeneity/homogeneity of the document collection. Instead of advocating a universal model, we insist that an index term weighting model would function well only in association with a concrete system environment. The other is its extendibility to incorporate a new factor of term significance, if discovered in the future, by adding a new term of general weighting function to the linear combination of the composite weighting function.

For applying the model, we propose a number of strategies to set up benchmarks in order to select the appropriate constants in the formula. Theoretically, our model is a generalization of some simple weighting models, such as the inverse document frequency(IDF) scheme. The actual performance is largely dependent upon how the

coefficients are chosen to produce a system that works fairly well under a concrete operational environment. In this sense, expertise is critical to the ultimate performance of the indexing system. Future work in this area involves development of a systematic, if not analytic, way to determine the effect of the indexing environment on term weight assignment. For instance, we expect to know exactly the impact of the homogeneity/heterogeneity of the document collection on the factors of term significance. In addition, expertise is also required for reorganization or modification of an indexing system. Some researchers have presented such ideas based on immediate feedback from on-line information users [74,75]. An inductive reasoning mechanism could be explored on the basis of periodic on-line retrieval experiences in order to build a more sophisticated indexing system.

The second topic was to develop a methodology for the design of a comprehensive document retrieval system. We proposed a composite query language, a composite indexing model, a query processing model, a matching model, and a ranking model. The composite query language mainly consists of the Phrase-Oriented Fixed-Level Expressions(POFLE), which can be viewed as a variation of Boolean expressions; however, there are more than Boolean operators incorporated in the POFLE, and a retrieval by synonyms can be performed through a selective element, a retrieval by context can be performed through a conditional element, and a retrieval by co-existence relationship can be performed through a joint element.

The composite indexing model was developed under a new strategy of creating a stem-based index term file, a phrase-based document description file, and a knowledge-based user classification file. The composite weighting model was applied

to assign numerical weights to both index terms and document descriptors. In the creation of the document description file, both linguistic and statistic methods were used to generate a set of raw phrases, and then a set of decomposition rules and a set of selection rules were developed to build into the document descriptors a hierarchical relationship so that a partial matching mechanism between query descriptors and document descriptors could be enforced. Hence, in our model, query descriptors presented by a user need not be completely consistent with document descriptors produced by the system. A user classification model was described in mathematical terms, including four types of rules in association with each user profile to depict how the factors of quality and recency would affect the users' preference for the documents with respect to their information needs.

The query processing model was proposed to perform a task of query reformulation to ensure an appropriate retrieval and to function as a screen to eliminate those hopeless documents from an initial response set. An interactive procedure was suggested for the query reformulation so that the new formulated query would be verified by the user and the new information drawn from the user could be used to modify the various dictionaries.

A delicate matching model was developed to assign a topical relevance measure to each document in the initial response set. The matching model was characterized by a partial matching mechanism based on the indexing structure, along with a retrieval by synonyms, a retrieval by context, and a retrieval by co-existence relationship through the selective elements, conditional elements, and joint elements, which are possibly used in POFLE. The matching function proposed denoted our model as

being more general than the traditional Boolean model and the term vector space model.

The ranking model was developed to rank topically relevant documents according to their preferable relevance score evaluated by means of four factors of user preference, including quality, recency, fitness, and reachability. The ranking model was designated by a weighted Boolean expression of the above four factors so that the evaluation could be done in a similar way to that of the evaluation of the logical expression of index terms. The documents finally presented to a user were in descending order of the retrieval status value, which was implemented as a preferable relevance score.

The composite retrieval model presented in this paper has demonstrated a number of attractive features which are lacking in the current competing models. However, what we have achieved is still a methodology for the system design; many problems yet need to be solved at the time of an implementation under a specific system environment. For instance, in the ranking model, although our theory has indicated the possibility of combining four preference factors such that the ranking process can be controlled by selecting appropriate coefficients of the evaluation function, the actual performance will be affected by how to weight different factors according to a user's information needs being reflected by means of a user profile. Expertise is needed in creating a user classification file, and the system must be able to simulate expert thinking in order to weight different factors and select appropriate coefficients. This will have to be developed further in the future. In addition, we are considering the problem of one document being more relevant than the other, not simply either

relevant or non-relevant; more factors regarding the quality score, such as novelty, and more factors regarding the preferable relevant score may be explored and incorporated into the model. We suggest that more research efforts be made to implement a document retrieval system as an expert system in the future experiments. Several of the main jobs that remain are listed below, in no particular order.

