ADDIS ABABA UNIVERSITY

GRADUATE STUDIES

FACULTY OF COMPUTER AND MATHEMATICAL SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

AMHARIC DOCUMENT IMAGE RETRIEVAL USING LINGUSTIC FEATURES

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

BY: TILAHUN YESHAMBEL

ADVISOR: YAREGAL ASSABIE (PhD)

OCTOBER 21, 2011

ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY

GRADUATE STUDIES

FACULTY OF COMPUTER AND MATHEMATICAL SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

AMHARIC DOCUMENT IMAGE RETRIEVAL USING LINGUSTIC FEATURES

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

BY: TILAHUN YESHAMBEL

Name and Signature of Members of the Examining Board

	Name	**Signature**
Advisor:	Yargal Assabie (PhD)	_____
Chairperson:	_____	_____
Examiner:	_____	_____

ACKNOWLEDGMENT

This thesis could not have been completed without the help of many people. First and foremost, I would like to give sincere honour to my advisor, Dr. Yaregal Assabie for his guidance, his excellent and tireless support in this research, the patience and the confidence he showed to me and to my work. He provided me with the proper tools and motivation throughout my thesis work.

I feel also grateful to my families, Ato Zerihun Mekonnon and all his families who are living inside and outside the country for their support and encouragement for the completion of this work.

This thesis is the result of the effort of many instructors, students and other members at Addis Ababa University. So I would like to thank Ato Getachew Endale, who is the member of Amharic Department in Addis Ababa University, for providing me with the excellent facilities to complete this thesis. Thanks also to my friends who have contributed to the system implementation and its testing. I thank the reviewers for their valuable comments.

On top of that, I would also like to express my gratitude to Addis Ababa University for financial support and Department of Computer Science for providing me with tools such as laptop and the excellent facilities such as printing, to complete this thesis.

Finally, I am deeply grateful to my mother Yeshitla Esheti and all my sisters Emahoye W/S,

Misrak and Abebe and her families for their love and care over the years.

Dedication

This thesis is dedicated to my Mother with her love for her children and the late Ato Gashew Gebeyaw for his love and efforts to change my life. You are in my heart and in my mind for ever and also I would like to say thank you so much Zemedi. My Mom I never forget those harsh times you face to grow us.

ABSTRACT

The advent of modern computers play important roles in processing and managing electronic information that are found in the form of texts, images, audios and videos, etc. With the rapid development of computer technology, digital documents have become popular options for storage, accessing and transmission. With the need of current fast evolving digital libraries, an increasing amount of historical documents, newspaper, books, etc. are being digitized into an electronic format for easy archival and dissemination purposes. Optical Character Recognition (OCR) and Document Image Retrieval (DIR), as part of information retrieval paradigm, are the two means of accessing document images that received attention among the IR community. Amharic is the official language of Ethiopia since 19th century and as a result so many religious and government documents are written in Amharic. Huge collections of Amharic machine printed documents are found in almost every institution of the country. It is observed that accessing those documents has become more and more difficult. To address this problem, very few number of research works have been attempted recently by using OCR and DIR methods.

The aim of this research is to develop a system model that enables users to find relevant Amharic document images from a corpus of digitized documents in an easy, accurate, fast and efficient manner. So this work presents the architecture of Amharic DIR which allows users to search scanned Amharic documents without the need of OCR. The proposed model is designed after making detailed analysis of the specific nature of Amharic language. Amharic belongs to the Semitic languages and is morphologically rich language. Surface words formation involves prefixation, suffixation, infixation, circumfixation and reduplication.

