RETRIEVING SEMANTICALLY RELEVANT DOCUMENTS USING

LATENT SEMANTIC INDEXING

CHUE WUT YEE

M.C.Sc. **JANUARY 2020**

RETRIEVING SEMANTICALLY RELEVAT DOCUMENTS USING LATENT SEMANTIC INDEXING

By

CHUE WUT YEE

B.C.Sc. (Hons:)

A dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

M.C.Sc.

University of Computer Studies, Yangon

JANUARY 2020

ACKNOWLEDGEMENTS

I would like to express my gratitude and appreciation to the following persons who supported and contributed towards the success of the thesis.

First and foremost, I would like to express my gratitude and my sincere thanks to **Dr. Mie Mie Thet Thwin,** Rector, University of Computer Studies, Yangon, for her kind permission to submit this thesis.

Firstly, I am honored to express my deepest gratitude and heartfelt thanks to my dearest supervisor, **Dr. Zon Nyein Nway**, Lecturer, Faculty of Information Science, University of Computer Studies, Yangon, for her guidance, encouragement, and continued support throughout this study. Her extensive knowledge and creative thinking have been an invaluable help throughout my thesis.

I would like to express my special appreciation to **Dr. Thi Thi Soe Nyunt**, Professor and Head of Faculty of Computer Science, for her administrative support and encouragement in the thesis development.

I would like to express my gratitude and appreciation to **Daw Htet Htet Aung**, Lecturer, the Department of Language for her advice, editing and suggestion from the language point of view.

I would like to thanks to all my teachers from childhood to the present time for their support not only for the fulfillment of the degree of M.C.Sc. but also for my life.

Moreover, I would like to extend thanks to the Division of Atomic Energy, Department of Research and Innovation, Ministry of Education that allowed me to study the degree of M.C.Sc.

I am deeply grateful to my parents and my beloved husband for their immense love and support, without whom I would not have been able to complete my Master's degree successfully. Lastly, I would also like to thank all my friends and colleagues for their support and encouragement.

STATEMENT OF ORIGINALITY

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

------------------------------ ------------------------------

Date Chue Wut Yee

ABSTRACT

Nowadays, with the development of the internet, it is available to collect very large amounts of data and searching effective information from these data develop into an essential work. The major purpose of an Information Retrieval system is to retrieve all the relevant documents, which are relevant to the user query. Popular search engines such as Google, Yahoo, Alta Vista and Bing give the services of the form of modern information retrieval.

Term matching techniques may retrieve irrelevant or inaccurate results because of synonyms and polysemys words, so effective concept-based techniques are needed. This system examines the utility of conceptual indexing to improve retrieval performance of a domain specific information retrieval system using Latent Semantic Indexing (LSI). LSI is an indexing and retrieval method that uses a mathematical technique called Singular Value Decomposition (SVD) to figure out patterns in the relationship between the term used and the meaning they convey. LSI makes use of the words that occur together in documents to capture the hidden related meanings among documents and thus can improve the ability to rank relevant documents.

This system is able to accept a user query such as a phrase or sentences, search the most semantically related documents and rank and retrieve such documents according to their similarity values. In this system, Cosine Similarity Method is used to find the relevancy and also let the user to view the results by descending order of the similarity values. This system ensures to support the searching time and provide the rate of latent semantic relevancy. The accuracy result of the system is calculated by precision, recall and f-measure. This system introduces to search the symptoms and signs of disease which are collected from https://www.medicinenet.com/symptoms_and_signs/symptomchecker.htm#introView. It basically works as a web page search system. The proposed system expected that it helps the people who want to find the information about biomedical diseases.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

Information Retrieval (IR) deals with the representation, storage, organization of and access to information items. The representation and organization of the information provides the users with easy contact to the information in which they are interested. For hundreds of years, people have understood the importance of archiving and searching information. With the large quantity of information usable online, the World Wide Web is a productive field for information retrieval research.

1.1 Overview of the System

The fast growth of world-wide information system develops in new requirements for text indexing and retrieval. The main purpose of an information retrieval system is to retrieve all the relevant documents, which are relevant to the user query [20]. There are a variety of techniques to improve efficiency in information retrieval. Information is a basic resource in today’s society. So, information retrieval is a critical role for various organizations all over the world. Information is being looked for among objects, e.g., text, picture, sound, or multimedia objects. Some people have been searching the information they need in some place, especially in library, manually. In particularly, it is not convenient for the students who are searching the book’s information they want. If users search this information manually, they can’t get semantically relevant information in the short time.

The proposed system describes a approach to automatic indexing and retrieval. It is studied to overcome a main problem of previous term matching retrieval system that attempt to match terms of queries with terms of documents. The users want to retrieve based on conceptual concept, and singular terms not provide accurate information about the conceptual meaning of a document. There are many things to describe a given concept, so the actual terms in a user’s query may not match those of a relevant document. Most terms have different meanings, so terms in a user’s query will actually match terms in documents that are not of interest to the user.

Semantic Search in Information Retrieval procedures an essential part of individual search engines. Powerful search engines such as, Google, Yahoo, Bing etc.

use the concept of semantic search between the terms and documents. A profitable process of developing this technique is the Latent Semantic Indexing (LSI), which plans the words under study on a conceptual space. The conceptual space suspends on the queries and the document collection [19]. It applied a mathematical technique to figure out the relationship between the terms called as Singular Value Decomposition (SVD). LSI is used for finding results, documents clustering, spam filtering, speech recognition, patent searches and automatic essay grading.

In this system, familiar method for latent semantic similarity, latent semantic indexing (LSI) is used to retrieve semantically related documents rather than term based search. This system considers the conceptual indexing in an attempt to improve retrieval performance of a domain specific information retrieval system. This system is able to accept a user query, search the most related documents and rank and retrieve such documents according to their similarity values. This system introduces to aid the sign and symptoms of disease searches which are collected from biomedical data. It is expected that the system helps the people who want to find easily and correctly the information about the popular signs and symptoms of different diseases.

1.2 Theoretical Area of Information Retrieval

Information Retrieval (IR) is the area of computer science that the processing of text documents such as scientific papers or electronic textbooks so that they can be speedy retrieved based on terms specified in a user's query. The widely spread of the Worlds Wide Web, IR is most of the information on the Web is textual. Search engines such as Yahoo, Google, Alta Vista and Ask Jeeves are used by millions of users to search information on Web pages on any matter.

M.Al-Qahtani, A.Amira and N.Ramzan [10] discussed that effective information retrieval technique for e-Health systems. This system stated that data in computer systems is in coded format. However, user comments cannot be coded because it is stored in the form of free text. Extracting the valuable information from such free text is a challenge due to the complexity of the stored data. This system used the latent semantic indexing algorithm on the Health Improvement Network. The LSI algorithm uses the computational power of multiprocessor in performing the retrieval process. The representation of the patient's data in the form of term document matrix is transformed. By using this method, processing time will be reduced and the rate of relevancy was accurate.

Nang Ei Khaing [14] presented the similarity based document retrieval system by computing the weight of term based on term frequency and its inverse document frequency (idf) in the document collection. And Jaccard Coefficient similarity method is deployed to find the similarity measurement between words. Almost 200 documents with pdf format from University of Computer Studies, Yangon are trained for the retrieval model. In this paper, the author stated that although the cosine similarity takes less amount of time as compared to Jaccard coefficient method because of using mathematical formula, the Jaccard is applied in the system as this similarity measure is suitable with the tf-idf method.

R.Anita, C.N.Subalalitha and A.Dorle [15] proposed about the semantic search using latent semantic indexing and wordnet. Most of the famous search engines use the concept of semantic search. The general method (keyword search) similarity search is the sequential search which involves various noise effects. An efficient sequential search is latent semantic indexing (LSI) which maps the terms under the conceptual space. That conceptual space built upon the queries and the document collections. And the results obtained from LSI are free of some semantic such as polysemy and synonym etc. So, the system is integrated with WordNet, a large lexical database of English language to increase the search results.

1.3 Motivation of the Thesis

Most of concepts or documents can be described in many different ways due to the background and people's vocabulary natures. The keyword search (term based search) is that the conceptual meaning of the document is ignored during retrieval process. The key words are obtained from the query provided by the user and the documents are mined for these keywords. This technique is not entirely reliable, as it does not take conceptual meaning of documents and the individual words do not provide direct information about the meaning of the documents. Therefore, documents that do not involve the keywords but are semantically relevant to the given query are not retrieved. This causes low recall. A sufficient retrieval system should retrieve many relevant documents possible. At the equal time, it may retrieve not many non-relevant documents. The proposed system use Latent Semantic Indexing (LSI) to overcome this problem.

1.4 Objectives of the Thesis

The objectives of the thesis as follows:

- To implement the semantically related documents retrieval system.
- To learn the effectiveness of latent semantic indexing method in retrieval system.
- To give the information that LSI is suitable for retrieving from different kinds of documents.
- To describe the use of LSI as an improvement over general keyword search.
- To help the user by providing relevance feedback for their search results by calculating similarity scores.

1.5 Organization of the Thesis

This thesis is organized into five chapters.

Chapter 1 includes introduction, overview of the system, theoretical area of information retrieval, motivation and objectives of the thesis.

Chapter 2 describes the web mining, detail about of information retrieval, similarity measurement and evaluation of system performance.

Chapter 3 explains detail about of Latent Semantic Indexing, Singular Value Decomposition, and sample calculations of the system.

Chapter 4 presents design and implementation of proposed system which includes system flow diagram, database design, screen designs of the proposed system and experimental results.

Finally, chapter 5 describes the conclusion, advantage, limitation and future extensions of the system.

CHAPTER 2

BACKGROUND THEORY

Information retrieval (IR) is supports the user search necessity information from a huge collection of text documents. The users also suggest some problematic in the retrieval process generating a semantic space between their requirement. The achievement of an individual information need on the web is supported by search engines and other tools desired at helping users collect information from the web. Due to the advance of internet information, search engines have a prominent role in information retrieval and web mining applications.

2.1 Web Mining

Web mining is the application of data mining techniques to extract knowledge from Web data, including web documents, hyperlinks between documents usage of web sites. Web mining basically deals with providing solution to different problems and finding relevant information form the World Wide Web by using suitable algorithm [20]. The Web mining research relates to distinct research communities such as database, data mining, information retrieval, data warehousing, information extraction, natural language processing and artificial intelligence. Web mining techniques have practical application in m-commerce, e-government, e-commerce, distance learning, knowledge management, e-learning, digital libraries and organizational learning [16].

Web mining is currently developed towards three main research directions, which are related to the type of data: web content mining, web usage mining and web structure mining.

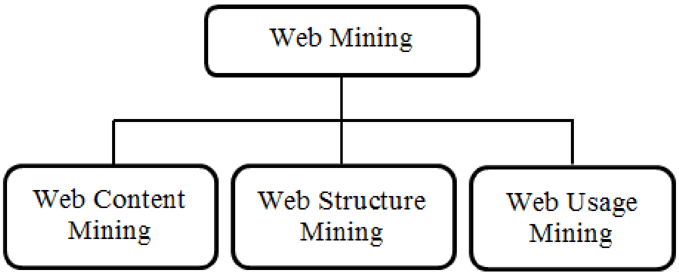


Figure 2.1 Web Mining Structure

2.1.1 Web Content Mining

Web content mining is the extraction and integration of suitable data, information and knowledge is available in many different formats – textual, metadata, links, multimedia objects, hidden and dynamic pages. Web content mining research area has prevalently focused on unstructured documents such as free text and semi-structured documents such as HTML documents. The main applications of web content mining are oriented to the text categorization, classification and to the event tracking and detection [20].

2.1.2 Web Structure Mining

Web Structure Mining is discovered the structure from the Web. Wen graph structure contains of web pages as nodes, and hyperlinks as edges related pages. The links allow to access to the desired information from the web pages and are included into hyperlink and document structure [20].

Hyperlinks: A Hyperlink is a structural unit that connects in a web page to different location, either within the same web page or on a different web page.

Document Structure: Web page can be constructed in a tree-structured format, which based on the HTML and XML tags within the page. Mining have focused on automatically extracting document object model (DOM) structures out of documents.

2.1.3 Web Usage Mining

Web Usage Mining is to discover usage patterns from web data and web-based applications. Web usage mining can be classified depend on the usage data [20].

Web Server Data: The user logs are collected by Web server data such as IP address, page reference and access time.

Application Server Data: Various commercial application servers like Web logic provide the ability for tracking various kinds of business events and log them in application sever logs.

Application Level Data: New kinds of events can be defined in logging and application can be turned on for them thus generating histories of these specially defined events.

2.2 Basic Concepts of Information Retrieval

The vast amount of information available in these days, it cannot be effectively and efficiently search. The purpose of information retrieval is to find relevant documents to a given user query. Today, people are rare to go to the libraries and more and more search on the web. Information retrieval researches the storage, acquisition, distribution, organization, and retrieval of information.

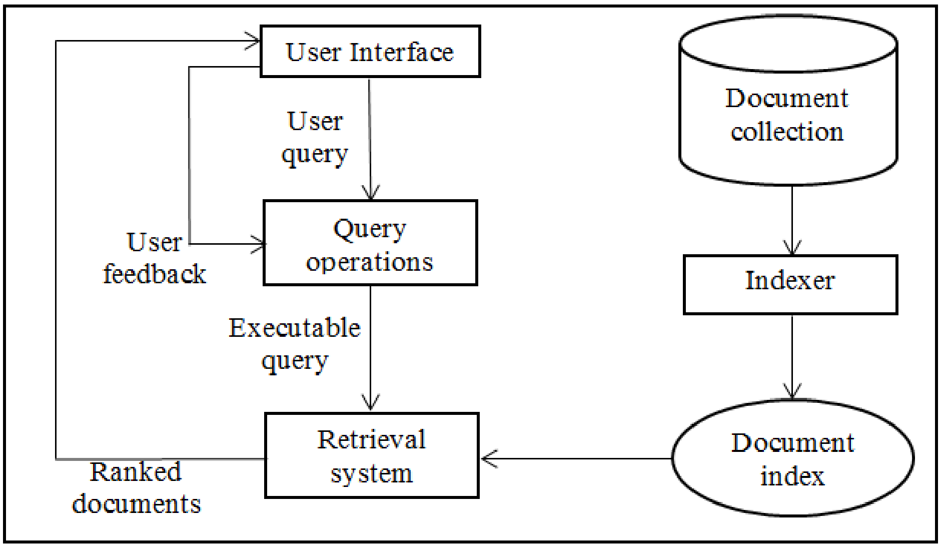


Figure 2.2 A general information retrieval system architecture

As in Figure 2.2, the user enters a query to the IR system through the query operations module. The retrieval module uses the document index to retrieve those documents that contain some query terms, calculate relevant scores for that query, and then rank the retrieved documents according to the similarity values. Previous IR assumes that a document is basic information and large document collections become the text database, which is indexed by the indexer for efficient retrieval.