1. More work needs to be done with the application of the composite weighting model. This includes the exploration of an quantitative relationship between the measures of homogeneity/heterogeneity of the document collection and the values of constants to be specified in the formula of the composite weighting function.
2. More work needs to be done with the development of a complete set of linguistic rules in extracting the raw phrases from the given document collection. The types of raw phrases may be expanded to cover more semantically valid phrases, and extend the decomposition rules and selection rules accordingly.
3. More work needs to be done with the implementation of the algorithms in the query processing model. This include a complete set of rules for the query reformulation and for the screen test, and a set of criteria in order to control the retrieval by synonyms, retrieval by context, and retrieval co-existence relationship.
4. More work needs to be done with the construction of an inferential engine for the ranking model. This includes a complete set of rules for the specification of the sensitivity measures that reflects the differences of users' views of preference between different categories and for the specification of the coefficients in the evaluation function of the ranking model.

5. A comparison must be made between the composite retrieval system and other experimental/commercial systems in both matching and ranking facilities. Since all the current evaluation measures for retrieval effectiveness are based on a binary judgement, i.e., a document is judged either relevant or non-relevant, it is also a task for the future research to develop an evaluation mechanism based on a fuzzy measure.

Once we have completed the above jobs, we shall have a clearer picture about the composite retrieval system. As our methodology is capable of incorporating more expertise, the composite retrieval system is expected to be improved towards an expert system.

## **BIBLIOGRAPHY**

1. G. Salton, " Mathematics and information retrieval", Journal of Documentation, 35(1): 1-29, March 1979
2. K. Spark Jones, " A statistical interpretation of term specificity and its application in retrieval", Journal of Documentation, 28(1):11-20, March 1972
3. H.P. Zipf, " Human behavior and the principle of least effort", Addison-Wesley, Cambridge, Massachusetts, 1949
4. G. Salton and M.J. McGill, " Introduction to modern information retrieval", New York, McGraw-Hill, 1983
5. C.J. van Rijisbergen, " Information retrieval", Butterworths, London, 1979
6. G. Salton(Ed), " The SMART retrieval system- experiments in automatic document processing", Prentice-Hall, Englewood Cliffs, New Jersey, 1971
7. M. Dillon and A.S. Grey, " FASIT: a fully automatic syntactically based indexing system", Journal of the American Society for Information Science, 34(2):99-108, 1983
8. S. Braun and C. Schwind, " Automatic, semantics-based indexing of natural language texts for information retrieval systems", Information Processing and Management, 12:147-153, 1976

9. G. Salton and R.K. Waldstein, " Term relevance weights in on-line information retrieval", *Information Processing and Management*, 14(1):29-35, 1978
10. W.S. Cooper and M.E. Maron, " Foundations of probabilistic and utility theoretic indexing", *Journal of the ACM*, 25(1):67-80, Jan. 1978
11. S.P. Harter, " A Probabilistic approach to automatic keyword indexing, Part1: On the distribution of speciality words in a technical literature, Part2: An algorithm for probabilistic indexing", *Journal of the American Society for Information Science*, 26:197-206, 280-289, 1975
12. K. Spark Jones, " Index term weighting", *Information Storage and Retrieval*, 9:619-633, 1973
13. A. Bookstein and W. Cooper, " A general mathematical model in information retrieval", *Library Quarterly*, 46(2): 153-167, April 1976
14. T. Noreault, M. Koll and M.J. McGill, " Automatic ranked output from Boolean searches in SIRE", *Journal of the American Society for Information Science*, 28(6):333-339, Nov. 1977
15. G. Salton, A. Wong and C.T. Yu, " A vector space model for automatic indexing", *Communications of the ACM*, 18:613-620, 1975
16. A. Bookstein, " Explanation and generalization of vector models in information retrieval", in G. Salton & H.J. Schneider(Eds.): *Research and Development in*



