ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF INFORMATICS DEPARTMENT OF INFORMATION SCIENCE

AMHARIC TEXT RETRIEVAL: AN EXPERIMENT USING LATENT SEMANTIC INDEXING (LSI) WITH SINGULAR VALUE DECOMPOSITION (SVD)

By

TEWODROS HAILEMESKEL GEBERMARIAM

A thesis submitted to

the School of Graduate Studies of Addis Ababa University

in partial fulfillment of the requirements for the Degree of Master of Science in Information Science

July 2003

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF INFORMATICS DEPARTMENT OF INFORMATION SCIENCE

AMHARIC TEXT RETRIEVAL: AN EXPERIMENT USING LATENT SEMANTIC INDEXING (LSI) WITH SINGULAR VALUE DECOMPOSITION (SVD)

BY

TEWODROS HAILEMESKEL GEBERMARIAM

Signature of the Board of Examiners for Approval

ACKNOWLEDGMENT

I would like to thank every one at the Faculty of Informatics, Department of Information Science, AAU, who has encouraged and assisted me throughout the completion of this thesis. I would particularly like to thank my advisors, W/t Saba Amsalu and Ato Kibur Lisanu. Thank you for your patience, your support and your time. I would also like to thank my friend Henok for his unreserved assistance in going along with Delphi. Lastly, I would like to thank my family for supporting me in so many ways.

DEDICATION

To my family

TABLE OF CONTENTS

ACKNOWLEDGMENT.....iv

DEDICATION.....v

TABLE OF CONTENTS vi

LIST OF FIGURES xi

LIST OF TABLES xii

LIST OF APPENDIX. xiii

ABSTRACT xiv

CHAPTER ONE.....1

INTRODUCTION.....1

1.1 BACKGROUND.....1

1.1.1 Vector Based Information Retrieval Method and Latent Semantic Indexing).....2

I. The Vector Space Model (VSM).....2

II. Latent Semantic Indexing IR 4

1.2 STATEMENT OF THE PROBLEM.....6

1.3 OBJECTIVES OF THE STUDY.....10

1.3.1 General Objective.....10

1.3.2 Specific Objectives.....10

1.4 METHODOLOGY.....11

1.4.1 Literature Review.....11
1.4.2 Data Source selection.....12
1.4.3 Experimentation method.....12
1.4.4 Evaluation Method.....13
1.5 SIGNIFICANCE OF THE STUDY.....14
1.5 SCOPE AND LIMITATION OF THE STUDY.....14
1.6 OUTLINE OF THE THESIS.....15

CHAPTER TWO.....16
LITERATURE REVIEW.....16
2.1 INTRODUCTION.....16
2.2 AUTOMATIC INDEXING.....17
2.3 AUTOMATIC TERM EXTRACTION AND WEIGHTING.....18
2.3.1 Index Term Extraction.....18
2.3.2. Term Weighting.....20
I. Term Frequency Weighting Function.....21
II. Inverse Document Weighting Function.....21
III. Signal-Noise (Log-Entropy) Weighting Function.....22
IIII. Term Discrimination Value Weighting Function.....23
2.4 QUERY MATCHING.....25
2.5 IR SYSTEM EVALUATION.....27
2.6 THE AMHARIC WRITING
SYSTEM.....299

2.6.1 The Amharic Alphabets.....29
2.6.2 The Punctuations.....30
2.6.3 The Amharic Number system.....30
2.6.4 Problems of Retrieving Amharic text.....31
I. Redundancy of Some Characters.....31
II. Formation of Compound Words.....32
III. Existence Irregular Spelling.....33
2.7.6 Amharic Software.....34

2.7 LATENT SEMANTIC INDEXING (LSI) IR.....35
2.7.1 Shortcomings of Term Matching Retrieval Systems.....35
2.7.2 Existing Approaches to Solve the Problem of Synonym and Polysemy.....36
2.7.3 Latent Semantic Indexing Approach.....39
2.7.4 Query Representation.....43
2.7.5 Review of Related Researches.....44
I. LSI: In Information Retrieval.....44
II. LSI: In Information Filtering.....48
III. LSI: In Cross-Language Retrieval.....50

CHAPTER THREE.....52

DESIGN, DEVELOPMENT AND TESTING OF THE PROTOTYPE LSI RETRIEVAL MODEL.....52

3.1 INTRODUCTION.....52

3.2 DESCRIPTION OF THE PROTOTYPE SYSTEM.....53

3.3 SELECTING DOCUMENT PART FOR INDEXING.....56

3.4 DOCUMENT AND QUERY PRE-PROCESSING.....56

3.4.1 The Natural Language Pre-processing.....57

I. Identification of Individual Words.....57

II. Changing Redundant Character to a Common Form.....59

III. Removal of Common Words.....60

IIII. Term by Document Matrix Generation.....62

3.4.2 The Numerical Data-Preprocessing/Term Weighting.....63

3.5 K-DIMENSIONAL SINGULAR VALUE DECOMPOSITION (SVD).....65

- Choosing the Number of Dimensions.....66

3.6 QUERY PROJECTION, MATCHING AND RANKING OF RELEVANT DOCUMENTS.....68

3.6.1 Query Projection.....68

3.6.2 Matching and Ranking of Relevant Documents.....69

3.7 TESTING.....71

3.8 DISCUSSION.....76

CHAPTER FOUR.....79

CONCLUSION AND RECOMMENDATION.....79

4.1 CONCLUSION.....79

4.2 RECOMMENDATION.....81

REFERENCE.....83

APPENDIX.....91

DECLARATION.....99

LIST OF FIGURES

Figure 3.1 The design of the prototype LSI model.....54

Figure 3.2 Precision at recall levels of 0.25, 0.50 and 1.00 67

Figure 3.3 Average recall-precision graph for two indexing methods74

LIST OF TABLES

Table 1.1 Example of term by document matrix.....2

Table 3.1 Sample Stop-words for two categories: common terms of the language and news specific 62

Table 3.2 Sample raw term by document matrix.....63

Table 3.3 Sample weighted term by document matrix.....65

Table 3.4 Dimension vs. Average precision at recall levels of 0.25, 0.50 and 0.75 72

Table 3.5 Some characteristics of the test collection.....76

Table 3.6 Sample queries and the ranked output for the queries.....77

Table 3.7 Precision for standard recall levels for the sample queries of table 4.4 77

Table 3.8 Average recall-precision results for two indexing Methods 78

Table 3.9 Sign test for LSI and Vector space retrieval approaches..80

LIST OF APPENDIX

Appendix: I The Amharic character set (Bender *et al.*, 1976). 96

Appendix: II Amharic numbers 97

Appendix: III List showing the symbols used in Visual Ge'ez font for Amharic Fidel.....98

Appendix: IV The queries and their corresponding relevant document numbers 100

Appendix: V Precision at Standard recall points for the LSI Method ...102

Appendix: VI Precision at Standard recall levels for the Vector Space Method 103

ABSTRACT

The increase in the amount of electronic information has caused increasing need for efficient information retrieval techniques. Most techniques to retrieving textual materials from databases depend on exact term match between terms in user's query and terms by which documents are indexed. However, since there are usually many ways to express the same concept, the terms in the user's query may not appear in a relevant document. Alternatively, many words can have more than one meaning. Due to these facts term matching methods are likely to miss relevant documents and also retrieve irrelevant ones (Dumais, 1992; Berry, Dumais & Letsche, 1995). The Latent Semantic Indexing (LSI) technique of information retrieval can partially handle these problems by organizing terms and documents into a "semantic" structure more appropriate for information retrieval. This is done by modeling the inherent higher-order pattern in the association of terms with documents.

In this thesis, the potential of LSI approach in Amharic text retrieval is investigated. 206 Amharic documents and 25 queries were used to test the approach. Automatic indexing of the documents resulted in 9256 unique terms which are not in the stop-word list used for the research. A 110-factor SVD of the term by document matrix is used for indexing and retrieval. Finally, the performance of the LSI approach is compared with the standard vector space. Except at one standard recall level (0.80) precision of the LSI approach was above that of the standard vector space.

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

As storage becomes more ample and less expensive, the amount of information retained by businesses and institutions is likely to increase. Searching that information and deriving useful facts, however, will become more awkward unless new techniques are developed to automatically manage the data (Baeza-Yates, 1999; Salton, 1983; Chowdhury, 1999). The pace by which documents are created is much faster than the growth rate of the technology that manages and makes them accessible to the users (Rijsbergen, 1996). The result of this phenomenon is that relevant information gets buried since it is never revealed.

Therefore, information retrieval systems, which can search large information collection and return only the relevant information to user's information need, will become more and more important. Users often describe their information need using a query which consists of a number of words. Information retrieval systems compare the query with the documents in the collection and return the documents that are likely to satisfy the information need.

In information retrieval systems, representation of user's queries and documents in a way that facilitates effective retrieval is a central issue (Salton, 1983). Several techniques have been devised and integrated into information retrieval systems

(Baeza-Yates, 1999). Latent Semantic indexing (LSI) information retrieval is a new technique which is based on vector representation of documents and user's query.

1.1.1 Vector Based Information Retrieval Method and Latent Semantic Indexing

I. The Vector Space Model (Vsm)

Standard vector space model is one of the classic models of information retrieval. In this model, a vector is used to represent each item or document in a collection. Each component of the vector reflects a particular concept, keyword, or term associated with the given document. The value assigned to that component reflects the importance of the term in representing the semantics of the document. Typically, the value, which is often called the weight, is a function of the frequency with which the term occurs in the document and/or in the document collection as a whole (Berry, Drmac & Jessup, 1999; Carmel & Soffer, 2003; Salton, 1983).

A document collection containing a total of 'd' documents described by ' t' terms is represented as a 't X d' term-by-document matrix. The columns of the matrix are the document vectors, and the rows of the matrix are the term vectors.

The term by document matrix is represented as follows:

Term list	Doc 1	Doc 2	Doc3	Doc4	.....
Term 1	W11	W12	W13	W14	.....
Term 2	W21	W22	W23	W24	.....
Term 3	W31	W32	W33	W34	.....
Term 4	W41	W42	W43	W44	.....
. ....	....	...	...	.....	.....

Table 1.1 Example of term by document matrix.

From the IR viewpoint (Carmel & Soffer, 2003; Salton, 1983), this kind of representation enables us to exploit the geometric relationship (distance and/or angle) between term vectors and document vectors.

In the vector space IR model, a query is a set of terms, perhaps with weights, represented just like a document (Berry, Drmac, & Jessup, 1999; Salton, 1983). It is likely that many of the terms used to represent the document collection do not appear in the query, meaning that many of the query vector components are zero. In this model, the documents returned as a response to a user query are those geometrically closest to the query according to some similarity measure (Kolda & O'leary, 2002; Baeza-Yates, 1999; Salton, 1983).

Similarity in vector space model is determined by using associative coefficients based on the inner product of the document vector and query vector, where word overlap indicates similarity (Husband, Simon & Ding, 2001; Rijsbergen, 1996). The inner product is usually normalized. The most popular similarity measure is the cosine coefficient, which measures the angle between the document vector and the query vector (Berry, Drmac & Jessup, 1999; Salton, 1983). There are other measures that are widely used in the literature to measure vector similarities including the Jaccard's and Dice's similarity measures (Chowdhury, 1999; Salton, 1983).

The standard vector space model of information retrieval suffers from the problem of literal term mismatch (Berry, Dramc, Jessup, 1999). The model treats the terms as unrelated objects in the semantic space. That is, if the query and documents in text collection share no common words, may be due to variability in word choice between users and authors, zero similarity values will be the result and no document will be retrieved, even some documents may be actually relevant to the query. Another drawback of the vector space model is that for large text collection the term-by-document matrix is always a huge and sparse one, which needs a lot of disk storage space (Penenovic, 1997, Berry, Drmac & Jessup, 1999).

Over the years, alternative modeling paradigms, which are geared towards solving these drawbacks of standard vector space models, have been proposed. The generalized vector, the latent semantic indexing and the neural network models are the major once (Baeza-Yates, 1999).

II. Latent Semantic Indexing IR

LSI information retrieval model is a variant of the standard vector retrieval model (Berry, Dumais & O'Brien, 1995; Baeza-Yates, 1999) in which the dependencies between terms are explicitly taken into account in the representation of documents and queries. This is done by simultaneously modeling all the interrelationships among terms and documents. Through the pattern of co-occurrences of words, LSI is able to infer the hidden or "latent" structure of relationships between documents and words (Dumias, 2002; Deerwester et al., 1990).

Latent semantic indexing often uses statistical techniques called Singular Value Decomposition (SVD) to capture this latent structure of word usage in the documents of the collection (Berry, Dramc & Jessup, 1999; Ozel, 2002). Singular value decomposition uses the term by document matrix produced in any vector space model to generate the indexing spaces. The term by document matrix is decomposed and reduced in dimension to a product of three separate matrices. The model uses these three matrices to store all the information of the text collection and do the retrieval. By using the reduced dimensional representation, the disk space usage is reduced and much of the problems that arise due to the variability in word usage between users and authors are eliminated. The major advantage of LSI is that it uses concepts or topics instead of individual words to index and retrieve the documents, consequently allowing a relevant document to be retrieved even when it shares no common words with the query (Berry, Dramc & Jessup, 1999).

Experiments in English language (Berry, Dumais & Letsche, 1995; Deerwester, et al., 1990) showed that LSI retrieval model performs better than exact term matching techniques, such as the standard vector space models. This is especially true in situations where documents and queries share no terms due to the possibility of expressing the same concept using different words.

Amharic, the official language of the federal government and a language most widely used in Ethiopia, is a language that is characterized by this feature of synonyms (Bender, et al., 1976). The inspiration for this research comes from the desire to test performance of latent semantic indexing in Amharic text retrieval.

1.2 STATEMENT OF THE PROBLEM

A lot of valuable information is being produced in Ethiopia, most of them written in Amharic. The documents contain information related to: research in many fields; particularly agriculture and water resource development; information on the development of the tourist and business sectors; government policies; and bulk of information produced by offices in day to day work. Informative bulletins, magazines and newsletters are also regularly produced by most government ministries, UN agencies, and NGOs.

Nonetheless, there is no a deep-rooted culture of information exchange and dissemination established in the country. The lack of accurate and fast accessing method for relevant information is acknowledged as a major hurdle affecting the success and quality of research and development activities, trade and industry (ICT Focus, Vol1, Issue 6).

Since for most people English is a second or third language the ability of information storage and retrieval systems to represent, store and retrieve Amharic documents written in Ge'ez alphabet is important. Although small in number, some experiments

on information processing on Amharic documents have been made in several areas. Some of the experiments carried in this area are: The application of information retrieval techniques to Amharic documents (Saba, 2001; Bethlehem, 2002); Automatic Classification of Amharic documents (Zelalem, 2001); the application of OCR techniques on computer printout, type written and handwritten documents (Worku, 1997; Ermias, 1998; Dereje, 1999; Million, 2000 and Nigussie, 2000, Yaregal, 2002); Amharic Word Parsing (Abyoit, 2000); and others.

Saba's work uses Extended Boolean technique, which is a Boolean model with vector functionality. Bethlehem's work, on the other hand, is based on N-gram indexing using vector space model for representation and retrieval. Considering the drawbacks of the standard vector space model and the high number of word variants in Amharic language, concept based retrieval systems, such as LSI retrieval system, are likely to perform more effectively than term matching systems.

In Amharic language lexical variation is very common (Bender et al., 1976). Brief examination of some of the Amharic dictionaries (example, those written by Kesate-Birhane Tessema and by Desta Teklewolde) gives indications of this characteristic of the language. A study conducted in four Amharic speaking regions in the North and North central part of the country (ibid) identified different types of lexical variations.

In the study, some words are found to be used with different meaning in different regions. For instance, the word 'zämän' (ዘመን) is used to mean 'year' in Gojjam, where as in Addis Ababa and its environs it is used to mean 'long period of time'. Some genuine dialectal difference between pairs of synonyms is also observed, in which one of the synonym is used to the exclusion of the other in a given area. For example, 'kärrä' and 'billäwä' are used in exclusion in different regions to refer to 'knife'. There are many words that can have different meaning by stressing some characters in the word while reading. For example, the word 'Wana' (ዋና) which means 'swimming' can have another meaning 'the main' when 'n' is stressed.