2.3 Information Retrieval Models

An information retrieval model manages how a user query and documents are represented and how the relevance of a user query to a document is defined. Information retrieval that have three main models: Boolean model, Vector Space model and Probabilistic model. The three models represent documents and queries differently that they are used the same framework. A document is represented by a set of different terms and each term is associated with a weight. With the vector representation, a collection of documents is described as a relational table (or a matrix) [16].

2.3.1 Boolean Model

The Boolean model is one of the earliest and transparent information retrieval models [16]. In the Boolean model, documents and queries are represented as sets of terms and based on Boolean algebra such as AND, OR, and NOT. The Boolean model based system retrieves every document that produces the query logically true based on the binary decision criterion, i.e., a document is either relevant or irrelevant that is called exact match. Boolean retrieval is usually more effective than ranked retrieval because documents can be rapidly erased from consideration in the scoring process. The major disadvantages of the Boolean model are not accepted of partial match and do not returned ranked documents, which leads to poor retrieval results. To overcome these limitations researchers developed models such as the vector space model that incorporate ranking.

2.3.2 Vector Space Model

This model is the best known and most widely used information retrieval. A document in the vector space model is represented as a weight vector, in which each component weight is computed based on some variation of term frequency (TF) or term frequency inverse document frequency (TF-IDF) scheme. The weight w_{ij} of term t_i in document d_j is no longer in {0, 1} as in the Boolean model, but can be any number.

Then, the normalized term frequency of t_j in document d_j is given by

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots, f_{|v|j}\}} \tag{2.1}$$

The inverse document frequency of term t_j is given by:

$$idf_i = log\frac{N}{df_i} \tag{2.2}$$

The final TF-IDF term weight is given by

$$w_{ij} = tf_{ij} \times idf_i \tag{2.3}$$

The term weight w_{iq} of each term t_j in query q can also be computed in the same way as in a general document or minor differently.

$$w_{iq} = \left(0.5 + \frac{0.5\, f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots, f_{|v|j}\}}\right) \times log \frac{N}{df_i} \quad (2.4)$$

A Vector Space model is used for many processes of information retrieval, information filtering, indexing and relevancy rankings. The model has improved retrieval performance in general respect to Boolean models. A shortcoming of the vector space model is independent upon the terms weight and dimensions of the document space so it ignores the relationship among terms. Some other approaches such as Latent Semantic Indexing (LSI) based on the Vector Space model have overcome these limitations [5].

2.3.3 Latent Semantic Indexing

Latent Semantic Indexing (LSI) was proposed at the end of 80's as a way to solve the problem of vector space model [17] [19]. LSI also known as latent semantic analysis (LSA) is based on the basis that terms used in the same context attends to have the same meaning. LSI proposed to address the some of the shortcomings of traditional lexical matching problem. LSI analyzes the degree of semantic relationship that exists between documents. LSI can retrieve documents conceptually related to the search criteria, even if the document contains none of the user search terms. So, LSI addresses some of the retrieval problems in ‘thinking with concepts but communicating with words'.

LSI extends Vector Space Model by using reduced dimension representation computed by the matrix rank reduction technique of Singular Value Decomposition (SVD). The reduction results in the approximation of the original data matrix as semantic space, which reflects the major associative patterns in the data while ignoring the noise caused by word usage. Subsequently the queries are projected and processed in the lower dimensional space to find similarities with the documents [6].

The use of LSI is that it solves two keyword searches: the different words which have the same meaning (synonymy) and the individual word that can be used to express two or more different meanings (polysemy). Another use of LSI is that, it does not depend on any knowledge about the text. This means that LSI works well with any language and it can even be used to find documents across languages. LSI is also accepts to misspelled words, and it modifies well to changes in the vocabulary used in the data collection.

2.4 Similarity Measures

A key factor in the information retrieval system is the similarity measure between the query and document in the document collection. It is the mathematical measure of the degree of which two items are similar. Items which are more alike have higher similarity between them. Similarity measures are often non-negative numbers, which are normally in the range of [0, 1]. 1 implies complete similarity and 0 for no similarity [16]. There are many similarity measures

1. Cosine similarity
2. Jaccard similarity
3. Adjusted cosine similarity
4. Dice similarity
5. Correlation based similarity
6. Extended Jaccard similarity
7. Overlap similarity
8. Asymmetric similarity

Among them, the proposed system uses the Cosine similarity measure.

The cosine similarity is the very popular measure to use for information retrieval model. The Cosine similarity examines documents and queries to be vectors in a term space, is that it can be easily calculated with vector operations [16]. In cosine similarity, the lower angle presents higher similarity and higher angle represent dissimilarity between query and as set of documents. Documents are ranked according to their proximity to the query in that space by using their similarity values. The top ranked documents are represented as more relevant to the query.

$$\mathrm{Sim}_{cosine} = \frac{\sum_{k=1}^{n} w_{kj} * w_{kq}}{\sqrt{\sum_{k=1}^{n} {w_{kq}}^2}\sqrt{\sum_{k=1}^{n} {w_{kj}}^2}} \qquad (2.5)$$

2.5 Evaluation of System Performance

If there is no system performance evaluation, it is impossible to know how well the system is performing. Measuring system accuracy means how well the proposed system can retrieve relevant documents from non-relevant documents for the given user query. An effectiveness of IR system can be calculated by execution efficiency, storage efficiency and retrieval effectiveness [16].

Execution efficiency is calculated by the time it takes a system to perform a computation. This factor is main interest for IR systems because a lengthy retrieval time will make with the helpfulness of the system. Storage efficiency is calculated by the number of bytes used to store the data. Relevance is a fundamental concept in that the goal is to satisfy the human users' needs. A user cannot discuss why one document is more relevant than the other, i.e. relevance judgments for a user query often differ when defined by different users. Therefore, it is necessary to introduce a method to evaluate the performance of a retrieving process by evaluating the degree of relevance at which the retrieved information matches the query. Precision, recall and f-measure are commonly used measures to determine the effectiveness of the retrieval system [16].

The achievement of an information retrieval system, have been describe retrieval performance. Precision, recall and f-measure, which are extensively used to measure IR success based on the concept of relevance. An effective retrieval system should retrieve as numerous relevant documents as possible, i.e. have high recall, at the same time it should retrieve as some non-relevant documents. 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 the definitions, the following equations represent the precision, recall and f-measure.

$$\text{Precision} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}} \tag{2.6}$$

$$\text{Recall} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents}} \tag{2.7}$$

$$\text{F\_measure} = 2\,\frac{\text{precision}\,.\,\text{recall}}{\text{precision} + \text{recall}} \tag{2.8}$$

The prospect of the users may differ from one person to another. Some users connect more importance to precision, i.e., they want to relevant information without going through a lot of trash. Others take recall as a desire, 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 referenced. Due to this reason, some methods that evaluate the IR performance in terms of precision and recall simultaneously have been developed. F-measure which combines the precision and recall to give a single score is defined to be the harmonic mean of the precision and recall. There are several reasons that the F-measure can be criticized in particular circumstances due to its bias as an evaluation metric. This is also known as the F-measure, because recall and precision are evenly weighted [16].

CHAPTER 3
THE PROPOSED SYSTEM METHODOLOGY

Many approaches or articles are described in many ways due to the vocabulary and people's language natures. If a user query uses distinct words used in a document, is not be retrieved although it may be relevant document because the document uses some synonyms of the words in the user query. This causes low recall. For example, "photo", "image" and "picture" are synonyms in the context of digital cameras. If the user query only has the word "picture" relevant documents that contain "photo" or "image" but not "picture" will not be retrieved.

3.1 Semantically based Information Retrieval System

Semantically based retrieval system finds semantically related documents based on a set of common concepts. The output of such retrieval should be based on the degree of relevance, where relevance is measured based on the closeness of the concepts. It is difficult to provide a precise measure of the degree of relevance between a set of terms. Therefore, the statistical method, Latent Semantic Indexing (LSI) is used as a semantically base information retrieval system.

3.2 Indexing by Latent Semantic Indexing

Latent Semantic Indexing (LSI) aims to solve the problem through the identification of statistical associations of terms. LSI extends vector space model by modeling the term-document relationship using reduced dimension representation computed by the matrix rank reduction technique of singular value decomposition (SVD), to estimate this latent structure and to remove the "noise". The results of this decomposition are based on the latent semantic structure is also called the hidden "concept" space, which associates syntactically different but semantically similar terms and documents. Furthermore, the query is also transformed into the "concept" space before retrieval [4].

Let D be the text collection, the number of words in D be m and the number of document in D be n. LSI starts with and $m \times n$ term-document matrix A. Each row of A represents a term and each column represents a document. The matrix may be computed in various ways, e.g., using term frequency of TF-IDF values. Term frequency is used as an example in this section. Thus each entry or cell of the matrix *A*, denoted by A_{ij}, is the number of times that term *i* occurs in document *j* [5].

3.3 Singular Value Decomposition

Singular Value Decomposition (SVD) is the matrix rank reduction technique. This reduction results in the approximation of the original data matrix as semantic space and semantically related documents draw closer together. SVD reduces large dataset into a concentrated dataset containing only the important information from the original data. This method reflects the major associative patterns in the data while ignoring the noise caused by word usage. Subsequently the queries are designed and processed in the lower dimensional space to find similarities with the documents [6].

As a first step, a matrix *A* is generated which is a term-by-document *(m* × *n)* matrix for this set of documents. Each cell in the matrix represents frequency of occurrence of term *m* in document *n*. A query vector *q* is constructed in the similar way as *A*. This vector contains frequency of occurrence of words from matrix *A*.

In the second step, this matrix *A* is decomposed into three matrices by using SVD as a term matrix with unit-length columns (U), document matrix with unit-length columns (V) and diagonal matrix of singular values (S) are always arranged in decreasing order.

$$\mathrm{A = USV^T} \tag{3.1}$$

The third step is dimensionality reduction step where a low rank approximation of *A* is produced by retaining top *k* singular values ($k \leq min\ (m,n)$), from *S* matrix and their associated columns in *U* and *V* matrix and changing remaining values to zero. The value chosen for *k* should be small enough to favor speed but large enough to capture important information from the documents (k is the dimension rank of A). Since zeros were introduced in these matrices, they can be represented as smaller matrices by deleting their corresponding rows and columns.

$$A_k = U_k S_k {V_k}^T \tag{3.2}$$

The new space is called the *k-concept* space. Figure 3.1 presents the original matrices and the reduced matrices representation. LSI method does not re-produce the original term-document matrix *A* completely. The truncated SVD takes most of the important basically structures in the association of terms and documents at the same time removes noise caused by the word usage in keyword matching retrieval.

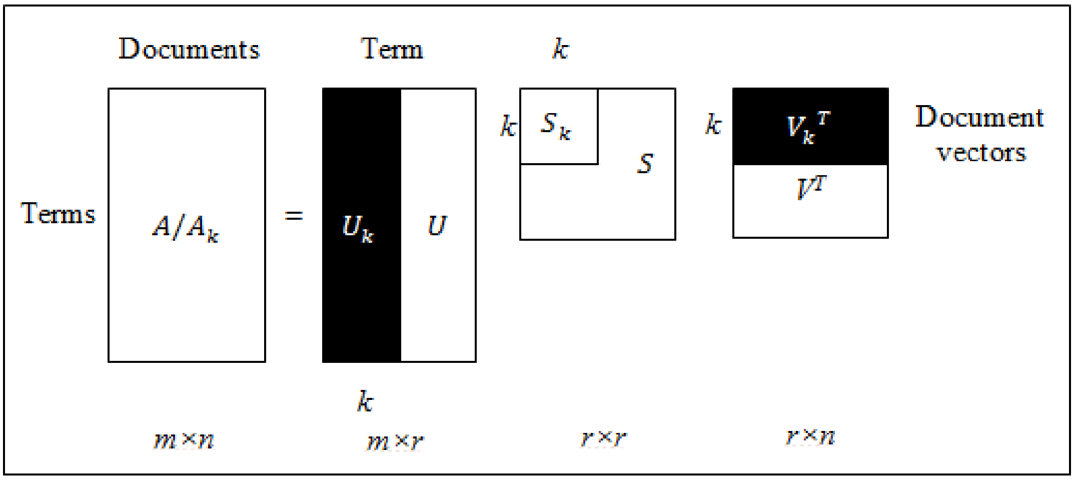


Figure 3.1 Representation of original matrices and reduced matrices

Given a user query q, firstly converted into the *k-concept* space which is expressed by q_k. This changing is required because SVD is converted the original documents into the k-concept space and stored them in V_k. The query q is gave as a new document in the original space represented as column in A and then projected to q_k as an extension document in ${V_k}^T$. The equation 3.3 is get for query in the reduced term-document space.

$$q_k = q^T U_k {S_k}^{-1} \tag{3.3}$$

Finally, q_k is simply compared with each document in V_k using as similarity measure. Alternatively, $S_k {V_k}^T$ represents the documents in the converted k-concept space and the similarity of the query document can be compared in the converted space which is $S_k {q_k}^T$ and each converted document in $S_k {V_k}^T$ for retrieval [13] [18].

3.4 Sample Calculation of the Proposed System

There are six main steps to conduct the LSI process. The following steps explain how to calculate the LSI with example. Let query term for information retrieval system, **q** is "Association Rule Mining". The document collection contains of the following three documents.

D1: Querying XML data based on improved prefix encoding

D2: Scalable approach for Association rule mining from structured XML data

D3: Implementation and application of Apriori and FP-Growth algorithm based on MapReduce

Tokenizing: In the above document and query, terms are tokenized into individual word such as query **q** is 'Association', 'Rule', 'Mining' and documents are
D1: 'Querying', 'XML', 'data', 'based', 'on', 'improved', 'prefix', 'encoding'
D2: 'Scalable', 'approach', 'for', 'Association', 'rule', 'mining', 'from', 'structured', 'XML', 'data'
D3: 'Implementation', 'and', 'application', 'of', 'Apriori', 'and' 'FP-Growth', 'algorithm', 'based', 'on', 'MapReduce'

Lowercase Changing: In the above document terms array and query term array are changed to small letter. So that, query and document terms are:
q is 'association', 'rule', 'mining'
D1: 'querying', 'xml', 'data', 'based', 'on', 'improved', 'prefix', 'encoding'
D2: 'scalable', 'approach', 'for', 'association', 'rule', 'mining', 'from', 'structured', 'xml', 'data'
D3: 'implementation', 'and', 'application', 'of', 'apriori', 'and', 'fp-growth', 'algorithm', 'based', 'on', 'mapreduce'

Stop word Removing: Words in a document that are frequently occurring but meaningless in terms of information retrieval are called stop word. After removing strop words, query and document terms are:
q: 'association', 'rule', 'mining'
D1: 'querying', 'xml', 'data', 'based', 'improved', 'prefix', 'encoding'
D2: 'scalable', 'approach', 'association', 'rule', 'mining', 'structured', 'xml', 'data'
D3: 'implementation', 'application', 'apriori', 'fp-growth', 'algorithm', 'based', 'mapreduce'

Stemming: Stemming eliminates grammatical variations of the same word by reducing it to the root word. After tokenization and stop words removing tasks, stemming is used. After using stemming algorithm, query and document terms are:
q: 'associate', 'rule', 'mine'
D1: 'query', 'xml', 'data', 'base', 'improve', 'prefix', 'encode'
D2: 'scale', 'approach', 'associate', 'rule', 'mine', 'structure', 'xml', 'data'
D3: 'implement', 'applicate', 'apriori', 'fp-growth', 'algorithm', 'base', 'mapreduce'

Processing Steps for Latent Semantic Indexing (LSI)

Step1: Set term weights and construct the term-document matrix **A** and query matrix **q**.