Information Retrieval, Proc. Berlin 1982, Lecture Notes in Computer Science, Vol 146, Springer-Verlag, Berlin, 1983

17. M.E. Maron and J.L. Kuhns, " On relevance, probabilistic indexing and information retrieval", Journal of ACM, Vol 7, pp.216-244, July 1960
18. S.E. Robertson, M.E. Maron and W.S. Cooper, " Probability of relevance: a unification of two competing models for document retrieval", Information Technology: Research and Development, Vol 1, pp.1-21, Jan. 1982
19. C.J. van Rijisbergen " A theoretical basis for the use of co-occurrence data in information retrieval", Journal of the Documentation, 33(2):106-119, June 1977
20. A. Bookstein and D.R. Swanson, " A decision theoretic foundation for indexing", Journal of the American Society for Information Science, 26(1):45-50, Jan.-Feb. 1975
21. W.S. Cooper and P. Huizinga, " The maximum entropy principle and its application to the design of probabilistic retrieval systems", Information Technology: Research and Development, 1:99-112, 1982
22. William S. Cooper, " Exploiting the maximum entropy principle to increase retrieval effectiveness", Journal of the American Society for Information Science, 34(1):31-39, Jan. 1983

23. S.E. Robertson, " On the nature of fuzz: a diatribe", Journal of the American Society for Information Science, 29(6):304-307, 1978
24. V. Tahani, " A fuzzy model of document retrieval systems", Information Processing and Management, 12:177-188, 1976
25. T. Radecki, " Conceptual model of an information retrieval system based on the concept of fuzzy thesaurus", Information Processing and Management, 12:313-318, 1977
26. A. Bookstein, " Fuzzy requests: an approach to weighted Boolean searches", Journal of the American Society for Information Science, 31(4):240-247, July 1980
27. P.B. Kantor, " The logic of weighted queries", IEEE Transactions on Systems, Man, and Cybernetics, SMC11(12):816-821, Dec. 1981
28. D.A. Buell and D.H. Kraft, " A model for a weighted retrieval system", Journal of the American Society for Information Science, 32(3):211-216, May 1981
29. W. Goffman, " An indirect method of information retrieval", Information Storage and Retrieval, 4(4): 361-373, 1968
30. S.F. Dennis, " The design and testing of a fully automatic indexing-searching system for documents consisting of expository text", in Information Retrieval: A Critical Review, G. Schechter(Ed.), Thompson Book Co., Washington, D.C., pp.67-94, 1967

31. H.P. Luhn, " The automatic creation of literature abstract", IBM Journal of Research and Development, 2:159-165, 1958
32. C.E. Shannon, " A mathematical theory of communication", Bell System Technical Journal, 27(3): 379-434, 1948
33. V.V. Raghavan and S.K. Wong, " Critical analysis of vector space model for information retrieval", Journal of the American Society for Information Science, 37(5): 279-287, 1986
34. W.B. Croft and D.J. Harper, " Using probabilistic models of document retrieval without relevance information", Journal of Documentation, 35(4): 285-295, Dec. 1979
35. D.B. Cleveland, " An n-dimensional retrieval model", Journal of the American Society for Information Science, 27: 342-347, 1976
36. T. Saracevic, " Relevance: a review of and a framework for the thinking on the notion in information science", Journal of the American Society for Information Science, 26(6):321-343, 1975
37. W.S. Cooper, " A definition of relevance for information retrieval", Information Storage and Retrieval, 7(1):19-37, June 1971
38. A. Bookstein, " Relevance", Journal of the American Society for Information Science, pp.269-273, Sep. 1979