In this work a model for searching Amharic document images is proposed and word image features are systematically extracted for automatically indexing, retrieving and ranking of document images stored in a database. A new approach that applies one of the NLP tools which is Amharic word generator is incorporated in the proposed system model. By providing a given Amharic root word to this Amharic specific surface word synthesizer, a number of possible surface words are produced. Then, the descriptions of these surface word images are used for indexing and searching purposes. On the other hand the system passes through various phases such as noise removal, binirization, text line and word boundary identification, word segmentation and resizing to normalize different font types, sizes and styles, feature extraction

and finally matching query word image against document word images. The proposed method was tested on different real world Amharic documents from different sources like magazines, textbooks and newspapers with various font styles, types and sizes. Precision-recall measures of evaluation had been conducted for sample queries on sample document images and promising results have been achieved.

List of Figures

List of Tables

List of Algorithms

List of Acronyms

ADIRS	Amharic Document Image Retrieval System
ADIR	Amharic Document Image Retrieval
CV	Consonant vowel
DBMS	Database Management System
DR	Document Retrieval
ID	Document Identification
IR	Information Retrieval
DIR	Document Image Retrieval
DTW	Dynamic time warping
IDF	Inverse Document Frequency
MT	Machine Translation
NCC	Normalized Cross-correlation
NLP	Natural Language Processing
OCR	Optical Character Recognition
POS	Part Of Speech
SE	Search Engine
SOV	Subject-Object-Verb
TF	Term Frequency
USA	United State of America
VSM	Vector Space Model
WWW	World Wide Web

CHAPTER ONE

INTRODUCTION

1.1 Background

Information access is one of the research topics of information society and it has become more important since the advent of the Web. The society relies on information both for professional and personal goals. Nowadays, it is considered as one of the most valuable and strategic goods: knowing the right information, at the right moment, as soon as its availability has become important [35].

The World Wide Web (WWW) may be seen as a huge collection of documents freely produced and published by a very large number of people, without any solid editorial control. This is probably the most democratic, lawless and widespread means for anyone to express feelings, comments, convictions and ideas, independently of ethnics, sex, religions or any other characteristics of human societies [51]. Information on the web can be found in different forms such as texts, audios, videos, images and so on, and they can be accessed using search engines. Information retrieval is concerned with the processes involved in the representation, storage, searching and finding of information that are relevant to human users [54]. The representation and organization of the information items should provide the users with easy access to the information in which they are interested. Information retrieval systems are used for management of unstructured and semi-structured representations of full-text documents and support similarity search within text documents. Indexing and retrieval are the two important components of an Information Retrieval System. The basic ideas behind document indexing and retrieval are:

- to provide the ability to characterize the document corpus in a meaningful way;
- to allow users to provide a query as a set of terms and;
- to provide mechanisms to retrieve the most relevant documents for that query.

Given a collection of documents, indexing describes documents using an index language [30]. Retrieval uses the results of indexing and finds related documents corresponding to a user's

query. There are two major categories of documents, machine editable and document images. Unlike in machine-editable, there are no explicit objects in document images, so processing of document images rely on document image analysis techniques to find the text and image objects [25].

The amount of visual information is increasing in an accelerating rate in many diverse application areas. Nowadays, digital document has become popular formats for storage and transmission instead of traditional paper documents. Document imaging is the process of converting paper documents into electronic documents that are exact replicas of their paper counterparts. In an attempt to move towards a more paperless office, large quantities of printed documents are digitized and stored as images in a database [4]. Each document image in a database has index that uniquely represent it. Scanning and storing documents as images in a database has advantages over storing hard copy. Since it is beneficial to maintain a copy of documents in image form the main issue then becomes the need for robust ways to access or index the information these document images contain.

If we are working everything on the paper, it accumulates quickly. Our files become larger and larger from time to time. Folders and filing systems make it somehow easier to find our documents. Record managers organize archive and retrieve our information. However, he/she couldn't manage well when the amount of paper keeps growing. Paper files are often hard to find and require wide space. Records may not be there in their proper folder or they may be borrowed and then lost on somebody's desk. Computerized document imaging offers a better way to manage the records we rely on. However, maintaining and accessing a database of document image is a challenging problem. For document images database, the incoming data is typically less structured so it is difficult to index and retrieve document images. One way of characterizing a document's content is to filter out common stop words that have a negligible effect on the content, and then represent the document by a term vector consisting of the frequencies of meaningful terms. Once document indexing has been constructed it provides easy and fast access. Accessing noisy document images or/and documents with different font type, size and style are anther two main challenges in scanned document image retrieval.