These characteristics of the language causes inconvenience for exact term match retrieval systems, such as standard vector space and Boolean models, as different people with different background, different needs or different linguistic habit use different words to express the same concept.

Information retrieval systems that are based on exact term matching traditionally dealt with the problem of synonymy by using term expansion and thesaurus development. Of course, these techniques are helpful in improving recall, but they lead to decrement of precision, due to polysemy (word with different meaning). That is because the words generated from term expansion or from a thesaurus may have more than one meaning. Thus, using those additional terms can only improve recall, but will also retrieve documents irrelevant to the original query, decreasing precision (Dumias, 2002, Witter & Berry, 2001).

Dealing with the problem of polysemy has been found more difficult as it involves ambiguity inherent to natural languages (Hong, 2000; Berry, Dramc & Jessup). One approach is to use a controlled vocabulary (a specified set of vocabulary from which to choose index terms) to resolve ambiguity in query terms, but this has the obvious problem of usability, since a controlled vocabulary is likely to be unnatural and difficult to remember (Salton, 1983). In addition to that, the controlled vocabulary might not even be in a desired document, meaning that the document may not have been indexed with the correct term from the vocabulary.

LSI partially overcomes some of the deficiencies of exact term matching retrieval systems and provides a way of dealing with synonymy and polysemy automatically without the need for manually constructed thesaurus, a grammar, or ontology (Berry, Drmac & Jessup, 1999). As a result, it reduces costs related to the development, implementation, and maintenance of these auxiliary structures.

Because it operates on the basis of concepts, not keywords; searches are not constrained by the specific words that users choose when they formulate queries. By using statistical techniques, like singular value decomposition, the LSI model can retrieve relevant documents even when those documents do not share any words with a query (Deerwester, et al.; 1990). Thus, this research is initiated by the desire to examine and explore these capabilities of LSI in Amharic text retrieval.

1.3 OBJECTIVES OF THE STUDY

The general and specific objectives of the research are discussed in this section:

1.3.1 General Objective

The general objective of this study is to examine the potential of Latent Semantic Indexing technique in the retrieval of Amharic text documents and compare its performance with that of exact term matching technique, the standard vector space.

1.3.2 Specific Objectives

To accomplish the aforementioned general objective, the following specific objectives are formulated.

- Review concepts in automatic indexing and vector representation of documents and user queries.
- Review literature on Latent Semantic Indexing and Singular Value Decomposition in the context of text retrieval.
- Review the features of Amharic writing system pertinent to the purpose of this research.
- Setup the test set by selecting Amharic documents and queries, and collecting relevance judgment from expert in the particular area to identify which documents are relevant to a specific query.
- Apply the classic vector space method to the selected Amharic documents and queries in order to derive indexing terms and represent the documents as a vector of term by document matrix.
- Apply Singular value decomposition to the term-document matrix in order to decompose the matrix into a concept space of K dimensions. The value of K

is determined by evaluating the performance of the LSI space for different values of K, and selects the one that performs best.

- Use the developed K-dimensional concept space for the purpose of retrieving relevant documents to the users' queries.
- Use the recall and precision measures to evaluate the performance of the system.
- Compare the result with the performance results obtained for standard vector space method.
- Draw conclusions and forward recommendations for further study

1.4 METHODOLOGY

The methods employed to achieve the above stated objectives of the research are presented below.

1.4.1 Literature Review

Extensive literature review was conducted to get deeper understanding on information retrieval systems and in particular on latent semantic indexing approach to text retrieval. A review of literature was also conducted to get familiarized with the basic Amharic text features in relation to computer representation and retrieval. Printed materials like books, journal articles, and previous related research work as well as electronic materials on the Web have been consulted for this purpose.

1.4.2 Data Source selection

The test data are 206 Amharic local news articles available in electronic form. The news articles are obtained from the archive on the Web site of Walta information center. Walta Information Center is a government information center that produces and distributes news for broadcast over television and radio (Saba, 2001).

The fact that the news articles are easier to access, in sufficient amount and in electronic form, was the major motive to use them. The entire article (the title and the body of the news) was used for indexing, because the entire text of the articles is better representative of the content of the news. Besides, the small size of the news articles (in most cases half a page) supported the decision to use the entire document.

1.4.3 Experimentation method

The process of developing a latent semantic indexing model was broken into three phases:

i) Pre-processing and indexing (this includes extracting terms, term-document matrix generation and calculating term weighting)

ii) K-dimensional Singular Value Decomposition (SVD) (K represents the dimension of the LSI space)

and iii) Query Projection, Matching and Ranking of Relevant documents.

A program code using Borland Delphi 5.0 is developed to automatically generate index terms. Delphi allows writing Windows programs more quickly and easily. Its string manipulation facilities were the major incentives to use Delphi programming language.

The weighting and all the activities in the last two phases are done on Matlab software Release 12. The name MATLAB is an abbreviation for 'Matrix Laboratory'. It is an interactive, matrix-based system for scientific and engineering calculations. Matlab was chosen because it is freely accessible and suffices for this research.

1.4.4 Evaluation Method

In information retrieval, the classic recall and precision are often used to measure the performance of information retrieval systems. Recall is the fraction of the relevant documents that are returned by the system. Precision is the fraction of relevant documents in the set of returned documents (Salton, 1983; Van Rijsbergen, 1996; & Baeza-Yates, 1999). In this research too, Precision-recall curves are used to evaluate the performance of the system. Precision-recall curve reflects precision at different standard recall levels.

1.5 SIGNIFICANCE OF THE STUDY

Besides being an academic exercise to carry out the requirement of the Masters program, this research is believed to produce results that can indicate the merit of applying latent semantic indexing in Amharic text retrieval. The technique can be applied to any problem where there is a need to retrieve Amharic documents from a large collection. Especially, the LSI model can be more effective in situations where high recall is need and text description are short. The method can also be extended to other application on the Amharic language like information filtering. The research is also believed to have significant contribution in an attempt to use latent semantic indexing for cross-language retrieval in which Amharic is one of the components

1.6 SCOPE AND LIMITATION OF THE STUDY

Due to the requirement of larger time and memory space in SVD computation the number of test documents used in this research is small (206). Through the use of external sever (that store and operates on data) and Matlab's object oriented features one can handle large number of documents.

In this research, stemming of index terms is not done. By its very nature LSI partially handles the problems that arise due to morphological variation of index terms. If words with the same stem are used in similar documents they will be have similar vectors in the reduced LSI space (Berry, Dumais & Letsche, 1995). Of course, had stemming been used, it would have at least reduce the number of

index term (if not improved the performance) and in turn allow to add some additional documents into the test collection.

1.7 OUTLINE OF THE THESIS

The remainder of this thesis is organized as follows; chapter two is devoted for literature review. General concepts on information retrieval and on Amharic writing system applicable to the research are discussed. Latent semantic indexing and related researches on latent semantic indexing are also reviewed. Chapter three describes the test collection used in this thesis, discusses the development of prototype LSI model and the evaluation metrics. Chapter four discusses conclusions and recommendations.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter is broadly divided in to three sections. The first section discusses concepts related to indexing and the series of activities involved in document and query indexing process. Concepts like term extraction and term weighting are briefly introduced. Evaluation of information retrieval system is also dealt with. The two conventional measures of effectiveness of IR systems, recall and precision, are discussed in detail.

In the second section the features of Amharic writing system pertinent to information retrieval is discussed. The Amharic alphabets, numbers and punctuation marks are also introduced.

The third section discusses about LSI (Latent Semantic Indexing) technique of information retrieval. It presents the basic tasks involved in LSI information retrieval, such as: Singular Value Decomposition (SVD) of term by document matrix and query representation. Related works are also reviewed.

2.2 AUTOMATIC INDEXING

Most computer-based information retrieval systems store only a representation of the documents or queries. Describing text documents based on their contents, called *indexing*, is an important process in an IR system (Salton, 1983). When the process is done with the help computer, it is often called *automatic indexing*. In this thesis the terms 'indexing' and 'automatic indexing' are used interchangeably.

The indexing process in IR systems involves analysis of text documents in order to assign identifiers that are capable of representing the content of the information items. These identifiers are supposed to allow obtaining access to an item whenever it is wanted (Salton, 1983; Rijsbergen, 1999).

The original idea that suggests the use of certain terms automatically extracted from document text to represent document content was forwarded by Luhn (Salton, 1983). Still today, the search engines that operate on the Internet index the documents based on this principle (Baeza-Yates, 1999). The task of indexing is therefore concerned with identifying those words that properly reflect the content of a document they are taken from. This is because, all words in a text are not good in representation of the content of a document and also words that are good do not contribute equally in defining the content (Salton, 1983; Rijsbergen, 1999).

2.3 AUTOMATIC TERM EXTRACTION AND WEIGHTING

Automatic indexing process can be considered to be composed of two major tasks: first, assigning terms or concepts capable of representing the content of each document in the database, *term extraction* and second, assigning to each term a weight or value that signifies its importance for purposes of content description, *term weighting* (Radu, 2001; Salton, 1983).

2.3.1 Index Term Extraction

The task of extracting content descriptors itself is composed of a sequence of activities (Salton, 1983):

- **Lexical Analysis**

Lexical analysis involves identification of individual words that constitute the input text. It also incorporates a sort of text cleaning process, such as conversion of abbreviations and acronyms into their full text, conversion of cases, removal of numbers and symbols (example $, @, %, #, etc) which do not make up good index terms (Matsumura, 2002). Lexical analysis process produces candidate index terms that can be further processed, and eventually selected as index terms.

- **Use of stop-List**

The words of a document text do not have equal value for indexing purpose. High-frequency function words often appear in almost all documents in a

collection. Function words are lexical devices that serve grammatical purposes and do not refer to objects or concepts of the world (Moens, 2000). They often belong to syntactic classes such as articles, pronouns, particles, and prepositions. These words are characterized by poor ability to identify relevant items and to distinguish them from the non-relevant once since they exist in every document (Salton, 1983; Rijsbergne, 1996; Baeza-Yates, 1999). Thus, they could be removed from the text by comparing each term in the text with a list of common words developed for a particular language and some times for a particular domain.

- **Stemming**

The process of stemming is designed to reduce different morphological variants of words to their stem/root form. The basic assumption behind stemming is that words with the same stem are semantically related and have the same meaning to the user of the text (Salton, 1983; Rijsbergen, 1996).

In IR environment, stemming is performed in order to boost the recall of the system (Croft, 1995; Salton, 1983). Since different variants of the same word, in both documents and queries, are mapped to the same stem the chance of matching between the query and documents will be enhanced. Additionally, stemming reduces the number of index terms and consequently the size of the vocabulary in the text representations, having the benefit of reducing storage cost.

The effect of stemming on retrieval effectiveness has been evaluated on different languages (In English Frakes, 1992 and Hull, 1996; In Dutch Klraij & Puhimann, 1996 cited in Moens, 2000). The generalization reached from the experiments is that stemming either has a positive or no effect on retrieval effectiveness.

2.3.2. Term Weighting

All the terms included in the index list are not equally important in reflecting the content of a specific text. An importance indicator or a term weight should be associated with each index term. Term weighting is important to select good index terms for inclusion in the text representation or to better discriminate the index terms when matching a query in a retrieval environment (Rijsbergen, 1999; Salton, 1983). Weighting index terms increase the precision of retrieval (Dumais, 1992).

Many weighting functions have been proposed and tested (Dumais, 1992). Most of the weighting functions rely upon the distribution pattern of the terms with in a document as well as in the document collection as a whole. The functions use these distribution statistics to compute the weight of each term in each document and query (Salton, 1983). The major weighting functions that are widely used in IR environment are discussed below.

I. Term Frequency Weighting Function

The basic premise for this function is that the degree of treatment of a topic in a text is reflected by the frequency of occurrence of terms in it (Spark Jones & Willet, 1997; Salton, 1983). A content term (after stop-word removal) that appears more frequently in a text is more important than a rarely appearing term. Hence, term frequency (tf) weighting measures the frequency of an index term in the document text.

Raw term frequency measure does not give any distinction between the occurrences of a rare term in a short text and in a long text. However, the occurrence of a rare term in a short text is more significant than its occurrence in a long text. The logarithmic term frequency is often used to smooth this bias (Spark Jones & Willet, 1997).

$$Wt_i = \text{Log}\ (tf_i + 1) \quad \text{(equ 2.1)}$$

i.e. the weight of term i is the logarithm of its occurrence frequency

II. Inverse Document Weighting Function

It is perhaps the most commonly used term weighting function. The function is based on the notion that a term that occurs in many documents is less important. The more rarely a term occurs in individual texts the more discriminating that term is (Rijsbergen, 1999; Spark Jones & Willet, 1997; Salton, 1983). Specifically, the number of documents in which a term i occurs is counted. This gives the document frequency df(i) of term i, representing the number of

documents to which term i is assigned. Then, a measure of the inverse document frequency can be written as

$$\mathbf{idf(i) = Log\ N/df(i) + 1} \quad \text{(equ 2-2)}$$

where N is the total number of documents

Hence, a composite expression measuring the weight of a term i in document j (W_{ij}) is given by;

$$\mathbf{W_{ij} = tf(i, j) * Log_2\ N/df(ti) + 1} \quad \text{(equ 2-3)}$$

This scheme gives more importance to less frequent and more discriminating terms.

III. The Signal-Noise Ratio (Log-Entropy) Weighting Function

The signal-noise ratio is another popular term weighting scheme based on Information Theory (Spark Jones & Willet, 1997; Salton, 1983). This scheme treats broad and nonspecific terms as noisy and gives them low weight. It favors those terms that distinguish specific documents from the remainder of the collection (Rijsbergen, 1999; Salton, 1983). Terms that occur only in small number of documents in the collection are considered to have high signal value.

In this weighting scheme, the local term weight is the logarithm of term frequency and the global weight or the signal value uses Entropy E(i) and they are specified by: (Salton, 1983; Dumais, 1992)

$$\mathbf{L(i, j) = Log\ _2 tf(i, j) + 1} \quad \text{(equ 2-4)}$$

$$\mathbf{G(i) = 1 - E(i)} \qquad \mathbf{E(i) = -} \sum_{j=1, m} \frac{P_{ij} * Log_2(P_{ij})}{Log_2 N} \quad \text{and} \quad P_{ij} = \frac{tf(i, j)}{\sum_j tf(i, j)}$$

where 'm' is the number of documents in the collection, P_{ij} is the probability of term i in document j and N stands for the total frequency of term i.

Therefore the weight of term i in document j (W_{ij}) is given by

$$W_{ij} = L(i, j) * G(i, j) \quad \text{(equ 2-6)}$$

IV. Term Discrimination Value Weighting Function

This weighting scheme estimates the importance of a term as a document discriminator by using the type of change occurring in the space configuration when a term is assigned to documents of a collection (Salton, 1983; Rijsbergen, 1996). Hence, good discriminators are those index terms that increase the average distance between documents or lower space density upon their assignment to the documents in the collection (ibid). Consequently, those index terms that decrease the average distance between documents or increase space density upon their assignment are poor discriminators.

The discrimination value of a term (say term i), Dv_i, is given by:

$$Dv_i = Q - Q_i \quad \text{(equ 2-7)}$$

where Q denotes space density before assignment of term i to the document collection

and Q_i denotes space density after assignment of term i.

Here, space density (Q and Q_i) represents the average pair-wise document similarity in the collection. i.e.

$$Q = 1/N(N+1) \sum_{i=1, N} \sum_{j=1,N\ i\neq j} Sim (D_i, D_j), \quad \text{(equ 2-8)}$$

for some constant 1/N(N+1) where N is the number of documents

Alternatively, the space density can be computed more efficiently by constructing an average document located in the center of the document space, called document Centroid, $C=(c_1,\ldots c_N)$ (Salton, 1983). The terms of the document centroid are assumed to have average frequency characteristics.

$\mathbf{C_k = 1/N \sum_{i=1,M} tf_{ik}}$ for k=1, ...N where N- number of index terms and M – Number of documents (equ 2-9)

The space density is then computed as the sum of the similarity of each document with the centroid

$\mathbf{Q = K \sum_{i=1,N} Sim\ (C, D_i)}$, for some constant K (equ 2-10)

This simplified formula significantly reduces the number of similarity coefficient computations.

From experiment, (Salton, 1983) it has been found that high frequency terms have got negative discrimination values. Low frequency terms are characterized by near zero discrimination values. And medium frequency terms are found to be good discriminators, with positive discrimination values.

