INFORMATION RETRIEVAL BY MULTI-PERSPECTIVE REPRESENTATION

by

Jia Zeng

B.E., Huazhong University of Science and Technology, 2003

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

MATHEMATICAL, COMPUTER AND PHYSICAL SCIENCES

(COMPUTER SCIENCE)

THE UNIVERSITY OF NORTHERN BRITISH COLUMBIA

July 2005

Your file Votre référence
ISBN: 978-0-494-28380-6
Our file Notre référence
ISBN: 978-0-494-28380-6

Abstract

This thesis proposes an innovative document representation approach, called multi-perspective representation (MPR) model, for textual information retrieval (IR) purposes. It assumes that a document can be observed from multiple perspectives to obtain its derivatives and it can be associated with multiple representations obtained through the derived documents. We propose a generalized MPR solution that is applicable to the MPR model. We also present the differential latent semantic indexing (DLSI) method proposed in the literature, which is considered to be a special application of the MPR model. Experiments conducted on standard collections using the latent semantic indexing method and the term vector approach with the generalized MPR solution and the traditional representation method have shown that the generalized MPR solution makes significant contribution to the retrieval performance of both IR methods. The applicability of the DLSI method is also explored and the experiment on a benchmark database has demonstrated its effectiveness.

TABLE OF CONTENTS

List of Tables

List of Figures

Acknowledgement

I would like to thank my supervisor, Dr. Liang Chen, for his inspiration, support and guidance throughout my study and research work at UNBC. I also wish to thank Dr. Charles Brown, Dr. Jianbing Li for their encouragement and valuable advice on my thesis. As well, I would acknowledge the external reviewer for his/her willingness to serve on my thesis committee.

I would also give special thanks to Kevin Brammer who helps me along the way. Personally, I would like to thank my parents, without whom, I cannot have reached this point.

This thesis contains the contents of our previous publications for the subject. (Please refer to the appendix for a list of the publications.) I thank all the coauthors.

Chapter I

Introduction

This chapter presents an introduction to the topic of this thesis and the proposed methodology. The organization of the rest of the paper will be provided as well.

1 Problem Statement

With the exponential development of information technologies, more and more intellectual resources have been recorded in numerical forms of information that can be digitally transmitted or processed. The evolution of communication networks has made an immense number of the data sets available in the public domain, which can be obtained by ordinary users. However, due to the tremendous volume of the data collections, it has become more and more difficult for the users to find the information that interests them. This has led to a huge demand for the information management technologies, which can facilitate easy access of the information for the users.

Information retrieval (IR) is a discipline which looks into developing algorithms that can retrieve relevant information from storage in response to a user's request. Information can be recorded in many forms: textual, image, audio and video, etc. In this thesis, our research focuses on a major branch of information retrieval, called *textual retrieval* or *document retrieval*, which seeks to retrieve free-form natural language textual records that possibly satisfy the user's information needs from an organized repository. By convention, we call the textual records *documents*, the information requests *queries* and the repository a *collection* or *corpus*.

A number of approaches have been proposed for textual retrieval in the literature. One common method for document representation and indexing is called the *vector space model*. It considers each document as a set of *terms* and represents it as a *term vector* in a *document space*. Based on this model, a series of IR methods have been developed. Most of these approaches employ the *traditional representation model*, i.e., each document is associated with one and only one representation.

Are there any other possible solutions to represent a document other than the traditional model? How will they affect the performance of the IR methods? These questions are the inspirations for our research. The observation that multi-perspective descriptions have always been beneficial in visual and acoustic recognition has led to our belief that perceiving objects from two or more perspectives is always an advantage. Our research objective was to find out if it is helpful to observe a document from two or more perspectives, i.e., to associate the document with multiple representations.

2 Our Contribution

In this thesis, we have proposed a *multi-perspective representation* (MPR) method for document representation. Two sample solutions that are applicable to the MPR method have been provided. Experiments based upon these solution strategies have been conducted and their results have been reported as well. The following is a list of our contribution.

- The multi-perspective representation model we propose is an innovative document representation approach that attempts to imitate a human's manner of depicting

objects, that is, to describe them from different perspectives. It is assumed that each document can be observed from multiple perspectives to obtain its multiple derivatives, called *perspective documents*, and the *original document* can be represented by all the perspective documents that are derived from it. By applying the multi-perspective representation on documents, the procedure of measuring the similarity between a query and a document is converted to a process of integrating values that indicate the similarity degrees between the query and each of the perspective documents derived from that document.

- We have provided a generalized MPR solution, which is expected to be applied to most IR methods that are based upon the vector space model. Experiments have been conducted on several standard corpora using two IR methods: the *latent semantic indexing* (LSI) method and the *term vector* method, with the generalized MPR solution and the traditional representation model. The results have shown that both IR methods achieve significant improvement of retrieval performances with the MPR model over that with the traditional model [CZT05].

- We also discovered that the *differential latent semantic indexing* (DLSI) method that was proposed in the literature exhibits attributes that fall into the category of multi-perspective representation. We have conducted experiments that employ the DLSI method to solve an IR-related problem: medical data classification. Its performance on a benchmark data set has demonstrated the advantage of this approach [CZP04].

3 Overview

The purpose of this thesis is to present the multi-perspective representation model, to explore its applicability and to demonstrate its effectiveness. The following chapters outline the background, development and application of the MPR approach. Chapter II provides an overview of the basic IR background and reviews the approaches that are related to the proposed method in the literature. Chapter III presents the rationale of the MPR method and offers some guideline of its implementation. In Chapter IV, the experimental procedures and results are demonstrated. A summary of this thesis and a discussion of future work are provided in Chapter V.

Chapter II

Background and Literature Review

In this chapter, we will provide a brief overview to the field of information retrieval in order to facilitate the introduction of the proposed method. A historical review of a number of IR approaches will be presented, with the emphasis on the cognate methods that are relevant to the thesis topic. Evaluation strategies that are used to estimate the retrieval performance of an IR method will also be covered. As well, we will conduct a discussion over some special topics that are closely relating to our proposed approach.

1 Introduction to Information Retrieval

In Chapter I, we stated that "Information retrieval is a discipline which looks into developing algorithms that can retrieve relevant information from storage in response to a user's request". Before we proceed, it would be useful to conduct a discussion over this definition and clarify a few points.

1.1 Information Retrieval vs Data Retrieval

When the term *information retrieval* is mentioned in this paper, we refer to the automatic information retrieval systems that can search the database for data records that are relating to the user's information need, and inform the user on the existence/non-existence as well as the whereabouts of the document. Although the term *information* has a close association with the term *data*, they are not equivalent concepts. The field of *data retrieval* (DR) possesses some properties that are distinguishable from information

retrieval. Table 1 illustrates some differences between the two topics [vR79].

Table 1: Information Retrieval vs Data Retrieval

	Information Retrieval	Data Retrieval
query language	natural language	artificial language
query specification	incomplete	complete
matching	best match	exact match
satisfied retrieval	relevant	exact matching
model	probabilistic	deterministic

In IR, the query statement is usually expressed in natural language and does not necessarily need to be complete. On the contrary, DR usually requires the request to comply to the specified syntax and to provide as complete a description of the information need as possible. With regard to evaluating the retrieved records, IR considers the items that are relevant to the query to be good matches and among them the most relevant one judged by the user is determined to be the best match, whereas DR only regards the exact matches to be successful. Due to the differences between the characteristics of IR documents (unstructured records) and DR documents (structured records), information retrieval engines are possibly able to, but not guaranteed to find out all the relevant documents from the storage. Even if they manage to retrieve a list of relevant records, such a list might not be complete. On the other hand, DR systems are guaranteed to output all occurrences of the records satisfying a match.

1.2 Information Retrieval Systems

An information retrieval system is a device interposed between a potential user of information and the information collection itself [Har86]. For a given information problem, the purpose of an IR system is to capture wanted items and to filter out unwanted items [Har86]. A typical IR system deals with representation, storage and retrieval of unstructured data, thus should contain some/all of the following parts: indexing, query operation, matching, output module, feedback module and user interface.

The indexing component usually contains two primary processes. The first process is to conduct conceptual analysis on the information resources in the collection to obtain the concepts that are contained. These concepts, usually called *effective terms*, make up a *system vocabulary* that is applicable to all the information pieces in this system. The output of the first process flows to the second stage, in which a translation mechanism is employed and a database of information representations can be obtained. When an information request is posed, the query operation process will parse it into its constituent elements. An analysis will be conducted over its conceptual content and the query will be transformed into a representation that is consistent with all the information items in storage. Given the query representation, the matching mechanism evaluates the relatedness of all the potential targets to the query and obtains a rank of relevance. An ordered set of information items will be returned to the user by the output module. When interaction between the user and the information retrieval system is available, he/she can communicate with the system through the feedback module by refining the query during one search session in the light of a sample retrieval. The user interface serves as a bridge

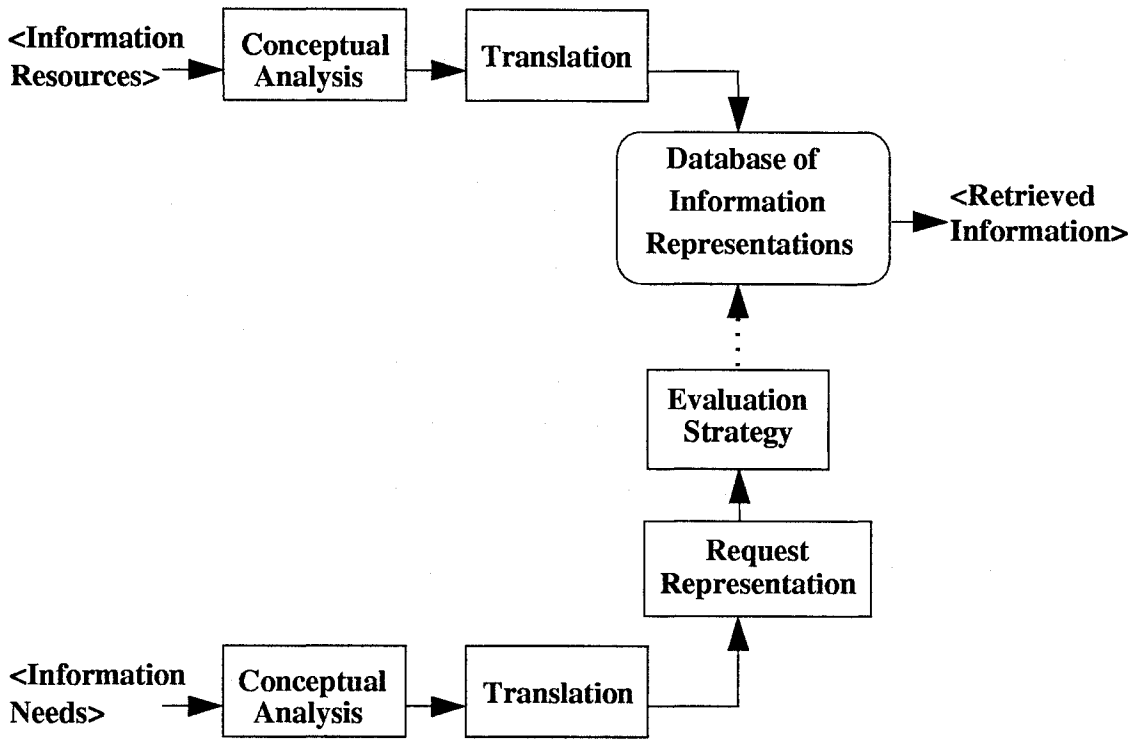

Fig. 1: Framework of An Information Retrieval System

connecting the client to the other modules of the system. The infrastructure of a typical IR system is depicted in fig. 1.

2 Information Retrieval Approaches

In the field of information retrieval, there are two major categories of techniques: statistical analysis and semantic analysis. The statistical approaches consider the free-form natural language documents to be pure data records and index them in terms of some statistical measure. The assessment of the relevance between a pair of documents is also based on a certain statistical metric. The semantic approaches, however, attempt to reproduce to some degree the understanding of the natural language text that a human may provide. This section selectively reviews some of the IR methods in the literature.

2.1 Statistical Analysis

Statistical methods break documents and queries into terms, which make up a population that is counted and measured statistically. Most commonly, these terms are words that occur in a collection and/or a given query. Textual records often undergo *preprocessing* when words are usually converted into a standard form, e.g., lowercase. Other techniques such as *stemming* and *stop-listing* are also frequently employed in the preprocessing stages. The stemming algorithm, first introduced by Porter [Por80, Por97], is to extract the root of each word in order to eliminate the variation that arises from the occurrence of different grammatical forms of the same word. By replacing the terms by their base forms, the number of terms can be reduced in the meanwhile. The objective of stop-listing is to remove the common words that have little discriminatory power. The list that describes such words is always called a *stop list* or *negative dictionary.*

In addition to using words as terms, some sophisticated methods also extract phrases as terms, where a phrase is a combination of adjacent words which may be recognized by frequency of co-occurrence in a given collection or by presence in a phrase dictionary. Some other methods break documents into *n-grams*, i.e., arbitrary strings of n consecutive characters [Dam95]. It is worth noting that the use of n-grams has proved to be language independent and appear to be relatively insensitive to degraded text, e.g., typos, errors due to poor print quality in *optical character reader* (OCR) transmission, etc [PN96].

By far, the greatest amount of work has been devoted to the statistical approaches, which fall into four categories: *classical Boolean*, *extended Boolean*, *vector space* and *probabilistic.* The rest of the section presents some representative methods of these

categories.