Organizations are currently using and are dependent on document image databases, especially if they use document images extensively. Modern technology has made it possible to produce, process, store and transmit document images efficiently. The mainstream now concentrates on how to provide highly reliable and efficient retrieval functionality over these digital document images produced and utilized in different services. It is difficult to retrieve user-relevant information from image data than from text data. Thus, the study of information retrieval in document image database is an important subject in knowledge and data engineering. To make billions of traditional and legacy documents available and accessible on the Internet, they are scanned and converted to digital images using digitization equipments. Document images retrieval is a task of searching relevant document images from the database based on user query images.

In business climate where organizations are seeking ways to cut costs and increase productivity, document imaging systems are providing the most dramatic impact on productivity since the copy machine replaced carbon paper. A number of approaches for document image retrieval have been proposed. They can be categorized in two main kinds of methods based on the feature they used. The methods are Optical Character Recognition (OCR) and Document Image Retrieval (DIR).

OCR is the process of converting text contained in a scanned image into text that is machine-editable. It recognizes characters in a scanned document and makes it possible for the user to edit. After a document image has been converted into text, text search can be executed using words and phrases known to be included within it. The OCR based techniques are suitable for applications where the documents are of good quality and involve relatively small lexicons [10]. However, it cannot give good performances on poor quality documents and complex nature of the scripts, high time consuming and the requirement of different OCR system to deal with different language characters [25, 39, 44].

DIR or recognition free approach avoids explicit recognition during indexing and is important for processing historical documents [63]. It aims at finding relevant document images from a corpus of digitized pages. The basic idea of document image retrieval is to find documents relying on document image features only [25]. DIR approach treats each word object as a single,

indivisible entity and attempts to recognize it using features of the word as a whole [39]. In this approach after preprocessing of the document image and word segmentation, feature vectors are extracted from word images and stored in a database. This approach has been exploited for several tasks such as word indexing and keyword spotting, retrieval of graphical items, retrieval of hand writing, layout retrieval [63].

Amharic is one of the Semitic language in the world and the working language of the Federal Democratic Republic of Ethiopia (a country with more than 70 million populations). It has been the working language of the government, the organization, non-organization and private institutions throughout modern times. Outside Ethiopia, Amharic is the language of 2.7 million emigrants (notably in Egypt, Israel, Sweden and USA) [32]. Thus, it is the official language and is spoken by many people as their native and second language. Even though the language is spoken by millions of people on different parts of the world and so many things are written by it since the past, the percentage of Amharic content on the web is less compared to English and some other languages.

A lot have been done on English and some other languages document image retrieval in the past, but very few researches have been attempted on Amharic document image retrieval [1], [40], [47]. Million [47] proposed a document image retrieval method employing information about word image features. Mesfin [40] proposed a document image retrieval system that can responds to user query without addressing the language features well. Abreham [1] describes a system with the ability of searching using single word query without taking into account noisy and degraded documents. Though searching need good indexing file, the techniques used so far to construct index file are not efficient as the nature of Amharic language had not been taken into account.

Therefore, the area of Amharic Document Image Retrieval needs further investigation to come up with practical system that searches efficiently. Having Amharic Document Image Retrieval System will have many advantages over traditional papers. This includes:

- to process large document collection quickly;

- to modify indexing easily ;
- to search document image fully and allow ranked retrieval ;
- for better management ;
- to share files easily;
- to reduce the damage and lose of information and;
- to save space and ;

1.2 Statements of the Problem and Justifications

In this paper, we present a system for searching document images in Amharic document images database based on using word image matching technique. In recent years, some researches on Amharic Document Image Retrieval have been attempted. However, previously proposed system;

- are relying on good quality document images;
- do not use the application of natural language processing tools to information retrieval;
- do not apply good indexing techniques;
- use only a single word for searching.