Therefore, the weight of a specific term i in document j is computed by

$\mathbf{W_{ij} = tf(i, j) * Dv_i}$ (equ 2-11)

2.4 QUERY MATCHING

Another important activity in an IR system is comparison of query and document representations in order to retrieve documents that are more similar to the specific query. The three classic models of Information retrieval: the Boolean model, the Vector space model and the probabilistic model are often used to accomplish this task. The Boolean and probabilistic models are briefly discussed below. The vector space model is already discussed in chapter one.

I. The Boolean Model

The Boolean model of Information retrieval is the oldest of the three classic retrieval models and it relies on the use of Boolean operators in combination with set theory (Baeza-Yates, 1999). The terms in a query are linked together with AND, OR and NOT. This method is often used in search engines on the Internet (eg. Google) because it is fast and can therefore be used online (ibid).

Unfortunately, the Boolean model has got its own drawbacks. It requires the users to have some knowledge of the search topic for the search to be effective (Carmel & Soffer, 2003; Baeza-Yates, 1999). A wrong word in a query could rank a relevant document non-relevant. In addition to that, all retrieved documents are considered to be equally important.

Another problem of the Boolean model is that, most users find it difficult to translate their information need into a Boolean expression (Carmel & Soffer, 2003; Salton, 1983). This means most users need an intermediary to work with

the Boolean retrieval model, which in turn brings its own problems, such as misunderstanding and possibly misrepresentation of the information needs of the user.

II. The Probabilistic Model

In the probabilistic model, the retrieval is based on the Probability Ranking Principle (Carmel & Soffer, 2003; Rijsbergen, 1996). The probability ranking principle asserts that the best retrieval effectiveness will be achieved when documents are ranked in decreasing order according to their probability of relevance.

Given a user query, the document collection can be divided in to two sets; a set which contain exactly the relevant documents and a set which contains the non-relevant documents. Hence, the querying process can be seen as specifying the properties of the relevant document set using index terms. However, since the properties of the relevant document set is not known in advance, there is always the need to guess at the beginning the descriptions of this set. The user then takes a look at the retrieved documents for the first query and decides which ones are relevant and which ones are not. By repeating this process iteratively the probabilistic description of the relevant document set can be improved (Chowdhury, 1999; Baeza-Yates, 1999).

The characteristic of the probabilistic model to rank documents in decreasing order of their probability of relevance to the user query is taken as the most important feature of the model (Rijsbergen, 1996; Salton, 1983).

However, lack of initial information on the relevant and non-relevant documents in the collection with respect to specific query; its ignorance to the frequency with which an index term occurs in a document and the assumption of index terms independence are taken as its major drawbacks. (Moens, 2000; Rijsbergen, 1996; Carmel & Soffer, 2003)

2.5 IR SYSTEMS EVALUATION

Recall and Precision are two commonly used metrics to measure the retrieval effectiveness of IR systems (Rijsbergen, 1996; Salton, 1983; Chowdhury, 1999). Recall is defined as the proportion of relevant documents that the retrieval system found and precision is the proportion of documents that the system found that are relevant.

To illustrate these metrics, suppose we have a document collection D, a query Q, and a retrieval system S. Out of the documents in the collection, let us assume that R of them is relevant to query Q. The relevance of a document to a query is often determined by domain experts. Finally, let us suppose that system S retrieves a set of 'n' documents for Q and 'r' of the 'n' retrieved documents is relevant to Q. The recall and precision for Q and D are defined as:

$$Recall(S) = r/R$$

$$Precision(S) = r/n$$

However, most modern IR systems do not return a set of documents for a query, hence, not possible to directly apply these formulas. Instead, they output a list of documents ranked in descending order of probability of relevance to the query (Baeza-Yates, 1999; Rijsbergen, 1996). To come up with a solution to this problem, a recall/precision pair is calculated for each relevant document in the ranked output and then interpolate to find the precision at standard recall points 0.0, 0.1, 0.2, ..., 1.0. Hence, precision for recall point 0.0 means the precision for the first relevant document in the ranked output. Often the average of the precision values at the standard recall points is used as a single metric to measure the quality of the ranked output (Chowdhury, 1999; Salton, 1983; Rijsbergen, 1996; Baeza-Yates, 1999).

When a set of query is considered, the precision at a recall point for the query set takes the average of the precision values of all queries at that recall point. The average precision for the query set takes the average of the average precision values of all queries.

2.6 THE AMHARIC WRITING SYSTEM

In this section the features of Amharic writing system relevant to the purpose of this research are discussed.

2.6.1 The Amharic Alphabets

The current Amharic writing system consists of a core of thirty-three characters (ðdL, fidel) each of which occurs in a basic form and in six other forms known as orders (Getachew, 1967). The non-basic forms are derived from the basic forms by more-or-less regular modifications. Thus there are 231 different characters. The seven orders represent syllable combinations consisting of consonant and following vowel. This characteristic according to Bender (1976) makes the Amharic writing system a syllabic writing system. A character or a symbol is used to represent a phoneme, which is a combination of a vowel and a consonant.

In a syllabic system, like Amharic, the number of characters (symbols) needed by the language is determined by the number of basic sounds used (Bender, 1976). In addition to the 231 characters there are nearly forty others which contain a special feature usually representing labialization, for example, ኈ (k^{w}ä) from ከ (kä) and ቈ (q^{w}ä) from ቀ (qä). Only about twenty of these are common and are usually listed as an appendix to the main list (ibid).

2.6.2 The Punctuations

The Amharic writing system consists of as many as 10 punctuation marks in addition to the characters (Daniel, 1994). However, only few of them are practically used, especially in computer-written system. The word-separator (Hulet Neteb), two square dots arranged like colon, :, and sentence-separator, four square dots arranged in a square pattern :: , are the basic punctuation marks in Amharic writing system. Hulet Neteb is oddly used more in hand written practices today than in modern typesetting. Its place is almost completely taken over by space.

Lists in Amharic text are separated by an equivalent of comma, 'netela sereze' (#) followed by ASCII space and 'derib sereze' (\), which is the equivalent of semi-colon, may also be found in use as a list separator. In addition to these, the writing system has borrowed some punctuation marks from foreign languages. For instance, the exclamation mark '!' and the question mark '?' are used in the language (Bender et al., 1976).

2.6.3 The Amharic Number System

The Amharic number system consists of twenty (20) single characters. They represent numbers one to ten, multiples of ten (twenty to ninety), hundred, and thousand. These characters are derived from Greek letters (Bender et al., 1976), and in order to make them look like the Amharic characters the symbols are modified by adding a horizontal stroke above and below.

The system has no place value and there is not symbol representing the number zero (0). In addition, the number system does not use commas or decimal points. These situations make arithmetic computation using this system very complicated (Bender et al., 1976).

Both Amharic and Western numerals are in use today. Though the Amharic has long since been retired to a reserved use primarily for calendar dates and demarcation of sections in literature, while Western numerals are used everywhere else following western practices (ibid).

2.6.4 Problems in Retrieving Amharic Text

Here, some of the problems in Amharic text retrieval systems that are caused by the nature of the writing system of the language are discussed.

I. Redundancy Of Some Characters

In Amharic Alphabets there are some different symbols having the same pronunciation (sound). Although in Ge'ez language, these different symbols give each word different meanings, in the Amharic language they have are used interchangeably (Getachew, 1967; Bender et al., 1976). The presence of these redundant characters with the same sound in the language creates problem, especially in term matching retrieval systems. Literally different word can be formed by combining the different form of the same sound character. For

instance, the same word 'tsehay']hY, can be written differently as ጸሀይ,]¼Y,]`Y, ]¦Y, [¼Y, [`Y, etc.

The class of symbols with the same sound falls into two. The first class includes characters with the same sound for the first and fourth order. These are h and ¦, ¼ and ˆ, ^ and `, x and ¨, and ; and >. The second class includes characters with different alphabets that share the same sound. These characters are h, ¼, and ^, s and ¿, x and ;, and] and [.

II. Formation Of Compound Words

In Amharic writing system there is no agreed upon standard in spelling compound words (Bender et al., 1976). They are sometimes written as a single word and some other time as two separate words. For example, the word 'megneta bet' which means "bed room" can possibly be written as 'መኝታቤት' and 'መኝታ ቤት' and also the word 'bet mekides' which means "temple" can be written as 'ቤተመቅደስ' and 'ቤተ መቅደስ'.

Occasionally, the constituent terms may have completely different meaning from the compound word formed from them. For example, the word 'hode-sefee' (ሆደ-ሰፊ) which means 'tolerant' has different meaning from the constituent terms 'hode' which means 'stomach' and 'sefee' which means 'wide'.

In literal term matching retrieval systems, the constituent terms of the compound noun are considered as independent and a document which contains one of these terms is treated as relevant. This phenomena result in retrieval of irrelevant materials for a query which contains one of the constituent terms. However, concept based retrieval systems, like LSI, can partially handle this problem, as the co-occurrence frequency of the constituent terms is taken into account in determining the relation between the terms (Dumias, 1992; Deerwester et al,. 1990).

III. Existence of Irregular Spelling

A number of words in Amharic can be written with different spelling (Bender et al., 1976). For example, the word "samtoal' which means 'he has heard' may be spelled as ሰምቷል, ሰምትዋል and ሰምቶአል and also the frequently used term "Ethiopia" can also be spelled as ኢትዮጵያ and ኢትዮጲያ (Bender et al., 1976). Literal term matching retrieval systems also suffer from this irregularity in spelling, as the same word can have different spelling.

Transliteration of foreign words into Amharic writing system is one of the main causes of this irregular spelling of words (ibid). Amharic language lacks some basic English sounds. As Bender (1976) stated, about six vowels and three consonant sounds common to English are absent in Amharic. Due to this a native Amharic speaker may fail to correctly pronounce some English words. The

situation is similar to other foreign languages. Hence, each writer has a tendency to write a foreign word the way he/she pronounces it.

2.6.5 Amharic Software

Fundamentally, computers just deal with numbers. They store letters and other characters by assigning a number for each one (Stalling, 1993). Amharic alphabets do not have a representation in the ASCII (American Standard Code for Information Interchange) code table. Apparently, different Amharic word processing software makes use of the ASCII code for writing Amharic by associating the English keyboard buttons with Amharic symbols. Since the number of Amharic character together with punctuation marks is much greater than English, two and three keys are used to represent a single Amharic symbol (Worku, 1997).

Different Amharic word processing software have been developed since 1987 (e.g Power Ge'ez, Geez, Agafari, Visual Ge'ez etc.) (Zelalem, 2001). These software use the same English keyboard differently. That is, two Amharic word processors can use the same button to represent two different Amharic characters. As a result, whenever data is passed between different Amharic word processing software, that data always runs the risk of corruption. ECoSA (Ethiopian Computer Standards Association), a professional association, is working to solve the problems that result from the inconsistency in the available different Amharic software (ICT Focus, Vol 2. Issue 1).

Most of the software are written to work only with Microsoft word. However, there are few which can work in other programs. Visual Ge'ez is one of the exceptions. Visual Ge'ez has two versions; VG2 and VG2000 developed for different versions of Microsoft office products. Both the test collection and the sample queries used in this research are written in VG2 version of Visual Ge'ez.

2.7 LATENT SEMANTIC INDEXING (LSI) IR

This section discusses about LSI (Latent Semantic Indexing) technique of information retrieval. It presents the basic tasks involved in LSI information retrieval, such as: dimension reduction (using Singular value decomposition (SVD)) of raw term by document matrix and query representation. Related works are also reviewed.

2.7.1 Shortcomings Of Term Matching Retrieval Systems

The classic information retrieval models (Boolean, standard vector space and probabilistic) are all based on term matching. Typically, only text documents that contain one or more words in common with those in users query are returned as relevant. However, there are a number of issues which can prevent the results of an exact term matching search from being accurate. Two of these issues include *polysemy* and *synonymy* (Deewester et al., 1990; Furnas et al., 1988; Berry, Dramc & Jessup, 1999).

Polysemous words are characterized by different meaning based on the context. For instance, the word "አለም" in Amharic can mean 'the world', or it can also mean 'every one' or 'Happiness', or 'wealth' (Taken from Desta Teklewold Amharic Dictionary). Polysemy causes terms in the query to be matched with words in irrelevant documents and for the search results to be too broad (Berry, Dramc & Jessup, 1999; Dumais, 1992).

Words that are synonyms have equivalent meaning. Synonyms allow people to use different ways to refer to the same thing. It happens to the everyday searcher, even though, different people have same object in mind to search for, each issues different queries (Baeza-Yates, 1999). They are all valid as they are synonyms. For example, in Amharic the words 'ገረድ', 'አሽከር' and 'አገልጋይ', all can be used to mean 'servant' (from Desta Teklewold Amharic Dictionary). Thus, in a literal search, only documents containing the literal query terms will be matched, causing the search results to exclude relevant documents (Berry, Dramc & Jessup, 1999). Therefore, synonyms tend to reduce the recall of IR systems.

2.7.2 Existing Approaches To Solve The Problem Of Synonym And Polysemy.

A number of approaches have been employed to solve the problems of polysemy and synonymy in information retrieval systems (Salton, 1983; Baeza-Yates, 1999; Moens, 2000). Some of these methods are, stemming, thesaurus construction, relevance feedback and dimensionality reduction.

Stemming: - stemming is one of the methods used to solve word mismatches in user's queries and in documents in the collection that are caused due to morphological variation (Moens, 2000). For instance, if the user of a retrieval system uses the word 'vegetable" in his query, it is most likely that documents that contain the word "vegetables" are of interest to the user. Thus, stemming algorithms, or stemmers, reduce words in queries and documents to common roots so that documents containing both morphological variants as well as exact words in the query are retrieved.

Thesaurus construction: - thesaurus is used to handle the problem of synonymous by representing a group of synonym terms by a single group identifier (Salton, 1983). For example, if a query contains the word "computer", documents that contain the words "Laptop" and "PC" are likely to be relevant. Hence, the query needs to be expanded by including the synonym words.

Relevance feedback: - relevance feedback is a method that enhances a query depending on the relevance judgment on the retrieved documents (Dumais, 1992; Berry, Drmac & Jessup, 1999; Salton, 1983). Words that frequently appear in relevant documents are included into the query and also the weights of the query words are adjusted based on their appearance in the relevant and non-relevant documents. Relevance feedback techniques can find valuable words which do not have obvious semantic relationship with the original query words (Salton, 1983). For example, "Sadam", and "chemical and biological weapon"

may be added to the query "War for Iraq Freedom" after observing the documents returned in the first search.

Dimensionality Reduction: - Dimensionality reduction technique seeks to resolve the problems of synonymy and polysemy by examining the "latent" structure of a document and the terms within it (Deerwester, 1990; Dumais, 1992; Berry, Drmac, & Jessup, 1999). These techniques decompose words and documents into vectors in a low dimensional space.

The variability in word choice between the system users and authors of documents is addressed, because any word can be matched with another word to some degree in the low dimensional space (Dumais, 1992; Deerwester, 1990; Berry, Dumais & Letsche, 1995). Latent semantic indexing (LSI) is an example of these techniques.

2.7.3 Latent Semantic Indexing Technique

Latent semantic indexing is one of the many recent information retrieval techniques that are based on a vector space model (Dumais, 1992; Berry, Dumais & Letshe, 1995; Berry, Drmac, Jessup, 1999). However, LSI adds additional processing on the term by document matrix, perhaps the weighted one, generated by standard vector space model.

The term by document matrix is decomposed into a set of 'r', rank of the matrix, orthogonal factors from which the original matrix can be approximated by linear combination. Typically, any rectangular matrix, X, for instance, a tXd matrix of terms and documents can be decomposed into the product of three other matrices

$$\underset{tXd}{X} = \underset{tXr}{U_o} * \underset{rXr}{S_o} * \underset{rXd}{V_o^T} \qquad \text{(equ 2-12)}$$

such that columns of U_o and V_o define the left and right singular vectors of matrix X, which are the orthogonal eigenvectors associated with XX^T and X^TX respectively. S_o is a diagonal matrix representing the singular values of X arranged in decreasing order, which are non-negative square roots of the eigenvalues of X^TX (Dumais, 1992; Wall, Rechtsteiner & Roche, 2003). This is called Singular Value Decomposition (SVD) of X.