2.1.1 Classical Boolean Approach

The classical Boolean approach is based upon the theory of Boolean algebra. A conventional Boolean query combines terms with the classical Boolean operators *AND*, *OR* and *NOT*, and is evaluated by the classical rules of Boolean algebra. Such a model is very straightforward and easy to implement.

However, due to the characteristics of the standard Boolean model, the classical Boolean method encounters some major limitations in the field of information retrieval. Like any Boolean expression, the query only has two values: *true* or *false*. Correspondingly, a document is either relevant to a query or non-relevant to it. Therefore, no ranking strategy is possible. With regard to effective term weighting, only two values are available: 0 for an absent term and 1 for a present term. Such an all-or-nothing condition tends to have the effect that either an intimidatingly large number of documents or none at all are retrieved [Har92]. As well, the classical Boolean rules tend to produce counter-intuitive results because of this all-or-nothing characteristic. For example, in response to a multi-term OR operation, "a document containing all (or many of) the query terms is not treated better than a document containing one term" [SB88]. Similarly, in response to a multi-term AND operation "a document containing all but one query term is treated just as badly as a document containing no query term at all" [SB88]. These features of the classical Boolean model have emerged as a considerable issue that needs to be overcome.

2.1.2 Extended Boolean Approach

As we mentioned above, the classical Boolean scheme is based upon the assumption of all or nothing, and expresses the relevant relationship between documents in a crude way. Due to this reason, a number of extended Boolean models have been proposed in the literature [WK79, SM83, Pai84, Zim91, GCT97] trying to integrate a ranking strategy into the Boolean model.

The extended Boolean models employ *extended Boolean operators*, also called *soft Boolean operators*. These operators make use of the weights assigned to the terms in each document. They also extend the truth value range from a discrete two-element-set: $\{0, 1\}$ in the case of classical Boolean model to a consecutive range: $[0, 1]$. In other words, the operators evaluate their arguments to a number, corresponding to the estimated degree to which the given logical expression matches the given document. By doing this, the extended Boolean methods are able to provide a ranked output allowing some documents to satisfy the query condition more closely than others [Lee94]. Therefore it manages to overcome the limitation of the classical Boolean approach. Experiments have shown that the extended Boolean model can achieve better IR performance than either the classical Boolean or the vector space model [Gre01]. However, there is a big price for this performance improvement. Formulating effective extended Boolean queries involves more thought and expertise in the query domain than either the classical Boolean method or the vector space approach [Gre01].

2.1.3 Vector Space Approach

One common solution to document representation for statistical purposes is to represent each document as a set of terms. The vector space approaches have achieved great success in IR by applying the theory of linear algebra on this representation model.

In the *traditional vector space method*, also called the *term vector method*, the union of the effective terms defines a *document space* so that each distinct term represents one dimension in this space. For a given document, each term is assigned a numeric weight, which indicates an estimate of the usefulness of the term as a descriptor of the given document, i.e., the discriminatory power of the term for this document. By exploiting the weights of all the terms for a document, the document is then encoded as a term vector in the document space. It is worth noting that a query is usually considered to be a *pseudo-document* and can also be represented as a term vector.

Sometimes, it is also desirable to define a *term space*, where each document corresponds to one dimension. Accordingly, a term is represented by a document vector in this space.

We can combine the perspectives of document space and term space by viewing the entire collection as a *term-by-document matrix*, also called an *indexing matrix*. Each row of this matrix represents a term and each column of this matrix represents a document. The element m_{ij} at row i, column j reflects the importance of term i in representing the characteristics of document j. A data set of d documents and t terms can be represented by a matrix shown below.

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1d} \\ m_{21} & m_{22} & \cdots & m_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ m_{t1} & m_{t2} & \cdots & m_{td} \end{pmatrix}$$

Please note that any defined denotation, e.g. d, t and M, or abbreviation, will be applicable to the rest of the text.

Term Weighting

One significant issue that any vector space model needs to consider is *term weighting*, i.e., how to assign a weight to a certain term for a given document so that it properly estimates the contribution this term makes to the document in the respect of distinguishing it from other documents. A variety of weighting schemes have been proposed, which basically fall into two categories: *local weighting* and *global weighting.*

Local weighting schemes attempt to reflect the importance of a term within a given document by document-specific statistics. Usually, the local weights assigned to the same term vary from document to document. Some popular local weighting functions include the *raw term frequency*, *binary* and *logarithm of the term frequency* (or logarithm for short). Let us denote L_{ij} to be the local weight of term i in document j and denote tf_{ij} to be the frequency with which term i appears in document j. The local weights by the three types of schemes are evaluated as follows.

- Raw Term Frequency: $L_{ij} = tf_{ij}$
- Binary: $L_{ij} = \begin{cases} 1 & \text{if } tf_{ij} \geq 1 \\ 0 & \text{otherwise} \end{cases}$
- Logarithm: $L_{ij} = log(1 + tf_{ij})$

It is worth mentioning that the logarithm weighting scheme exploits the logarithmic function to transform the raw term frequency so that it can dampen effects of large differences in frequencies.

In addition to estimating the document-specific statistics, it could also be useful to characterize a term's overall importance in the whole collection. Global weighting strategies are designed to measure the distribution of a term within the given collection, thus are able to estimate how likely a term occurs in a certain document by chance. Generally, they give less weight to terms that occur frequently or occur in many documents because these terms are not considered to be strong descriptors for any specific document in which they appear. Four well-known global weighting schemes are: *normalized term frequency* (or normal for short), *inverse document frequency* (or idf for short), *global frequency-idf* (or gf-idf for short) and *entropy.* Let G_i be the global weight assigned to term i, gf_i be the frequency term i occurs in the entire collection, df_i be the frequency of documents in which term i occurs and d be the number of documents in the whole collection. The evaluation of G_i by the four types of global weighting methods are represented as follows.

- Normal: $G_i = \sqrt{\frac{1}{\sum_{j=1}^{d} tf_{ij}^2}}$
- Idf: $G_i = log(\frac{d}{df_i}) + 1$
- Gf-idf: $G_i = \frac{gf_i}{df_i}$
- Entropy: $G_i = 1 - \sum_{j=1}^{d} \frac{p_{ij} log(p_{ij})}{log(d)}$, where $p_{ij} = \frac{tf_{ij}}{gf_i}$

All of the global weighting schemes share a principle of assigning less weight to terms that occur frequently or in many documents. The ways in which this is done involve

variations in the relative importance of local frequency, global frequency and document frequency. The normal weighting scheme normalizes the length of the vector for a term to 1. This has the effect of giving high weight to infrequent terms. However, it only depends on the sum of the squared frequencies and not the distribution of those frequencies per se. Gf-idf and idf are closely related because both schemes weight terms inversely by the number of different documents in which they appear. Gf-idf also increases the weight of frequently occurring terms. Neither method depends on the distribution of terms in documents but the number of different documents in which a term occurs. The entropy scheme is based on information theoretic ideas and is the most sophisticated weighting scheme. It takes into account the distribution of terms over documents. The main idea is to assign little weights to terms that are equally distributed over documents and assign big weights to terms which are concentrated in a few documents [Dum91].

After all, weighting scheme contains advantages and limitations. There is not a fixed solution for choosing a term weighting scheme. In the cases when both local weights and global weights are used to measure the term weights, the value of the ith row, jth column element can be evaluated as follows.

$$m_{ij} = L_{ij} \times G_i \tag{1}$$

Normalization

Another technique that is always employed in vector space approaches is *normalization*. It is based on the recognition that the size of documents in the collection may be different from each other. In order to allow for the variation, it normalizes each term vector so that its length is 1 for a document of any size.

Similarity Measurement

We have discussed about how to obtain term vectors to represent documents (including queries), and how to construct the representation for the entire document collection accordingly. The next step is to apply a similarity measurement that can estimate the similarity between a pair of vectors. A practical use of this metric is to evaluate the similarity between a pair of term vectors representing a document and a query respectively. This similarity value is used as an indicator of how relevant the document is to the query.

A usual similarity measure employed in the vector space model is the *inner product* [Sal89]. It can be computed by multiplying one element in one of the two vectors by the corresponding element in the other vector and summing these products over all dimensions in the vector. Suppose a term vector has t dimensions. Let us denote the vector representation of a document (query) by DR and denote the ith element of the vector DR by DR_i. The inner product of document D_j's vector, denoted by DR_j and query q's vector, denoted by DR_q, is defined as follows:

$$DR_j \cdot DR_q = \sum_{i=1}^{t} DR_{j_i} \cdot DR_{q_i} \tag{2}$$

Another popular metric derived from the inner product measure is called *cosine similarity*. Denote $|DR|$ to be the length of vector DR. The cosine similarity between vector DR_j and DR_q, denoted by $cos(DR_j, DR_q)$, can be evaluated by the following equation:

$$cos(DR_j, DR_q) = \frac{DR_j \cdot DR_q}{|DR_j||DR_q|} \tag{3}$$

Two strategies are commonly used to utilize the similarity values for retrieving relevant documents. One is called *range queries*, which is to retrieve all documents up to a distance threshold. The other one is called *nearest-neighbour queries*, which is to retrieve

the N best matches. Although we do not expect any similarity metric to be a perfect model that corresponds exactly with the human judgement of relevance, the measurement should somehow be able to assign higher values to the documents with a higher proportion of the relevant text as well as assigning lower values to the documents with fewer relevant content. By integrating the ranking strategy in IR systems, the human user can restrict his/her attention to a set of documents of manageable size instead of having to go through every single document in the corpus.

Discussion

As an efficient model, the traditional vector space scheme is becoming very popular in the IR research. Since it has a sound mathematical foundation, a variety of similarity measures can be developed based on this model and some of them could be transformed into a linear form. It is also accounted as an advantage that there exists a probabilistic interpretation of this model.

The traditional vector space approach provides an effective way to approximate the statistical properties of the document set, however, it is obviously an oversimplification. The major problem that exists with this method is that it assumes all the terms are independent, orthogonal dimensions of the document space so it simply ignores the relationship among terms. However, it is a fact that there are strong associations among terms in natural languages, the above assumption is never satisfied [Hul94]. Another feature this model has that can be a drawback in some applications is that the number of terms, which occur in a collection can be quite large, the traditional term-based document space tends to have a large number of dimensions. Some alternative approaches based

on the traditional vector space model have been developed to overcome these limitations. Latent semantic indexing method is one of them.

Latent Semantic Indexing

Motivation In the research of retrieving free-form natural language data, it is always useful to analyze the features of human verbal behavior. There are two issues that are discussed the most: *synonymy* and *polysemy*. Synonymy refers to the states when two or more words or expressions have the same or nearly the same meaning in some or all senses [MW98]. Polysemy describes the cases when one word has multiple meanings. These characteristics of natural languages result in the deficiencies of some IR algorithms that do not offer comparison methods on terms.

Latent semantic indexing method was proposed in order to capture the statistical relationship among terms and accordingly provide an effective solution to the problems of synonymy and polysemy that cannot be tackled by either word-based approaches or the traditional vector space approach.

Theory Latent semantic indexing, also called the *latent semantic analysis* (LSA) method, was first proposed by Deerwester *et al.* [DDL+90]. It assumes that, in the textual data, there is some underlying latent semantic structure that is partially obscured by the randomness of word choice with respect to retrieval. This structure can be estimated by statistical techniques and the obscuring noise can be removed.

Like the traditional vector space approach, the LSI method starts with a term-by-document matrix that represents the association of terms to documents. It applies a

dimensional reduction scheme, *singular value decomposition* (SVD), on the matrix to construct a reduced unified semantic space for retrieval. This smaller space, called *LSI space*, consists of derived dimensions that are assumed to convey truly independent concepts. By employing the dimensional reduction strategy, LSI not only captures most of the important underlying semantic structure in associating terms with documents, but also has a better chance in removing the noise or possible variability in word usage.

Singular Value Decomposition Singular value decomposition is an effective dimensional reduction scheme. It is closely related to a number of mathematical and statistical techniques that have been widely used in other fields, such as the *principal component analysis* (PCA) for image processing and face recognition [TP91]. It has been proved to be a very good choice for uncovering latent semantic structure. (See [DDL+90] for a further discussion of SVD and the other alternative models.)

It begins with an arbitrary rectangular matrix with different entries on the rows and columns. The matrix is then decomposed into three matrices containing singular vectors and/or singular values. These three matrices with special forms show a breakdown of the original matrix into linearly independent components or factors. Many of these components are very small, leading to an approximate model that contains many fewer dimensions. Thus, for information retrieval purposes, SVD provides a reduced model for representing the term-to-term, document-to-document and term-to-document relationships. By dimension reduction, it is possible for documents with somewhat different profiles of term usage to be mapped into the same vector of factor values [DDL+90]. This property helps to eliminate the noise in the original data, thus improving the reliability

of the algorithm.

Suppose we obtained a $t \times d$ term-by-document matrix M from the collection indexing process of the traditional vector space method (page 12). We can apply SVD on M, which is then decomposed into three special matrices U, S and V. The decomposition can be written as:

$$M = USV^T \tag{4}$$

U is the $t \times t$ orthogonal matrix having the left singular vectors of M as its columns, and V is the $d \times d$ orthogonal matrix having the right singular vectors as its columns, and S is the $t \times d$ diagonal matrix having the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{min(t,d)}$ of M in order along its diagonal. It should be noted that for any arbitrary matrix, such a factorization exists. (For details, see [Str98] for reference.)