Our proposed system index document image by integrating Amharic word synthesizer, removing non-functional words from indexing and decreasing the size of index terms. Moreover, the system tries to handle font variations by resizing different fonts into a common form. On the query side it also removes stop-words to reduce query representation and apply rendering technique. On top of that, it enables users search by using phrase in addition to single word.

1.3 Objectives

1.3.1 General Objective

The general objective of this research work is to design and develop a generic model for Amharic Document Image Retrieval.

1.3.2 Specific Objectives

The specific objectives of this research work are to:

- study and analyze the language specific features of Amharic and explore natural language processing techniques for Amharic Document Image Retrieval;
- reviewing the current researches in document image retrieval and to select an efficient document image indexing and ranking techniques;
- design Amharic document image search engine model;
- explore distinct features which are employed to describe the Amharic word's image and word image matching;
- indexing and retrieval of Amharic document image without using OCR;
- develop a prototype to demonstrate the effectiveness of the proposed model.

1.4 Scopes and Limitations

In this research work, the researcher reviewed techniques for image preprocessing on document and query side, rendering query to image, automatic indexing and retrieval of document images. Even though documents can be written in different languages, scripts and available in different kinds of formats, this research work considers only machine printed Amharic document images with the assumption that skew free and on a database. Natural language application to information retrieval is applied in this research. Even though a number of Natural Language Processing(NLP) techniques namely stemming, part of speech tagging, words synthesizing, etc. that are used in IR system to improve the quality searching, in this work only stemming, stop

words removal and synthesizing are used in Amharic Document Image Retrieval (ADIR). On the other hand, word image is treated as a single entity and recognized as a unit without applying character segmentation. Due to less capacity of processing machine, it is not possible to show the effects of each and the combination of two or more word image feature values on searching document image.

One of the main contributions of this work is integrating Amharic word synthesizer to the proposed system architecture. This system can produce more than 1000 words for a given Amharic root word. However, this system is domain specific that is it only considers Amharic perfective verbs. Hence, the search space does not contain all possible words. So our system lexicon or word dictionary is restricted to our dataset.

1.5 Applications of the Result

DIR has critical applications in different sectors. Digital library is one of the important applications areas of DIR. On top of that it is applicable in any other private and governmental institutions which are more or less extensively depending on documents. Thus, by using DIR these institutions can move towards paperless offices and can manage well their files efficiently. Once Amharic Document Image Retrieval has been implemented well, it is important to increase the contents of Amharic on the web and to make different Amharic documents freely available for the users. Furthermore, historical Amharic documents can easily available to the users in an electronic format by processing, managing, accessing them using ADIR.

1.6 Methodologies

To achieve the objectives of this research, the researcher used different techniques that are described below.

1.6.1 Literature Review

Different literatures that are considered to be relevant for the research are reviewed and adopted for this work. Since this research work is on document image retrieval system, it touches a number of areas like natural language processing, rendering, image processing, word image feature extraction, word image matching, information retrieval, indexing and ranking. The

characteristics of Ethiopic character encodings and their phonetic representations are studied, and related works are reviewed to properly design and develop the proposed ADIRS. The existing document image search engines developed for Amharic and other languages are revised to understand the way in which searching and retrieving Amharic document images can be performed.

1.6.2 Data Collection and Preparation

To demonstrate the effectiveness of our proposed system the researcher collects Amharic document image and scan some other documents to prepare document image corpus for testing our system. The corpus encompasses all types of data in machine printed format. The collection contains documents that vary in font types, sizes and styles; and real-life documents such as books, magazines and newspapers with different image quality degraded/broken characters, noisy and clean documents.

On the other side, Amharic word synthesizer is used to produce a number of Amharic words for building Amharic words corpus. The outputs of word synthesizer are converted into images. Finally, features are extracted from these word images and stored in a database for the purpose of indexing and searching.