Terms		D1	D2	D3		Query
algorithm		[0	0	1		[0
applicate		0	0	1		0
approach		0	1	0		0
apriori		0	0	1		0
associate		0	1	0		1
base		1	0	1		0
data		1	1	0		0
encode		1	0	0		0
fp-growth	A =	0	0	1	q =	0
implement		0	0	1		0
improve		1	0	0		0
mapreduce		0	0	1		0
mine		0	1	0		1
prefix		1	0	0		0
query		1	0	0		0
rule		0	1	0		1
scale		0	1	0		0
structure		0	1	0		0
xml		1	1	0]		0]

Step2: Decompose matrix **A** and find **U**, **S** and **V** matrices for equation $\mathbf{A=USV^T}$

Where, U= term concept matrix,

S= concept matrix,

V= document concept matrix and

V^T= transport of V.

Perform SVD on the given matrix A.

Step 2.1: In order to find U, we have to start with A^TA.

$$A^TA=\begin{bmatrix}7 & 2 & 1\\ 2 & 8 & 0\\ 1 & 0 & 7\end{bmatrix}$$

Step 2.2: Find the determinant such that $|A^TA - \lambda I| = 0$ where I is the identity matrix and λ is a scalar, to obtain the Eigenvalues and Singular values which will be used to construct the S matrix.

$$A^TA - \lambda I = \begin{bmatrix} 7 & 2 & 1 \\ 2 & 8 & 0 \\ 1 & 0 & 7 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 7 - \lambda & 2 & 1 \\ 2 & 8 - \lambda & 0 \\ 1 & 0 & 7 - \lambda \end{bmatrix}$$

$|A^TA - \lambda I| = (7-\lambda)\,[(8-\lambda)(7-\lambda) - (0*0)] - 2\,[2(7-\lambda) - (1*0)] + 1\,[(2*0) - (1(8-\lambda)]$

$= -\lambda^3 + 22\lambda^2 - 156\lambda + 356$

$|A^TA - \lambda I| = 0$

$-\lambda^3 + 22\lambda^2 - 156\lambda + 356 = 0$

$\lambda_1 = 9.7093$

$\lambda_2 = 7.1939$

$\lambda_3 = 5.0968$

λ_1, λ_2 and λ_3 are eigenvalues $|\lambda_1| > |\lambda_2| > |\lambda_3|$

The Singular Value would be: $s_1 = \sqrt{9.7093} = 3.116$

$s_2 = \sqrt{7.1939} = 2.6821$

$s_3 = \sqrt{5.0968} = 2.2575$

$$S = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix} = \begin{bmatrix} 3.116 & 0 & 0 \\ 0 & 2.6821 & 0 \\ 0 & 0 & 2.2575 \end{bmatrix}$$

$$S^{-1} = \begin{bmatrix} 1/3.116 & 0 & 0 \\ 0 & 1/2.6821 & 0 \\ 0 & 0 & 1/2.2575 \end{bmatrix} = \begin{bmatrix} 0.3209 & 0 & 0 \\ 0 & 0.3728 & 0 \\ 0 & 0 & 0.443 \end{bmatrix}$$

Step2.3: Compute the Eigenvectors by evaluating $(A^TA - \lambda_i I)\,X_i = 0$, where λ_i corresponds to each of the Eigenvalues that were computed in the previous step.

Calculating the Eigenvector for the Eigenvalue $\lambda_1 = 9.7093$ we get

$$A^TA - \lambda_1 I = \begin{bmatrix} 7 - \lambda & 2 & 1 \\ 2 & 8 - \lambda & 0 \\ 1 & 0 & 7 - \lambda \end{bmatrix}$$

$$= \begin{bmatrix} 7-9.7093 & 2 & 1 \\ 2 & 8-9.7093 & 0 \\ 1 & 0 & 7-9.7093 \end{bmatrix}$$

$$= \begin{bmatrix} -2.7093 & 2 & 1 \\ 2 & -1.7093 & 0 \\ 1 & 0 & -2.7093 \end{bmatrix}$$

$$(\mathrm{A^TA} - \lambda_1 \mathrm{I})\, \mathrm{X}_1 = \begin{bmatrix} -2.7093 & 2 & 1 \\ 2 & -1.7093 & 0 \\ 1 & 0 & -2.7093 \end{bmatrix} \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \mathrm{x}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$-2.7093\mathrm{x}_1 + 2\mathrm{x}_2 + \mathrm{x}_3 = 0$ Eq. (1)

$2\mathrm{x}_1 - 1.7093\mathrm{x}_2 = 0$ Eq. (2)

$\mathrm{x}_1 - 2.7093\, \mathrm{x}_3 = 0$ Eq. (3)

In Eq. (3) If $\mathrm{x}_1 = 1,\ \mathrm{x}_3 = 0.3691$

By substituting $\mathrm{x}_1 and\ \mathrm{x}_3$ In Eq. (2) $\mathrm{x}_2 = 1.1701$

The Eigenvector of $\lambda_1 = \begin{bmatrix} 1 \\ 1.1701 \\ 0.3691 \end{bmatrix}$

Normalized the vector by the length;

$\mathrm{L} = \sqrt{(-1)^2 + (-1.1701)^2 + (-0.3691)^2} = 1.5828$

The normalized Eigenvector of $\lambda_1 = \begin{bmatrix} 1/1.5828 \\ 1.1701/1.5828 \\ 0.3691/1.5828 \end{bmatrix} = \begin{bmatrix} 0.6318 \\ 0.7392 \\ 0.2332 \end{bmatrix}$

Using similarity approach for calculating the Eigenvector for the Eigenvalue λ_2 = 7.1939 we get

The Eigenvector of $\lambda_2 = \begin{bmatrix} 1 \\ -2.4811 \\ 5.1573 \end{bmatrix}$

Normalized the vector by the length;

$\mathrm{L} = \sqrt{(1)^2 + (-2.4811)^2 + (-5.1573)^2} = 5.8098$

The normalized Eigenvector of $\lambda_2 = \begin{bmatrix} 1/5.8098 \\ -2.4811/5.8098 \\ 5.1573/5.8098 \end{bmatrix} = \begin{bmatrix} 0.1721 \\ -0.4271 \\ 0.8877 \end{bmatrix}$

Using similarity approach for calculating the Eigenvector for the Eigenvalue $\lambda_3 = 5.0968$ we get

The Eigenvector of $\lambda_3 = \begin{bmatrix} 1 \\ -0.5254 \\ -0.6889 \end{bmatrix}$

Normalized the vector by the length;

$$\mathrm{L} = \sqrt{(1)^2 + (-0.5254)^2 + (-0.6889)^2} = 1.3231$$

The normalized Eigenvector of $\lambda_3 = \begin{bmatrix} 1/1.3231 \\ -0.5254/1.3231 \\ -0.6889/1.3231 \end{bmatrix} = \begin{bmatrix} 0.7558 \\ -0.5207 \\ -0.3971 \end{bmatrix}$

Step 2.4: Constructing V matrix by using calculation of Eigenvectors as column in V.

$$\mathrm{V} = \begin{bmatrix} 0.6318 & 0.1721 & 0.7558 \\ 0.7392 & -0.4271 & -0.5207 \\ 0.2332 & 0.8877 & -0.3971 \end{bmatrix}$$

When we really want its transpose

$$\mathrm{V^T} = \begin{bmatrix} 0.6318 & 0.7392 & 0.2332 \\ 0.1721 & -0.4271 & 0.8877 \\ 0.7558 & -0.5207 & -0.3971 \end{bmatrix}$$

Step 2.5: Constructing U matrix by using equation $\mathrm{U} = \mathrm{AVS^{-1}}$ we get;

$$\mathrm{U} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0.6318 & 0.1721 & 0.7558 \\ 0.7392 & -0.4271 & -0.5207 \\ 0.2332 & 0.8877 & -0.3971 \end{bmatrix} \times \begin{bmatrix} 0.3209 & 0 & 0 \\ 0 & 0.3728 & 0 \\ 0 & 0 & 0.443 \end{bmatrix}$$

$$U = \begin{bmatrix} 0.0748 & 0.3309 & -0.1759 \\ 0.0748 & 0.3309 & -0.1759 \\ 0.2372 & -0.1592 & -0.2307 \\ 0.0748 & 0.3309 & -0.1759 \\ 0.2372 & -0.1592 & -0.2307 \\ 0.2776 & 0.3951 & 0.1589 \\ 0.4400 & -0.0951 & 0.1041 \\ 0.2027 & 0.0642 & 0.3348 \\ 0.0748 & 0.3309 & -0.1759 \\ 0.0748 & 0.3309 & -0.1759 \\ 0.2027 & 0.0642 & 0.3348 \\ 0.0748 & 0.3309 & -0.1759 \\ 0.2372 & -0.1592 & -0.2307 \\ 0.2027 & 0.0642 & 0.3348 \\ 0.2027 & 0.0642 & 0.3348 \\ 0.2372 & -0.1592 & -0.2307 \\ 0.2372 & -0.1592 & -0.2307 \\ 0.2372 & -0.1592 & -0.2307 \\ 0.4400 & -0.0951 & 0.1041 \end{bmatrix}$$

Step 3: Implementation of a rank 2 approximation by keeping the first two columns of U and V and the first two columns and rows of S. Find $\mathrm{U_k}, \mathrm{S_k}, \mathrm{V_k}$ and ${\mathrm{V_k}}^{\mathrm{T}}$.

$$U_k = \begin{bmatrix} 0.0748 & 0.3309 \\ 0.0748 & 0.3309 \\ 0.2372 & -0.1592 \\ 0.0748 & 0.3309 \\ 0.2372 & -0.1592 \\ 0.2776 & 0.3951 \\ 0.4400 & -0.0951 \\ 0.2027 & 0.0642 \\ 0.0748 & 0.3309 \\ 0.0748 & 0.3309 \\ 0.2027 & 0.0642 \\ 0.0748 & 0.3309 \\ 0.2372 & -0.1592 \\ 0.2027 & 0.0642 \\ 0.2027 & 0.0642 \\ 0.2372 & -0.1592 \\ 0.2372 & -0.1592 \\ 0.2372 & -0.1592 \\ 0.4400 & -0.0951 \end{bmatrix}$$

$$S_k = \begin{bmatrix} 3.16 & 0 \\ 0 & 2.6821 \end{bmatrix}$$

$$V_k = \begin{bmatrix} 0.6318 & 0.1721 \\ 0.7392 & -0.4271 \\ 0.2332 & 0.8877 \end{bmatrix}$$

Step 4: Finding the new document vector coordinates in this reduced 2-dimensional space.

$\mathrm{d1}\ (0.6318, 0.1721)$

$\mathrm{d2}\ (0.7392, -0.4271)$

$\mathrm{d3}\ (0.2332, 0.8877)$

Step 5: Finding the new query vector coordinates in the reduced 2-dimensional space

$$\mathrm{q} = \mathrm{q}^{\mathrm{T}}\mathrm{U}_{\mathrm{k}}\mathrm{S}_{\mathrm{k}}{}^{-1}$$

$$\mathrm{q}^{\mathrm{T}} = [0\quad 0\quad 0\quad 0\quad 1\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 1\quad 0\quad 0\quad 1\quad 0\quad 0\quad 0]$$

$$\mathrm{S}_{\mathrm{k}}{}^{-1} = \begin{bmatrix} 0.3165 & 0 \\ 0 & 0.3728 \end{bmatrix}$$

$$\mathrm{q} = [0\quad 0\quad 0\quad 0\quad 1\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 1\quad 0\quad 0\quad 1\quad 0\quad 0\quad 0] \times \begin{bmatrix} 0.0748 & 0.3309 \\ 0.0748 & 0.3309 \\ 0.2372 & -0.1592 \\ 0.0748 & 0.3309 \\ 0.2372 & -0.1592 \\ 0.2776 & 0.3951 \\ 0.4400 & -0.0951 \\ 0.2027 & 0.0642 \\ 0.0748 & 0.3309 \\ 0.0748 & 0.3309 \\ 0.2027 & 0.0642 \\ 0.0748 & 0.3309 \\ 0.2372 & -0.1592 \\ 0.2027 & 0.0642 \\ 0.2027 & 0.0642 \\ 0.2372 & -0.1592 \\ 0.2372 & -0.1592 \\ 0.2372 & -0.1592 \\ 0.4400 & -0.0951 \end{bmatrix} \times \begin{bmatrix} 0.3165 & 0 \\ 0 & 0.3728 \end{bmatrix}$$

$$\mathrm{q} = [-0.2221\quad -0.1781]$$

Step 6: Ranking documents in decreasing order of query-document Cosine similarities with

$$\mathrm{sim}(\mathrm{q}, d_j) = \frac{\sum_{k=1}^{n} w_{kj} * w_{kq}}{\sqrt{\sum_{k=1}^{n} {w_{kq}}^2}\sqrt{\sum_{k=1}^{n} {w_{kj}}^2}}$$

$$\mathrm{sim}(\mathrm{q}, d_1) = \frac{\big((-0.2221) * (0.6318)\big) + \big(\,(-0.1781) * (0.1721)\big)}{\sqrt{(-0.2221^2) + (-0.1781)^2}\sqrt{(0.6318)^2 + (0.1721)^2}} = 0.5883$$

$$\mathrm{sim}(\mathrm{q}, d_2) = \frac{((-0.2221) * (0.7392)) + ((-0.1781) * (-0.4271))}{\sqrt{(-0.2221^2) + (-0.1781)^2}\sqrt{(0.7392)^2 + (-0.4271)^2}} = 0.9885$$

$$\mathrm{sim}(\mathrm{q}, d_3) = \frac{((-0.2221) * (0.2332)) + ((-0.1781) * (0.8877))}{\sqrt{(-0.2221^2) + (-0.1781)^2}\sqrt{(0.2332)^2 + (0.8877)^2}} = -0.4068$$

Finally, Sorting and ranking the documents in descending order is done according to the similarity values based on user query.