Generally, in Eq. 4, the matrices U, S and V must all be of full rank. However, one great facilitation that SVD offers is to allow a simple strategy for optimal approximate fit using smaller matrices [DDL+90]. If the singular values in S are ordered by size, the first k largest values may be kept and the remaining smaller ones are set to zero. The product of the resulting matrices is a matrix M_k which is only approximately equal to M, and is of rank k. Since zeros were introduced into S, the representation can be simplified by deleting the zero rows and columns of S to obtain a new diagonal matrix S_k, and then deleting the corresponding columns of U and V to obtain U_k and V_k respectively. The rank-k model with the best possible least-squares-fit to M can be written as follows:

$$M_k = U_k S_k V_k^T \tag{5}$$

where M_k is a matrix of size $t \times d$, U_k is of size $t \times k$, S_k is of size $k \times k$, and V_k is of

size $k \times d$.

SVD provides an optimal solution to dimensionality reduction in that it derives an orthonormal space, where the dimensions are ordered. Therefore, projecting the set of documents onto the k lowest dimensions is guaranteed to have, among all possible projections to a k dimensional space, the lowest possible least-square distance to the original documents [SS97].

Retrieving Documents Using LSI The process of retrieving is to sort the documents according to their similarity degrees to the query and to return a ranked list of documents to the user. This involves the consideration of representing documents and queries in the same manner and applying a certain function to estimate the similarity between them.

In the LSI method, each document/query is projected onto the LSI space that is obtained by using SVD. LSI then exploits the cosine measurement between the projections of a pair of term vectors in the LSI space to make comparison between the two documents. Thus, the similarity can be obtained by computing the cosine value of the angle between the document's term vector and the query's term vector. All the documents can be ranked according to their similarity values and an ordered set of documents will be retrieved.

2.1.4 Probabilistic Approach

Theory

There is no clear line which separates probabilistic methods from statistical methods,

as probabilities are often calculated on the basis of some statistical evidence. The biggest feature of probabilistic methods that distinguishes them from other statistical methods is that, in probabilistic methodology, formal probability theory and statistics are used to arrive at the estimates of probability of relevance by which the documents are ranked. On the other hand, in pure statistical approaches, a similarity measure is used, whose value is not necessarily directly interpretable as probabilities [CGD92].

In a probabilistic method, one usually computes the conditional probability $P(D|R)$ that a given document D is observed on a random basis given event R, that D is relevant to a given query [Sal89, vR79]. Typically, queries and documents are represented by sets of terms, $P(D|R)$ is then calculated as a function of the probability of occurrence of these terms in relevant versus non-relevant documents [Gre01].

The term probabilities are analogous to the term weights in the vector space model and may be computed using the same statistical measures. A probabilistic formula is then employed to evaluate $P(D|R)$ and it depends on the specific model used as well as on the assumptions made about the distribution of terms, etc.

In a more general case, $P(D|R)$ may be computed based on any clues available about the document. This has led to the idea of a *staged* computation, in which a probabilistic model is first applied to each composite clue (stage one), and then applied to the combination of these composite clues (stage two) [CGD92]. The *logistic regression* analytical method is derived from this idea. Another scheme originating from this idea is the *inference net* in which rules can be specified for combining different sources of evidence to compute a *belief* that an information need has been satisfied by a given document [TC91]. *Bayesian inference network model* is a famous application of this scheme.

Probabilistic Latent Semantic Indexing

Probabilistic latent semantic indexing (PLSI) is a probabilistic approach, which was proposed by Hofmann [Hof99]. He pointed out that, although LSI has been applied with remarkable success in different domains, it has some deficits due to its unsatisfactory statistical foundation. The PLSI approach is based on a statistical latent class model for factor analysis. It has a solid statistical background since it is based on the likelihood principle and defines a proper generative model of the data [Hof99]. Moreover, the factor representation obtained by PLSI allows us to deal with polysemous words and to explicitly distinguish between different meanings and different types of word usage [Hof99].

The core of PLSI is a statistical model called *aspect model* [SP97]. It exploits a latent variable model for general co-occurrence data that associates an unobserved class variable z with each occurrence of a word w in a document d. Let $P(d)$ denote the probability of selecting a document d. Let $P(z|d)$ denote the probability of picking a latent class z. Let $P(w|z)$ denote the probability of generating a word w. A likelihood function combines all the aspects of $P(w|z)$ and eventually obtains a pair (d, w), while discarding the latent class variable z. In the procedure for maximum likelihood estimation in latent variable models, Hofmann proposed a generalization of maximum likelihood for mixture models, called *tempered expectation maximization* (TEM), which is based on entropic regularization and is closely related to a method known as *deterministic annealing* (refer to [RGF90] for details).

Hofmann claims that with respect to retrieval performance, PLSI yields better result than direct term matching methods as well as over LSI. The combination of models with different dimensionalities has proven to be advantageous.

Discussion

Despite all the advantages that a truly probabilistic design methodology can offer, advocates of non-probabilistic methods regard the formulation of exact statistical assumptions as an unnecessary theoretical burden on the researcher. They maintain that the time and effort spent on such analysis would be better spent on ad hoc experimentation using formalisms looser and friendlier than probability theory [CGD92].

2.2 Semantic Analysis

Semantic analysis, also called *natural language processing* (NLP), refers to the methods that are based on knowledge of the syntax and/or semantics of the natural language in which the document text is written. Unlike statistical approaches, which merely use statistical measurement, semantic approaches attempt to address the structure and meaning of textual documents directly.

The idea of retrieving relevant information based on the understanding of the context seems to be very intuitive to human beings. However, in automated systems, it is much more difficult to implement than the statistical methods. As a result, within the realm of IR, semantic approaches are often used as a supplement to the statistical analysis since even the best statistical method to date will fail to grasp all the features of the documents that a semantic approach may be able to discover. Some research work has been done trying to combine both semantic and statistical methods together hoping that this combo will perform better than the statistical methods alone [Ril95, CH95, LPY94].

3 Evaluation of Information Retrieval Performance

In previous sections, we presented the basic architecture of an IR system and introduced some representative approaches that can be used to implement it. In this part of the chapter, we will discuss criteria for evaluating an information retrieval system.

Frakes *et al.* [FBY92] provided a summary of the evaluation process: an information retrieval system can be evaluated in terms of many criteria, including execution efficiency, storage efficiency, retrieval effectiveness and the features they offer a user. The system designers look into the relative importance of these factors based on the particular environment and expectation in order to make appropriate selection of data structures and algorithms.

Execution efficiency is measured by the time it takes a system, or part of a system, to perform a computation. This can be measured in C based systems by using profiling tools such as prof on UNIX [Ear84]. This factor has always been a major concern for the interactive IR systems because a long retrieval time will interfere with the usefulness of the system. The requirements of such IR systems usually specify maximum acceptable times for searching, and for database maintenance operations such as adding and deleting documents.

Storage efficiency is measured by the number of bytes needed to store the data. Space overhead, a common measure of storage efficiency, is the ratio of the sizes of the index files plus the size of the document files over the size of the document files.

Most IR experimentation has focused on retrieval effectiveness, which is based on *relevance judgement*. Relevance is an inherently subjective concept in that the ultimate

goal is to satisfy the human users' needs. Due to variation of the user's personal background, it is impossible to design a perfect system that meets all of the expectations for all users. Therefore, it is necessary to introduce some measures to evaluate the performance of a retrieving process by estimating the degree of relevance at which the retrieved information matches the query.

3.1 Precision and Recall

Two widely used parameters to measure IR success, which are based on the concept of relevance, are *precision* and *recall*. Precision is the ratio of relevant items retrieved to all items retrieved. Recall is the ratio of relevant items retrieved to all the relevant items. To facilitate the understanding of these definitions, we present an evaluation contingency table, shown as below:

Table 2: Evaluation Contingency Table

	Relevant	Irrelevant
Retrieved	A	B
Not Retrieved	C	D

where A denotes the number of relevant items retrieved, B denotes the number of irrelevant items retrieved, C denotes the number of relevant items not retrieved but in the database, and D denotes the number of irrelevant items not retrieved but in the

database.

$$Precision = \frac{A}{A+B} \tag{6}$$

$$Recall = \frac{A}{A+C} \tag{7}$$

The expectation of the users may vary from one person to another. Some users attach more importance to precision, i.e., they want to see relevant information without going through a lot of junk. Others take recall as a preference, i.e., they want to see all the documents that are considered to be highly relevant. Hence the evaluation that involves only one of the two parameters may be biased. Due to this reason, some methods that evaluate the IR performance in terms of precision and recall simultaneously, have been developed. They are discussed in the following sections.

3.2 E Measure

A method that combines both measures of recall and precision, called *E measure* (short for *effective measure*), was proposed by van Rijsbergen [vR79]. It allows the user to specify the relative importance of precision and recall. The evaluation measure is defined as,

$$E \approx 1 - \frac{1}{\alpha\left(\frac{1}{P}\right) + (1-\alpha)\frac{1}{R}} \tag{8}$$

where P is precision, R is recall, and α is a parameter which varies from 0 to 1. The higher the value of α, the more important the measure of precision is considered and vice versa.

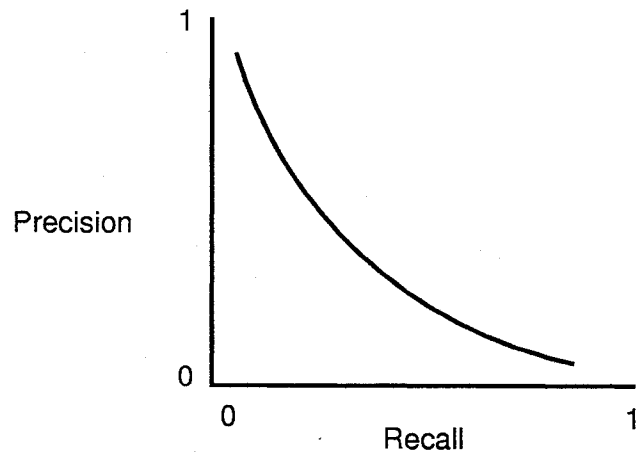

Fig. 2: Recall-Precision Graph

3.3 Recall-Precision Graph

Another method, called *recall-precision graph*, depicts the tendencies in which recall and precision values change and demonstrates the inter-relationship between them. It is illustrated by a bivariate plot where one axis is recall and the other precision. The recall-precision plots show that recall and precision are inversely related. That is, when precision goes up, recall typically goes down and vice versa. Fig. 2 illustrates this curve.

3.4 Average Precision

Although the recall-precision graph is a good indicator of the trade-off between the precision and recall, it is difficult to make a comparison of retrieval effectiveness with it. A more common type of measure that is widely used in the research community, called *average precision*, attempts to summarize this kind of curve as a single value in order to make different IR algorithms or the same algorithm on different document collections

comparable.

For a given query, a ranked list of documents ranging from the most relevant to the least relevant is obtained. An average precision for this query can be computed. This is done by computing the precision after a new relevant document is retrieved and then averaging the precision values that are computed in this way. These query average precisions are then combined (averaged) across all query topics in the collection to create the average precision for that collection.

One variant of the average precision is called *interpolated average precision.* It introduces the concept of *standard recall level* by defining a set of recall values, e.g, $0.25, 0.5, 0.75$. However, among the set of experimental recall values, some or all of these standard recalls might not exist. In order to solve this problem, we employ interpolation. For a given query, to interpolate the precision at a standard recall level ρ, we take the highest precision after an experimental recall is greater than or equal to ρ. The interpolated average precision for this query can be computed by averaging the interpolated precision values at all standard recall levels.

4 Special Topics

4.1 Representation Models

Most of the current IR research employs a traditional model to represent a document, i.e., regarding it as a bag of terms. This paper proposes an innovative representation method, called multi-perspective ("stereo") document representation, which assumes each document is associated with multiple representations that are obtained by observing

the document from different perspectives. By reviewing some recent research work in the area of information science, we were able to find some traces of the multi-perspective idea [CZT05].

Researchers in the area of face recognition have realized that by associating each person with more than one picture in the database, the system usually achieves better performance [WZ02]. As well, data fusion, a method which is based on the idea of integrating many answers to one question into a single best answer, has been successful in its application in meta-search [Mon02]. The main idea of meta-search is to combine the results from many different search engines, each employing different search strategies, to produce a single list. We can interpret this as each document being viewed in many different ways by many different search strategies. Experiments have shown that meta-search can significantly improve the raw performance of the input search engines [Mon02]. Researchers have also achieved success in information retrieval for structured documents such as HTML documents by using language models derived from multiple sources of structural information, such as in-links, title, URLs and out-degrees [OC03].

A successful automated tutoring system, AutoTutor, evaluates a student's response to a question by selecting the optimal value out of a set of matching values that result from comparing the response with each correct answer (aspect) and all possible combinations of the correct answers (aspects) [GWHWH+99]. An intelligent essay assessor, developed by researchers at University of Colorado, attempted to find out if it is valuable to evaluate a student's essay by representing it with two vectors, one based on the technical words only and the other based on the non-technical words only, and found that nothing was gained by doing so [RSW+98].

4.2 Retrieval Tasks

Based on the traditional information retrieval modes, there are two major types of tasks: an *ad hoc* task and a *routing* task. In the ad hoc task, it is assumed that new questions are being asked against a static set of data. In other words, the user can formulate any number of arbitrary queries but applies them to a fixed search target. In the routing task, it is assumed that the same questions are always being asked, but that new data is being searched [Har95]. That is, the queries cover a fixed number of topics, and when a new message comes, it is routed to the class corresponding to the topic that fits most closely to the message. In many routing experiments, there is just one topic or query, therefore, there are just two classes, relevant and non-relevant [Gre01].