1.6.3 Implementation and Testing

After completing the development of the proposed prototype, an experiment is conducted to evaluate the performance and efficiency of the system. The work has been developed using three different programming languages namely Java, Visual Basic and MATLAB. Visual Basic is used for producing variant forms of a given Amharic root word to construct our words corpus. Java programming language is used to render the output of Amharic word synthesizer to word image. On top of that, Java is used to create graphical user interface for user query input and convert the query into image. MATLAB is used for document image processing and word recognition purposes. The recognition passes through several stages including image capturing or scanning, converting it to binary, segmentation, noise reduction, resizing each word in the document into each word in the corpus, feature extraction, applying matching algorithm, indexing, etc. Finally, the developed system is tested based on its performance in retrieving relevant documents from a

document images corpus using recalling technique and its ability in removing irrelevant items from searching results is evaluated using precision technique.

1.7 Organization of the Paper

Chapter one of this paper presents background of the paper, problems that initiate the researcher to do this work, major contributions of this research and the methodologies followed to address the problems. **Chapter Two** discusses the details of Amharic language features and Natural Language Processing techniques that are important to improve performance of IR systems. Amharic language script analysis and word formation are presented under this chapter. **Chapter Three** discusses background about Information Retrieval and Information Retrieval models. This chapter outlines document images processing operations such as scanning, preprocessing, various similarity measure approaches, etc. along with reviewing some related works. It also explains word image features that can be used for recognition purpose. **Chapter Four** shows the proposed system architecture and work flows including detail descriptions of algorithms. **Chapter Five** shows the implementation of the proposed system prototype, overall experimental setup, and analysis of our experimental results and shows the performance of the system. Finally, conclusions are drawn and some research directions for further research works are proposed in **Chapter six**.

CHAPTER TWO

AMHARIC LANGUAGE AND NATURAL LANGUAGE PROCESSING

2.1 Introduction

Ethiopia is located in Eastern Africa with a population of about 80 million [19]. The country is divided into nine fairly independent regions, each with its own nationality language. It is a linguistically diverse country with more than 80 languages for every day communication among people and there are more than 200 different dialects spoken [67]. However, Amharic is the language for country-wide communication and also served as a medium of instruction for education for a long period of time. As pointed out in [28] the Ethiopian languages are divided into four major language groups as Cushitic, Omotic, Nilo-Saharan and Semitic.

Language is one of the fundamental aspects of human behavior and it constitutes a crucial component of our lives. It can be used in the form of written or spoken. In its written form it serves as a means of recording information and knowledge on a long term-basis and transmitting what it records from one generation to the next. In its spoken form it serves as a means of coordinating our day-to-day life with others. In different parts of the world there are a number of languages.

Amharic is a Semitic language and is spoken as mother tongue in the Amhara National Regional State and also widely spoken in Ethiopia. It is the official and working language of the country. Amharic is the second most spoken Semitic language in the world (after Arabic) [14] and related to Hebrew, Arabic, and Syrian in some aspect. It has been the working language of government, the military and of the Ethiopian Orthodox Church throughout modern times. Accordingly, there are a number of printed documents (such as letters, books, newspapers, and magazines) available in government and private offices, libraries and museums. Amharic has its own script known as FIDEL that is borrowed from Geez, which is another Ethiopian Semitic language [53]. The language has 34 basic characters and each character has 7 forms for each consonant-vowel combination, and extra characters that are consonant-vowel-vowel combinations for some of the basic consonants and vowels [15]. It also has a unique set of

punctuation marks and digits. The language is written from left to right and from top to bottom. Manuscripts in Amharic are known from the 14th century and the language has been used as a general medium for literature, journalism, education, national business and cross-communication [15]. Amharic has a complex morphology. As [58] stated Amharic word formation involves prefixation, suffixation, infixation, reduplication and Semitic stem inter digitations, among others. Like any other Semitic languages Amharic verbs and their derivation constitute a significant part of the lexicon. In Semitic languages, words, especially verbs, are best viewed as consisting of discontinuous morphemes that are combined in a non-concatenative manner. According to [15] an Amharic verb could have over 150 different forms. Syntactically, Amharic is a Subject-Object-Verb (SOV) language. The concepts of upper-case and lower-case letters are absent in Amharic writing system.