39. D.A. Kemp, " Relevance, pertinence and information system development", *Information Storage and Retrieval*, 10:37-47, 1974
40. A. Bookstein and D.R. Swanson, " Probabilistic model for automatic indexing", *Journal of the American Society for Information Science*, 25(5):312-318, 1974
41. C.T. Yu and G. Salton, " Precision weighting-an effective automatic indexing method", *Journal of the ACM*, 23(1):76-88, Jan 1976
42. S.E. Robertson and K. Spark Jones, " Relevance weighting of search terms", *Journal of the ACM*, 27(3):129-146, May-June 1976
43. G. Salton and C.S. Yang, " A theory of term importance in automatic text analysis", *Journal of the American Society for Information Science*, 26:33-44, 1975
44. G. Salton, A. Wong and C.T. Yu, " Automatic indexing using term discrimination and term precision measurements", *Information Processing and Management*, 12:43-51, 1976
45. A. Bookstein, " A comparison of two systems of weighted Boolean retrieval", *Journal of the American Society for Information Science*, 32:275-279, July 1981
46. H. Wu and G. Salton, " A term weighting model based on utility theory", in *Information Retrieval Research*, R.N. Oddy, S.E. Robertson, C.J. van Rijsbergen and P.W. Williams(Eds.), Butterworths, Lodon, pp.9-22, 1981

47. W.G. Waller and D.H. Kraft, " A mathematical model of a weighted Boolean retrieval system", *Information Processing and Management*, 15(5):235-245, 1979
48. C.J. Date, " An introduction to database systems", Reading, Massachusetts, Addison-Wesley Pub. Co., 4th ed., 1986
49. J.D. Ullman, " Principles of database systems" Maryland, Computer Science Press, 2nd ed., 1983
50. Ronald R. Yager, " A note on weighted queries in information retrieval systems", *Journal of the American Society for Information Science*, 38(1):23-24, Jan. 1987
51. C.T. Yu, W.S. Luk and M.K. Siu, " On models of information retrieval", *Information Systems*, 4(3):205-218, 1979
52. D. J. Harper and C.J. van Rijsbergen, " An evaluation of feedback in document retrieval using co-occurrence data", *Journal of Documentation*, 34(3):189-206, Sep. 1978
53. D.A. Buell and D.H. Kraft, " Threshold values and Boolean retrieval system", *Information Processing and Management*, 17:127-136, 1981
54. D.H. Kraft and D.A. Buell, " Fuzzy sets and generalized Boolean retrieval systems", *International Journal of Man-Machine Studies*, 19:45-56, 1983
55. D.J. deSolla Price, " The citation cycle", in "Key Papers in Information science",

- B.C. Griffith(Ed), Knowledge Industry Publications Inc., White Plains, New York, pp.195-210, 1980
56. J. Virgo, " A statistical procedure for evaluating the importance of scientific papers", Ph.D. Dissertation, Univ. of Chicago(1974)
57. C.D. Hurt, " Identification of important authors in science: a comparison of two methods of identification", Information Processing and Management, 21(3): 177-186, 1985
58. G. Salton, " Automatic indexing using bibliographic citations", Journal of Documentation, 27(2):98-110, June 1971
59. Elliot Noma, " Co-citation analysis and the invisible college", Journal of the American Society for Information Science, 35(1):29-33, 1984
60. R.G. Crawford, " The relational model in information retrieval", Journal of the American Society for Information Science, 33:51-64, Jan. 1981
61. I.A. Maclead, " Towards an information retrieval language based on a relational view of data", Information Processing and Management, 13:167-175, 1977
62. P. Wilson, " Situational relevance", Information Storage and Retrieval, 8:457-471, 1973
63. Ovad Mansur, " On selection and combining of relevance indicators", Information

Processing and Management, 16:139-153, 1980

64. M.H. Heine, " Incorporation of the age of a document into the retrieval process",  
Information Processing and Management, 13:35-47, 1977
65. P.M. Morse, " Library effectiveness", Mit Press, Cambridge, Mass., 1968
66. G. Salton, E.A. Fox and H. Wu, " Extended Boolean information retrieval", Com-  
munications of the ACM, 26(11), 1983
67. A. Hindle, " Markov model of book obsolescence", Information Processing and  
Management, 15:17-18, 1979
68. Z.W. Ras, " An algebraic approach to information retrieval systems", International  
Journal of Computer and Information Sciences, 11(4):275-293, 1982
69. S.E. Wiberley, " Journal rankings from citation studies: a comparison of national  
and local data from social work", Library Quarterly, 52(4):348-359, 1982
70. G.A. Miller, " The magical number seven plus or minus two: some limits on our  
capacity for processing information", Psychology Review, 63:81-97, 1956
71. P.H. Klingbiel, " Machine-aided indexing of technical literature", Information  
Storage and Retrieval, 9:79-84, 1973
72. T. Radhakrishnan, " Selection of prefix and postfix word fragments for data  
compression", Information Processing and Management, 14(2):97-106, 1978