It is worth mentioning that an information retrieval task has a very close association with a classification task. The problem a classifier tries to address is to compare an unlabeled data to all its existing clusters and group it into the cluster that shares the most commonalities with the data. If we consider the unlabeled data as an arbitrary query input by the user and clusters as the documents in the collection, we can see that the way, in which a classifier finds the cluster that is closest to the data, is analog to how an IR system manages to find the document that is the most relevant to a query. Thus the classification process can be viewed as a combination of information retrieval and database updating processes. Due to this reason, an IR method can have a direct application onto a classification system.

Chapter III

Methodology

This chapter looks into the methodology of the multi-perspective representation method. It presents an overview of the MPR approach. It also provides a generalized MPR solution that we have proposed in [CZT05] as well as a differential latent semantic indexing method that was proposed in the literature [CTN01, CTN03].

1 Overview of the MPR Method

Throughout the humans' experience of recognizing objects, it has been noted that multi-perspective descriptions have always been beneficial. Take visual and acoustic recognition for example. We know that two eyes are required for the stereographic recognition of objects, which give us our sense of depth perception. As well, two ears are necessary for distinguishing subtle differences among different music styles. Our experience in reading a document also seems to support this idea. We believe that, it is not necessary for us to read the entire context of one document before we can make a decision on its relevance to a posed query. Reading one part of a document may be enough for judging whether the document is relevant to a query or not. Reading two or more parts can enhance the confidence of the decision. These observations have led to our belief that perceiving objects from two or more perspectives is always an advantage.

In order to explore the applicability of the idea of multi-perspective descriptions in the field of information retrieval, we have proposed the *multi-perspective representation* (MPR) method. Traditionally in information retrieval, a document is associated with

one and only one representation. However, in the MPR method, it is assumed that a document can be observed from different perspectives to obtain its multiple *derivatives* (which we call the *perspective documents*). The *original document*, also called the *origin*, can then be represented by all the perspective documents that are derived from it. We classify any solution that uses multi-perspective representations of a document as a *multi-perspective representation solution.*

2 A Generalized MPR Solution

This section presents a generalized solution to the multi-perspective representation model. This solution is based upon the vector space model and its basic strategies are described as follows. Each document is perceived from different points of view to obtain its perspective documents. These derivatives are then temporarily regarded as ordinary documents and analyzed by some vector space based IR method. At the completion of these processes, each perspective document will be assigned a value that indicates its similarity to the given query. Then the similarity values of the perspective documents, which are derived from the same origin, are combined together to arrive at one single similarity value for the original document. In other words, by exploiting the perspective documents as *intermediates*, the analysis of an original document can be indirectly constructed by integrating the results obtained from the analysis conducted on its derivatives.

The following sections provide a guideline of how this solution can be developed as well as presenting some sample implementations that can facilitate the understanding of the proposed methodology.

2.1 Guidelines for Implementation

2.1.1 Obtain Multi-perspective Representations

In order to apply the MPR method on the vector space model, it is essential that we have a look into how to generate distinct perspective documents from an original document and how to obtain their vector representations accordingly. We call the vector representation of a perspective document the *perspective document's term vector*, or *perspective term vector* for short. The following paragraphs may offer you some idea on how this process can be implemented.

We split each original document into several perspective documents, where each perspective document conveys partial information of the original document. We do not have a fixed solution for creating the sub-files. The only requirement that we have is that the created perspective document should contain enough information for representing its origin.

When we analyze different pictures taken of the same object in order to construct a visual perception of the full dimensions, we are only interested in the ones that share some commonalities. The concept of perspective documents is similar to the above picture example, thus we suggest that perspective documents derived from the same origin should contain overlapped content as well.

Having generated perspective documents from each original document, the perspective documents can then directly obtain their vector representations in the document space by following the indexing procedure of the vector space model (page 12).

2.1.2 Combination Strategies

To effectively reconstruct the analysis for each original document through the intermediates' contribution, we have to employ a combination strategy. Two sample schemes are presented as follows.

The first scheme is called *multi-perspective representation average scheme*, or MPR-A for short. It exploits a direct strategy, which takes the average of the similarity values that are evaluated between a query q and any of the perspective documents that are used to represent the original document D_j. Let us denote D_{j_i} to be the perspective document of original document D_j from the ith perspective, denote p to be the number of different perspectives applied to D_j. We define the similarity between query q and original document D_j as:

$$sim(D_j, q) = \frac{\sum_{i=1}^{p} sim(D_{j_i}, q)}{p} \tag{9}$$

The second strategy is called *multi-perspective representation voting scheme*, or MPR-V for short. It is based on the idea of interpreting the original document D_j as a cluster and its associated perspective documents as the member documents in this cluster. The retrieval process can be converted into a classification problem by regarding the query to be the unlabeled document and its most relevant document to be the cluster the unlabeled document belongs to. We can interpret the similarity between a member document in a cluster and the query to be the probability that this member document votes for the query to be labeled by this class. We can then use the noisy-or operation to combine the similarity values of all the member documents in one cluster to arrive at one single value per class [CH88]. Accordingly, the similarity between query q and original document D_j

is defined as:

$$sim(D_j, q) = 1 - \sum_{i=1}^{p}(1 - sim(D_{j_i}, q)) \tag{10}$$

We would like to point out that, the MPR-V scheme is associated with probabilities, whose value range is from 0 to 1. It is advisable that before you decide to employ this voting strategy, make sure that the similarity metric of the IR method outputs a value that is within range $[0, 1]$.

In addition to the two sample strategies we proposed above, there are many other possible solutions for integrating the results of assessment between a query and the intermediates to compute the similarity value between the query and the original document. Making a comparison of the effectiveness of all the alternatives is beyond the scope of this thesis. (For more alternatives, refer to [RSW+98].)

2.2 Applications

We have applied the multi-perspective document representation method on two vector space model related IR approaches: the standard latent semantic indexing method and the standard term vector method.

As described in Chapter II, the LSI method performs SVD to the standard term vector method (i.e., traditional vector space model) in order to derive an LSI space of k dimensions, where k is less than or equal to the full rank of the term-by-document matrix. It should be noted that when k is equal to the full rank of the matrix, the LSI method is equivalent to the standard term vector approach. From this perspective, we consider the standard term vector method to be a special case of the LSI method and we will use the standard LSI method as the representative for vector space based methods

to demonstrate the implementation of the MPR model.

In order to highlight the characteristics of our representation method, we will start with an introduction to a standard LSI retrieval system followed by an LSI-MPR retrieval system, which adds the multi-perspective representation idea onto the standard LSI method. The complexity analyses of the standard LSI system and the LSI-MPR system will be discussed in terms of the analysis on their components' complexities.

2.2.1 LSI Retrieval System

In the previous chapter, we presented the methodologies of the latent semantic indexing method. The following paragraphs will present a detailed description of its implementation, together with some important formulae for computing the vector representation for documents using SVD matrices.

A typical LSI retrieval system may consist of six major components. They are *parser*, *indexer*, *deriver*, *transformer*, *evaluator* and *ranker*. Fig. 3 illustrates the system's basic framework. We name a sub-system consisting of the first five components (the parts inside the dotted box of fig. 3), the *LSI base*.

Parser

Given a document collection, the parser conducts lexical analysis on each document and converts it into a set of terms. We exploit the technique of stop listing, which helps to remove the punctuation and common words in the files. A term list can then be generated from the remaining terms. This term list, also called the system vocabulary, will be applicable to the following processes.

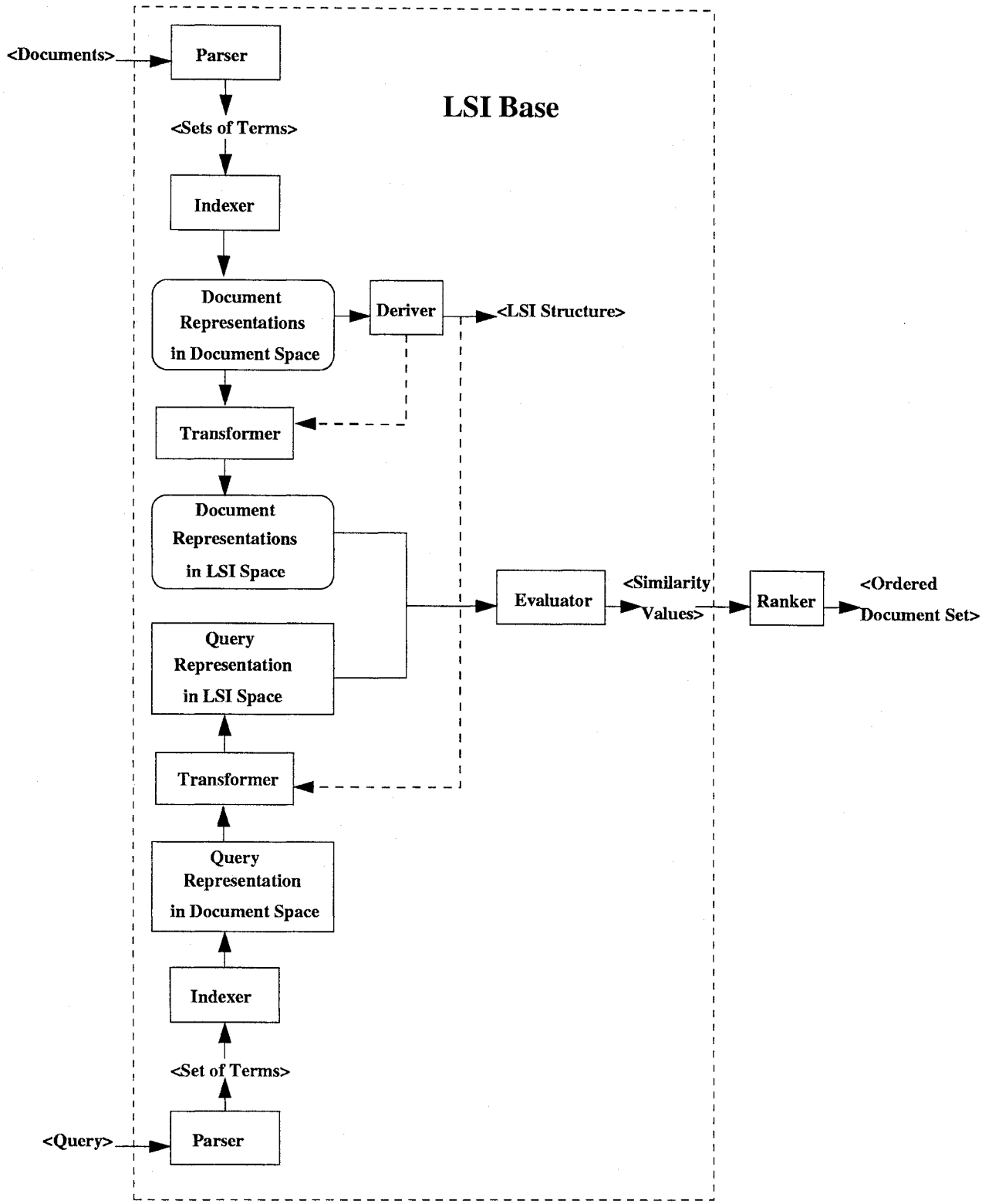

Fig. 3: LSI Retrieval System

Lexical analysis is expensive because it requires examination of every input character. Although no studies on the cost of lexical analysis in information retrieval systems have been done, lexical analysis has been shown to account for as much as 50% of the computational expense of compilation [FBY92]. The specific data structure our lexical analyzer utilizes is the *finite state machine.*

Indexer

In LSI, or any other vector space based approach, documents are represented in terms of vectors. The LSI system includes an indexing process to analyze the association between terms and documents. The indexer can simply apply raw frequency analysis or employ more complicated term weighting techniques. At the completion of indexing, each document is represented as a vector of terms. A term-by-document matrix is then constructed where each column corresponds to a term vector of a document.

The basic data structure that we employ for implementing the indexer is the C++ *standard template library* (STL) map class template. We use the terms as the keys for the map and store the statistical information of a given term in its related entry.

While the C++ standard does not specifically require that the map container be implemented using any specific data structure, the time complexity requirements imposed by the standard on each map operation suggest a balanced binary search tree. Many STL implementations use a red/black tree to implement a map. Map operations such as searching for an element or adding an element require $O(logn)$ operations [1] . Similarly, the time complexity for the major map operation in our implementation, *operator*[ ] is

[1]Retrieval Time: July 2005. http://www.mtsu.edu/~csjudy/STL/Map.html

$O(logn)$.

In STL, the operation of comparing two strings has been implemented, and the map container with a key of string type will be sorted alphabetically. Therefore, the indexer will output a list of terms in a dictionary format, which is beneficial. Another benefit of using STL map container is that the common operations in map take logarithmic time, which is very suitable for storing a collection of any size, thus it is an advantage for information retrieval systems.

Deriver

As we learned from Chapter II, LSI assumes that a latent semantic structure is underlying all the textual data and it can be estimated by some statistical techniques. In order to derive such a higher-order structure, which we call the *LSI structure* or *LSI space*, we apply the singular value decomposition algorithm, which has been generally applied in many of the LSI related approaches in the deriver.