If only the 'k' largest singular values of S_o (where K<= r (rank of A)) are kept along with their corresponding columns in the matrices U_o (u_1, u_2... u_k) and V_o (v_1, v_2... v_k), and the rest avoided, yielding matrices U_k, S_k and V_k , the resulting matrix X_k is the unique matrix of rank 'k' and is closest in the least squares sense to the original matrix X (Witter &Berry, 1998; Dumais, 1992; Husbands, Simon, & Ding, 2001)

$$\underset{tXd}{X} \sim \underset{tXd}{X_k} = \underset{tXk}{U_k} * \underset{kXk}{S_k} * \underset{kXd}{V_k^T} \qquad \text{(equ 2-13)}$$

The point here is that the reduced matrix X_k, by keeping only the first 'k' independent linear components of X, captures the major associational structure in the matrix and also much of the noise caused by, such as, polysemy and synonymy, that leads to poor information retrieval is removed with the reduction in dimensionality (Berry, Drmac & Jessup, 1999; Deerwester et al., 1990).

Hence, the selection of the right dimensionality or the value of 'k' is a crucial issue. Of course, the value of 'k' should be large enough to integrate all the real structure in the document collection, but also it should be small enough so that noises that are caused by the variability in word usage are not included (Deerwester, 1990; Berry, Dumais, & Letsche, 1995; O'Leary & Kolda, 1998).

There is no hard and fast rule that can be used determine the optimal value of 'k'. Currently, an operational criteria is used, that is, a value of 'k' which yields good retrieval performance (Deerwester, et al., 1990; Dumais, 1992).

In this reduced model, the similarity of documents is determined by the overall pattern of term usage instead of exact term match, so documents can be near each other regardless of the precise words that are used to describe them (Deerwester, 1991; Witter & Berry, 1998). A document which has no words in common with a user's query may be near to the query if that is consistent with the major pattern of word usage

Geometrically, the location of terms and documents in the approximated k-dimensional space is given by the row vectors from the U_k and V_k metrics respectively (Berry, Drmac & Jessup, 1999; Dumais, 1992). The cosine or dot product between vectors in this space corresponds to their estimated similarity. The representation of both term and document vectors in the same space makes the computation of the similarity between any combination of terms and documents very easy (Witter & Berry, 1998; Berry, Drmac & Jessup, 1999).

From IR point of view, three basic similarity comparisons in the reduced matrix X_k are important (Deerwester et al., 1990; Dumais, 1992; Berry, Dumais & Letsche 1995).

I. Term By Term Comparison

The term to term comparison is used to determine the extent to which two terms have a similar pattern of occurrence across the document.

Using the reduced matrix X_k in equ (2-13) and the dot-product similarity measure, the term-to-term similarity values can be computed as

$$X_k \cdot X_k^T = (U_k.S_k.V_k^T).\ (U_k.S_k.V_k^T)^T = U_k \cdot S_k \cdot V_k^T \cdot V_k \cdot S_k \cdot U_k^T$$

$$= U_k \cdot Sk \cdot (V_k^T \cdot V_k) \cdot S_k \cdot U_k^T$$

$$= (U_k \cdot S_k) \cdot (U_k \cdot S_k)^T \text{ since } V_k \text{ is orthonormal} \quad \text{(equ 2-14)}$$

Here, we can see that the i, j cell of the square matrix $X_k \cdot X_k^T$ can be obtained by taking the dot-product between the i^{th} and j^{th} rows of the matrix $U_k \cdot S_k$. We can consider $U_k \cdot S_k$ as coordinates for terms, because since S_k is a diagonal matrix, it is merely used to stretch or shrink the axis of the reduced vector space without affecting the position of U_k in the space.

II. Document By Document Comparison

This comparison is used to determine the degree of similarity between the documents by identifying the extent to which the two documents have a similar term occurrence pattern (Berry, Dumais & Letsche, 1995). This can be attained by computing the dot-product of column vectors of the reduced matrix X_k i.e.

$$X_k^T \cdot X_k = (U_k.S_k.V_k^T)^T \cdot (U_k.S_k.V_k^T)$$

$$= V_k \cdot S_k \cdot (U_k^T \cdot U_k).S_k.V_k^T$$

$$= (V_k \cdot S_k)\ (V_k \cdot S_k)^T \quad \text{(equ 2-15)}$$

The i, j cell of $X_k^T . X_k$, which is a document-by-document square matrix, is obtained by taking the dot product between the i^{th} and j^{th} row of the matrix $V_k.S_k$. However, the space of $V_k . S_k$ is just a stretched or shirked version of the space of V_k, in proportion to the corresponding diagonal elements of S_k.

III. Term By Document Comparison

The third comparison is between a term and a document. However, the comparison between a term and a document is just the value of an individual cell of X_k. Hence,

$$X_k = U_k . S_k . V_k^T = U_k . S_k^{½} . S_k^{½} V_k^T = (U_k . S_k^{½}) (V_k . S_k^{½})^T \qquad \text{(equ 2-16)}$$

This implies that the i, j cell of X_k is obtained by taking the dot-product between the i^{th} row of the matrix $U_k . S_k^{½}$ and the j^{th} row of the matrix $V_k . S_k^{½}$ (or the j^{th} column of $(V_k . S_k^{½})^T$). Here again, the effect of the factor $S_k^{½}$ is stretching or shrinking of the axes by a factor of its value.

2.7.4 Query Representation

In LSI based IR systems, a user's query is often considered as a pseudo-document and is represented as a vector in the reduced term-by-document space (Witter & Berry, 1998; Berry, Dumais & Letsche, 1995; Kolda & O'Leary, 1998). First, the terms used by the searcher are represented by an (m X 1) vector 'q' whose elements are either zero or correspond to the frequency of terms that exists in the database of reduced vector space. The local and global term weights used for the document collection are applied to each non-zero

elements of the query vector q. Then the query vector is represented in the reduced LSI space by the vector

$$q\hat{} = q^T . U_k . S_K^T \quad \text{(equ 2-17)}$$

where $q^T . U_k$ is the sum of term vectors specified by vector q scaled by S_k^T (Witter & Berry, 1998; Berry, Dumais & Letsche, 1995).

Finally the query vector can be compared to all existing document vectors, and the documents ranked by their similarity to the query. One common measure of similarity is the cosine between the query vector and document vector. Typically, the first 'n' closest documents or all documents exceeding some cosine threshold are returned to the user (Deerwester et al., 1990).

2.7.5 Review Of Related Researches

In this section a review of several researches related to the application of LSI is made.

I. LSI: In Information Retrieval

Originally, LSI was developed for IR application (Berry, Dumais &Letsche, 1995). In its first application, automatic matching of queries with document abstracts, LSI provided a remarkable improvement over prior methods (Landauer, Foltz & Laham, 1998).

The first tests for evaluating the performance of LSI in IR environment were against standard collection of documents having well formulated queries and relevance judgment obtained from knowledgeable domain experts (Berry, Dumais & Letsche, 1995, Landauer, Foltz & Laham, 1998; Deerwester et al., 1990; Dumais et al., 1988). The results of the experiments were in the worst case equivalent to the best prior methods and reach up to 30% better than that obtained using standard keyword vector method. It was also observed that the LSI method performs best relative to standard vector method when the queries and relevant documents do not share many words, and at high levels of recall.

Even in recent projects, sponsored by the National institute of standards and Technology, LSI was compared with a large number of other experimental and commercial retrieval systems (Landauer, Foltz & Laham, 1998). Although, direct quantitative comparison among the many retrieval systems was not appropriate, due to variations in preprocessing, overall performance results of LSI models were quite similar to earlier ones.

In another experiment carried out to compare the standard vector method with LSI approach; it was found that LSI was a 16% improvement (Dumais, 1994). The experiment was conducted by keeping the preprocessing actives the same (like using the same stop Lists, getting rid of case differences and spelling errors, identifying proper nouns as special etc.).

Furnas et al. (1988) conducted experiments to examine the effectiveness of LSI in two standard document sets to which user queries and relevance judgments were available. The first database was consisted of 1033 medical reference abstracts and titles and 5823 automatically extracted terms which appear in more than one document. After approximating the term by document matrix by a 100 factor SVD, the retrieval effectiveness of the system is evaluated against 30 queries available with the dataset. The result was a 13% improvement over the basic inner product term matching. This improvement was attributed to the LSI ability to capture some structure in the data which was missed by raw term matching. The improvements were large, particularly at higher levels of recall.

The second experiment was found to show no improvement over the literal word matching method. The research was conducted on 1460 information science abstracts (CISI). Through automatic indexing 5135 terms occurring in more than one document were extracted. 35 queries available with the dataset were used to evaluate a 100-factor SVD approximation of the term by document matrix. For this particular dataset, precision for all retrieval systems was below 0.30, even for the lowest levels of recall.

Dumais and his co-workers (1992) studied several techniques used to improve the basic LSI method, including term weighting, selecting the optimal value of dimension (k) and relevance feedback.

Selection of the appropriate number of dimensions for the reduced dimensional representation is a challenging problem (Berry, Dumais & Letsche, 1995). Dumais and his co-workers have evaluated retrieval performance of the LSI model using a range of dimensions. From their experiment, on several Information science test collections for which queries and relevance assessments were available, it was observed that the performance of the model improves considerably as the number of dimensions increase from a very low dimension, reach the highest point between 70 and 100 dimensions, and then begins to reduce slowly approaching the level of performance attained by standard vector method. It is obvious that with sufficiently large dimensions, SVD will exactly reconstruct the original term by document matrix. This pattern was consistent for all databases.

The effects of several local and global weighting schemes on the performance of the LSI model were also examined by the same research in five different Information science test collections. Six different term weighting schemes were explored in each of the test collection: raw term frequency with no global weighting, four combinations of raw term frequency as a local weight and one of GFIDF (global frequency *I* document frequency), IDf, Entropy or Normal ($1/\sqrt{tf_{ij}}$) as a global entropy weight, and one combination of a local Log weight (Log (tf_{ij} + 1)) and a global entropy weight (Log-Entropy). The results of the experiments were consistent on all test collections. Normal and GFIDF were worse than no global weighting and IDF; Log-Entropy resulted in substantial performance

improvements. Averaged over the five test collections, Log-Entropy weighting was 40% more effective than raw weighting.

The effect of relevance judgment was also examined in the research. In the experiment retrieval is first performed using the original query. Then this query is replaced by

i) the first relevant document in the ranked list of retrieved documents.

ii) the average of the first three relevant documents.

Replacing the original users' query with the first relevant document improved the performance of the LSI retrieval model by an average of 33%. The replacement of the original query by the average of the first three relevant documents resulted in 67% performance improvement.

The users' were supposed to view a median of only one document in order to find the first relevant document , and a median of 7 documents had to be viewed by the user in order to get the first three relevant documents. Here, one can easily observe that this substantial performance improvement is obtained with little burden on the side of the users.

II. LSI: In Information Filtering

In addition to IR, LSI has also been shown to be effective in Information filtering environment (Foltz, 1990). In information filtering tasks, as opposed to IR, a user has a relatively stable long-term interest or profile (ibid). New information sources are constantly received and presented to the specific user by matching the sources with the somewhat stable user profile. To apply LSI to information filtering, first an initial sample of documents will be parsed into a keyword–document matrix and then approximated by a reduced dimensional space using LSI/SVD. One or more vectors will be used to represent users' profile in this reduced dimension LSI space, as a user may have many different types of topics that he/she is interested in (Foltz & Dumais, 2001).

To determine if a document is relevant, it is first folded into the semantic space on the basis of its contained terms. If the document appears close enough to the users' profile vector/s, it would be considered likely to be interesting and presented to the corresponding user.

Foltz (1990) compared LSI and standard keyword matching models for the task of filtering news articles (Netnews). First users were permitted to judge articles as to whether they are of interest or not, and based on these judgments, both LSI-generated model and literal keyword matching model predicted whether incoming article would be judged interesting. The result was that LSI offered a significant improvement (13%-25%) over the standard keyword matching. This implies that

LSI is capable of capturing more of the semantic structure that is useful in predicting user preferences than the keyword matching.

Two years later Foltz & Dumais (1992) made similar experiment. The research was designed to test several information retrieval methods for filtering Technical Memos. 34 regular users of the technical memos were selected and sent monthly personalized list of new technical Memo abstracts that were predicted to best match their interest. However, the predictions were made using two methods for describing user profiles; one based on sets of keywords that the users provided, and the other using feedback about previous abstracts which are judged relevant by the users. Two information retrieval methods were tested to make the predictions; one using standard keyword matching, and the other using LSI. In this particular research all four methods effectively selected relevant abstracts. But the best method for filtering used LSI with feedback about previous relevant abstracts.

III. LSI: In Cross-Language Retrieval

In cross-language retrieval, documents will be written in several language and users specify their request in any one of these languages and get access to the relevant documents written in any of the languages (Berry, Dumais & Letsche, 1995).

Landauer & Littman (1990) were able to show the effectiveness of LSI in a fully automatic cross-language retrieval environment. In their experiments term-document matrix is formed using a collection of abstracts that have both English and French versions. The two versions of a specific abstract are combined and treated as a single document. A reduced dimensional approximation of the terms by the combined abstract matrix is computed by using truncated SVD. Once this LSI space is generated, abstracts in either French or English are simply folded in into the existing space. Then queries in any one of the language can be matched to French or English abstracts.

The experiment showed that retrieval of French documents in response to English queries and vice versa was equally effective with first translating the queries into French and searching a French only document collection.

Another experiment by Young (1994) has shown comparable performance for retrieving English abstracts and Japanese kanji ideographs, and for cross-language translations of the Bible in English and Greek languages.

CHAPTER THREE

DESIGN, DEVELOPMENT AND TESTING OF THE PROTOTYPE LSI RETRIEVAL MODEL

3.1 INTRODUCTION

This chapter discusses the design, development and testing of the LSI Model for Amharic Text Retrieval (LSIMoATR). The peculiar characteristics of Amharic writing system in relation to text retrieval and basic concepts in LSI text retrieval discussed in chapter three are taken into account in the process.

In this research, 206 Amharic News articles were included in the test set. 100 of them were used by Bethlehem (2002) in her research on 'N-gram Based Indexing for Amharic Text Retrieval'. The remaining 106 news articles are obtained from the Archive in the web site of Walta Information Center. Each news article is kept in a separate text file written in a VG2 Main font under a common folder. In addition, 32 queries, 16 of them used by Bethlehem (2002) and the other 16 used by Saba (2001) in her research on 'The Application of Information Retrieval Techniques to Amharic Documents on the Web' were selected. Bethlehem used 100 out of the 313 test collection saba used.

To identify which news articles in the test collection are relevant for a specific query a relevance judgment for each of the documents were made by two journalists (from 'Addis Zänä', senior editor and reporter). However, 7 of the queries, those taken from Saba (2001), were found not to have a relevant

document in the collection. Hence, the remaining 25 queries were used in this research. Like the news articles in the test set, all text files each containing a single query, are kept in a separate folder.

Basically, there were three main stages involved in the processing of documents and queries, and development of the LSI retrieval system.

1. Pre-processing and indexing (this includes extracting terms, term-document matrix generation and calculating term weighting)
2. K-dimensional Singular Value Decomposition (SVD)
3. Query Projection, Matching and Ranking of Relevant documents.

Finally, comparison of the performance of the LSI model against the standard vector model was made. The performance of each method is evaluated by measuring precision at several different levels of recall.

3.2 DESCRIPTION OF THE PROTOTYPE SYSTEM

The design of the prototype LSI Amharic text retrieval system has the components depicted in figure 3.1 below. The different components represent the major activities involved in the prototype LSI retrieval model development.