$\mathrm{D2}: 0.9885 > \mathrm{D1}: 0.5883 > \mathrm{D3}: -0.4068$

Output documents for 'association', 'rule', 'mining' are:

D2: Scalable approach for Association rule mining from structured XML data..

D1: Querying XML data based on improved prefix encoding.

D3: Implementation and application of Apriori and FP-Growth algorithm based on MapReduce.

It can be seen that document D2 scores higher than D1 and D3. Its vector is closer to the query vector than the other vectors. In this example, document 2 (D2) will be appeared first, after that document 1 (D1), document 3 (D3).

In the above sample calculation, according to the review of the previous system, D2 is expected to be retrieved as closest to query because it contains the word ‘association rule mining’. Document D3 also talks about (concept) 'association rule mining' algorithm such as 'Apriori and FP-Growth algorithm" so it is expected to be ranked after D2. Document D1 does not mention anything related to query ‘association rule mining’ so we expect it to be at the lowest retrieved rank.

In this approach D2 is the closest to query which can be observed from LSI results. However, D3 should have been ranked after D2 but LSI ranks D1 after D2. The query ‘association rule mining’ is present in D2 and the only word common between D2 and other document is ‘XML data’ which is present in D1. So LSI gives preference to D1 over D3. LSI uses word-document concept to retrieve documents instead of using actual word occurrence.

CHAPTER 4
IMPLEMENTATION AND DESIGN

The proposed system is a semantically related document retrieval system. In this system, the latent semantic indexing is implemented to describe patterns between terms and concepts contained in an text documents. As the functional requirements, LSI algorithm is applied and the cosine similarity between the query and document in the latent space is computed. As the non-functional requirements, at least one computer (32 or 64 bit operating system) with the specification of Intel® Core™ i5-4200M CPU @2.50GHz processor and 8.00GB memory is required.

4.1 Overview of the Proposed System

The proposed system is a semantically relevant documents retrieval system based on Latent Semantic Indexing method. The method finds the correlation pattern of words between documents for retrieval.

The proposed system comprises of two phases as follow:

For training phase,

1. Get the collection of biomedical data with pdf format.
2. Preprocess the documents collection such as tokenize, stop word removal and stemming.
3. Perform LSI approach on documents collection.

For testing phase,

1. Get the testing phrase or sentence.
2. Find the similar documents for that phrase or sentence.

4.2 The Proposed System Design

This system intends to implement the information retrieval model by LSI. The goal of this retrieval model is to find documents that are relevant to the user query. The detail designs of the proposed system are described in Figure 4.1 and Figure 4.2. The proposed system uses the combination of Vector Space model and dimension reduction method. The proposed system consists of two modules: building retrieval model (training phase) and finding similar documents for query terms (testing phase).

For the training module, the system uses almost 200 documents approximately

2286 words. These documents are unstructured documents with pdf formats. Therefore, preprocessing step is necessary to reduce the whole document by removing words void of semantic content. Among the remaining words, the same word may be in the forms of grammatical variations. So, these words must be eliminated by reducing them to the single words with the help of Potter Stemmer. After preprocessing steps will be carried out and built the retrieval model by using statistical method, singular value decomposition (SVD) for latent semantic indexing.

For the similarity finding module, the query is also carried out the pre-processing stages. In database, all documents are transformed into document vectors by LSI. The query vector is compared to all documents vectors. The input of the proposed system is user query (phrases or lines of sentences) and the output is the related documents title of the collection of documents stored in database together with similarity values. Finally, the system displays results in descending order with similarity values.

To evaluate the system performance, analysis results are emphasized on training time and system accuracy results. The performance value is accomplished by employing precision, recall and f-measure.

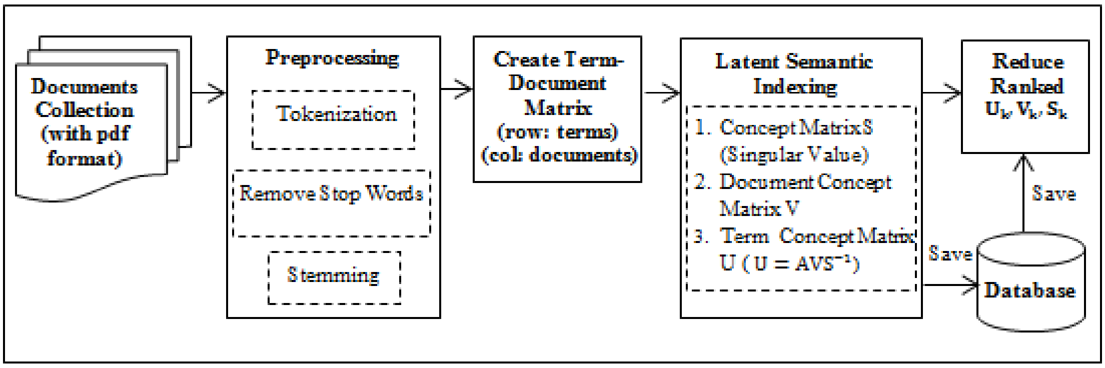

Documents Collection (with pdf format)
Preprocessing
Tokenization
Remove Stop Words
Stemming
Create Term-Document Matrix (row: terms) (col: documents)
Latent Semantic Indexing
1. Concept Matrix S (Singular Value)
2. Document Concept Matrix V
3. Term Concept Matrix U (U = AVS⁻¹)
Save
Database
Save
Reduce Ranked U_k, V_k, S_k

Figure 4.1 Process flow diagram for the Training Phase

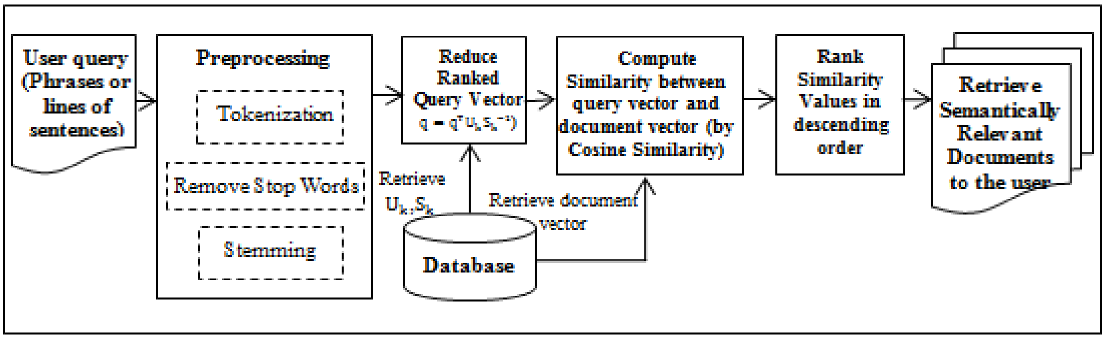

User query (Phrases or lines of sentences)
Preprocessing
Tokenization
Remove Stop Words
Stemming
Reduce Ranked Query Vector q = q^T U_k S_k^-1
Retrieve U_k,S_k
Database
Retrieve document vector
Compute Similarity between query vector and document vector (by Cosine Similarity)
Rank Similarity Values in descending order
Retrieve Semantically Relevant Documents to the user

Figure 4.2 Process flow diagram for the Testing Phase

4.3 Database Design of the Proposed System

The proposed system needs to use the following six tables.

1. Document Table
2. Word-List Table
3. Term-Document Matrix Table (Matrix A)
4. Singular Matrix Table (S)
5. Document Concept Matrix Table (Matrix V)
6. Term Concept Matrix Table (Matrix U)

Document table stores about all description of the documents such as Document_Code, Document_Title, Author_Name and the File_Path of the stored document. Word-list table stores the descriptions of the word for each document such as Word_ID, Document_ID, Word_Name and Word_Count. That table has composite primary key (Word_ID and Document_ID) to join with Document table. Based on these two tables, the main matrix for LSI: MatrixA table is built. In this table, ID, Word_ID and Colunm_Data (Document_ID) are attributes. After applying the LSI, the result matrixes such as matrixU, singular matrix and matrixV are stored in MatrixU table, Singular_Matrix and MatrixV tables. The database design of the proposed system is shown in Figure 4.3.

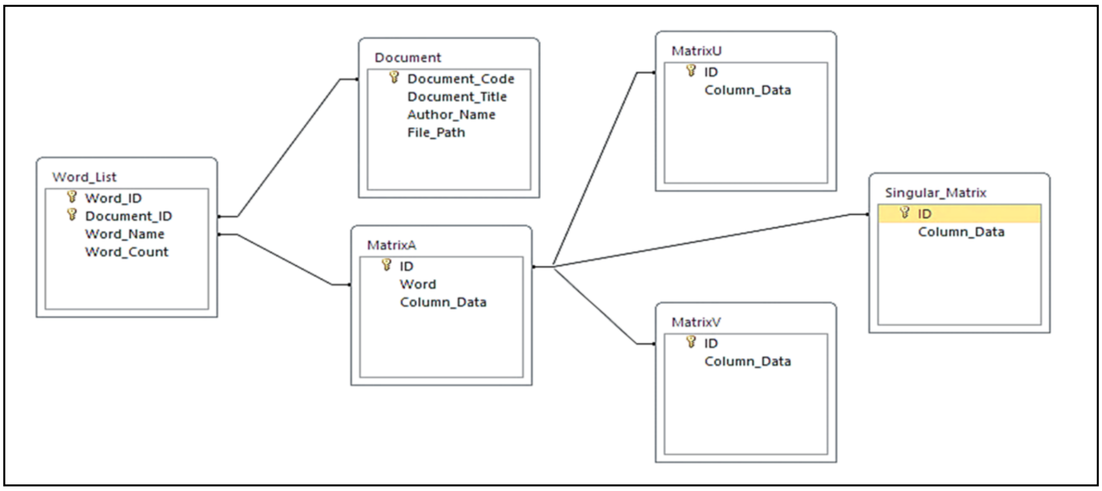


Figure 4.3 Database Design of the Proposed System

4.4 Training Phase of the Proposed System

For the training phase of the proposed system, the preprocessing step is necessary to reduce the whole document by removing words void of semantic content. The remaining words eliminates grammatical variations of the same words by reducing it to the stem or root word form with the help of Potter Stemmer. After preprocessing, the remaining words are the terms. Then the original term document matrix is created. LSI approach is applied based on that matrix.

4.4.1 Training Data Description

The collected documents are paper format with pdf extension. One document has at least one page. The collected biomedical related documents are from https://www.medicinenet.com/symptomsandsigns/symptomchecker.htm#introView. The main topics are the popular signs and symptoms of different diseases (approximately 200). According to the nature of biomedical diseases, some signs and symptoms are common for some kind of diseases. And there are many diseases that are totally different from one another.

These documents are unstructured documents with pdf formats. For each document, the list of terms and the frequency of each term are required to create the *m x n* term document matrix. According to the literature, some preprocessing steps should be performed in order to improve the retrieval process and to reduce the computational time and storage requirements.

4.4.2 Preprocessing for Information Retrieval

Preprocessing is used to increase the processing speed as well as the efficiency of the IR process [11]. The proposed system implements the preprocessing functions as follows: tokenization, stop word removing and stemming.

Tokenizing: The process of breaking up the document into words, phrases, symbols or other meaningful elements. Machines do not know the structure of natural language document and cannot automatically recognize words and sentences. So, humans programed the computer to identify the distinct word and referred to as a token. Such a program is commonly called a tokenizer or parser or lexer [16].

Stop Word Removing: In document, words that are frequently occurring but meaningless in terms of information retrieval are called stop word [16]. Stop list contain stop words**:** prepositions, articles, pronouns, some adverbs and adjectives and some frequent words. It reduces the size of indexing file and also improves the overall efficiency and makes effectiveness.

Stemming: Stemming eliminates grammatical variations of the same word by reducing it to the stem or root word form [16]. Stemming improves retrieval performance by reducing specific index terms. For example, the words “produce”, “produced” “producing”, “producer” and “production” would all be stemmed to the word “produc”. A well-known Porter Stemming Algorithm is typically used for this purpose.

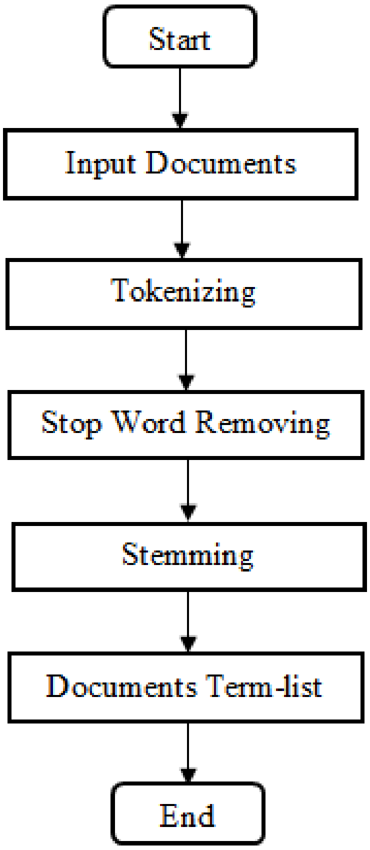
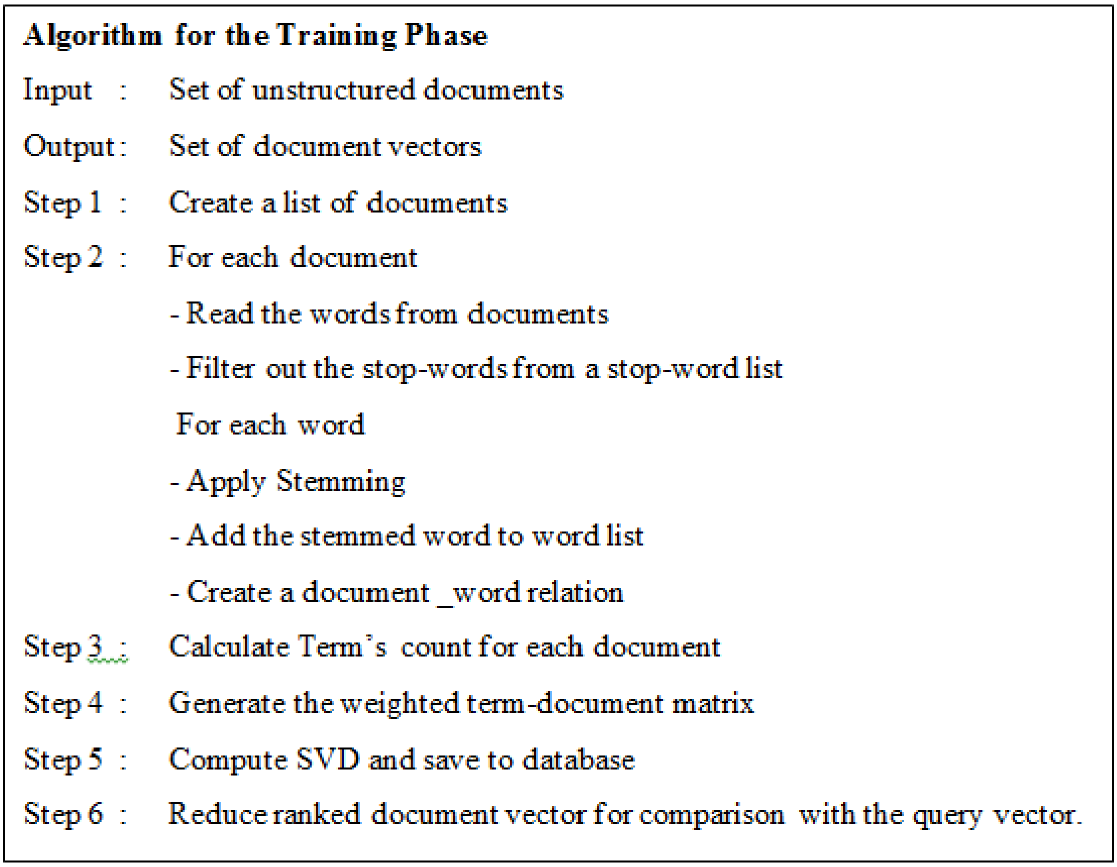


Figure 4.4 Preprocessing steps of text retrieval

4.4.3 Steps for the Training Phase

After preprocessing, the remaining words are the terms, which are numbered from 1 to *m*. Then *m x n* term document matrix is created. Based on that matrix, LSI approach is applied by the Figure 4.5 algorithm.