SVD is a technique that highly demands considerable computing resources. Its algorithm has been well developed and implemented in a number of programming languages, such as Matlab [2] , C [BDO+96], C++ [3] and Java [4]. The experiments conducted by us have applied the svd function that is provided by the Matlab software. More precisely, we employ a function in Matlab called svds, to approximate the original term-by-perspective document matrix by truncating the components of the smallest singular values. Matlab SVD uses the LAPACK routines to compute the singular value decomposition and the

[2]Retrieval Time: July 2005. http://www.mathworks.com/access/helpdesk/help/techdoc/ref/svd.html

[3]Retrieval Time: July 2005. http://www.cs.utexas.edu/users/suvrit/work/progs/ssvd.html

[4]Retrieval Time: July 2005. http://jmat.sourceforge.net

solution will converge if the limit of 75 QR step iterations is exhausted.[5]

Although SVD is a very powerful tool for factor analysis, its computational complexity is too big to be ignored. This has become a bottleneck for the application of the information retrieval systems based on this technique.

Transformer

Having obtained the LSI structure by the deriver module, we exploit the transformer to obtain the document representations in the LSI space. There is a one-to-one correspondence between the representation in the document space and the representation in the LSI space. To implement the process of transforming, we project the vectors in document space onto the LSI space. The specific formulas that are used to complete the procedure are demonstrated as follows.

Let us start from the principle formula of SVD, as shown in Eq. 5 (page 20), and derive the formula that computes the representation of a document using the SVD components. Let U_k, S_k and V_k be the matrices that are approximate to U, S and V respectively. Let M_k be the approximate term-to-document matrix resulting from the product of U_k, S_k and V_k. M_k's rows correspond to terms and columns correspond to documents. Please note that the U_k and V_k are orthogonal matrices and S_k is a diagonal matrix. Due to the

[5]QR is a matrix decomposition method. Suppose M is an arbitrary rectangle matrix, whose size is $t \times d$. It can be factorized into QR matrices as: $M = QR$, where Q is a $t \times t$ orthogonal matrix and R is a $t \times d$ upper triangular matrix.

characteristics of these special matrices,[6] $M_k'M_k$ can be derived as follows:

$$\begin{aligned} M_k'M_k &= (U_kS_kV_k')'(U_kS_kV_k') \\ &= V_kS_k'U_k'U_kS_kV_k' \\ &= V_kS_k'S_kV_k' \\ &= V_kS_kS_k'V_k' \\ &= (V_kS_k)(V_kS_k)' \end{aligned} \tag{11}$$

From the above equations, it can be seen that the dot product between two columns of M_k reflects the extent to which two documents have a similar profile of terms. Observe that the matrix $M_k'M_k$ is a square symmetric matrix containing all the document-to-document dot products, i.e., the cell (i, j) of matrix $M_k'M_k$ equals to the dot product between the ith row and the jth row of the matrix V_kS_k. So we can consider rows of matrix V_kS_k as coordinates for documents.

Let e_j denote the jth canonical vector of $d \times d$ identity matrix. For a document j, the representation by the jth row of matrix V_kS_k is given by:

$$\begin{aligned} DR_j &= (V_kS_k)'e_j \\ &= S_k'V_k'e_j \\ &= S_kV_k'e_j \end{aligned} \tag{12}$$

The time complexity for loading the SVD matrices: U_k, S_k and V_k, will be $O(Max(t, d) \times k)$ (see page 21 for a more detailed description of these matrices). To transform one vector

[6]For an orthogonal matrix A: $AA' = A'A = I$, where I is an identity matrix and A' denotes the transpose matrix of A. For a diagonal matrix B: $B = B'$.

from the document space to the LSI space, we need to apply the multiplication operation on matrices V_k and S_k. This procedure costs $O(d \times k)$ time.

Process Queries

Up to this point, all the documents have been pre-processed and represented by vectors in the LSI space. The next step is to generate similar representations for query statements. Like in most IR systems, a query is considered as a pseudo-document in the LSI retrieval system. The process of analyzing a query is very close to the one applied to documents, however, there are still some differences between these two processes in our LSI system that are worth mentioning.

When a query comes, it is first processed by the parser to obtain a set of terms. These terms are compared to the system vocabulary established when the documents are parsed. We keep the terms that are accepted by the system environment and simply discard the ones that never appeared before. The indexer then takes the filtered set of terms as input and outputs its representation as a vector of terms.

When the transformer was used to compute the documents' representations in the LSI space, each document corresponds to a point in the LSI space and its coordinates can be computed in terms of the SVD component matrices. A query, however, can be a completely new textual object that was not considered in the original analysis. Thus we need to find a way to fold the query into the system.

Suppose there is a column vector V_{k_q} in matrix V_k which corresponds to query vector

q. Then the representation of V_{k_q} can be given as follows:

$$\begin{aligned} q &= U_k S_k V'_{k_q} \\ q' &= V_{k_q} S_k U'_k \\ V_{k_q} &= q'(U'_k)^{-1} S_k^{-1} \\ V_{k_q} &= q' U_k S_k^{-1} \end{aligned} \tag{13}$$

Apply Eq. 12 (page 42), the representation for query q can be obtained by

$$\begin{aligned} DR_q &= S_k V'_{k_q} \\ &= S_k (q' U_k S_k^{-1})' \\ &= S_k (S_k^{-1})' U'_k q \\ &= U'_k q \end{aligned} \tag{14}$$

By exploring the algebraic attributes of the orthogonal and diagonal matrices, we can find a way to represent a new document based on the existent LSI structure. Thus given any document, whether or not it exists in the database, we will be able to represent it in a standard format.

Evaluator

Once vectors have been computed for the query and for all the documents in the collection, the next target is to assess the similarity between a pair of vectors, most often, a query's term vector and a document's term vector. The evaluator is designed for this purpose.

The evaluation criterion that we use is the cosine similarity (page 16). Denote DR_j to be the vector representation for document D_j in the LSI space, DR_q to be the query

vector representation in the same space, and θ_j to be the angle between the above two vectors. By applying the Eq. 12 (page 42) and Eq. 14 (page 44), the similarity between D_j and q can be estimated by

$$\begin{aligned} sim(DR_j, DR_q) = \cos\theta_j &= \frac{DR_j' DR_q}{|DR_j||DR_q|} \\ &= \frac{(S_k V_k' e_j)'(U_k' q)}{|S_k V_k' e_j||U_k' q|} \\ &= \frac{(e_j' V_k S_k)(U_k' q)}{|S_k V_k' e_j||U_k' q|} \end{aligned} \tag{15}$$

At the completion of the evaluation process, each of the documents in the collection will be assigned a similarity value indicating its similarity (i.e., relevance) to the given query.

Suppose all d documents in the collection and the query have been represented in the LSI space by k-dimensional vectors. The time complexity for the evaluator to estimate the similarity between a query and all the documents in the collection is $O(d \times k)$.

Ranker

In order to allow the human user to restrict his/her attention to a set of documents of manageable size, the output of the retrieval system needs to be an ordered set of documents, with the most relevant documents appearing prior to the least relevant ones. This is why we have introduced the ranker component to our system.

The ranker employs a ranking strategy, called nearest-neighbour queries (page 17) on all the similarity values assigned to all the documents. The documents can be sorted accordingly and the ones with top N ranks will be selected, which are considered to have a greater potential of meeting the user's information request.

The specific sorting algorithm we employ is the STL sorting method. The time complexity is $\Theta(N \times log(N))$ for average case and $O(N \times N)$ for the worst case. Although the cost is very high for the worst scenarios, it is very rare too. In general, the algorithm has been very efficient, where the average complexity applies.

It should be noted that no matter how good an IR method can be, there is no guarantee that all the retrieved documents are relevant nor is it certain that all the relevant documents will be returned.

2.2.2 LSI-MPR Retrieval System

In order to integrate the concept of perspective documents and to reconstruct the analysis for each original document based on the analytical results conducted on its derivatives, the LSI-MPR system introduces two new components onto the standard LSI system: the *observer* and the *combiner*. The structure of the new system is depicted in fig. 4.

Observer

The observer aims at generating perspective documents for an ordinary document collection. It perceives each document in the original collection from different perspectives and outputs a derived collection consisting of all the perspective documents.

Having observed the fact that perspectives of the same object should share some properties, we make sure that in the process of deriving multiple variants from the same document, overlapped content exists among all of its associated derivatives, i.e. the perspective documents. There are many possible solutions for implementing an observer

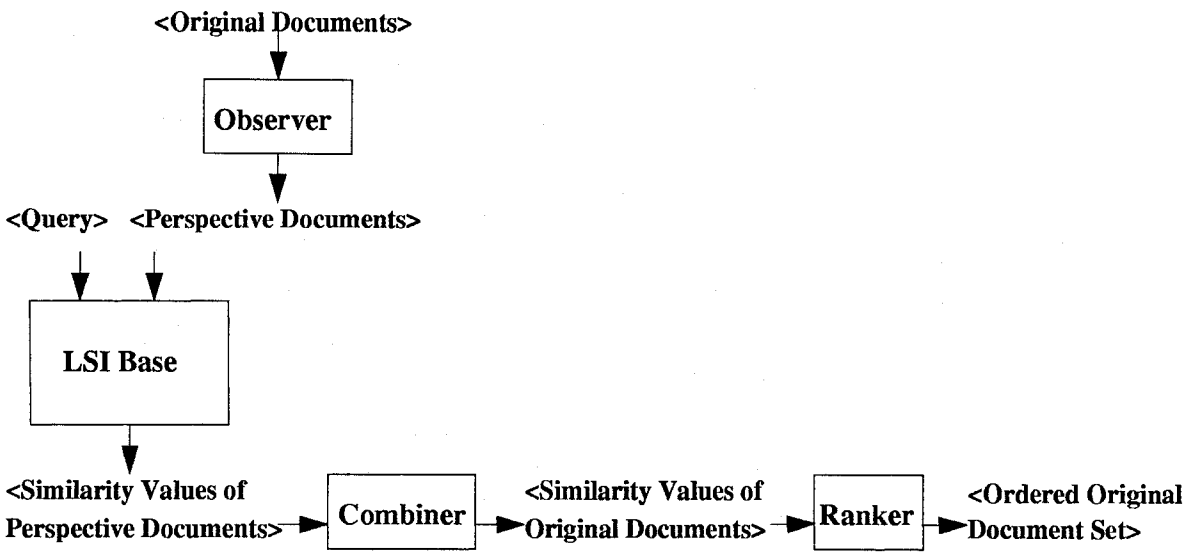

Fig. 4: LSI-MPR Retrieval System

component that meets the requirement for our guideline in Section 2.1.1.

A very straightforward idea is to collect abstracts of each original document from different individuals and regard different versions of abstracts to be the different perspective documents for the original document. This solution rigidly follows the linguistic definition of the word *perspective*, i.e., the interrelation in which a subject or its parts are mentally viewed [MW98] and it certainly makes sure that all the derived documents share some properties. However, it involves a lot of human interference, which conflicts with our expectation of developing a fully automated system.

Another solution that leads to a much wider application is to evenly distribute the content of each original document to its variants. We introduce a definition of *overlapping rate*, denoted by r, to represent the quota of the overlapped content in one perspective document. r can be defined as the ratio of the overlapped content in any perspective document to the entire content of this perspective document.

In the LSI-MPR system we have developed, each original document is divided by sentences and it is observed by p different perspectives. The time complexity for the observer to process the original collection and to generate the perspective document collection is $O(d \times p)$.

LSI Base

The LSI base is an abstraction for the sub-system of the LSI system we discussed earlier (Section 2.2.1). It includes the components of parser, indexer, deriver, transformer and evaluator. In our LSI-MPR system, the LSI base takes the collection of perspective documents generated by the observer and the user's query as input. It considers perspective documents to be ordinary documents and utilizes the parser, indexer, deriver and transformer to conduct analysis on them. It also exploits the evaluator to compare these perspective documents with the query. This leads to the output of the similarity values for the perspective documents.

Combiner

Having obtained the results from the analysis conducted on the perspective documents, the combiner employs a combination strategy on the intermediate results. Examples were presented in Section 2.1.2. By doing so, it outputs the similarity values that indicate the relevance of the original documents to a given query.

The time complexity for the combiner to compute the similarity values for all the original documents from their intermediates' similarity values is $O(d \times p)$.

Ranker

The ranker of the LSI-MPR system has the same functionality as LSI's. Please refer to page 45 for further details.

3 Differential Latent Semantic Indexing Method

We discovered that in the literature, the *differential latent semantic indexing* (DLSI) method, proposed by Chen *et al.*, exhibits attributes that are falling into the category of multi-perspective representation. In this section, we will provide a brief overview of this method.

3.1 Motivation

As we mentioned in Chapter II, the standard latent semantic indexing method applies a dimension reduction method SVD to the traditional vector space model and has shown to have a distinct advantage in dampening the effect of synonymy and polysemy problems. However, it also has some drawbacks that should not be neglected.

In the LSI method, the term vector of each document is projected onto the same reduced dimensional space. The projection is measured and used as the representation of the document, whereas the distance from the term vector to the reduced space is neglected. Fig. 5 illustrates the process of projecting. Accordingly, to evaluate the similarity between a pair of documents, the LSI method only measures the similarity between their term vectors' projections. As pointed out by Schutze *et al.* [SS97], the LSI method is indeed a global dimensional reduction (global projection) approach and because of this, it encounters a difficulty in adapting to the unique characteristics of each

document. The DLSI method was proposed in order to overcome this disadvantage of the standard LSI method. Besides projection of term vectors on a reduced dimensional space, the DLSI method also makes use of the distance from these vectors to the reduced space.