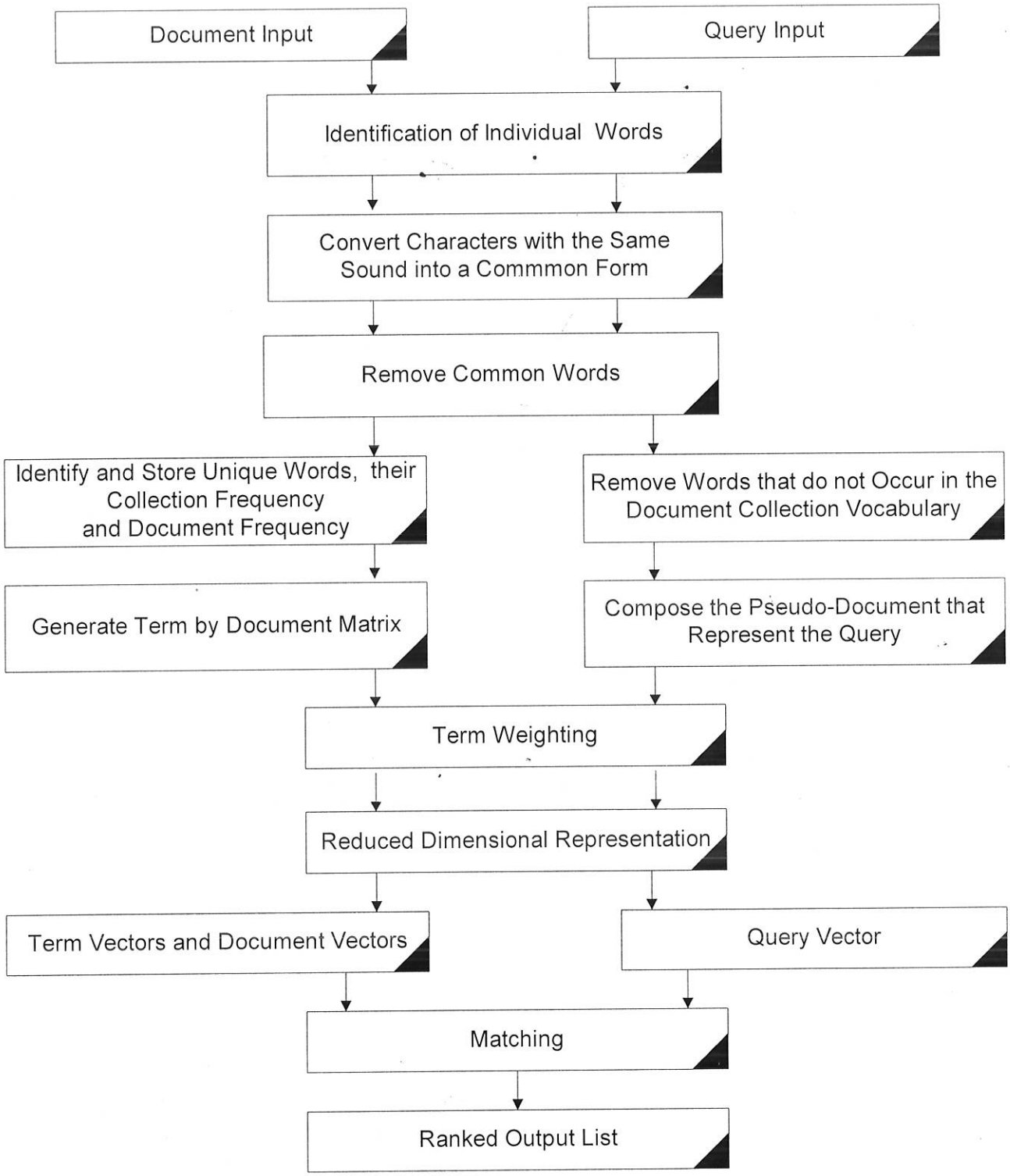

Figure 4.1 The design of the Prototype LSI Retrieval Model

The ultimate objective of this prototype model is to represent index terms, documents and queries in the same reduced-dimensional space so that comparison of the documents, terms and queries in this reduced-dimensional space is possible. In the middle of the prototype model, a standard vector space model is obtained. This makes comparison of the standard vector space model and the LSI model straightforward.

The test collection and sample query files stored in their respective folders are read and individual words which are not in the stop-word list are extracted. The term by document matrix and the query vector are generated. In order to increase or decrease the importance of each term based on their ability to represent the content of a document and also discriminate it from other documents in the collection; both the term-document matrix and the query vectors are weighted.

The weighted term-document matrix is then given to the SVD algorithm as an input and a reduced dimensional representation of the matrix is generated. The reduced-dimensional representation of the query vectors (pseudo-documents) is also obtained through projection into the reduced space. After the terms, documents and queries are represented in the same reduced space, cosine similarity measure is used to identify the ranked list of relevant documents for each query.

3.3 SELECTING DOCUMENT PART FOR INDEXING

To obtain index terms which can be used to represent documents, one may use the entire content of a document, or some delimited portion of it, such as the title or abstract, or a paragraph. Each news article used in the experiment is composed of a title and the body of the news. In this experiment, the entire document (both title and body) was use for indexing, because the researcher believes that the entire text of the articles is better representative of the content of the articles. Besides, as the news articles are of small size (in most cases half a page) taking the entire document will not consume much memory space.

3.4 DOCUMENT AND QUERY PRE-PROCESSING

The pre-processing stage, which involves preparing the text documents for SVD algorithm has been done in two phases. The first phase is a natural language pre-processing. This includes: identification of individual words; removal of extraneous characters; changing redundant characters (characters with the same sound) into a common form; removal of common terms by referring to a previously developed stop-word list (negative dictionary); and generation of a raw term by document matrix (where each number in the cell of the matrix indicates simple frequency of the term in the corresponding document). Each activity is discussed in the next section.

The second phase is a numerical data-preparation. In this phase, each term in the term by document matrix, generated in the first phase, is weighted (using log-entropy weighting scheme) in order to emphases or de-emphases its relative importance within the document as well as in the document collection.

A program in Borland Delphi was developed to handle all the activities indicated in the natural language pre-processing. A small weighting function was also written in Matlab software (Release 12) to accomplish the weighting task.

3.4.1. The Natural Language Pre-Processing

I. Identification Of Individual Words

A natural language text often contains numbers and various punctuation marks. These symbols do not have their own meaning. For instance, a number '123' can be used to indicate 'number of cars' in one text, it might be used to indicate 'page number' some where in the same text or in the text of other document, it might also be used to indicate the 'number of causalities' in a certain accident, etc.

Likewise, punctuation marks are used for syntactic, grammatical or structural purposes. Often, they are used to clarify meaning in a text, to help convey emphasis and breathing pauses, to indicate sentence structure, and also to enhance readability (Microsoft®Encarta®Encyclopedia99). Due to these facts, numbers and punctuation marks in the text of the test documents and sample queries were not considered for indexing.

As discussed in section 3.6, Amharic words in a text are separated by punctuation marks, spaces, tabs, carriage return and line feed characters. However, since the test documents in the collection as well as the queries are written in VG2 Main Amharic writing software, the punctuation marks considered in this research are those that are supported by the writing software. These comprise: ፡, ።, ፣, ፤, -, ?, /, ., and « ».

Consequently, the codes that represent these punctuation marks in the implementation of the software are identified and used in the process of word identification. Some decisions were made concerning the punctuation marks '-' and '.'. In Amharic texts the punctuation mark '-', equivalent of hyphen in English, is used to form compound words. However, in the test collection this punctuation mark was not used consistently. The same compound words were found written both as separate words without the hyphen mark and as compound words with hyphen (example, **ጸረ ኤድስ** and **ጸረ-ኤድስ**). To keep consistency throughout the test collection, a decision was made to replace the hyphen mark with one character space and split compound words into their constituent terms.

Likewise, the symbol '.' is used for abbreviation purpose. Through analysis of the document collection, it was found that the symbol was used only with two abbreviations: **አ.ም** and **አ.አ**, which are equivalent to A.D and A.A (for Addis Ababa). Because, their frequencies in the collection were high, removing them

from the text was believed to cause no significant difference in the performance of the models.

The algorithm given below, developed by Zelalem (2001), was adopted for the actual word identification purpose.

1. *initialize the variables to hold the word*
2. *read a character from the sentence (document)*
3. *check if the character is any one of the Amharic delimiter (punctuation mark, space, tab, carriage return, and line feed characters)*
4. *if not, concatenate the character to the variable*
5. *else if the character length is above one character report the word*
6. *if there is more data to process, go to step 1.*

The identified words were also checked for whether they contain a number or not. This is because, words such as 1ኛ(1st), በ20ኛው (during the 20th), የ1977ቱ (the 1977), etc. were not considered for indexing. Besides identifying individual words, the algorithm was made to produce the frequency count of each word in each document. Finally, unique words together with their frequency were stored in a text file.

II. Changing Redundant Characters to a Common Form.

As pointed out in section 2.6.4, the presence of several Fidels (symbols) with the same sound in the Amharic writing system creates a problem in text retrieval. The same word written using different symbols (letters) with the same sound will be considered as different words by the retrieval system. In this research, choosing one letter for the group of letters with the same sound and replace the remaining ones is taken as a solution to the problem. Therefore, if a character is

any of ሐ, ኀ, ኃ, ሐ, ኻ or ሃ (all with the sound 'h') then, it is replaced by 'ሀ'. Also the different orders of ሐ and ኀ are changed to their corresponding equivalent orders of ሀ. Similarly, all orders of ሠ (with the sound 's') are changed to their corresponding equivalent orders of ሰ, all order of ዐ (with the sound 'a') are changed to their corresponding equivalent orders of አ, all orders of ፀ (with the sound 'tse') are changed to their corresponding equivalent orders of ጸ. Zelalem's algorithm is used to accomplish this task.

The algorithm is stated as follows:

1. *read the character*
2. *if the character is any one of*
   ሐ, ኀ, ኃ, ሐ, ኻ *or* ሃ *or any other order thereof then*
   change it to ሀ *exit*
   else if it is ሠ *or any other order thereof*
   change it to ሰ *exit*
   else if it is ፀ or any other order thereof
   change it to ጸ *exit*
3. *if the character that follows is a diacritic marking, attach it to the changed base character*

III. Removal Of Common Words

All terms in a text are not equally important in allowing retrieving relevant documents from a document collection (Salton, 1983; Rijsbergen, 1996). It is a common practice to remove the non-content bearing terms, which are not useful to specifically identify some portions of the document collection, from the index term list. These terms are often high frequency terms (ibid). Mostly, non-content bearing terms are removed by either removing high frequency terms from the indexing term list or by using a negative dictionary (stop word list) for that language.

In this research both alternatives were used. A list of 85 Amharic common terms, which were used by Nega (2002) in his PhD dissertation entitled 'Stemming on Amharic words for Information Retrieval'; are included as stop-words. Then, two kinds of stop words were identified from the word list generated in section 4.3.1(I); common words which are not in Nega's list and news specific stop words. The terms are selected in light of the fact that stop words often occur frequently in the majority of the documents in the collection. Using this concept, all terms that occur in fifty (50) and above fifty documents (which is almost a quarter of the collection) with collection frequency greater than 100 were retrieved and examined to identify stop words of both types. In determining the above thresholds the two journalists, who made the relevance judgment for the documents, were involved. The frequency of the terms contained in the word list of section 4.3.1 (I) ranges from 1 to 318.

Stop words that are categorized as news specific are those terms which are frequently used by Journalists and reporters while reporting an incident to the public. For example, 'inform' or 'notify' (ገለጹ, ተገለጸ, አሳወቀ, አሳወቁ, አስታወቁ, አስታወቀ, አስታውቀዋል, አስታውቀው, ጠቅሰው, እንደገለጹት etc.). Similarly, terms that are frequently used in the text and have no specific relation with the content discussed in the document are categorized as common words. The two categories together produced a total of 745 stop-words. This constitutes about 8% of the unique words in the collection.

Common Stop words	News Specific Stopwords
ነው·	እንደተገለጸው·
ናቸው·	ዋልታ·
ላይ	ኢንፎርሜሽ
አቶ	ተናግረዋል
ደግሞ	አስገንዝበው·
ጋር	አስረድቷል
ጊዜ	ተከናውኗል
የጊዜ	አውስተው·
ግን	አመልክተዋል

Table 3.1 Sample Stop words of each category

IV. Term by Document Matrix Generation

Latent semantic indexing being a variant of vector space method, the documents as well as the queries are represented mathematically as vectors in some vector space (Berry, Drmac & Jessup, 1999). To generate the term by document matrix, a code is written that first declares a matrix with dimensions; number of unique terms by number of documents in the collection. The entries in the cells of the matrix are initialized to 0. Then, a text file containing the list of all unique terms is created. The frequency of each term in each document is read from the text file generated in section 4.3.1(I) and substituted in the appropriate cell of the matrix.

Term List	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
የኢማተልፋ	1	0	0	0	0	0	0	0	0	0
ፕሮጀክቶች	3	0	0	0	0	0	1	0	0	4
ከእቅድ	1	0	0	0	0	0	0	0	0	0
እየተጠናቀቁ	2	0	0	0	0	0	0	0	0	0
ምንጭ	1	0	0	0	0	0	0	0	0	0
ማህበራዊ	1	0	0	0	0	0	0	0	0	0
ተሀድሶና	1	0	0	0	0	0	0	0	0	0
ፈንድ	1	0	0	0	0	0	1	0	0	0
ያስጀመራቸው	1	0	0	0	0	0	0	0	0	0
የንጹህ	3	0	0	0	0	0	0	0	0	0
መጠጥ	2	0	0	0	0	0	0	0	0	0
ከተያዘላቸው	1	0	0	0	0	0	0	0	0	0
ሰሌዳ	1	0	0	0	0	0	0	0	0	0
የሰሜን	1	0	0	0	1	0	0	0	0	0
አሞ	1	0	0	0	0	0	0	0	0	0
ሀብት	1	0	3	0	0	0	0	0	0	0
ማእድንና	1	0	0	0	0	0	0	0	1	0
ኢንደስት	1	0	0	0	0	0	0	0	0	0
የመምሪያው	1	0	0	0	0	1	0	0	1	0

Table 3.2 Sample raw term by document matrix

3.4.2 The Numerical Data-Preprocessing/Term Weighting

The numerical data-pre-processing/weighting was done on Matlab software package. Only the body of the term by document matrix generated in the first phase, excluding the term list and document ID, is selected and saved into a text file (in a folder reachable by Matlab). Then, the matrix is imported into the Matlab workspace using the load command.

The basic rational behind weighting is that a term has high weight if it is frequent in the relevant documents but infrequent in the document collection (Salton, 1983). There are many popular local and global weighting schemes proposed so far (see section 2.3.2). However, it is found that the log-entropy weighting

scheme provided a 40% advantage over simple term frequency on several standard document test collection (Dumias, 1993). So, in my research the log-entropy weighting scheme is used. This weighting scheme modifies the value of the elements in the term by document matrix to a product of its local and global weight. In section 2.3.2 (III) the log-entropy weighting is discussed in detail.

Apparently, a code for a Matlab function that computes the weight of the term-document matrix is written. The function accepts the raw term-document matrix as input apply the weighting to each of its elements and return a weighted term-documents matrix as well as the global weights (a value that indicate the importance of the term in the document collection) used for index term. Here, the function is intentionally made to return the global weight of each term, because, the values are used later for weighting query vectors. The code for the function is given below.

```
 Function [Wmatrix, Gi]=weight(A)
% signal noise weighting scheme
% A function used to weight a term by document matrix A and return
% the weighted matrix W
% Gi is a vector which holds the global weight of each term in the
%vocabulary list

[M, N]=size(A)      % M-terms and N-documents
for i=1:M
      gfi=0;    % collection frequency of term i
  for j=1:N
      gif=gfi + A(i, j)   % adds the frequency of term i in each document
  end

Noise=0;
Denom=log2(N);
  for j=1:N
    pij=A(i,j)/gfi;
```

```
        if(pij~=0)
            Noise=Noise + pij *log2(pij)/denom;
        end
    end
Gi=1 – Noise;          % global weight of term i.

for j=1:N
    Wmatrix(i,j)=log2(A(i,j) +1)*Gi(i);
end
end
```

Term List	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
የኢማተልፉ	1	0	0	0	0	0	0	0	0	0
ፕሮጀክቶች	3.0943	0	0	0	0	0	1.5472	0	0	3.5924
ከእቅድ	1.1055	0	0	0	0	0	0	0	0	0
እየተጠናቀቁ	1.585	0	0	0	0	0	0	0	0	0
ምንጭ	1.25	0	0	0	0	0	0	0	0	0
ማህበራዊ	1	0	0	0	0	0	0	0	0	0
ተሀድሶና	1.3578	0	0	0	0	0	0	0	0	0
ፈንድ	1.4562	0	0	0	0	0	1.4562	0	0	0
ያስጀመራቸው	1	0	0	0	0	0	0	0	0	0
የንጹህ	2.6235	0	0	0	0	0	0	0	0	0
መጠጥ	2.2614	0	0	0	0	0	0	0	0	0
ከተያዘላቸው	1	0	0	0	0	0	0	0	0	0
ሰሌዳ	1	0	0	0	0	0	0	0	0	0
የሰሜን	1.3903	0	0	0	1.3903	0	0	0	0	0
አሞ	1.2062	0	0	0	0	0	0	0	0	0
ሀብት	1.4904	0	2.9808	0	0	0	0	0	0	0
ማእድንና	1.3021	0	0	0	0	0	0	0	1.3021	0
ኢንደስት	1	0	0	0	0	0	0	0	0	0
የመምሪያው	1.3652	0	0	0	0	1.3652	0	0	1.3652	0

Table 3.3 Sample weighted term by document matrix

3.5 K-dimensional Singular Value Decomposition (SVD)

The weighted term-document matrix 'W' of the previous stage is used as an input and its SVD is computed for a reduced dimension K, where K is less than both the number of terms and the number of documents in the matrix (i.e. the number of rows as well as the number of columns of the matrix).