2.2 Amharic Language

2.2.1 Amharic Writing System

The current Writing system in Amharic uses a unique script which has originated from the Ge'ez alphabet. In now a day's Ethiopic script, each syllable pattern comes in seven different forms or orders. These orders reflect the seven vowel sounds. The first order is the basic form and the rests are derived from it by regular modifications indicating the different vowels. Currently, there are 34 basic characters and each of which occurs in seven base orders [72] (see Appendix I for detail understanding). Hence, a total of 7*34 distinct FIDELS exist in the language. Two of the base forms represent vowels in isolation, but the rests are for consonants and thus correspond to consonant vowels (CV) pairs, with the first order being the base symbol with no explicit vowel indicator. In addition to these the writing system has 40 other symbols which contain special features usually representing labialization [15]. Atelach [15] also states these characters have consonant-vowel-vowel combinations for some of the basic consonants and vowels. e.g §ª, à ,úª, …, šª, Ì, Æª, –, —, etc.

According to Marvin et al. cited in [67], Ethiopia has three writing systems. These are the Amharic syllabary, the Roman alphabet and Arabic script. Geez, which is the originates of Amharic, follows rule in denoting vowels by a variety of changes in the structure of the

consonantal symbol. Vowels have thus become an integral part of Amharic writing which now assumed the character of a syllabary. The Amharic writing system uses multitudes of ways to denote compound words and there is no agreed upon spelling standard for compounds [53].

2.2.2 Problems in Amharic Spelling

Amharic orthography reflects the spoken phonetic features to a large extent. So that one can be leaded to believe that there is no notion of "spelling" in Amharic. The rule to Amharic language users generally follow in their writing system is that if the alphabet in a word sounds right when read aloud, then it is written. Due to this speakers of the language from different regions of the country write the same word differently. The absence of restricted rules for this lead to a number of problems to develop efficient Amharic information retrieval system. Some of the problems of Amharic writing systems are:

One of the problem is the complexities in Amharic spelling is the presence of Ge'ez loan words and words derived from a Geez root [28]. Daniel in this reference also says Geez is the ancient language of Ethiopia that is analogous in the role that Latin played for the Romance language of Europe. Geez had a richer phonemic inventory and required additional letters for its orthography. In Amharic orthography these additional letters from Geez would take on the phonemic value of its nearest neighbor. The result being two syllabic series for's' ('s' and '¿'), two series for 'ts' ('™' and '{'), two for 'a' ('x' and ';') and 3 for 'h' ('h', '¼', and 'ኀ'). These redundancies in Amharic become a source of confusion and the letters are treated as interchangeable by a person. Common Amharic spelling then becomes highly flexible and correctness is not a matter of precision but one of acceptable proximity. For example the word "sun" can be written as "{hY", "{¦Y", "™hY", "™¦Y", etc. all mean the same, although they are written differently. The fourth month of Ethiopian year, which is December, may have forms: "¬HúS" , "¬ÞúS" , "¬Hœ o " ,"¬ኀœ o " ,etc.

The second problem is formation of compound words [67]. These words can be written as two separate words or as a single word. For instance, the word "kitchen" can be written as "¥D b¤T" or "¥D b¤T".

The third problem in Amharic language is that it is common to write some words in shorter form by using '/' (forward slash) or '.' (dot). The short form of words which are linked by either of these symbols can be expanded as single or a combination of words. For instance, x.x is expanded as combination of two words xÄ‹S xbÆ (means Addis Ababa). m/R is a short form of the single word m mHR (means teacher).