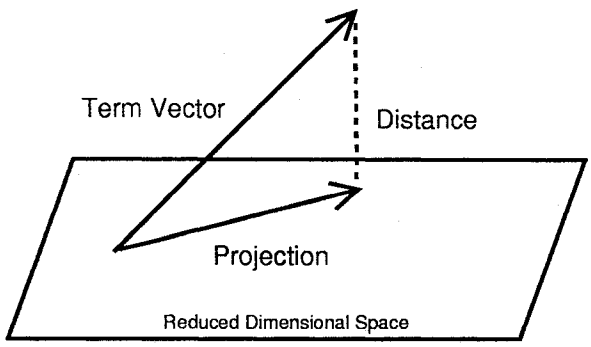

Fig. 5: Projecting a Vector

3.2 Theory

3.2.1 Differential Term Vector

Like the generalized solution strategy, the DLSI method assumes that each document is associated with multiple perspective documents. It introduces the concept of *differential term vector*. Denote DR_{i_j} to be the vector representation of a perspective document D_{i_j} (i.e., the jth perspective document of original document D_i). A differential term vector is defined as $DD = DR_{i_{1j_1}} - DR_{i_{2j_2}}$, where $DR_{i_{1j_1}}$ and $DR_{i_{2j_2}}$ are term vector representations of two perspective documents $D_{i_{1j_1}}$ and $D_{i_{2j_2}}$. In particular, when these two perspective documents are distinct derivatives of the same original document, DD is called an *intra differential term vector*; when they are distinct perspective documents

having different origins, DD is called an *extra differential term vector.*

The way to select two perspective documents for constructing a differential term vector can be arbitrary. A pragmatic solution is called *mean vector strategy.* A *mean vector* takes the average of the perspective documents' vectors that have the same origin. Denote S_i to be the mean vector of original document D_i. It can be defined as:

$$S_i = \frac{\sum_{j=1}^{p} DR_{i_j}}{p} \tag{16}$$

By introducing mean vectors, an intra differential term vector can be constructed by $D_{i_j} - S_i$ and an extra differential term vector can be constructed by $D_{i_{1_j}} - S_{i_2}$ $(i_1 \neq i_2)$. Please note that S_i is derived from the perspective documents' term vectors of the original document D_i, thus it is also considered as a perspective term vector of D_i and can be used for constructing the differential term vectors.

The intra and extra *differential term-by-document matrices*: M_I and M_E can be defined accordingly, each column of which is a differential intra or extra term vector respectively.

3.2.2 Posterior Model

DLSI applies SVD on both M_I and M_E matrices and obtains the intra and extra DLSI spaces. In order to estimate the similarity between a query q and a document D_i, we can construct a differential term vector x by assigning $q - D_{i_j}$ or $q - S_i$ to it, a likelihood function can be evaluated on both DLSI spaces, which exploits the vector's projection on and its distance to the space. A Bayesian posterior function then combines the likelihood values in both cases to arrive at an estimate of the similarity between the

two term vectors that are used to construct the differential term vector.

3.2.3 Geometric Interpretation

In the DLSI method, both the differential term vector and the likelihood function convey a rich geometric sense. As mentioned earlier, a differential term vector DD can be obtained by conducting subtraction over two different term vectors $D_{i_{1j_1}}$ and $D_{i_{2j_2}}$. In the likelihood function, the length of DD is exploited, which is approximately equivalent to the cosine value of the angle between $D_{i_{1j_1}}$ and $D_{i_{2j_2}}$, assuming that normalization has been applied to all the term vectors. In addition, the distance of DD to the DLSI space is also taken into consideration, providing a representation of the characteristics of an individual document, which LSI method is unable to offer.

3.2.4 Algorithm

The implementation of DLSI consists of two major components. One is to process the document collection and to set up the retrieval system. The other one is to apply the retrieval system on a query and to obtain an ordered document set accordingly. The algorithms of the procedures are shown as follows:

Set up System

(1) Generate perspective documents for each original document in the database and obtain their term vector representations.

(2) Construct an intra differential term-by-document matrix M_I such that each of its columns is an intra differential term vector.

(3) Decompose M_I, by the SVD algorithm, into $M_I = U_I S_I V_I^T$ where $S_I = diag(\delta_{I,1}, \delta_{I,2}, \dots)$. Find an appropriate k_I[7] and apply it to M_I in order to get an approximate matrix M_{I,k_I}, where $M_{I,k_I} = U_{k_I} S_{k_I} V_{k_I}^T$. Then evaluate the likelihood function [CTN03]:

$$P(x|M_I) = \frac{n_I^{1/2} exp(-\frac{n_I}{2} \sum_{i=1}^{k_I} \frac{y_i^2}{\delta_{I,i}^2}) exp(-\frac{n_I \varepsilon^2(x)}{2\rho_I})}{(2\pi)^{n_I/2} \prod_{i=1}^{k_I} \delta_{I,i} \cdot \rho_I^{(r_I - k_I)/2}}, \tag{17}$$

where $y = U_{k_I}^T x$, $\varepsilon^2(x) = (\|x\|)^2 - \sum_{i=1}^{k_I} y_i^2$, $\rho_I = \frac{1}{r_I - k_I} \sum_{i=k_I+1}^{r_I} \delta_{I,i}^2$, and r_I is the rank of matrix M_I.

(4) Construct an extra differential term-by-document matrix M_E such that each of its columns is an extra differential term vector.

(5) Decompose M_E, by the SVD algorithm, into $M_E = U_E S_E V_E^T$, $S_E = diag(\delta_{E,1}, \delta_{E,2}, \dots)$. Find an appropriate k_E, and apply it to obtain an approximate matrix M_{E,k_E}, where $M_{E,k_E} = U_{k_E} S_{k_E} V_{k_E}^T$. Then calculate the likelihood function [CTN03]:

$$P(x|M_E) = \frac{n_E^{1/2} exp(-\frac{n_E}{2} \sum_{i=1}^{k_E} \frac{y_i^2}{\delta_{E,i}^2}) exp(-\frac{n_E \varepsilon^2(x)}{2\rho_E})}{(2\pi)^{n_E/2} \prod_{i=1}^{k_E} \delta_{E,i} \cdot \rho_E^{(r_E - k_E)/2}}, \tag{18}$$

where $y = U_{k_E}^T x$, $\varepsilon^2(x) = (\|x\|)^2 - \sum_{i=1}^{k_E} y_i^2$, $\rho_E = \frac{1}{r_E - k_E} \sum_{i=k_E+1}^{r_E} \delta_{E,i}^2$, and r_E is the rank of matrix M_E.

(6) Define the posterior function [CTN03] as

$$P(M_I|x) = \frac{P(x|M_I)P(M_I)}{P(x|M_I)P(M_I) + P(x|M_E)P(M_E)}, \tag{19}$$

[7]The way how to choose a value k is not fixed. Although reduction in k can help removing noise, keeping too few dimensions may lose important information. Therefore, only by applying experiments on a certain data set and observing its performance on different ks, can we find the most appropriate value.

where $P(M_I)$ is set to $1/d$, and d is the number of original documents in the collection, and $P(M_E)$ is set to $1 - P(M_I)$.

Automated Document Search

(1) Given a query, set up its term vector. For each original document in the collection, repeat step (2) to (5) below.

(2) Construct a differential term vector x by subtracting the query's term vector by any of the perspective term vectors or the mean vector of the original document.

(3) Evaluate the intra document likelihood function $P(x|M_I)$ and the extra document likelihood function $P(x|M_E)$.

(4) Calculate the Bayesian posterior probability function $P(M_I|x)$.

(5) Conduct ranking strategy over the $P(M_I|x)$ values for all the original documents and return the most relevant ones.

4 Use MPR Solutions for Classifiers

In addition to the application on information retrieval systems, the multi-perspective representation method can also be utilized by classifiers. On page 31, we discussed about the close relationship between a retrieval task and a classification problem, and pointed out that an IR method can be applied to set up a classifier as well. In particular, there exists a direct classifiable interpretation of our multi-perspective representation model. This statement can be verified as follows.

In a typical classification data set, there are many different classes consisting of multiple different member items, and all the member items from the same class share some common attributes. This can be interpreted by the MPR method as: in this data set, there exist different documents (classes), and each of them is associated with multiple distinct perspective documents (member items from the same class) that are used to represent it. By doing so, the MPR method can be easily extended to the field of document classification and thus greatly widens the fields of its application.

Chapter IV

Experiments

This chapter focuses on presenting the experiments that have been conducted using the multi-perspective representation solutions we provided in Chapter III. The basic research procedures will be demonstrated. The experimental results as well as their evaluation will also be covered.

1 The Generalized MPR Solution Field Test

This series of experiments intends to conduct field tests over the generalized MPR solution, to apply it on the standard LSI and the standard term vector method and to evaluate its performances.

1.1 Document Collections

We conducted our experiments on two standard IR document collections, TIME [8] and ADI [9] where queries and relevance judgement are readily available. The TIME collection consists of articles from Time magazine's world news section in year 1963. ADI is a test collection of document abstracts from the library science and related areas. Table 3 shows some characteristics of these data sets. The vocabulary size refers to the number of terms that are accepted by the system. A document size is the total number of terms in this document. In cases where a term appears more than once, all its occurrences are counted.

[8]Retrieval Time: July 2005. http://www.cs.utk.edu/~lsi/

[9]Retrieval Time: July 2005. http://www.cs.utk.edu/~lsi/

Table 3: Characteristics of TIME & ADI

	TIME	ADI
number of documents	425	82
number of queries	83	35
average number of documents relevant to a query	4	5
vocabulary size	20959	1513
average document size	588.12	66.72

1.2 Procedure

Firstly, we set up an LSI retrieval system using the standard LSI method. Then we construct two LSI-MPR retrieval systems by integrating the multi-perspective representation into the LSI system. We call the variant using the MPR-A strategy the LSI-MPR-A system and the variant using the MPR-V strategy the LSI-MPR-V system (refer to page 35 for details of the MPR strategies). Following that, we conducted experiments on both TIME and ADI collections.

1.2.1 LSI Retrieval System

In the LSI system, the parser applies the SMART stop list to eliminate the stop words from the documents. The SMART stop list is a negative dictionary that is used in the vector space model based IR system SMART [Sal71].

The indexer then analyzes the output of the parser to generate the indexing matrix,

i.e., the term-by-document matrix. It is worth mentioning that in the literature, experiments have been done using the standard LSI method on the same test collections by Dumais [Dum91]. She not only removed the words that occur in the SMART stop list, but also eliminated the ones that occur only once in the collection. In order to make a fair comparison to her system, we followed her way of constructing effective terms. Consequently, in the process of indexing, only the terms that occur more than once are used to construct the term-by-document matrix. As well, Dumais employed the raw term frequency strategy for local term weighting and assigned the global weight to be 1 for all the terms. Therefore our indexer employs the same strategies.

The deriver performs SVD algorithm on the term-by-document matrix that is generated by the indexer in order to build up the LSI structure. As presented in Eq. 4 (page 20) and Eq. 5 (page 20), SVD provides a simple scheme to approximate the original term-by-document matrix with a smaller matrix of rank k. The amount of dimension reduction, i.e., the choice of k, is a very critical parameter to our experiments. Ideally, we want a value of k that is large enough to fit all the real structure in the data, but small enough so that the sampling error or noise data are filtered. However, the proper way to make such choices remains an open issue in the factor analytic literature. In practice, we are using an operational criterion. That is, a value of k which yields the best retrieval performance is the best. A rule that most researchers follow is to assign value k to be a certain percentage (e.g., 10%) of the full rank of the original term-by-document matrix, or assign it to be a value between [100, 300] for very large collections. To provide a proper representation of the dimension reduction, we introduce a term *dimension reduction rate*, which is defined as the dimension of the dimensional reduced LSI space (i.e., the value

k) over to the full rank of the original indexing matrix.

Based upon this higher-order structure, the transformer uses Eq. 12 (page 42) to construct the document representation in the LSI space by using the SVD components.

Each query in the query list undergoes similar processes to obtain its representation in the LSI space (Eq. 14, page 44). Then the evaluator employs cosine similarity metric (Eq. 15, page 45) to compute the similarity values for each document and the ranker outputs the documents in an order that the most relevant ones come prior to the least relevant ones.

1.2.2 LSI-MPR Retrieval Systems

To set up the LSI-MPR retrieval systems, we reuse the LSI base of the standard LSI system, and add in the observer, combiner and ranker components.

In the observer, we apply two different perspectives (i.e., $p = 2$) to observe each original document. Due to the fact that both collections consist of relatively short articles and no abstracts are available, the observer constructs the perspective documents in the following way. Each original document is divided by sentences. Denote o to be the number of overlapped sentences, and p to be the number of different perspectives. For every $o+p$ sentences in the document, o sentences are shared by all the perspective documents and the other p sentences are evenly distributed to the p perspective documents. Please note that the sizes of the perspective documents associated with the same origin do not have to be exactly the same. Thus the overlapping rate r, which is used to represent the quota of the overlapped content in the perspective document, equals to $\frac{o}{o+p}$.

For example, there is an original document D_j that has 7 lines, we assign $o = 1$ and

$p = 2$. The method to distribute D_j's content into its perspective documents: D_{j_1} and D_{j_2} is demonstrated as follows:

D_j: Line 1. Line 2. Line 3. Line 4. Line 5. Line 6. Line 7.

D_{j_1}: Line 1. Line 2. Line 4. Line 5. Line 7.

D_{j_2}: Line 1. Line 3. Line 4. Line 6. Line 7.