The K-dimensional SVD of the weighted matrix is performed using Matlab's built-in function, 'svds'. The function accepts the weighted matrix and the number of dimensions as an input and returns three matrices: the K-largest left singular vectors, Uk, of W, the K-largest right singular vectors, Vk, of W, and a diagonal matrix Sk whose non-zero entries in the principal diagonal are the singular values of W arranged in decreasing order.

In short, [Uk, Sk, Vk]=svds(W, k), where W is the weighted matrix and k is the reduced dimensionality.

- **Choosing The Number Of Dimensions**

As we have mentioned in section 2.7.3, there is no straightforward rule that can be used to select the optimal value of dimension K. In this research, the value for the dimension is chosen by the principle of 'What works best'. That means, the retrieval performance of the LSI model was examined for several different dimensions and the dimensionality that maximizes performance was selected.

Performance of the model was taken to be average precision over recall levels of 0.25, 0.50 and 0.75 for four different queries. The numbers represent equal interval in the range between the minimum and maximum recall levels (0 and 1). The queries are chosen in such a way that the number of words they contain and the number of relevant documents they have in the collection vary from small to large.

The figure as well as the table below shows the performance of the LSI retrieval models for different values of dimensions.

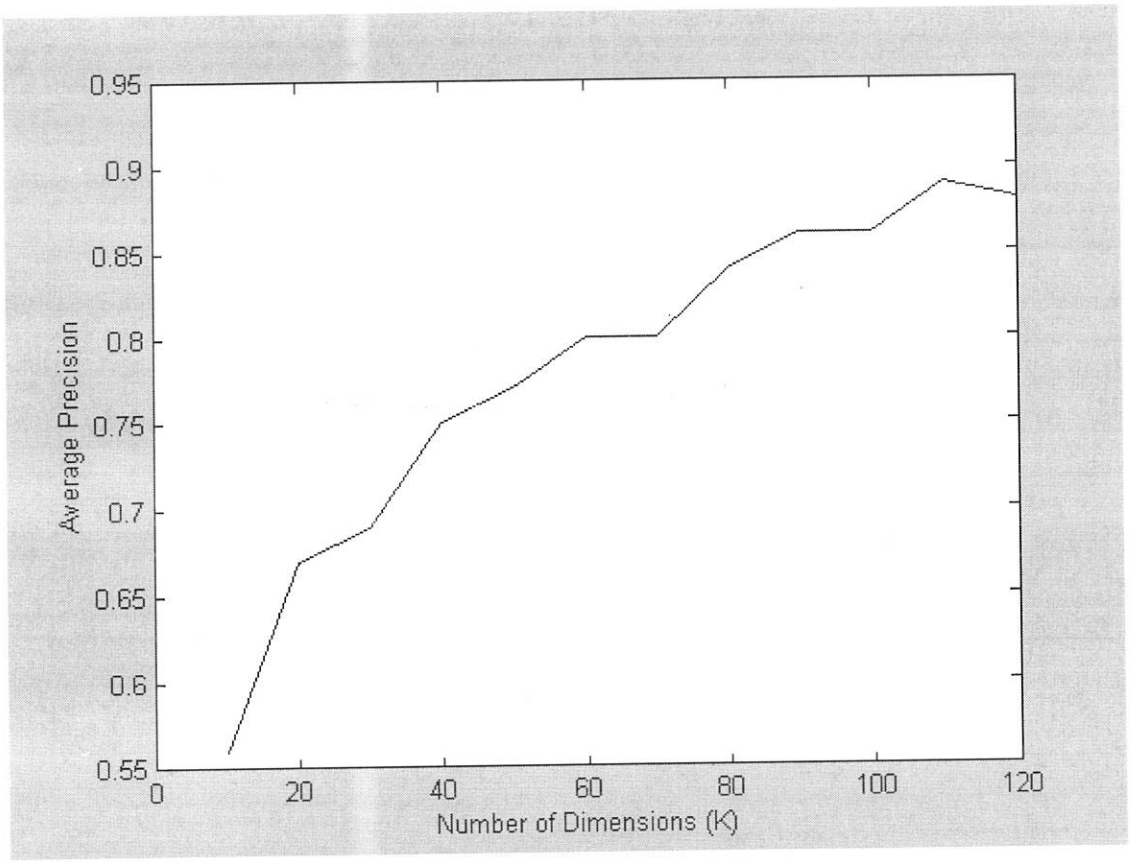

Figure 3.2 Average precision at recall points of 0.25, 0.50 and 0.75 for a range of LSI dimensions

Dimension	10	20	30	40	50	60	70	80	90	100	110	120
Mean Precision	0.56	0.67	0.69	0.75	0.77	0.80	0.80	0.84	0.86	0.86	0.89	0.88

Table 3.4 Dimension vs. Average precision at recall levels of 0.25, 0.50 and 0.75

From the figure it can be seen that performance continually increases from 10 or 20 dimensions, peaks between 100 and 110, and then began to diminish very slowly. Consequently, 110-dimensional representation is found to work well for the test collection. However, as compared to the number of documents in the collection, the dimension is really big. This might be due to the heterogeneous nature of the topics in the news articles in the test collection; smaller dimensions may not be adequate to capture the major patterns of word usage across the document collection. Large number of topics is discussed on the news in the collection. Even a single news article talks about three or more topics. For example document 92 only talks about construction of schools, roads, health care center, clinics and irrigation projects.

3.6 QUERY PROJECTION, MATCHING AND RANKING OF RELEVANT DOCUMENTS.

3.6.1 Query Projection

The queries are treated as pseudo-documents. Hence, the natural language pre-processing performed on the test document collection is also performed on the queries and the vectors indicating the frequency of each term in the queries are obtained. In addition to that, in order to represent the queries in the same space with the test document collection, they are weighted using the same scheme used in weighting the documents, and then projected into the space.

A Matlab function, designed to perform the weighting and project the queries into the reduced space using the formula "**q=qT *Uk * Sk^{-1}**" which is stated in section 2.7.4 (equ 2-17) is shown below.

```
Function newquery=queryproj(q, Gi, Uk, Sk)
% A function used to weight and project a query vector
% Gi is the global weight of each term produced from weighting the
% original term-document matrix
% Uk and Sk are the left singular vector and the diagonal matrix of the
% SVD of the original matrix

  for i=1:length(q)
      Wq(i,1)= log2(q(i)+1) *Gi(i);  %weighting of each term in the query
  end
 [M, N]=size(Sk);
T=Sk;
for i=1:M
      T(i, i)= 1/T(i, i);
end
newquery=Wq' * Uk * T;
```

3.6.2 Matching and Ranking of Relevant documents.

Now, that the queries and the documents are situated in the same reduced LSI space, it is possible to compute the Euclidean distance between the queries and each document and take the nearest documents as the best (relevant) ones for the specific query. However, as the term-document matrix was not normalized, a decision was made to use the cosine similarity measure, because by computing the angle, the lengths of the documents, which can affect the distance between the query and the documents in the space can be normalized (Salton, 1983).

Consequently, having the reduced dimensional representation of the query vector 'q' and the scaled document matrix 'D' (i.e. D=Vk*Sk), the angle between them can be computed by the formula

$$\text{Cos } x = \frac{q^T * D}{||q|| * ||D||}$$, where ||q|| and ||D|| are the norm of the query and document vector respectively.

After the cosine measure is computed, the documents are sorted according to the cosine coefficients, the larger the cosine coefficient, the more relevant the document.

The Matlab code developed to perform these tasks is given below.

```
Function [Svalue, Rank]=rankorder(D, q)
% A function used to compute the cosine of the angle between a query
% and each documents and rank the documents according to the
% similarity value
[M, N]=size(D);    % determines the dimensions (row and column) of D
  for j=1:N
    Di=D(1:M, j);   % select a column from the document matrix D
    R(i)= abs(Di' *q)/(norm(Di)*norm(q));
%R(i) computes the cosine of the angle between document vector Di and
% query q and store the value in a vector R
 end

[Svalue, Rank]= Sort(R)      % Rank the elements of R in ascending order
```

Here, the function 'rankorder' accepts the Document matrix 'D' and the query 'q' and returns two output vectors. The first vector 'Svalue' contains the cosine similarity values sorted in ascending order and the vector 'Rank' contains the corresponding document numbers.

The same function can also be used to compare any document matrix and query vector with compatible dimensions. The same function is used with the standard vector space model to compute the similarity of each vector in the weighted term-document matrix with the queries before projection into the reduced LSI space.

3.7 TESTING

The classic Recall and Precision measures are used to judge the retrieval performance of the two indexing approaches. An attempt has been made to compare the results of the Latent Semantic Indexing (LSI) method against the standard vector space approach. For the vector space model, the same term by document matrix, that was the starting point for the LSI method, is used.

The automatic indexing of the test document collection resulted in 9256 indexing terms that occur in at least one document, and not in a stop-word list prepared for this research. Some addition characteristics of the dataset are given in the table below.

Number of Unique terms	**Standard Vector space**	**LSI**
Number of Unique terms	9256	9256
Mean Number of terms per Document	186	186
Mean number of terms per query	4	4
Mean number of relevant documents per query	6	6

Table 3.5: Some characteristics of the datasets

To evaluate the ranked list of outputs returned by the different indexing approaches, precision is plotted against recall after each relevant document. However, to facilitate computing average performance over a set of queries – each with a different number of relevant documents – individual query precision values are interpolated to a set of standard recall levels, from 0 to 1 in steps of 0.1.

The particular rule used to interpolate precision at a specific standard recall level, say i, is to use the maximum precision obtained for the query for any actual recall level greater than or equal to i (Baeza-Yates, 1999). As an example, the following sample tables are used to calculate precision at standard recall levels using the above procedure.

Query No	Query	Relevant Document number	Ranked list of output documents
15	የሰብል ዝርያ ማሻሻያ ምርምር	47, 78, 115, 159, 176	**115**, **176**, 118, **47**, 3, **159**, 49, **78**, 80, 136, 179, 146, 132, 125, . . .
24	የጤና ተቋማት ግንባታ	22, 130, 117, 149, 92, 191, 205	**22**, **149**, 185, 40, **130**, **191**, 197, 183, **205**, 109, **117**, 140, . . .

Table 3.6 Query, relevant document list and ranked list of output documents

Note: - In the ranked list of output documents the numbers in bold represent relevant documents.

Query No	Standard Recall Points										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Doc15	1.00	1.00	1.00	1.00	1.00	0.75	0.75	0.67	0.67	0.63	0.63
Doc24	1.00	1.00	1.00	0.60	0.60	0.67	0.56	0.56	0.55	0.06	0.06

Table 3.7 Precision at standard recall levels for query 15 and 24.

The precision values were obtained after the implementation of the interpolation procedure.

Similarly, for each available query (25 in number) precision at the standard recall points are computed and then averaged over all queries. The following table presents the results obtained for the LSI and standard vectors space approaches.

Recall Level	Average Precision for 25 queries	
	LSI Method	Standard Vector Method
0.00	0.9232	0.8960
0.10	0.9000	0.8828
0.20	0.8968	0.8788
0.30	0.8540	0.8116
0.40	0.8428	0.7960
0.50	0.7960	0.7764
0.60	0.6780	0.6644
0.70	0.5832	0.5784
0.80	0.5352	0.5456
0.90	0.4460	0.3940
1.00	0.4180	0.3800

Table 3.8 Average recall-precision results for LSI and standard vector space indexing methods.

The figure below shows precision as a function of recall for LSI 110-factor and standard vector space methods.

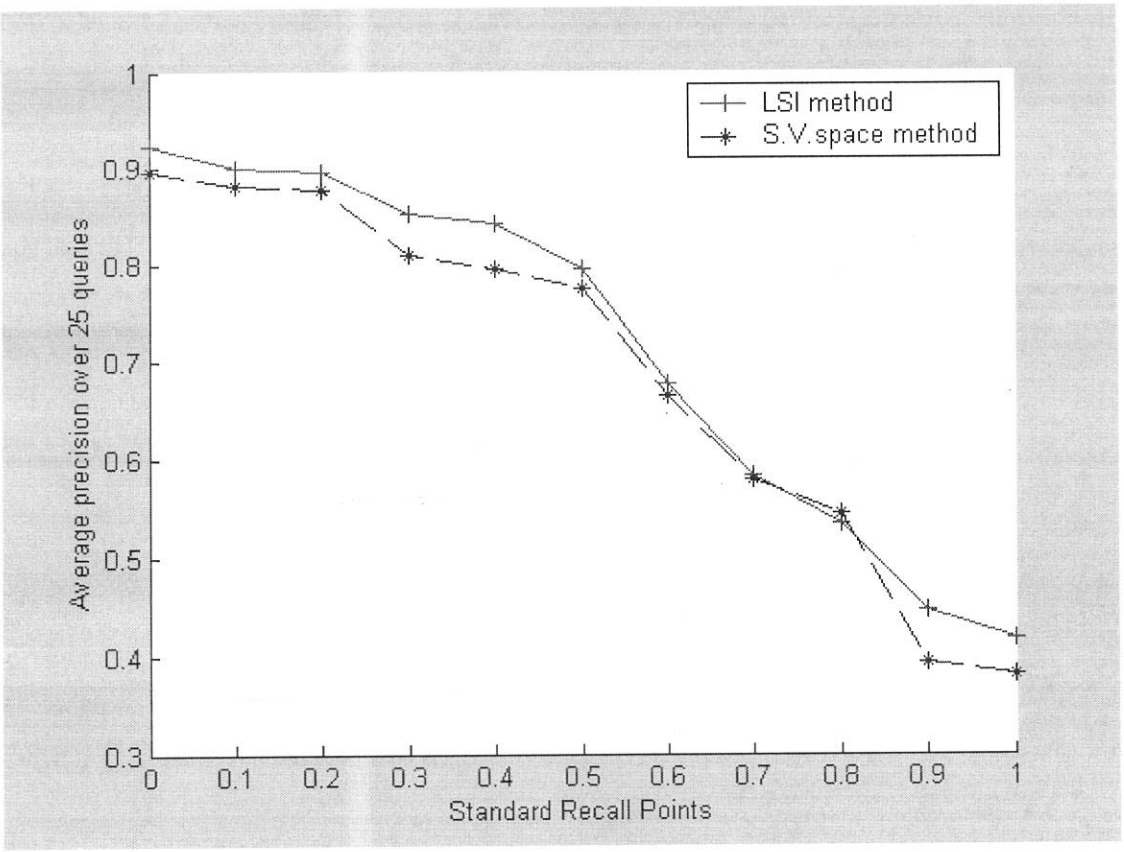

Fig 4.3 Average Recall-precision graph for two indexing methods

As can be seen from the graph for all recall levels, except at recall level 0.8, the precision of the LSI model is above that of the vector space.

The average difference in precision between the LSI and the standard vector method is 0.0244, which is the difference between the average precision for the LSI approach (0.7157) and the average precision of the vector space approach (0.6913). This represents a 2.4% improvement over the standard vector space method.

Since it is difficult to judge the significance of the differences between the two performance measures by simply citing percentage improvement, an attempt has been made to furnish statistical evidence indicating whether the resulted difference between the two averages is in fact significant.