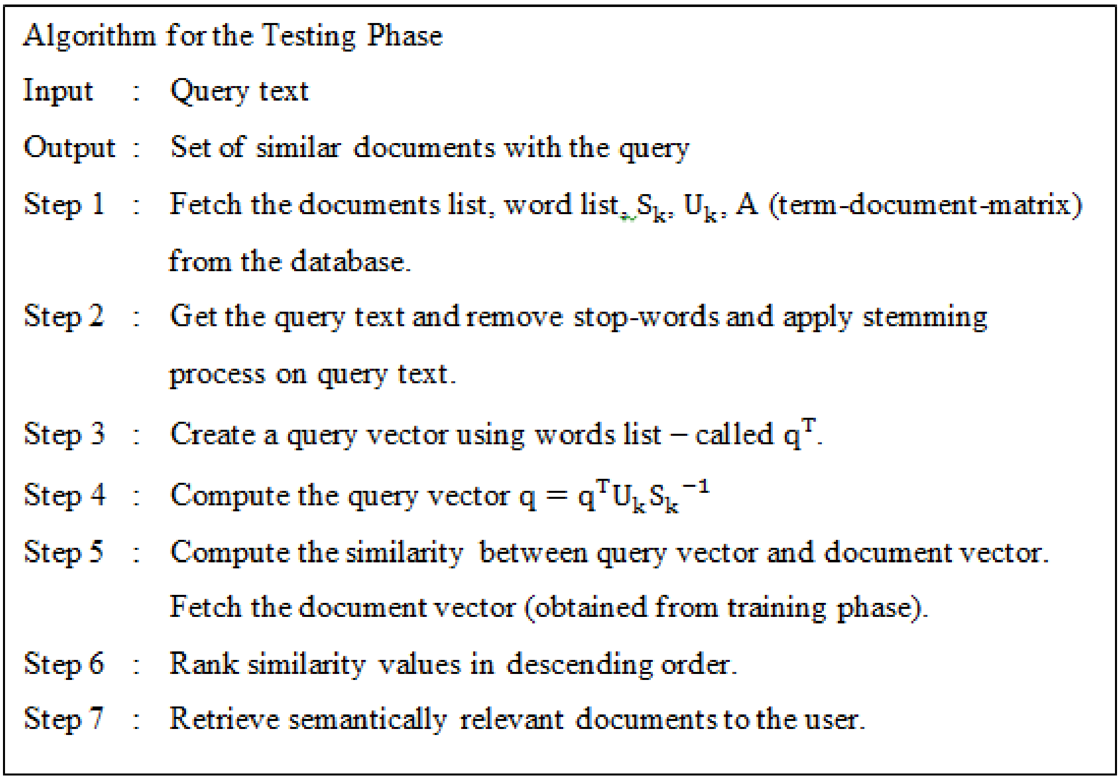


```
Algorithm for the Training Phase
Input   :   Set of unstructured documents
Output :    Set of document vectors
Step 1  :   Create a list of documents
Step 2  :   For each document
            - Read the words from documents
            - Filter out the stop-words from a stop-word list
             For each word
            - Apply Stemming
            - Add the stemmed word to word list
            - Create a document _word relation
Step 3  :   Calculate Term's count for each document
Step 4  :   Generate the weighted term-document matrix
Step 5  :   Compute SVD and save to database
Step 6  :   Reduce ranked document vector for comparison with the query vector.
```

Figure 4.5 Algorithm for the Training Phase

4.5 Testing Phase of the Proposed System

After the training phase, the retrieval model is available to use. The user input query is accepted to search the similar documents from the trained document collection. The query must be represented as a column vector. The largest score document is the most relevant. LSI replaces the term-document matrix is generated by the truncated singular value decomposition (SVD) and computes the most similar documents by the Figure 4.6 algorithm.



```
Algorithm for the Testing Phase
Input   :  Query text
Output  :  Set of similar documents with the query
Step 1  :  Fetch the documents list, word list, $S_k$, $U_k$, A (term-document-matrix)
           from the database.
Step 2  :  Get the query text and remove stop-words and apply stemming
           process on query text.
Step 3  :  Create a query vector using words list – called $q^T$.
Step 4  :  Compute the query vector $q = q^T U_k S_k^{-1}$
Step 5  :  Compute the similarity between query vector and document vector.
           Fetch the document vector (obtained from training phase).
Step 6  :  Rank similarity values in descending order.
Step 7  :  Retrieve semantically relevant documents to the user.
```

Figure 4.6 Algorithm for the Testing Phase

4.6 Analysis and Experiment Results

The computational difficulty of Singular Value Decomposition in Latent Semantic Indexing method is related on the number dimensions (k) and the number of terms and documents in the text collection. There are almost 200 pdf typed documents. Initially, 50 documents are trained and then another 50 documents are added up to the 200 documents. It is found that the training time is increasingly according to the training document size by the analysis results of Table 4.1 and Figure 4.7.

Table 4.1 Training Time Analysis

Number of Documents	Training Time (min)
50	35
100	72
150	157
200	315

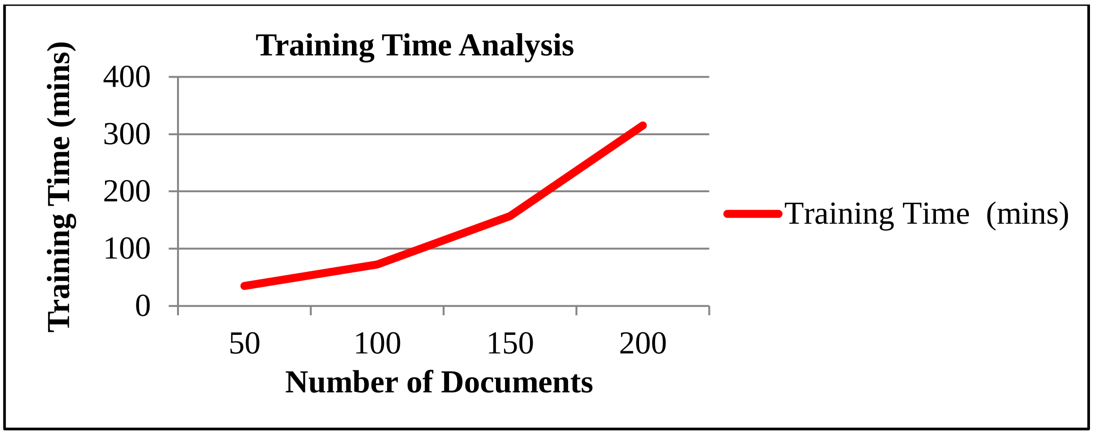


Figure 4.7 Line Chart for Training Time Evaluation

But only analyzing the time does not completely represent the powerful retrieval system. The efficiency should be considered. Therefore, the retrieval results must be analyzed for system effectiveness. Choosing the number of dimensions (k) is an important part of the SVD method. Table 4.2 shows the accuracy for different k value.

Table 4.2 Accuracy for Different k Value

k Value	Accuracy (%)
50	70
100	82
150	89
200	90

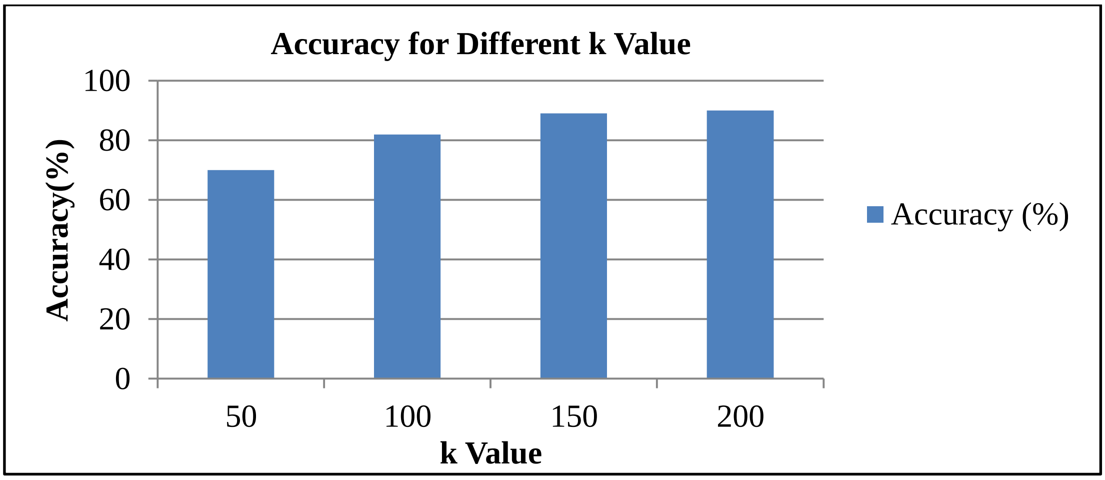


Figure 4.8 Accuracy for Different k Value

According to the Table 4.2, it is found that k=150 is the best value for this system. The above table shows the system correctness at different k values. Due to the desirable results, the training data size 200 documents are chosen and the rank value k is the most suitable at 150 because the correctness of the system accuracy is higher than the k value 100. For the earlier k value 100, the correctness of the system decreases a little. The column chart for accuracy of different k value is shown in Figure 4.8.

Table 4.3 System Accuracy Evaluation

Number of Documents	Precision (%)	Recall (%)	F-measure (%)
10	80	100	88
20	80	100	88
30	100	75	85
40	100	75	86
50	100	75	86
200	80	100	89

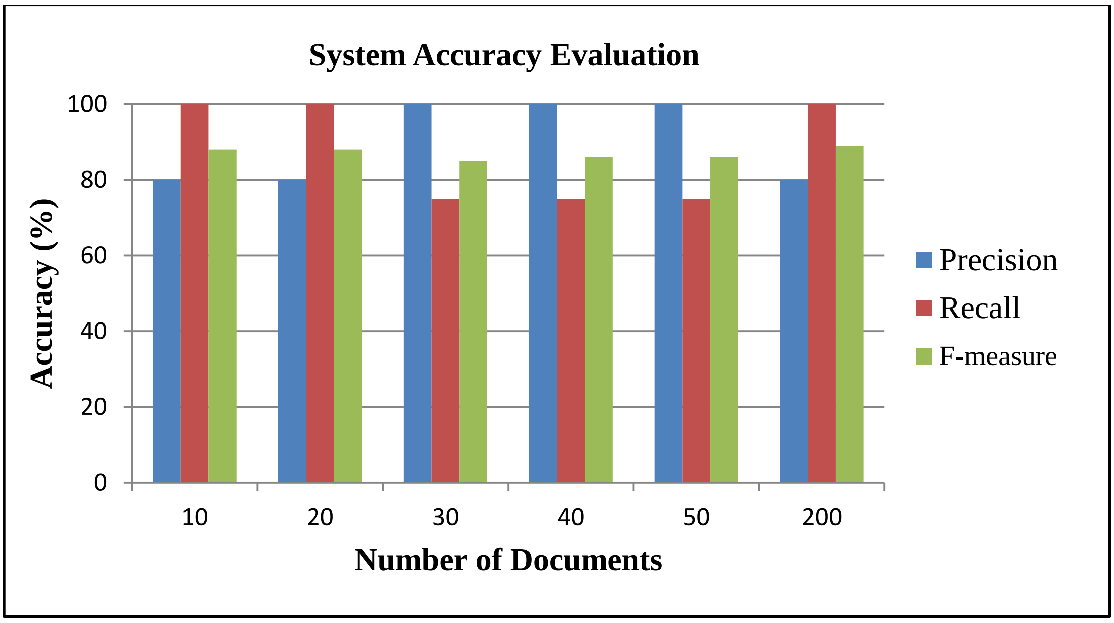
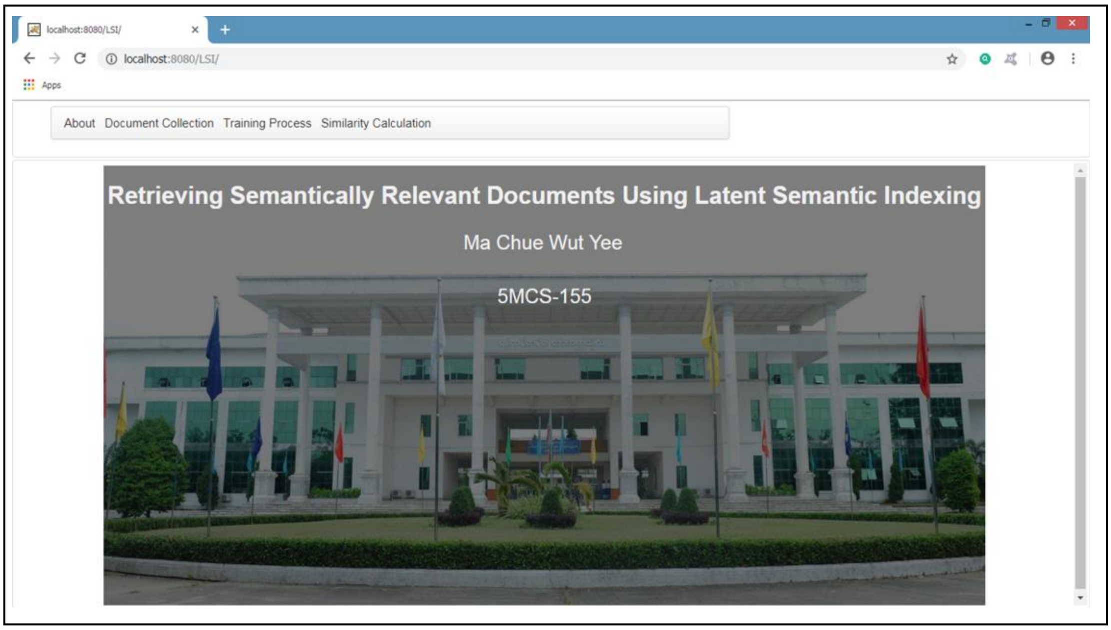


Figure 4.9 System Accuracy Evaluation

If there is no accuracy evaluation measure, it is impossible to know how well the system is performing. Finally, after selecting the k value (150) and the number of training documents (200), the effectiveness of the proposed system is analyzed by testing of 10, 20, 30, 40, 50 and 200 documents. Based on the number of correctly retrieved documents from total number of documents, the resulted value of precision, recall and f-measure are shown in Table 4.3 and column chart in Figure 4.9. Depending on the experimental result, the overall accuracy of the proposed system is 89% for the number of testing documents 200.