In order to make sure all perspective documents have reasonable sizes (i.e., contain enough information to represent their origin), we set $r = \frac{1}{2}$ for TIME collection and $r = \frac{5}{7}$ for ADI collection.[10]

In the parsing process, the LSI-MPR systems are implemented in a way that is a bit different from the LSI system. In order to keep a consistent system vocabulary for both systems, we filter the perspective documents with the effective term list that is generated by the standard LSI system. In other words, we only keep the words that are not in the stop list and the ones that occur more than once in the original document collection. The reason why we do not try to set up a different system vocabulary for the LSI-MPR systems is that we realize in the observer, we introduced overlapped content in the perspective documents collection, therefore, such a scenario becomes possible, that a word only occurs once in the original document collection might occur more than one time in the derived collection if it is used as the overlapped parts and is assigned to all the perspective documents having the same origin.

The LSI base processes the perspective document collection and the query to output the results of the relevance judgement analysis conducted on the perspective documents.

[10]For TIME collection, $o = 2$ and $p = 2$. For ADI collection, $o = 5$ and $p = 2$.

These intermediate results are then analyzed by the combiner, which exploits a combination strategy (Eq. 9, page 35 or Eq. 10, page 36) to integrate the similarity values of the associated perspective documents. It then outputs the reconstructed similarity values for the original documents.

The ranker then simply applies a sorting operation over the list of similarity values and according to that, an ordered set of original documents can be obtained.

1.3 Results and Evaluation

In our experiments, the criteria employed for evaluating the retrieval performance of the IR methods is the interpolated average precision (see page 29 for details). Following Dumais' report, we used three standard recall levels: 0.25, 0.50 and 0.75.

1.3.1 Standard LSI vs LSI-MPR

In order to compare the retrieval performance of the LSI method using traditional representation model and the multi-perspective representation model, we conducted experiments on the LSI system and the LSI-MPR-A, LSI-MPR-V systems with different dimension reduction rates (i.e., the dimension of the reduced space over the dimension of the rank of the term-by-document matrix, page 59): $0.1, 0.2, \cdots, 0.9$. The results on TIME and ADI collections are shown in fig. 6 and fig. 7 respectively, indicating that the retrieval performance of the LSI approach can be significantly improved by using the multi-perspective document representation method.

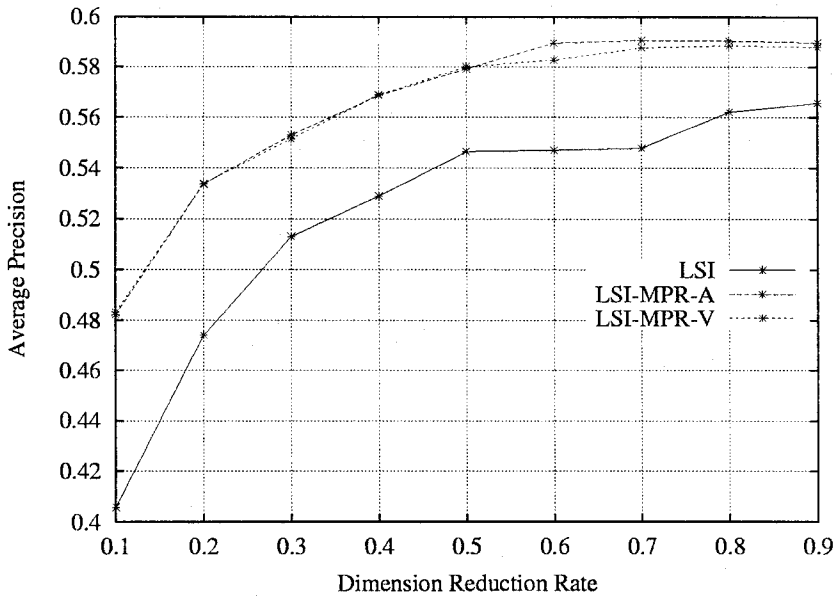

Fig. 6: Retrieval Performances on TIME using LSI, LSI-MPR-A and LSI-MPR-V Methods

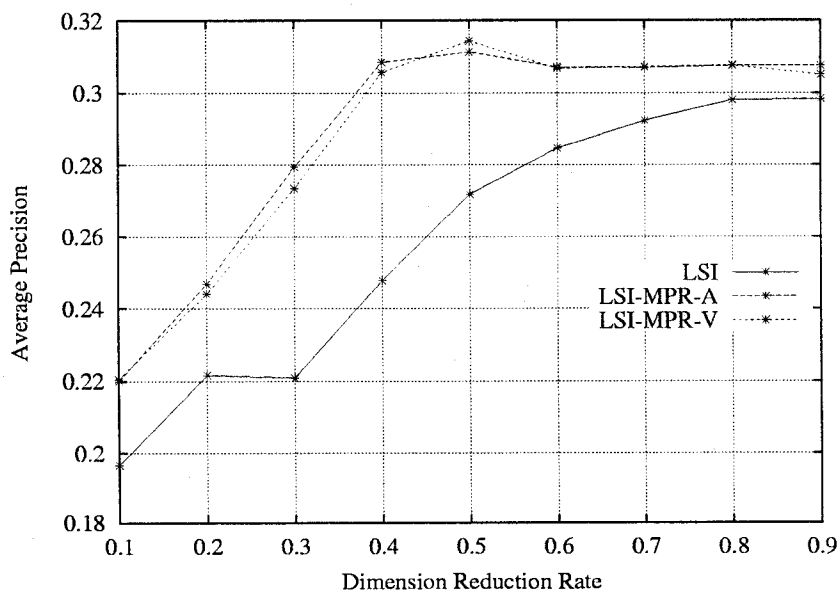

Fig. 7: Retrieval Performances on ADI using LSI, LSI-MPR-A and LSI-MPR-V Methods

Table 4: Retrieval Performances of Term Vector Approach using Traditional Representation Model & Multi-perspective Representation Model

Collection	Approach	Average Precision
TIME	standard term vector approach	57.92%
	term vector approach with MPR-A	58.95%
	term vector approach with MPR-V	58.87%
ADI	standard term vector approach	28.28%
	term vector approach with MPR-A	30.95%
	term vector approach with MPR-V	30.66%

1.3.2 Standard Term Vector Approach vs Term Vector Approach with MPR Schemes

As stated on page 36, we consider the standard term vector method to be a special case of the LSI method when the dimension of reduced space equals to the rank of the term-by-document matrix, i.e., the dimension reduction rate is 1.0. Table 4 demonstrates that the performance of term vector method can be greatly improved by employing the multi-perspective document representations.

1.3.3 Statistical Analysis

We have conducted statistical t-tests to examine if the improvement obtained from the multi-perspective representation model remains to be significant in order to ensure the significance of the results. Based upon the assumption that the difference between

the precision values obtained through the standard LSI/term vector approach and the LSI/term vector approach with the MPR model conforms to normal distribution, we exploit the tripled precision values for all queries at all standard recall levels on each dimension reduction rate using the standard LSI/term vector approach, and the LSI/term vector approaches with MPR schemes. The paired sample t-test has been applied to test the null hypothesis that the standard LSI/term vector approach and any of the two LSI/term vector approaches with MPR schemes do not have different performances on any dimension. The levels of significance, which are used to indicate whether the improved retrieval performance of our representation model has occurred merely by chance, are shown in fig. 8 and fig. 9. From the figures we can see that, the improvement of the multi-perspective representation schemes are significant in most cases.

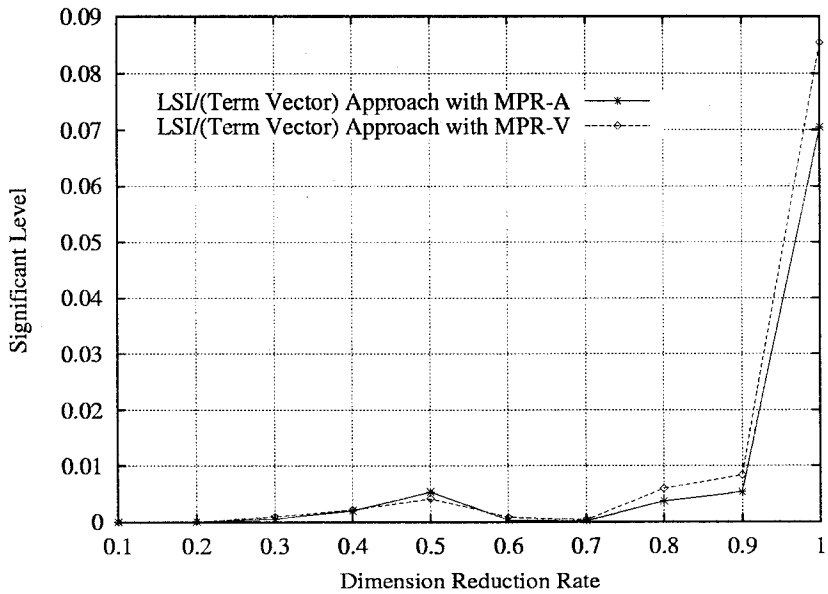

Fig. 8: Significance Level of LSI/(Term Vector) Approach with MPR Schemes on TIME

Furthermore, we have conducted t-tests on the tripled samples for all the dimension

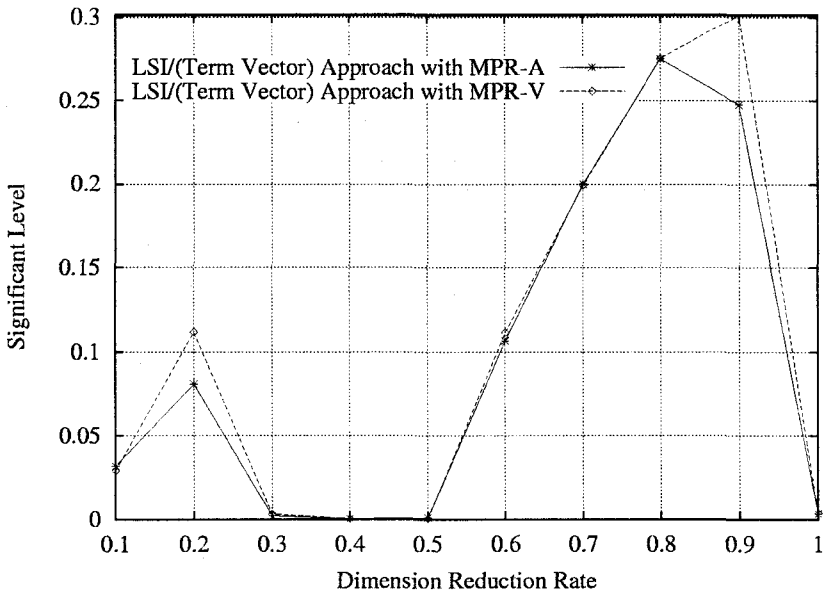

Fig. 9: Significance Level of LSI/(Term Vector) Approach with MPR Schemes on ADI

reduction rates we employed. The results of statistical analysis are demonstrated in table 5. The statistics of the significance levels (way beyond the 10^{-23} level and the 10^{-7} level for TIME and ADI respectively), have shown that the improved performances of the multi-perspective representation schemes are significant for both LSI and term vector approaches. By considering the overall performance in either the TIME or the ADI collection, the probability that the improvements come about by chance is extremely small. Consequently, we can claim that, the improvements gained through using the multi-perspective representation solutions are statistically significant.

Table 5: Significance Levels of t-tests on all the Samples on TIME & ADI

Significance Level	TIME	ADI
improved performances of LSI/(term vector) approach with MPR-A over the standard LSI/(term vector) approach	7.95×10^{-26}	4.29×10^{-9}
improved performances of LSI/(term vector) approach with MPR-V over the standard LSI/(term vector) approach	2.13×10^{-24}	1.13×10^{-8}

2 The DLSI Approach Field Test

2.1 Problem Statement

The differential latent semantic indexing method has been applied in the field of information retrieval and document classification. Experiments have shown that it demonstrates a significant improvement over the standard LSI method [CTN01, CTN03]. The intention of our experiments with the DLSI method is to test its effectiveness and explore its applicability in medical data classification problem.

We conducted our study on a well-known medical data set: *wisconsin breast cancer data* (WBCD), which is often used as a benchmark for testing effectiveness of classifiers [11] . WBCD is a breast cancer sample collection periodically collected by Dr. Wolberg in his clinical cases. Each sample has been assigned a vector of 9 elements, all of which are in the interval of 1 to 10, with value 1 corresponding to a normal state and 10 to the

[11]Retrieval Time: July 2005. http://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/cancer

most abnormal state [MW90].

The problem our classifier needs to address is how to classify an unlabeled data entry to one of the two classes: benign and malignant. The data set contains 699 entries, including 16 incomplete entries and 1 outlying entry and its class distribution is 65% for benign points and 35% for malignant points.

2.2 Procedure

In Chapter III, we provided an algorithm that applies the DLSI method to set up an information retrieval system. In order to utilize this algorithm in a classifier, we have to convert a classification process into a retrieval problem.

Suppose we have a data set of n classes, and for class i, it contains p_i member items. We can regard these classes as distinct original documents. The jth member entry in class i can be considered as a perspective document of original document i from the jth perspective.

We separated the entire WBCD data set into two subsets that do not have intersection: a training set and a testing set. To set up the classifier, we process each member item in the training data set and construct its vector representation accordingly. We then employ the mean vector strategy (see page 51 for details) to construct intra and extra differential term-by-document matrices, which are then decomposed by the SVD algorithm.