For this purpose a simple sign test is used. In table 3.9 a set of output data from a sign test process is shown for the two indexing methods LSI and Vector space. The table is based on the 11 standard recall levels. For each recall level, the table shows the number of queries favoring LSI and vector space approaches (ignoring ties), respectively, and the sign ('+' and '-'.) of the differences between the two values.

Recall level	Favoring		Tied	Sign of change from LSI to Vector space
	LSI	Vector Space		
0.00	3	0	22	+
0.10	4	1	20	+
0.20	4	1	20	+
0.30	7	2	16	+
0.40	6	3	16	+
0.50	4	5	16	-
0.60	8	7	10	+
0.70	10	6	9	+
0.80	9	7	9	+
0.90	13	6	6	+
1.00	14	6	5	+

Table 3.9 Sign test for LSI and Vector space retrieval approaches

To test the claim that the performance difference between the two methods did not occur by chance, the following hypothesis is formulated and tested at 0.05 levels of significance (95% confidence).

H_0= There is no difference between the performance the two approaches (i.e. the observed difference is not significant) (Null hypothesis)

H_1= LSI method is better than standard vector space

As can be seen from table 3.9 above, at 10 of the 11 recall levels the number of queries favoring LSI is greater.

For 11 recall points with 1 statistic (the number of times the less frequent sign occurs) and at 0.05 level of confidence a statistical table is referred to get the critical point (one tailed). The critical point is 2, which is greater than the statistic (less frequent sign). This resulted in the rejection of the null hypothesis. That is, the observed performance difference between the two methods is significant and for this research, LSI is statistically better than Vector space.

3.8 DISCUSSION

In order to understand the retrieval performance of the LSI method better, an examination of two kinds of failures of the method has been made. First, documents that the LSI method ranked highly but are judged to be irrelevant by the journalists (the persons who made the relevance judgment) were examined. Second, those documents that are classified as relevant, but are not in the top 30 returned by the LSI model were examined.

The observations presented below are based on preliminary analysis of some query topics on which the model performed relatively poorly.

The most common reason for the first kind of failure was lack of specificity. The highly ranked but irrelevant documents were generally about the topic of interest but not meet some of the restrictions described in the query topic. Some queries request answer to specific issue and demand the returned documents to focus exclusively on that particular issue. However, LSI systems were not designed to handle such kind of problems (Deerwester, et al., 1990). For instance, a query that requests news articles on the 'complain of Telephone customers', retrieved news articles which talks about telephone service expansion activities in the country in the top list.

For some of the top ranked but irrelevant documents it was not clear why they appear in the first row of the ranked output. Particularly, this phenomena was prevalent for smaller dimensional representation of the LSI space (k=20, 25, 30). As we have discussed in chapter two section 2.7, one advantage of LSI method of retrieval is that documents can match queries even when they share no word in common. This is because, LSI uses a statistically derived, 'semantic' space and not exact word overlap for matching queries to documents (Deerwester, et al., 1990; Dumias, 1993). Hence, this advantage of LSI method can some times result in spurious answers.

In order to analyze the second kind of problem, missing relevant documents, a random subset of relevant articles that were not in the top 30 returned by the LSI method were examined.

Many of the missed documents examined in the process represent news that were about a different topic than the query, but contains a few lines or sentences that are relevant to the query. In each document in the subset the majority of the terms were dedicated for the discussion of the topic advertised on the title. However, in chapter two section 2.7 it was mentioned that in LSI space documents are represented as the average of their constituent term vectors. This fact indicates that documents often tend to be near the dominant concept discussed in them.

CHAPTER FOUR

CONCLUSION AND RECOMMENDATION

4.1 CONCLUSION

Information retrieval (IR) is concerned with locating documents that are relevant for a user's information need or query from a large collection of documents. A fundamental problem with the classic information retrieval systems is that different ways of expressing the same concept may lead to missing of relevant documents and retrieval of irrelevant documents. A query is often a short and incomplete description of the information need. The users of IR systems and the authors of the documents often use different words to refer to the same concept.

The goal of the research is to examine the benefits of LSI text retrieval approach for Amharic text retrieval. Investigation on the retrieval performance of two text retrieval approaches has been made: standard vector space; and Latent Semantic Indexing, and the experimental results have been presented.

In this research, a test collection of 206 news articles were taken. The entire text of the news was used for indexing purpose. During the preparation of the text for indexing, punctuation marks, space, tab, carriage return and line feed character were used as word identifiers.

Each document is indexed automatically. All terms that do not occur in stop word list of 745 common words (including news specific common words) were included in the analysis. The automatic indexing resulted in a total 9256 unique words.

The analysis begins with a weighted term by document matrix in which each number in the cell of the matrix represent the importance of the term within as well as in the entire document collection. This weighted matrix is used for indexing and retrieval by the standard vector space method.

The weighted matrix was analyzed further by the Singular Value Decomposition to derive the latent structure model which was used for indexing and retrieval by the LSI retrieval model. The SVD of the matrix was computed for a range of dimensions and the optimum dimension (110 for this research) for the collection was chosen by the criteria of 'best retrieval performance'. Queries were placed in the resulting space at the resultant of their constituent terms.

Cosine similarity measure between the query vector and document vectors were used to identify documents which are near (relevant) to each queries and the documents are ranked according the their cosine measures.

At last, the outputs of the latent semantic indexing (LSI) method were compared against the outputs of standard vector space.

Precision at several different levels of recall was measured. This was done separately for each of the 25 queries and then averaged over all queries. Except at one recall level, the recall-precision graph of the LSI method was above the standard vector method. The biggest differences between the two approaches were observed at the last two high recall levels (0.90 and 1.00).

4.2 RECOMMENDATIONS

This research is just an attempt to see the advantageous of LSI in Amharic text retrieval. There are a number of directions in which the work in this thesis can be further pursued.

- The size of the test collection used in this research is too small. However, one can increase the test collection in many folds through the use of external servers.

- The index terms that are used to represent documents and queries were not stemmed. Using stemmed index terms, apart from enhancing performance it will significantly reduce the number of unique index terms in the collection.

- A retrieval system should allow addition of documents and terms and also removal of obsolete terms and documents into/from the collection. One can consider incorporating these capabilities into the model.

- One of the most important and robust technique of improving performance of retrieval systems is relevance feedback (Salton, 1983). One can consider enhancing the model, so that the model reformulates the query, submitted by user, automatically by altering it based on the user's feedback about which documents are relevant to the initial request.

- Applying LSI for cross language retrieval (Amharic vs. English) and information filtering on Amharic documents is highly advisable.

REFERENCES

Abyot, B. (2000). Design and Development of Amharic Word parser. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa.

Adriani, M. & Croft Bruce. (1997). Retrieval Effectiveness of Various Indexing Techniques on Indonesian News Articles. http://citeseer.nj.nec.com/459174.html

Baeza-Yates, R. & Ribeiro-Neto, B. (1999). Modern Information Retrieval. England. Addison Wesley Longman Limited.

Baye, Y. (1992). Ethiopian Writing System http://www.ethiopians.com/bayeyima.html

Bender, M. L., Sydeny W. Head, and Roger Cowley. (1976). The Ethiopian Writing System. In Bender et el (Eds.) Languages of Ethiopia. London: Oxford University Press.

Berry, M. W., Drmac, Z. & Jessup, R. (1999). Matrices, Vector Spaces, and Information Retireval. http://epubs.siam.org/sam-bin/gelfile/SIREV/articles/34703

Berry, M. W., Dumais, S.T. and Letsche, T. (1995). Computational Methods for Intelligent Information Access.

http://citeseer.nj.nec.com/cache/papers/cs/21177/http:zSzzSzmiles.cnuce.cnr.itzSz~palmerizSzdatamzSzarticleszSzSC95.pdf/berry95computational.pdf

Berry, M. W., Dumais, S. T., and O'Brien, G. W. (1995). Using Linear Algebra for Intelligent Information Retrieval. *SIAM Review, 37(4)*, 1995, 573-595.

http://citeseer.nj.nec.com/cache/papers/cs/143/http:zSzzSzwww.cs.utk.eduzSz~libraryzSzTechReportszSz1994zSzut-cs-94-270.pdf/berry95using.pdf

Bethlehem, M, A. (2002). The Application N-gram-Based Indexing in Amharic Text Retrieval. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa.

Brian C. Vickery & Alian Vickery. (1987). Information Theory and Practices. Great Britain: Anthony Rowe. Ltd. Chippenham.

Carmel, D. & Soffer, Aya. (2003). Probabilistic Models of Information Retrieval. Haifa University
http://cs.haifa.ac.il/courses/infor/Slides/ Probabilistic%20Models

Chowdhury, G. (1999). Introduction to Modern Information Retrieval. London: Library Association Publishing.

Croft, W. and Jinxi, Xu. (1995). Corpus-Specific Stemming Using Word Form Co-Occurrence.
http://citeseer.nj.nec.com/cache/papers/cs/97/http:zSzzSzciir.cs.umass.eduzSzinfozSzpsfileszSzirpubszSzunlv.pdf/croft95corpusspecific.pdf

Daniel, Y. (1994). The Difference Between the Geez and Ethopic Scripts.
http://www.abyssiniacybergateway.net/fidel/HISTORY. EthioSciences and soc.culture.african on April 13th.

Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A. (1990). Indexing by Latent Semantic Analysis. Journal of the Society for Information Science, 41(6), 391-407.
http://citeseer.nj.nec.com/cache/papers/cs/339/http:zSzzSzsuperbook.bellcore.comzSz~stdzSzpaperszSzJASIS90.pdf/deerwester90indexing.pdf

Dereje, T. (1999). Optical Character Recognition of Typewritten Amharic Text. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa.

Dumais, S. T. (1991). Improving the Retrieval of Information from External Sources: *Behavior Research Methods, Instruments and Computers, 23(2)*, 229-236.
http://lsi.research.telcordia.com/lsi/LSIpapers.html

Dumais, S. T., Letsche, T. A., Littman, M. L. and Landauer, T. K. (1997). Automatic Cross-Language Retrieval Using Latent Semantic Indexing.
http://citeseer.nj.nec.com/cache/papers/cs/2858/http:zSzzSzwww.cs.duke.eduzSz~mlittmanzSzpaperszSzx-lang-aaai97.pdf/dumais97automatic.pdf

Dumais, S. T. (1992). Enhancing Performance in Latent Semantic Indexing (LSI) Retrieval.
http://citeseer.nj.nec.com/cache/papers/cs/5986/http:zSzzSzsantana.uni-muenster.dezSzLibraryzSzInformationRetrievalzSzlsi.papers.brmic91.pdf/dumais92enhancing.pdf

Dumais, S.T. (1993). LSI meets TREC: A Status Report. "In: D. Harman (Ed.), *The First Text REtrieval Conference (TREC1), National Institute of Standards and Technology Special Publication 500-207,* pp. 137-152.
http://trec.mist.gov/pubs/trec1/papers

Dumais, S. T. (1994). Latent Semantic Indexing (LSI) and TREC-2: *The Second Text REtrieval Conference (TREC2), National Institute of Standards And Technology Special Publication 500-215,* pp. 105-116. http://citeseer.nj.nec.com/cache/papers/cs/728/http:zSzzSztrec.nist.govzSzpubszSztrec2zSzpaperszSzpszSzbellcore.pdf/latent-semantic-indexing-lsi.pdf

Dumais, S. T. (1995). Using LSI for Information Filtering TREC-3 Experiments: *The Third Text REtrieval Conference (TREC3) National Institute of Standards and Technology Special Publication.* http://lsi.research.telcordia.com/lsi/LSIpapers.html

Dumias, T. (2002). Massive Data Sets: Transcription of the Application presentation by Susan Dumias, Belicore. http://www.nap.edu/readingroom/books/massdata/media/sdumais-t

Dumais, S. T., Furnas, G. W., Landauer, T. K. and Deerwester, S. (1988). Using latent semantic analysis to improve information retrieval. *In Proceedings of CHI'88: Conference on Human Factors in Computing, New York: ACM, 281-285.* http://lsi.research.telcordia.com/lsi/LSIpapers.html

Ermias, A. (1998). Recognition of Formatted Amharic Text Using Optical Character Recognition (OCR) Techniques. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa.

Foltz, P. W (1990). Using Latent Semantic Indexing for Information Filtering. In R. B. Allen (Ed) *Proceedings of the Conference on Office Information Systems, Cambridge, MA, 40-47.* http://www-psych.nmsu.edu/~pfoltz/cois/filtering-cois.html

Foltz, P. W. and Dumais, S. T. (1992). Personalized Information Delivery: An Analysis of Information Filtering Methods: *Communications of the ACM, 35(12),* 51-60
http://www-psych.nmsu.edu/~pfoltz/cacm/cacm.html

Furnas, W., Deerwester, S., Dumais, S. T., Landauer, K., Harshman, A., Streeter, A. & Lochbaum. (1988). Information Retrieval Using a Singular Value Decomposition Model of Latent Semantic Structure.
http://www.ics.uci.edu/~pratt/courses/papers/p465-furnas

Getachew H. (1967). The Problem of Amharic Writing System. Unpublished

Gohaman, N. El-Ghazawi, T. & Grossman, D. (ND). Enterprise Text Processing: A Sparse Matrix Approach.
http://www.ir.iit.edu/publications/downloads/goharian_ITCC01

Hong, I. (2000). An Overview of Latent Semantic Indexing.
http://www.cs.berkeley.edu/~jasonh/classes/sims240/sims-240-final-paper-lsi

Husbands, P., Simon, H., Ding, C. & Berkeley, L. (2001). On the Use of Singular Value Decomposition for Text Retrieval.
http://citeseer.nj.nec.com/cache/papers/cs/26364/http:zSzzSzwww.nersc.govzSzresearchzSzSCGzSzcdingzSzpapers_pszSzhsd4.pdf/husbands00use.pdf

Isaacs, M. (2002). Discovery of Relationships Between Medical Entities Using Singular Value Decomposition.
http://www.cs.ucsb.edu/~isaacs/report

Jody S. Hourigan & Lynn V. McIndoo.(ND). Singular Value Decomposition.
http://online.redwoods.cc.ca.us/instruct/darnold/laproj/Fall98/JodLynn/report2.pdf

Kolda, G. & Dianne P. O'Leary. (1998). A Semidiscrete Matrix Decomposition for Latent Semantic Indexing in Information Retrieval. http://citeseer.nj.nec.com/cache/papers/cs/12127/http:zSzzSzcsmr.ca.sandia.gov zSz~tgkoldazSzpaperszSzsddlsi-acmtois.pdf/kolda97semidiscrete.pdf

Kolda, G. & O'leary, P. (2002). A Semidiscrete Matrix Decomposition for Latent Semantic Indexing in Information Retrieval. http://csmr.ca.sandia.gov/~tgkolda/papers/sddlsi-acmtois.pdf

Kontostathis, A. & Pottenger, M. (2002). A Mathematical View of Latent Semantic Indexing: Tracing Term co-occurrences. http://www.cse.lehigh.edu/techreports/2002/LU-CSE-02-006

Landauer, T. K. and Littman, M. L. (1990). Fully Automatic Cross-Language Document Retrieval Using Latent Semantic Indexing. In *Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research,* pp. 31-38. http://lsi.research.telcordia.com/lsi/LSIpapers.html

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. Discourse Processes, 25, 259-284. http://lsa.colorado.edu/whatis.html

Matsumura, N., Ohsawa, Y. and Ishizuka, M. (2002). PAI: Automatic Indexing for Extracting Asserted Keywords from a Document. http://www.kc.t.u-tokyo.ac.jp/~matumura/papers/matumuraNGC-PAI.pdf

Million, M. (2000). A Generalized Approach to Optical Character Recognition of Amharic Texts. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa.

Moens, M. (2000). Automatic Indexing and Abstracting of Document Texts. Kluwer Academic Publishers, London.

Nega, A. & Willet, P. (2002). Stemming of Amharic Words for Information Retrieval. http://www3.oup.co.uk/litlin/hdb/Volume_17/Issue_01/pdf/170001.pdf

Nigussie, T. (2000). Handwritten Amharic Text Recognition Applied to the Processing of Banking Cheques. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa.