4.7 Implementation of the Proposed System

Figure 4.10 shows the home page of the proposed system. In this page , the menu bar is included at the top left cornor of the screen. The menu items are About, Document Collection, Training Process and Similarity Calculation. The menu item "About" shows the home page information of the proposed system. The menu item "Document Collection" is the training documents page is shown in Figure 4.11. The menu item "Training Process" is the training phase of the proposed system. And the last menu item "Similarity Calculation" is about the finding documents similarity page.



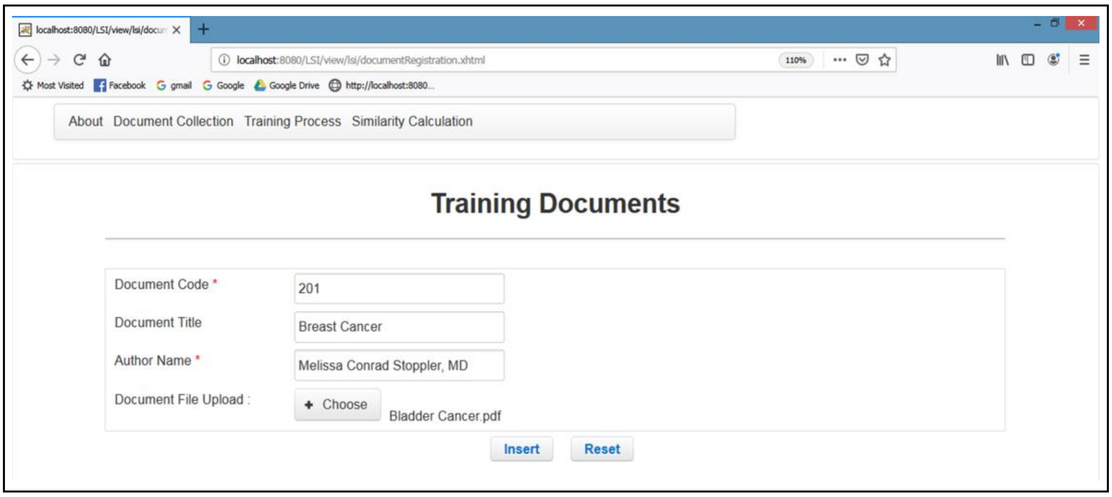
Retrieving Semantically Relevant Documents Using Latent Semantic Indexing

Ma Chue Wut Yee

5MCS-155

Figure 4.10 Home Page of the Proposed System

Figure 4.10 is the home page for retrieving semantically relevant documents retrieval system. It includes thesis title and author information.



Training Documents

Document Code * 201

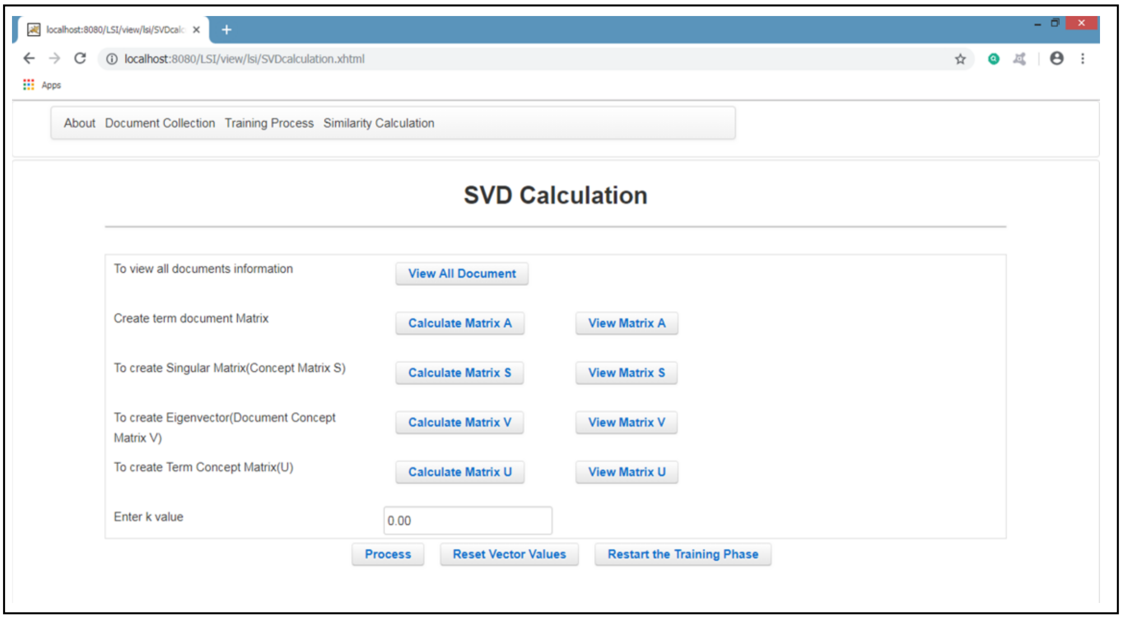
Document Title Breast Cancer

Author Name * Melissa Conrad Stoppler, MD

Document File Upload : Bladder Cancer.pdf

Figure 4.11 Training Documents Page

The Figure 4.11 is for document collection process. In this page, the user can register the new documents with pdf formats. To register the new documents, document title, author name and upload the whole document. There are two buttons – insert and reset. The "insert" button is to save the document information into the database and the "reset" button is to clear all input boxes for the next document registration. The document code field is implemented by the auto increment in the backend database. Therefore, the user does not need to input the document code. The system is automatically count the document code.



SVD Calculation

To view all documents information

Create term document Matrix

To create Singular Matrix(Concept Matrix S)

To create Eigenvector(Document Concept Matrix V)

To create Term Concept Matrix(U)

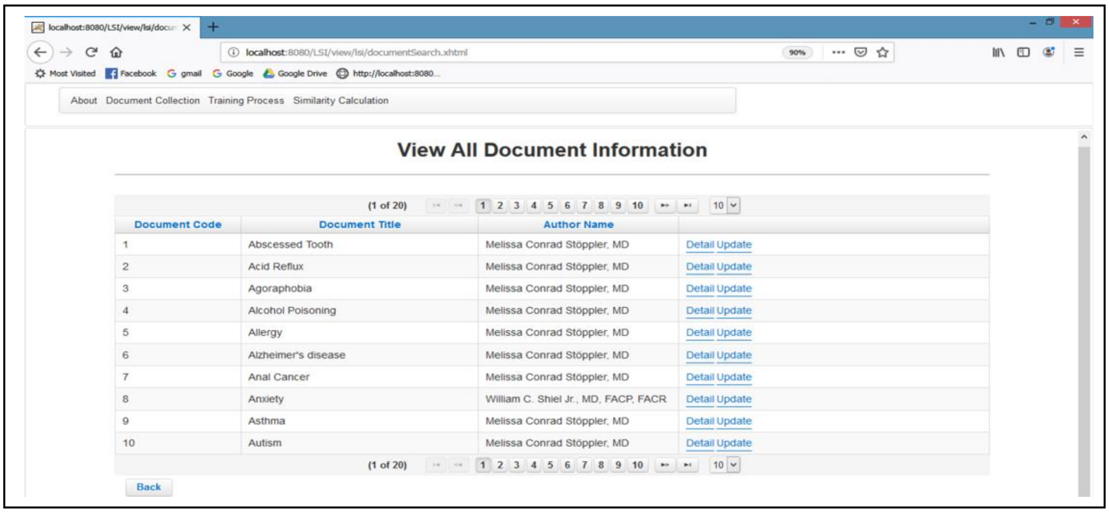
Enter k value 0.00

Figure 4.12 SVD Calculation Page

After finishing the document collection process, the next process is the training phase. Before the training phase is processed, the user can view all inputted document information via View all Document button as shown in SVD Calculation page of Figure 4.12. All document information can be viewed according to Figure 4.13. In the figure, the information such as how many documents are trained, what the title of documents are and who are the authors of these documents can be viewed.

The button "Calculate Matrix A" is to calculate the original term-document matrix. "Calculate Matrix S" butoon is to calculate the Singular Value Matrix. "Calculate Matrix V" butoon is to calculate the Document Concept Matrix V and "Calculate Matrix U" button is to calculate the Term Concept Matrix U. When these matrixes are calculated, the rank "k" value is given and then the "Process" butoon must be clicked to reduce the dimensions of the matrices. The result of these matrices are saved in database.

In Figure 4.13 View all documents information page, the two links Detail and Update are included. When Detail link is clicked, the detailed information about such document are shown as in Figure 4.14. And in this page, the user can download the document itself. When Update link is clicked, the user can update the document information such as document title and author name except the document code and the whole document as shown in Figure 4.15. There are three buttons in this page Update, Reset and Back. When the Update button is clicked, the updated information entered in input boxes are overwrited in database. When the Reset button is clicked, all text boxes are clear and the Back button redirected the user to the previous page.



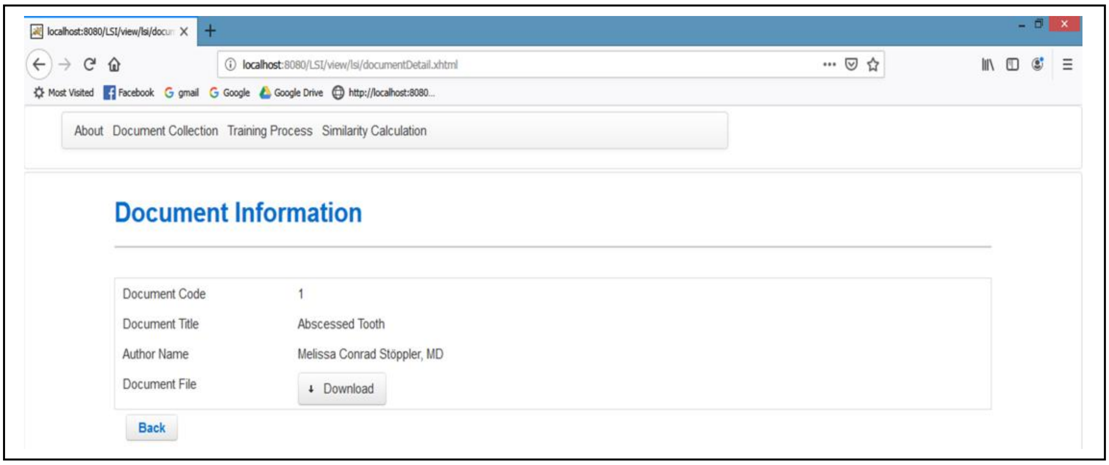
View All Document Information

(1 of 20) 1 2 3 4 5 6 7 8 9 10 10

Document Code	Document Title	Author Name	
1	Abscessed Tooth	Melissa Conrad Stöppler, MD	Detail Update
2	Acid Reflux	Melissa Conrad Stöppler, MD	Detail Update
3	Agoraphobia	Melissa Conrad Stoppler, MD	Detail Update
4	Alcohol Poisoning	Melissa Conrad Stöppler, MD	Detail Update
5	Allergy	Melissa Conrad Stöppler, MD	Detail Update
6	Alzheimer's disease	Melissa Conrad Stöppler, MD	Detail Update
7	Anal Cancer	Melissa Conrad Stöppler, MD	Detail Update
8	Anxiety	William C. Shiel Jr., MD, FACP, FACR	Detail Update
9	Asthma	Melissa Conrad Stöppler, MD	Detail Update
10	Autism	Melissa Conrad Stöppler, MD	Detail Update

(1 of 20) 1 2 3 4 5 6 7 8 9 10 10

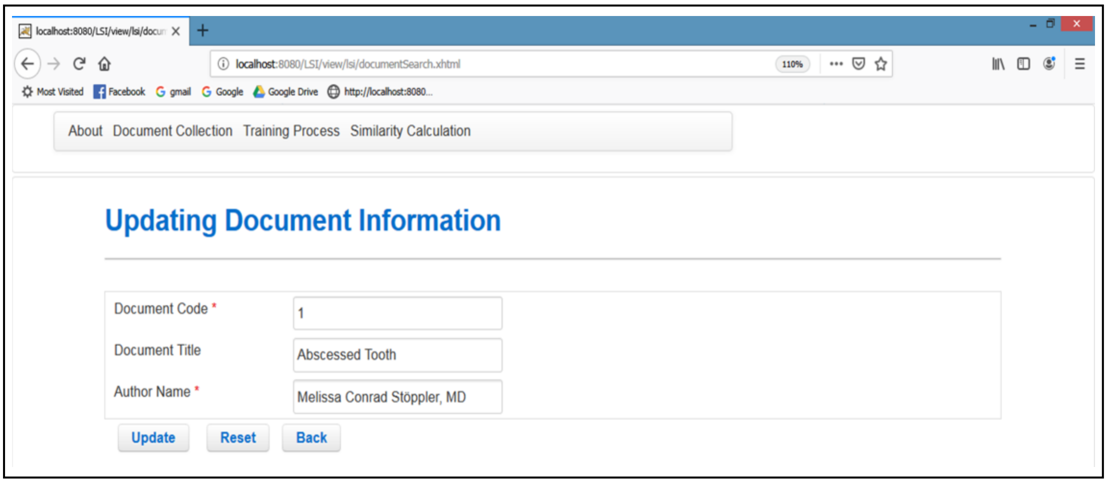
Figure 4.13 View all Document Information Page