73. J.L. Kolodner, " Indexing and retrieval strategies for natural language fact retrieval", ACM Transactions on Database Systems, 8(3):434-464, Sep. 1983
74. Lorrain M. Purgailis Parker, " Towards a theory of document learning", Journal of the American Society for Information Science, 34(1):16-21, 1983
75. K. Chores and C. Danilowuz, " Relative indexing", Information Processing and Management, 18(4):207-220, 1982
76. Steven Cater, " The topological information retrieval system and the topological paradigm: a unification of the major models in information retrieval", Ph.D. Dissertation, Department of Computer Science, Louisiana State University, 1986
77. M. J. McGill, L. Simith, S. Davidson, and T. Noreault, " Syracuse information retrieval experiment (SIRE): Design of an on-line bibliographic retrieval system", SIGIR Forum, 10(4): 37-44, Spring 1976
78. L. Jones, Edward Gassie, and S. Radhakrishnan, " INDEX: The statistical basis for an automatic conceptual phrase-indexing system", to appear in the Journal of the American Society for Information Science, 1988



## **VITA**

Zhen B. Zou was born in Zhejiang, China. He graduated from Xiao-Shi High School of Ningbo, Zhejiang province. In 1974, he entered Zhejiang University, where he received his B.S. degree in applied mathematics.

Upon graduation from Zhejiang University in 1977, he began to work as a technician in the institute of Application and Popularization of Electronic techniques in Beijing, where he remained for one year.

In the fall of 1978, after taking China's National Graduate Entrance Examination, he was admitted to the Graduate School of Chinese Academy of Science, where he studied for his M.S. degree in computing software.

He is currently a graduate student in Louisiana State University. His research areas of interest include information retrieval, database systems and artificial intelligence. He is going to receive his Ph.D. degree in Computer Science in December of 1988.

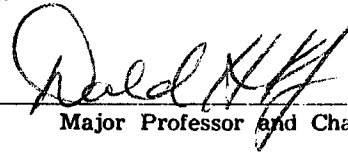
# DOCTORAL EXAMINATION AND DISSERTATION REPORT

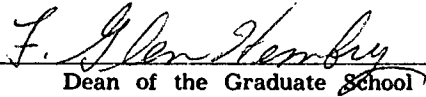
Candidate: Mr. Zhen-bao Zou

Major Field: Computer Science

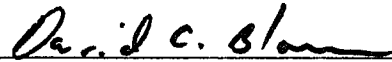
Title of Dissertation: A Hybrid Model for Document Retrieval Systems

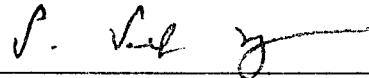
Approved:

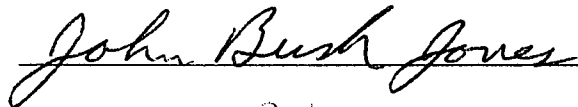
  
Major Professor and Chairman

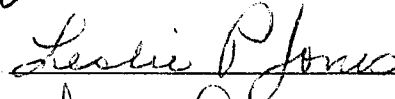
  
Dean of the Graduate School

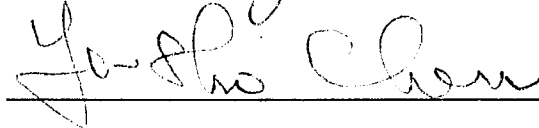
## EXAMINING COMMITTEE:

  
\_\_\_\_\_

  
\_\_\_\_\_

  
\_\_\_\_\_

  
\_\_\_\_\_

  
\_\_\_\_\_

Date of Examination: \_\_\_\_\_

August 4, 1988