Ozel, B. (2002). Latent Semantic Indexing by Singular Value Decomposition. http://cs.bilgi.edu.tr/pages/academic_staff/assistants/b-u-lent_ozel/LSI-Chapter

Penenovic, Z. (1997). Image Retrieval Using Latent Semantic Indexing. (Final Year Graduate Thesis). Department of Electrical Engineering AudioVisual Communication Laboratroy. http://lcavwww.epfl.ch/douments/final.html.

Punctuation. Microsoft®Encarta®Encyclopedia. © 1993-1998. Microsoft Corporation.

Radu, A. (2001). Implementation of Term Weighting in Simple IR systems. http://www.cs.helsinki.fi/u/popescu/docs/ir.pdf

Rijsbergen, V. (1996). Information Retrieval.
http://www.dcs.gla.ac.uk/keith

Saba, A. (2001). The Application of Information Retrieval Techniques to Amharic Documents on the Web. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa

Salton, G. (1983). Introduction to Modern Information Retrieval. U.S.A McGraw-Hill, Inc

Spark Jones, K. & Willet, P. (1997). Reading in Information Retrieval, San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Stalling, W. (1993). Computer Organization & Architecture: Principles of Structure and Functions. New York: Macmillan Publishing Company.

Vickery, C. & Vickery, A. (1987). Information Theroy and Practices. Great Britain, Anthony Rowe. Ltd. Chippenham.

Wall, E., Wall, Andreas Rechtsteiner & Luis M. Rocha. Singular Value Decomposition and Principal Component Analysis.
http://arxiv.org/ftp/physics/papers/0208/0208101.pdf

Witter, I. Berry, W. (2001). Down Dating the Latent Semantic Indexing Model for Conceptual Information Retrieval.
http://www3.oup.co.uk/computer-journal/subs/volume_41/Issue_08/witten

Worku Alemu (1997). The application of OCR Techniques to the Amharic Script. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa.

Yaregal, A. (2002). Optical Character Recognition of Amharic Text: An Integrated Approach. School of Information Studies for Africal. Addis Ababa University. Addis Ababa.

Zelalem, S. (2001). Automatic Classification of Amharic News Items: The Case of Ethiopian News Agency. School of Information Studies for Africal. Addis Ababa University. Addis Ababa.

ደስታ ተክለ ወልድ፤ 1970፤ ዐዲስ ያማርኛ መዝገበ ቃላት፤ አርቲስቲክ ማተሚያ ቤት፤ አዲስ አበባ

ከሣቴ ብርሃን ተሰማ፤ 1951፤ የዕማርኛ መዝገበ ቃላት፤ አርቲስቲክ ማተሚያ ቤት ታተመ፤ አዲስ አበባ

APPENDIX: I

Order							Labialized				
1st	2nd	3rd	4th	5th	6th	7th					
H	Hù	£	¦	ÿ	H	ç					
L	Lù	Lþ	Ş	l¤	L	lÖ			Ş*		
¼	¼ù	¼þ	ˆ	¼¤	Þ	‡					
m	Ñ	¸	¥	»	M	ä			à		
¿	¿ù	Œþ	œ	œ¤	o	ƒ					
R	„	¶	‰	Ê	R	é			…		
S	Sù	Sþ	ú	s¤	S	î			ú*		
¹	¹ù	¹þ	š	¹¤	>	ë			š*		
Q	qÜ	qE	Ý	q½	Q	ö	³	q&	Ì	Ì	q"
B	Bù	Bþ	Æ	b¤	B	ï			Æ*		
T	tÜ	tE	¬	t½	T	è			a		
C	cÜ	cE	Ò	c½	C	Ó			–		
^	^ù	^þ	`	^¤	~	‚	^Ö	^	°	°¤	^"
N	Nù	Nþ	Â	n¤	N	ñ			•		
ß	ßù	ßþ	¾	ß¤	Ÿ	®			—		
X	Xù	Xþ	¨	x¤	X	å					
W	Ý	êE	ê	ê½	W	ã					
;	;ù	>þ	>	>¤	:	â					
K	Kù	Kþ	µ	k¤	K	÷	kÖ	k&	µ*	µ¤	k"
¡	¡ù	¡þ	á	¡¤	<	ó					
Z	Zù	Zþ	²	z¤	Z	ø			²*		
¢	¢Ü	¢E	Ï	¢½	i	Î					
Y	†	'	Ã	ü	Y	×					
G	Gù	Gþ	U	g¤	G	¯	gÖ	g&	Ù	g¤	g"
D	Ç	Äþ	Ä	Á	D	ì			Ä*		
J	°	©þ	©	Ë	J	í					
«	«ù	«þ	È	«¤		õ			È*		
=	Œ	À	Å	~	u	ô			′		
™	™ù	™þ	Ú	™¤	A	Û			Ú*		
{	{ù	Éþ	É	É¤	I	ò					
′	′ù	′þ	Ô	′¤	e	Õ					
F	Û	Ð	Í	Ø	F	æ			Ð		
P	pÜ	pE	-	p½	P	±					
V	Vù	Vþ	Š	v¤	V	<					

APPENDIX: II

%	&	'	(	)	*	+	,	-	0
10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
1	2	3	4	5	6	7	8	9	
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	

APPENDIX: III

ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
H	h#	£	¦	ÿ	H	ç
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
L	l#	l!	§	l@	L	lÖ
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
/	/#	/!	ˆ	/@	?	‡
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
M	Ñ	¸	¥	»	**M**	ä
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
\	\#	œ!	œ	œ@	|	*f*
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
R	„	¶	‰	Ê	**R**	é
ስ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
S	s#	s!	ú	s@	S	î
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ
¹	¹#	¹!	š	¹@	>	ë
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
Q	q$	qE	”	³	Q	ö
ብ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
B	b#	b!	Æ	b@	B	ï
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
T	t$	tE	¬	t&	T	è
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ
C	c$	cE	Ò	c&	C	Ó
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
X	**x#**	**x!**	¨	x@	X	å
ን	ኑ	ኒ	ና	ኔ	ን	ኖ
N	n#	n!	Â	n@	**N**	ñ
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ
ß	ß#	ß!	¾	ß@	"	®
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
K	k#	k!	µ	k@	K	÷
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ
,	,#	,!	–	,@	<	—
ወ	:	ዊ	ዋ	ዌ	ው	ዎ
W	Ý	ê!	ê	ê&	W	ã
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
;	;#	›!	›	›@	:	â
ዝ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ

Z	z#	z!	²	z@	Z	ø
ገ	ገ	ገ	ገ	ገ	ጘ	ገ
¢	¢$	¢E	Ï	¢&	™	Î
ይ	ዩ	ዩ	ያ	ዬ	ይ	ዮ
Y	†	‘	Ã	ü	Y	×
ድ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
D	Ç	d!	Ä	Á	D	ì
ጅ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ
J	°	©!	©	Ë	J	í
ግ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
G	g#	g!	U	g@	G	¯
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
-	-#	-!	È	-@	_	õ
ጨ	ጩ	ጪ	ጫ	ጭ	ጭ	ጮ
=	=#	À	Å	~	+	ô
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
’	’#	’!	Ô	’@	ù	Õ
θ	θ	ፚ	ፃ	ፄ	ዕ	ፆ
]	]#	É!	É	É@	}	ò
ጸ	ጸ	ጺ	ጻ	ጼ	ጽ	ጾ
[	[	[!	Ú	[@	A	Û
ፍ	ፊ	ፋ	ፉ	ፌ	ፍ	ፎ
F	Û	Đ	Í	Ø	F	æ
ፕ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ
P	p$	pE	-	p&	P	±
ቨ	ቩ	ቪ	ቫ	ቬ	ቨ	ቮ
V	v#	v!	Š	v@	V	‹
ኧ	ፏ	ቿ	ኇ	ቖ	ኋ	ቷ
%	Đ	–	—	,	“	a
ቕ	ኗ	ሯ	ኟ	ሟ	ቜ	
¡	•	…	Þ	à	O	
ጪ	ቓ	ቚ	ቋ	ኃ	ኃ	ቧ
Þ	a	Ÿ	Ì	Ù	`	Æ*

The diacritic markings,

!	@	#	$	&	*	E

APPENDIX: IV

The queries and their corresponding relevant document numbers		
Query01	የኤድስን ስርጭት መግታት	Doc020, doc026, doc040, doc046, doc051, doc059, doc081, doc106, doc113, doc116, doc137, doc138, doc150, doc175, doc185', doc186, doc195, doc197,
Query02	የንጹህ መጠጥ ውሀ ፕሮጀክቶች	Doc001, doc007, doc022, doc075, doc092, doc104, doc105, doc117, doc120, doc122, doc123, doc127, doc140, doc143, doc158, doc164, doc189, doc190, doc191, doc204, doc205,
Query03	በኢትዮ ኤርትራ ጦርነት ከቤት ንብረታቸው የተፈናቀሉ ወገኖች	Doc029, doc068
Query04	የወረዳና የቀበሌ ምክር ቤቶች ምርጫ	Doc024, doc087, doc091, doc095,
Query05	ጎጂ ልማዶች	Doc032, doc50, doc73
Query06	የኤክስቴንሽን ልማት መአቀፍ	Doc080, doc122, doc148, doc152
Query07	የአማራ አቀፍ ልማት ማህበር የስራ እንቅስቃሴ	Doc027, doc031, doc205
Query08	የትምህርት ቤቶች ግንባታ	doc075, doc092, doc117, doc149, doc172, doc183, doc205
Query09	በድርቅ ለተጎዱ ተረጂዎች የሚደረግ እርዳታ	Doc037, doc059', doc084, doc085, doc089, doc123, (doc127), doc133, doc139, doc141, doc158, doc167, doc168, doc169, doc174, doc196
Query10	የወባ በሽታ መከላከል እና ቁጥጥር	Doc156, doc182,
Query11	የመስኖ ልማት	doc010, doc053, doc092, doc139, doc199, doc141
Query12	መንግስታዊ ያልሆኑ ድርጅቶች ድርጅቶች የሚያደርጉት የልማት እንቅስቃሴ	Doc013, doc045, doc059, doc074, doc075, doc084, doc106, doc116, doc141,
Query13	የቅርሶች እንክብካቤ እና ጥበቃ	Doc052, doc162, doc192
Query14	ጠለፋ እና አስገድዶ መድፈር	Doc021, doc036, doc039, doc50, doc106, doc138
Query15	የሰብል ዝርያ ማሻሻያ ምርምር	Doc176, doc159, doc115, doc78, doc047

Query16	የስልክ አገልግሎት ተጠቃሚዎች አቤቱታ	Doc004, doc028, doc096,
Query17	የቅድመ ጋብቻ የኤድስ ምርመራ	Doc046, doc175
Query18	የትራፊክ አደጋ	Doc099, doc203
Query19	የኢትዮጵያ ማህበራዊ ተሀድሶና ልማት ፈንድ	Doc001, doc007, doc022, doc027, doc054, doc092, doc149, doc204, doc205
Query20	ነጋዴ ሴቶች	Doc018, doc086
Query21	የውሀ ጉድጓዶች ቁፋሮ	Doc001, doc007, doc092, doc105, doc117, doc120, doc123, doc143, doc164, doc189, doc204, doc205
Query22	የኢንቨስትመንት ግንባታ	Doc109, doc126, doc144, doc146, doc155
Query23	ነጻው ፕሬስ	Doc102, doc110
Query24	የጤና ተቋማት ግንባታ	Doc092, doc191, doc205
Query 25	ሻአቢያ ያሰራቸው ኢትዮጵያውያን	Doc002, doc029

APPENDIX: V – Precision at Standard recall points for the LSI Method

Query 1	1	0.75	0.67	0.6	0.22	0.17	0.14	0.15	0.12	0.1	0.19
Query 2	1	1	1	1	0.91	0.92	0.68	0.65	0.33	0.29	0.24
Query 3	1	1	1	1	1	1	0.03	0.03	0.03	0.03	0.03
Query 4	1	1	1	1	1	1	1	1	1	1	1
Query 5	1	1	1	1	1	1	1	1	1	1	1
Query 6	1	1	1	1	1	1	1	1	1	0.04	0.04
Query 7	1	1	1	1	1	1	1	0.18	0.18	0.18	0.18
Query 8	1	1	1	0.5	0.5	0.5	0.3	0.3	0.14	0.15	0.15
Query 9	1	1	1	1	1	1	0.9	0.69	0.28	0.14	0.08
Query 10	1	1	1	1	1	1	1	1	1	1	1
Query 11	1	1	1	1	1	1	1	0.16	0.16	0.16	0.16
Query 12	1	1	1	0.75	0.8	0.22	0.19	0.18	0.08	0.06	0.06
Query 13	0.08	0.08	0.08	0.08	0.05	0.05	0.05	0.07	0.07	0.07	0.07
Query 14	1	1	1	1	1	1	1	0.63	0.65	0.32	0.32
Query 15	1	1	1	1	1	1	1	1	1	0.63	0.63
Query 16	0.5	0.5	0.5	0.5	0.67	0.67	0.67	0.6	0.6	0.6	0.6
Query 17	1	1	1	1	1	1	1	1	1	1	1
Query 18	1	1	1	1	1	1	0.67	0.67	0.67	0.67	0.67
Query 19	1	1	1	1	1	1	1	1	1	1	0.59
Query 20	1	1	1	1	1	1	0.08	0.08	0.08	0.08	0.08
Query 21	1	1	1	1	1	1	1	0.9	0.91	0.79	0.52
Query 22	1	1	1	0.67	0.67	0.2	0.2	0.25	0.25	0.04	0.04
Query 23	1	1	1	1	1	1	1	1	1	1	1
Query 24	1	0.67	0.67	0.75	0.75	0.67	0.71	0.71	0.5	0.47	0.47
Query 25	0.5	0.5	0.5	0.5	0.5	0.5	0.33	0.33	0.33	0.33	0.33

APPENDIX: VI- Precision at Standard recall levels for the Vector Space Method.

Query 1	1	0.67	0.57	0.32	0.28	0.24	0.16	0.14	0.1	0.11	0.11
Query 2	1	1	1	0.89	0.9	0.75	0.78	0.75	0.71	0.24	0.21
Query 3	1	1	1	1	1	1	0.01	0.01	0.01	0.01	0.01
Query 4	1	1	1	1	1	1	1	1	0.67	0.67	0.67
Query 5	1	1	1	1	1	1	1	1	1	1	1
Query 6	1	1	1	1	1	1	1	1	1	0.05	0.05
Query 7	1	1	1	1	1	1	1	0.16	0.16	0.16	0.16
Query 8	1	1	1	0.75	0.75	0.57	0.5	0.5	0.12	0.05	0.05
Query 9	1	1	1	1	1	1	1	0.76	0.76	0.58	0.26
Query 10	1	1	1	1	1	1	0.67	0.67	0.67	0.67	0.67
Query 11	1	1	1	1	1	1	1	0.38	0.38	0.05	0.05
Query 12	1	1	1	1	0.44	0.45	0.3	0.16	0.19	0.05	0.05
Query 13	0.07	0.07	0.07	0.07	0.1	0.1	0.1	0.02	0.02	0.02	0.02
Query 14	1	1	1	1	1	1	1	1	1	0.04	0.04
Query 15	1	1	1	1	1	0.75	0.75	0.67	0.67	0.5	0.5
Query 16	0.33	0.33	0.33	0.33	0.5	0.5	0.5	0.6	0.6	0.6	0.6
Query 17	1	1	1	1	1	1	1	1	1	1	1
Query 18	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Query 19	1	1	1	1	1	1	1	1	1	1	1
Query 20	1	1	1	1	1	1	0.18	0.18	0.18	0.18	0.18
Query 21	1	1	1	1	1	1	0.89	0.9	0.91	0.92	0.92
Query 22	1	1	1	0.33	0.33	0.38	0.38	0.17	0.17	0.06	0.06
Query 23	1	1	1	1	1	1	1	1	1	1	1
Query 24	1	1	1	0.6	0.6	0.67	0.56	0.56	0.54	0.06	0.06
Query 25	0.5	0.5	0.5	0.5	0.5	0.5	0.33	0.33	0.33	0.33	0.33

DECLARATIONS

This thesis is my original work, has not been presented for a degree in any other university and all sources of material used for the thesis have been duly acknowledged.

_____

Tewodros Hailemeskel Gebermariam

THE THESIS HAS BEEN SUBMITTED FOR EXAMINATION WITH OUR APPROVAL AS UNIVERSITY ADVISORS

_____

W/t Saba Amsalu

_____

Ato Kibur Lisanu