Document Information

Document Code	1
Document Title	Abscessed Tooth
Author Name	Melissa Conrad Stöppler, MD
Document File	Download

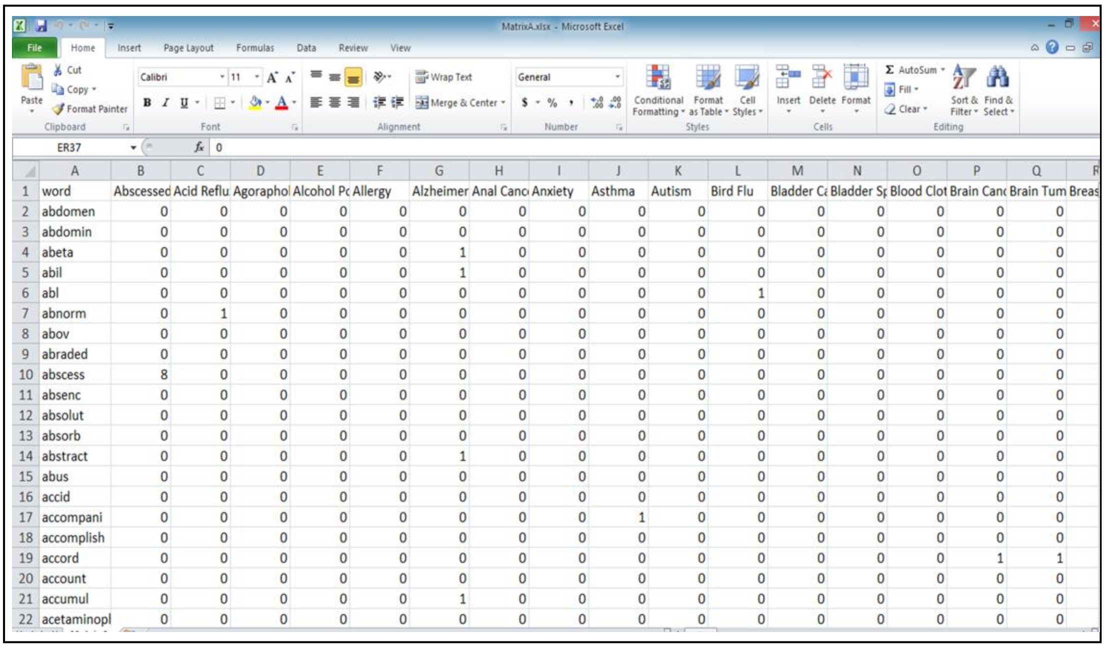
Figure 4.14 Document Information Page



Updating Document Information

Document Code *	1
Document Title	Abscessed Tooth
Author Name *	Melissa Conrad Stöppler, MD

Figure 4.15 Updating Docunment Information Page

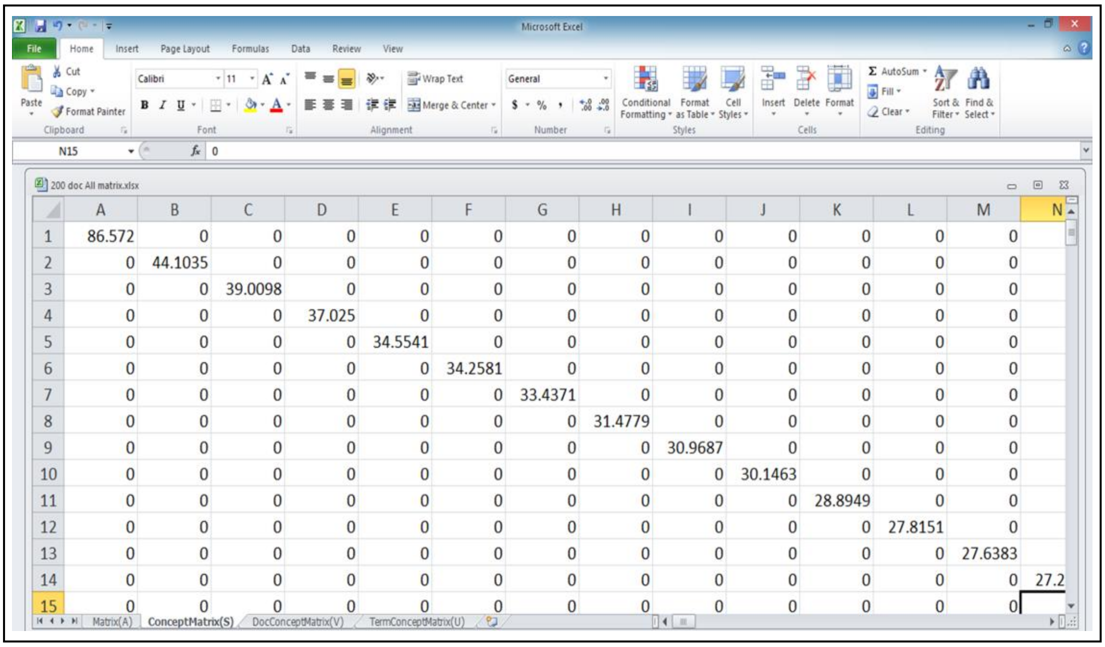


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	word	Abscessed	Acid Reflu	Agoraphol	Alcohol Pc	Allergy	Alzheimer	Anal Canc	Anxiety	Asthma	Autism	Bird Flu	Bladder Ca	Bladder Sp	Blood Clot	Brain Canc	Brain Tum	Breas
2	abdomen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	abdomin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	abeta	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
5	abil	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
6	abl	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
7	abnorm	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	abov	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	abraded	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	abscess	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	absenc	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	absolut	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	absorb	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	abstract	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
15	abus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	accid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	accompani	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
18	accomplish	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
19	accord	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	
20	account	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
21	accumul	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
22	acetaminopl	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 4.16 Viewing Excel File for Term-Document Matrix A

Figure 4.16 presents the original matrix A (term–document) matrix in excel format. There are 2286 rows and 200 columns. Row represents the terms and column represents the documents. The cell of row and column represents the number of existene of the terms in particular document. The cell valued "0" describes that a specific term is not occur in a particular document.

Figure 4.17, Figure 4.18 and Figure 4.19 represents the singular matrix(S), document-concept matrix(V) and term-concept matrix (U) in excel format.

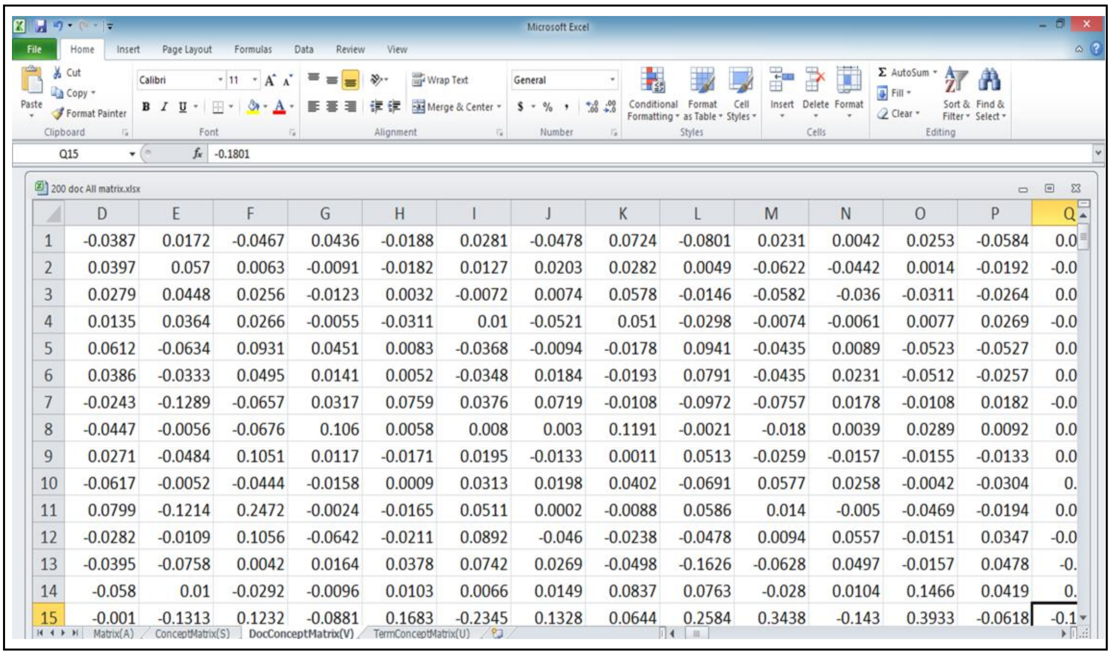


N15 f_x 0

200 doc All matrix.xlsx

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	86.572	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	44.1035	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	39.0098	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	37.025	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	34.5541	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	34.2581	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	33.4371	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	31.4779	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	30.9687	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	30.1463	0	0	0	
11	0	0	0	0	0	0	0	0	0	0	28.8949	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	27.8151	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	27.6383	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	27.2
15	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 4.17 Viewing Excel File for Singular Value Matrix S

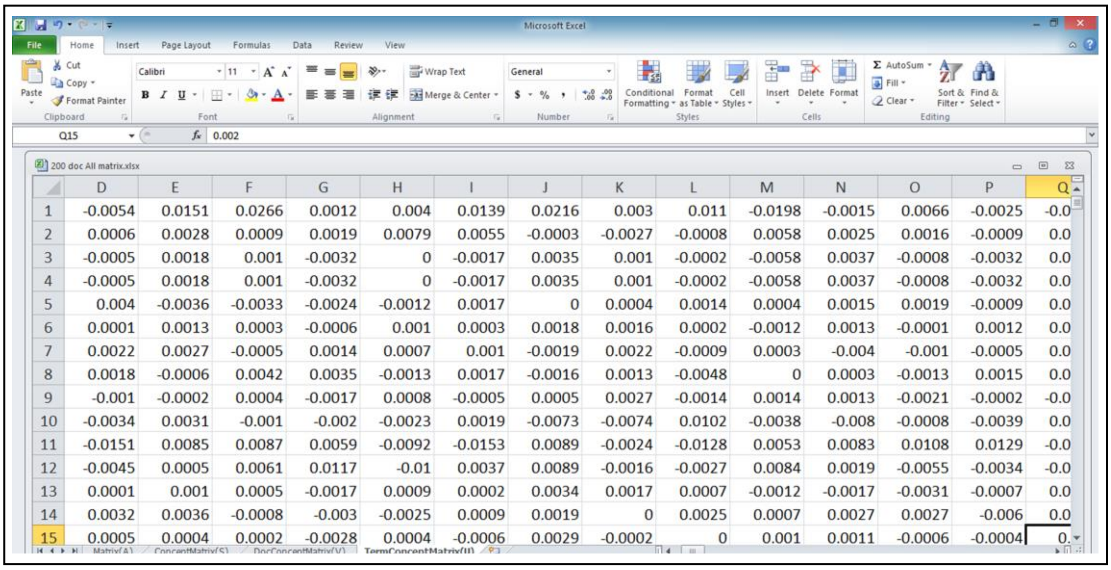


Q15 f_x -0.1801

200 doc All matrix.xlsx

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	-0.0387	0.0172	-0.0467	0.0436	-0.0188	0.0281	-0.0478	0.0724	-0.0801	0.0231	0.0042	0.0253	-0.0584	0.0
2	0.0397	0.057	0.0063	-0.0091	-0.0182	0.0127	0.0203	0.0282	0.0049	-0.0622	-0.0442	0.0014	-0.0192	-0.0
3	0.0279	0.0448	0.0256	-0.0123	0.0032	-0.0072	0.0074	0.0578	-0.0146	-0.0582	-0.036	-0.0311	-0.0264	0.0
4	0.0135	0.0364	0.0266	-0.0055	-0.0311	0.01	-0.0521	0.051	-0.0298	-0.0074	-0.0061	0.0077	0.0269	-0.0
5	0.0612	-0.0634	0.0931	0.0451	0.0083	-0.0368	-0.0094	-0.0178	0.0941	-0.0435	0.0089	-0.0523	-0.0527	0.0
6	0.0386	-0.0333	0.0495	0.0141	0.0052	-0.0348	0.0184	-0.0193	0.0791	-0.0435	0.0231	-0.0512	-0.0257	0.0
7	-0.0243	-0.1289	-0.0657	0.0317	0.0759	0.0376	0.0719	-0.0108	-0.0972	-0.0757	0.0178	-0.0108	0.0182	-0.0
8	-0.0447	-0.0056	-0.0676	0.106	0.0058	0.008	0.003	0.1191	-0.0021	-0.018	0.0039	0.0289	0.0092	0.0
9	0.0271	-0.0484	0.1051	0.0117	-0.0171	0.0195	-0.0133	0.0011	0.0513	-0.0259	-0.0157	-0.0155	-0.0133	0.0
10	-0.0617	-0.0052	-0.0444	-0.0158	0.0009	0.0313	0.0198	0.0402	-0.0691	0.0577	0.0258	-0.0042	-0.0304	0.
11	0.0799	-0.1214	0.2472	-0.0024	-0.0165	0.0511	0.0002	-0.0088	0.0586	0.014	-0.005	-0.0469	-0.0194	0.0
12	-0.0282	-0.0109	0.1056	-0.0642	-0.0211	0.0892	-0.046	-0.0238	-0.0478	0.0094	0.0557	-0.0151	0.0347	-0.0
13	-0.0395	-0.0758	0.0042	0.0164	0.0378	0.0742	0.0269	-0.0498	-0.1626	-0.0628	0.0497	-0.0157	0.0478	-0.
14	-0.058	0.01	-0.0292	-0.0096	0.0103	0.0066	0.0149	0.0837	0.0763	-0.028	0.0104	0.1466	0.0419	0.
15	-0.001	-0.1313	0.1232	-0.0881	0.1683	-0.2345	0.1328	0.0644	0.2584	0.3438	-0.143	0.3933	-0.0618	-0.1

Figure 4.18 Viewing Excel File for Document Concept Matrix (V)