For a testing data entry q, we assign x to be $q - S_i$, where S_i is the mean vector from cluster i. Follow Eq. 17 (page 53), Eq. 18 (page 53) and Eq. 19 (page 53), we calculate $P(x|M_I)$, $P(x|M_E)$ and $P(M_I|x)$. The cluster that has the largest $P(M_I|x)$ value is chosen as the cluster to which entry q belongs to.

2.3 Results and Evaluation

We conducted fields tests on different training sets. The ratio of the training set to the complete data set ranges from 20% to 80%. Fig. 10 depicts the performance of our classifier.

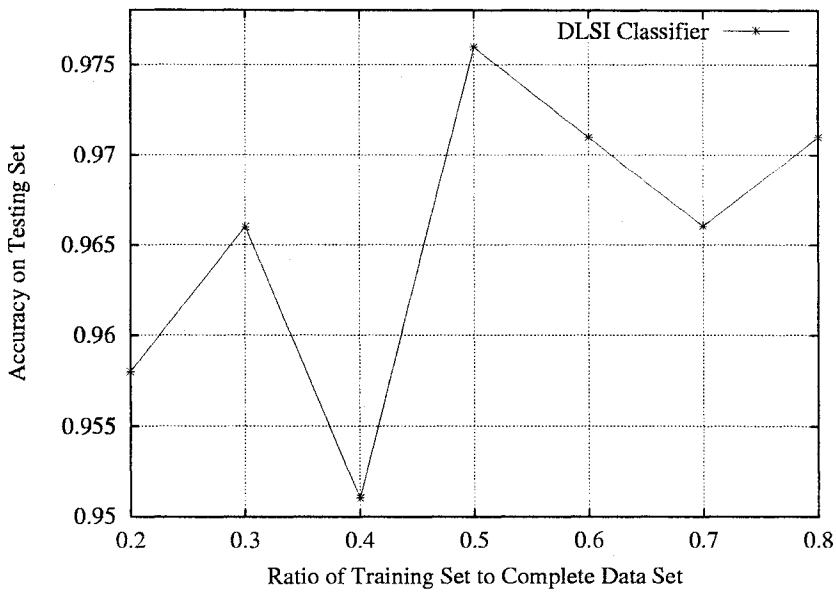

Fig. 10: Classification Accuracy of DLSI Classifier on WBCD

To date, much research work has been conducted on the WBCD data set. Methods such as rule generation approach, fuzzy-genetic approach, neural network approach have been presented in the literature [CZP04]. Taha and Gosh proposed a method using neural network to extract rules [TG96]. But it can be only applied to data with binary attributes. Setiono proposed a method based on the idea of finding a set of concise rules using a pruned neural network [Set96]. His classifier achieves good performance but the extraction of rules is manually processed, which involves much human intervention.

The DLSI method, however, can overcome this limitation. Our classifier is a fully

automated classification system and it obtains an average of 96.6% accuracy (derived from fig. 10). The best accuracy of 97.6% is obtained when we use 50% data for training. The classification performance still achieves 95.5% accuracy when only 20% of the data (i.e., 136 entries) are used for training. As far as we know, no known method has ever reached such a high performance with less than 200 training samples. The only comparable result using a small training set can be found in Wolberg's paper [WM90], where an accuracy of 93.5% is achieved by using 185 entries for training.

Chapter V

Conclusion

In this chapter, we will present a summary of the thesis as well as describing directions in our future research.

1 Summary

In this paper, we have proposed the multi-perspective representation (MPR) method, which is an innovative document representation model that imitates a human's manner of depicting objects, i.e., describing them from different perspectives. Two sample solutions were presented: a generalized MPR solution and the differential latent semantic indexing (DLSI) approach.

The generalized MPR solution is expected to be applied by many IR methods that are based on the vector space model. Experiments have been conducted employing the generalized MPR solution on the latent semantic indexing method and the term vector method. Two standard document collections (TIME and ADI) have been used in field testing. The experimental results and the statistical analysis conducted on them have demonstrated the effectiveness of the generalized MPR solution on both IR methods.

The DLSI method is an LSI-based IR approach proposed in the literature, which also exhibits the attributes of the MPR model. We have explored its applicability in an IR-related problem: medical data classification. Results from these experiments have demonstrated that the DLSI method achieves high performance on a benchmark medical data set (WBCD).

2 Future Work

It might be argued that the improved performance of the IR methods using the MPR model does not agree with a common understanding in IR, that retrieval performance normally increases with the size of the documents, because our MPR model proposes the use of perspective documents, which are necessarily smaller than their origin (the original document). We would like to point out that although the size of an individual perspective document is smaller than that of an ordinary document, the sum of all the perspective documents is equal to or greater than the original document. We also believe that the improvement of the MPR solution results from the final fusion process, which integrates all the perspective documents. We believe that the improvement of the MPR model should be more significant for longer documents than for shorter ones. The curves of significance level illustrated in fig. 8 and fig. 9 seem to support this idea (TIME's average document size is bigger than ADI's). However, since both TIME and ADI collections consist of relatively short documents, at our current stage of development, we still do not know whether the observed improvement resulting from the MPR model's contribution will decrease or even disappear as the corpus' size increases. Experiments on larger corpora will be necessary for further verification.

While presenting the generalized MPR solution, we provided some sample implementation strategies. We presented how we split the original document into multiple perspective documents and suggest an overlapped content should exist in all the perspective documents having the same origin. Yet, we do not know how different strategies of obtaining multi-perspective documents will affect the retrieval performance. Experiments

on more data collections may shed more light on these points.

We also introduced two sample combination strategies to reconstruct the analysis on the original document through its intermediates. Further research on other alternative strategies will be an interesting topic.

As well, using n-gram as a scheme for extracting terms might bring to the IR methods the advantages of language-independency as well as insensitivity to degraded text. In our future work, we can also explore the application of this scheme onto the MPR solutions.

References

[BDO+96] M. Berry, T. Do, G. O'Brien, V. Krishna, and S. Varadhan. Svdpackc (version 1.0) user's guide, 1996.

[CGD92] W. Cooper, F. Gey, and D. Dabney. Probabilistic retrieval based on staged logistic regression. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 198–210, Copenhagen, Denmark, 1992.

[CH88] W. Cohen and H. Hirsh. Text categorization using whirl. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 169–173, New York, 1988.

[CH95] A. Chakravarthy and K. Haase. Netserf: Using semantic knowledge to find internet archives. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, 1995.

[CTN01] L. Chen, N. Tokuda, and A. Nagai. Probabilistic information retrieval method based on differential latent semantic index space. *IEICE Trans. on Information and Systems*, E84-D(7):910–914, 2001.

[CTN03] L. Chen, N. Tokuda, and A. Nagai. A new differential lsi space-based probabilistic document classifier. *Information Processing Letters*, 88:203–212, 2003.

[CZP04] L. Chen, J. Zeng, and J. Pei. Classifying noisy and incomplete medical data by a differential latent semantic indexing approach. In *Data Mining in Biomedicine*. Springer Press, 2004. Accepted for publication.

[CZT05] L. Chen, J. Zeng, and N. Tokuda. A "stereo" document representation for textual information retrieval. *Journal of the American Society for Information Science and Technology*, 2005. Accepted for publication.

[Dam95] M. Damashek. Gauging similarity with n-grams. *Science*, 267:843–848, 1995.

[DDL+90] S. Deerwester, S. Dumais, T. Landauer, Furnas. G., and R. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41:391–407, 1990.

[Dum91] S. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23:229–236, 1991.

[Ear84] S. Earhart. *The UNIX Programming Language*, volume 1. New York: Holt, Rinehart, and Winston, 1984.

[FBY92] W. Frakes and R. Baeza-Yates. *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, 1992.

[GCT97] W. Greiff, W. Croft, and H. Turtle. Computationally tractable probabilistic modeling of boolean operators. In *Proceedings of the 20th Annual*

International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 119–128, 1997.

[Gre01] E. Greengrass. Information retrieval: A survey. Technical Report DOD Technical Report TR-R52-008-001, 2001.

[GWHWH+99] A. Graesser, K. Wiemer-Hasting, P. Wiemer-Hasting, R. Kreuz, and University of Memphis the Tutoring Research Group. Autotutor: A simulation of a human tutor. *Journal of Cognitive Systems Research 1*, pages 35–51, 1999.

[Har86] S. Harter. *Online Information Retrieval: Concepts, Principles, and Techniques.* Academic Press, Inc., 1986.

[Har92] D. Harman. User-friendly systems instead of user-friendly front-ends. *Journal of the American Society for Information Science*, 43(2):164–174, 1992.

[Har95] D. Harman. Overview of the third text retrieval conference. In *Third Text REtrieval Conference (TREC-3)*, pages 1–19. National Institute of Standards and Technology Special Publication 500-207, 1995.

[Hof99] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, 1999.

[Hul94] D. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *Proceedings of the 17th Annual International ACM*

SIGIR Conference on Research and Development in Information Retrieval, pages 282–291, 1994.

[Lee94] J. Lee. Properties of extended boolean models in information retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 182–190, 1994.

[LPY94] E. Liddy, W. Paik, and E. Yu. Text categorization for multiple users based on semantic features from a machine-readable dictionary. *ACM Transactions on Information Systems*, 12(3):278–295, 1994.

[Mon02] M. Montague. *Metasearch: Data fusion for document retrieval.* PhD thesis, Dartmouth College, Hanover, New Hampshire, 2002.

[MW90] O. Mangasarian and W. Wolberg. Cancer diagnosis via linear programming. *SIAM News*, 23(5):1–18, 1990.

[MW98] Merriam-Webster, editor. *Merriam-Webster's Collegiate Dictionary.* 10th edition, 1998.

[OC03] P. Ogilvie and J. Callan. Combining document representations for known item search. In *Proceedings of the Twenty Sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 143–150, Toronto, Canada, 2003. ACM.

[Pai84] C. Paice. Soft evaluation of boolean search queries in information re-

trieval systems. *Information Technology: Research and Development*, 3(1):33–42, 1984.

[PN96] C. Pearce and C. Nicholas. Telltale: Experiments in a dynamic hypertext environment for degraded and multilingual data. *Journal of the American Society for Information Science*, 47(4):263–275, 1996.

[Por80] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[Por97] M. Porter. An algorithm for suffix stripping. In K. S. Jones and Willett P., editors, *Readings in information retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., 1997.

[RGF90] K. Rose, E. Gurewitz, and G. Fox. A deterministic annealing approach to clustering. *Pattern Recognition Letters 11*, 11:589–594, 1990.

[Ril95] E. Riloff. Little words can make a big difference for text classification. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 130–136, 1995.

[RSW+98] B. Rehder, M. Schreiner, M. Wolfe, D. Laham, T. Landauer, and W. Kintsch. Using latent semantic analysis to assess knowledge: Some technical considerations. *Discourse Processes*, 25:337–354, 1998.

[Sal71] G. Salton, editor. *The SMART Retrieval System: Experiments in Auto-*

matic Document Processing. Prentice Hall, Englewood Cliffs, New Jersey, 1971.

[Sal89] G. Salton. *Automatic text processing: The transformation, analysis, and retrieval of information by computer.* Addison-Wesley, Reading, MA, 1989.

[SB88] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[Set96] R. Setiono. Extracting rules from pruned neural network for breast cancer diagnosis. pages 37–51, 1996.

[SM83] G. Salton and M. McGill. *Introduction to Modern Information Retrieval.* McGraw Hill Publishing company, New York, 1983.

[SP97] L. Saul and F. Pereira. Aggregate and mixed-order markov models for statistical language processing. In Claire Cardie and Ralph Weischedel, editors, *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 81–89. Association for Computational Linguistics, Somerset, New Jersey, 1997.

[SS97] H. Schutze and C. Silverstein. Projections for efficient document clustering. In *ACM/SIGIR*, pages 74–81, 1997.

[Str98] G. Strang. *Introduction to Linear Algebra.* Wellesley-Cambridge Press, 3rd edition, 1998.

[TC91] H. Turtle and W. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3), 1991.

[TG96] I. Taha and J Gosh. Characterization of the wisconsin breast cancer database using a hybrid symbolic-connectionist system. Technical Report UT-CVISS-TR-97-007, the Computer and Vision Research Center, University of Texas, 1996.

[TP91] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3, 1991.

[vR79] C. van Rijsbergen. *Information Retrieval.* Butterworths, London, 2nd edition, 1979.

[WK79] W. Waller and D. Kraft. A mathematical model of a weighted boolean retrieval system. *Information Processing and Management*, 15:235–245, 1979.

[WM90] W. Wolberg and O. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In *Proceedings of the National Academy of Sciences*, volume 87, pages 9193–9196, 1990.

[WZ02] J. Wu and Z. Zhou. Face recognition with one training image per person. *Pattern Recognition Letters*, 23(14):1711–1719, 2002.

[Zim91] H. Zimmerman. *Fuzzy Set Theory and its Applications.* Kluwer Academic Publishers, 2nd edition, 1991.

Appendix

List of Publications

1. Refereed Journal Article: Liang Chen, Jia Zeng and Naoyuki Tokuda, A "stereo" document representation for textual information retrieval, Journal of the American Society for Information Science and Technology (JASIST), Accepted for publication (2005).

2. Refereed Book Chapter: Liang Chen, Jia Zeng and Jian Pei, Classifying Noisy and Incomplete Medical Data by a Differential Latent Semantic Indexing Approach, in P. Pardalos et al. Eds: Data Mining in Biomedicine, Springer Press, Accepted for publication (2004).