Q15 0.002

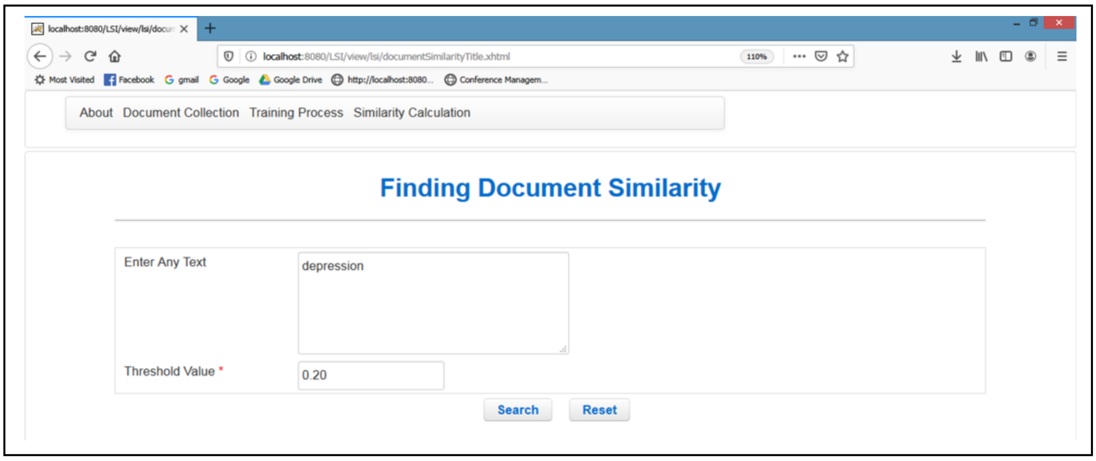
200 doc All matrix.xlsx

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	-0.0054	0.0151	0.0266	0.0012	0.004	0.0139	0.0216	0.003	0.011	-0.0198	-0.0015	0.0066	-0.0025	-0.0
2	0.0006	0.0028	0.0009	0.0019	0.0079	0.0055	-0.0003	-0.0027	-0.0008	0.0058	0.0025	0.0016	-0.0009	0.0
3	-0.0005	0.0018	0.001	-0.0032	0	-0.0017	0.0035	0.001	-0.0002	-0.0058	0.0037	-0.0008	-0.0032	0.0
4	-0.0005	0.0018	0.001	-0.0032	0	-0.0017	0.0035	0.001	-0.0002	-0.0058	0.0037	-0.0008	-0.0032	0.0
5	0.004	-0.0036	-0.0033	-0.0024	-0.0012	0.0017	0	0.0004	0.0014	0.0004	0.0015	0.0019	-0.0009	0.0
6	0.0001	0.0013	0.0003	-0.0006	0.001	0.0003	0.0018	0.0016	0.0002	-0.0012	0.0013	-0.0001	0.0012	0.0
7	0.0022	0.0027	-0.0005	0.0014	0.0007	0.001	-0.0019	0.0022	-0.0009	0.0003	-0.004	-0.001	-0.0005	0.0
8	0.0018	-0.0006	0.0042	0.0035	-0.0013	0.0017	-0.0016	0.0013	-0.0048	0	0.0003	-0.0013	0.0015	0.0
9	-0.001	-0.0002	0.0004	-0.0017	0.0008	-0.0005	0.0005	0.0027	-0.0014	0.0014	0.0013	-0.0021	-0.0002	-0.0
10	-0.0034	0.0031	-0.001	-0.002	-0.0023	0.0019	-0.0073	-0.0074	0.0102	-0.0038	-0.008	-0.0008	-0.0039	0.0
11	-0.0151	0.0085	0.0087	0.0059	-0.0092	-0.0153	0.0089	-0.0024	-0.0128	0.0053	0.0083	0.0108	0.0129	-0.0
12	-0.0045	0.0005	0.0061	0.0117	-0.01	0.0037	0.0089	-0.0016	-0.0027	0.0084	0.0019	-0.0055	-0.0034	-0.0
13	0.0001	0.001	0.0005	-0.0017	0.0009	0.0002	0.0034	0.0017	0.0007	-0.0012	-0.0017	-0.0031	-0.0007	0.0
14	0.0032	0.0036	-0.0008	-0.003	-0.0025	0.0009	0.0019	0	0.0025	0.0007	0.0027	0.0027	-0.006	0.0
15	0.0005	0.0004	0.0002	-0.0028	0.0004	-0.0006	0.0029	-0.0002	0	0.001	0.0011	-0.0006	-0.0004	0.

Figure 4.19 Viewing Excel File for Term Concept Matrix (U)

After processing the training phase, document similarity can be calculated as shown in Figure 4.20. In this page, there are two input boxes: one is to input the query terms by the user. The query terms may be phrases or one or more sentences. And the next input box is for threshold value for similarity to truncate the number of results as desired by the user. After inputting the inputs in input boxes, the search button must be clicked to retrieve the semeantically related documents for user query. The reset button is for next search process.

Figure 4.21 shows the result page of search button event from Figure 4.20. The documents with the highest similarity values are at the top list of the results. And if the user satisfied the similarity value for the search query, the user can also download the entire document.

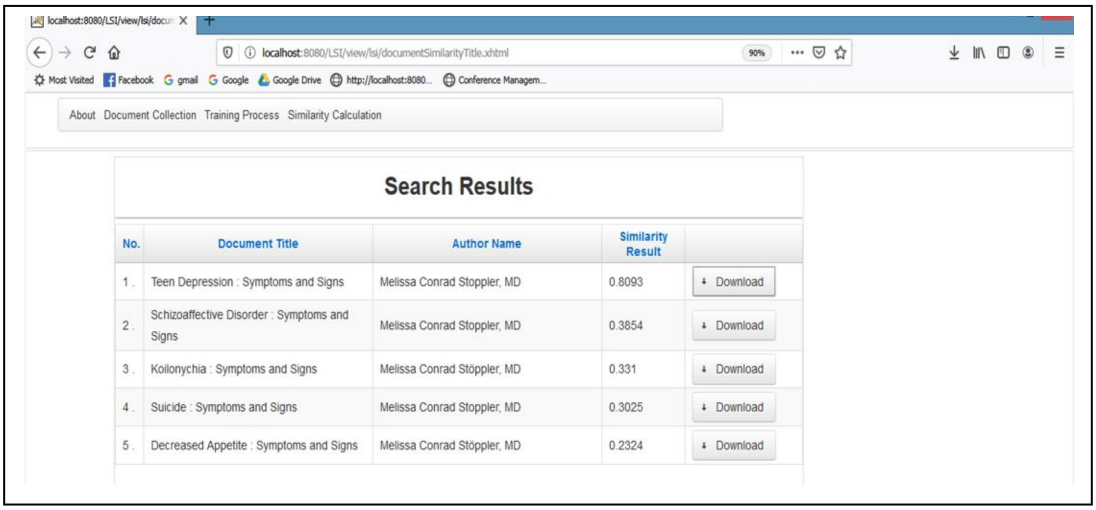


Finding Document Similarity

Enter Any Text: depression

Threshold Value *: 0.20

Figure 4.20 Finding Document Similarity Page



Search Results

No.	Document Title	Author Name	Similarity Result	
1 .	Teen Depression : Symptoms and Signs	Melissa Conrad Stoppler, MD	0.8093	Download
2 .	Schizoaffective Disorder : Symptoms and Signs	Melissa Conrad Stoppler, MD	0.3854	Download
3 .	Koilonychia : Symptoms and Signs	Melissa Conrad Stöppler, MD	0.331	Download
4 .	Suicide : Symptoms and Signs	Melissa Conrad Stoppler, MD	0.3025	Download
5 .	Decreased Appetite : Symptoms and Signs	Melissa Conrad Stöppler, MD	0.2324	Download

Figure 4.21 Viewing Similarity Results Page

CHAPTER 5
CONCLUSION

This thesis intends to develop retrieving semantically relevant documents in information retrieval system. The various components of this system are investigated and their contributions to the overall performance of the system are analyzed. In this chapter, the main contents of thesis are concluded. The advantages and limitation of the system and future work are suggested in this chapter.

5.1 Conclusion

The proposed system presents that the implementation is well suited to the retrieving semantically relevant documents by using Latent Semantic Indexing method. Latent Semantic Indexing is a useful technique to implement retrieval of documents on the basis of conceptual meaning and latent semantic analysis. This is the mathematical approach to retrieval and so it makes sure that the relevancy of the obtained documents is optimized. Not all the documents using generic key word search are relevant. This isn’t the case with latent semantic indexing as the documents that do not contain the key words in the query are also acquired, providing for relevancy.

The main purpose of this system is to store the documents collection and then retrieve the semantically relevant documents according to the user’s query. There is very tedious for matching the incoming query with the existing documents and is not grantee to retrieve the semantically related documents without using this system. This system can overcome the problem of term matching based information retrieval system. Therefore, this system can retrieve the most relevant documents and helps the people who want to find easily and correctly the information.

5.2 Advantages of the System

Latent Semantic Indexing (LSI) method adapts to represent large datasets and makes it usable for real world applications. LSI is capable of assuring acceptable results and much better than Vector Space model. It is a popular method that will get good community support and works well on diverse topics datasets. LSI uses Singular

Value Decomposition to reduce dimension and to remove noise. LSI can handle synonym problem.

The next advantage is due to similarity method. Cosine Similarity is best known technique for finding the similarity between queries to document. Cosine similarity is suitable for very large amount of data.

5.3 Limitation of the System

There is a little limitation in the proposed system. LSI is the computational overhead due to matrix factorization. LSI initially needs to construct a huge term by document matrix and then it has to perform a computational heavy SVD. When the term-by-document matrix is large in dimensions (thousands of rows/columns), factorization becomes memory inefficient. The decomposition algorithms require large quantities of processing power and extended processing times.

LSI passes only a partial solution of polysemy problem. It assumes that the same term means the same concept which causes problems that have multiple meanings depending on which contexts they appear in.

5.4 Further Extensions

Although there are some limitations, it can also be extended for this research. The proposed system is only tested by using the collected biomedical data as a dataset. So, this proposed system can be extended with other very large dataset.

A Few disadvantages of Latent Semantic Indexing can be overcome by Probabilistic Latent Semantic Analysis. Probabilistic Latent Semantic Analysis can handle polysemy problem. Detail description of Probabilistic Latent Semantic Analysis, which has not been included will be done as a future study for this paper.

AUTHOR'S PUBLICATIONS

[1] Chue Wut Yee, Zon Nyein Nway, *"Retrieving Semantically Relevant Documents using Latent Semantic Indexing"*, the Proceedings of the 10th Conference on Parallel and Soft Computing (PSC 2019), Yangon, Myanmar, 2019.

[2] Chue Wut Yee, Zon Nyein Nway, *"Implementing Information Retrieval System using Latent Semantic Indexing"*, the Second International Conference on Science, Technology and Innovation, Mandalay by IEEE, pp. 205-210, 2019.

REFERENCES

[1] A. Bellaachia and A. Mahajan, "Extraction of Text Summary Using Latent Semantic Indexing and Information Retrieval Technique: Comparison of Four Strategies", European Geothermal Congress, Volume. 2, pp. 453-464, 2004.

[2] A. Mirzal, "The Limitation of the SVD for Latent Semantic Indexing", IEEE International Conference on Control System,Computing and Engineering, 29 Nov – 1 Dec 2013.

[3] A. Moldovan, R.L.Bot and G. Wanka, "Latent Semantic Indexing for Patent Documents", International Journal of Applied Mathematics and Computer Science, pp. 551-56, January 2005.

[4] C.A. Kumar and S. Srinivas, "On the Performance of Latent Semantic Indexing-based Information Retrieval", Journal of Computing and Information Technology, Volume. 3, pp. 259-264, 2009.

[5] C. Aswani Kumar, M. Radvansky and J. Annapurna, "Analysis of a Vector Space Model, Latent Semantic Indexing and Formal Concept Analysis for Information Retrieval", Cybernetics and Information Technologies, Volume 12 No.1, 2012.

[6] D. Kalman, "A Singular Value Decomposition: The SVD of a Matrix", the American College Mathematic Journal, Volume. 27, No. 1, January 1996.

[7] D.S, Bhatia, "The Power of Latent Semantic Indexing in Review Retrieval", Software Engineering, Graduate Faculty of Texas Tech University, December 2016.

[8] J. Dobsa and B. Dalbelo-Basic, "Comparison of Information Retrieval Techniques: Latent Semantic Indexing and Concept Indexing", Journal of Information and Organizational Sciences, Volume 28, 1-2 November 2004.

[9] J. Geib, "Latent Semantic Indexing and Information Retrieval A quest with BosSE", Seminar for Computational Linguistics Institute of General and Applied Linguistics Heigelberg University, 18 January 2006.

[10] M. Al-Qahtani, A. Amira and N. Ramzan, "An Efficient Information Retrieval Technique for e-Health Systems", IEEE International Conference on System Signals and Image Processing, pp. 257-260, pp. 257-260, 2015.

[11] M. Fahsi and S.M. Benslimane, "Studying the Effects of Conflicting Tokenization on LSA Dimension Reduction", IEEE International Conference on Multimedia Computing and Systems, 14-16 April 2014.

[12] M.L. Fleur and F. Renstrom, "Conceptual Indexing using Latent Semantic Indexing", Institution for Information Technology, IT. 15067, pp. 58, 2015.

[13] M.W. Berry and R.D. Fierro, "Low Rank Orthogonal Decompositions for Information Retrieval Applications", Numerical Linear Algebra with Applications, Vol. 3, Issue 4, pp. 1-27, 1996.

[14] Nan Ei Khaing, "Similarity based Retrieval of Documents using Jaccard Coefficient Similarity Method", M.C.Sc Thesis, University of Computer Studies, Yangon, December 2018.

[15] R. Anita, C.N. Subalalitha, A. Dorle and K. Venkatesh, "Semantic Search Using Latent Semantic Indexing and WordNet", ARPN Journal of Engineering and Applied Sciences, 2017-2018.

[16] R.B- Yates and B.R- Nieto, "Modern Information Retrieval", British Library Cataloging-in-Publication Data, 1999.

[17] S. Deerwester, S.T. Dumais, T.K. Landauer and R. Harshman, "Indexing by Latent Semantic Analysis", Journal of the American Society for Information Science, Vol. 41, pp. 391-407, 1990.

[18] T.G. Kolda and D.P. O'Leary "A Semi-discrete Matrix Decomposition for Latent Semantic Indexing in Information Retrieval" ACM Transactions on Information Systems, Vol. 16, No. 4, pp. 322–346, October 1998.

[19] T.K. Landauer, P.W. Foltz and D. Laham, "An Introduction to Latent Semantic Analysis", Discourse processes, Vol. 25, Issue. 2-3: Quantitative Approaches to Semantic Knowledge Representations, pp. 259-284, 1998.

[20] W.B. Croft, D. Metzler and T. Strohman, "Information Retrieval in Practice", Pearson Education, Inc, 2015.