The fourth problem is different ways of writing a single word due to various reasons such as regional dialects and various ways of writing loan words [67]. Regional dialects have their own impact in word formation in the basic level where the words are more likely to be written by following their spoken form. For example, "S‰" vs " o ‰", "ካÉ" vs "ካ «", "Xnz‹H" vs "XnŸH", etc.

On top of the above problems, in Amharic loan (foreign) words can be written in different ways [67]. For example, the word computer can be written as "÷M p†t R", "÷M pWt R", "÷M pý t R" , "÷M pt R", etc.

2.2.3 Nature of Amharic Langue Script

In various time improvements have been made with respect to representation and manipulation of Amharic scripts using computer. Several fonts and encoding schemes have been proposed for scripts in the past, which results in production of non standard software package. Now a day, there are Unicode compatible fonts and companies updating their packages in this standard [28].

Amharic scripts formation has certain notable features. The script is a syllabary writing system where each character represents an open CV syllable [58]. The scripts are more or less orthographic representation of the phonemes in the language. The orthographic representation of the language is organized into orders. Each of the 34 consonants has seven orders. Six of them are CV combinations while the seventh is the consonant itself. Moreover, there are extra orthographic symbols in the language that are not organized as above. For each consonant C, the orthographic ordering is as follows C/e/ C/u/ C/i/ C/a/ C/ie/ C C/o/ [61].

2.2.4 Amharic Language Grammar

Amharic has different word units. These are phoneme, morpheme, root, stem and word [67]. The base characters are phonemes and a collection of phonemes form morphemes. The root is a sequence of three base characters and the collection of phonemes form word. The language has different word classes and word formation rules. These are described as follows:

Personal Pronouns: In some languages, there are a small number of basic distinctions of person, number and often gender that play important roles within the grammar of the language. In Amharic, the same distinctions appear within the grammar of the languages. For instance, English "I", Amharic "Xn", English "she", Amharic "Xú", English "we", Amharic "X¾", etc.

Subject-Verb Agreement: Amharic verbs agree with their subjects; that is, the person, number, and gender. The subjects of the verb are marked by suffix or prefix on the verb. Because the affixes that signal subject agreement vary greatly with the particular verb tens/aspect/mode.

Object Pronoun Suffixes: Amharic verbs often have additional morphology that indicates the person, number, and (2^{nd} and 3^{rd} person singular) gender of the object of the verb. For instance, xL¥ZN xy°T, which means "I saw Almaz".

While morphemes such as "-at"("-xT") in this example are sometimes described as signaling object agreement, analogous to subject agreement, they are more often thought of as object pronoun suffixes because, unlike the markers of subject agreement, they do not vary significantly with the tense/aspect/mood of the verb. For arguments of the verb other than the subject or the object, there are two separate sets of related suffixes, one with a benefactive meaning ('to', 'for'), the other with an adversative or locative meaning ('against', 'to the detriment of', 'on', 'at').

Possessive Suffixes: Amharic has a set of morphemes which are suffixed to nouns and signaling possession. As [64] notes Possessive expressions in Amharic are rendered by using possessive suffix pronouns attached to a head noun or prefixing ***l'*** to the personal pronouns or nouns indicating the possessor. For example, b¤T "house", b¤tI "my house", b¤a "her house".

In each of the above four aspects of the grammar, independent pronouns, subject-verb agreement, object pronoun suffixes, and possessive suffixes, Amharic distinguishes eight combinations of person, number, and gender. For first person, there is a two-way distinction between singular ('I') and plural ("we"), whereas for second and third persons, there is a distinction between singular and plural and within the singular a further distinction between masculine and feminine ("you masculine singular", "you female singular", "you plural", "he", "she", "they").

Reflexive Pronouns: For reflective pronoun ("myself", "yourself", etc.), Amharic adds the possessive suffixes to the noun ‰S: ‰s¤ "myself", ‰ú‡ "herself", ‰sì "himself", ‰úcW "themselves", etc.

Demonstrative Pronouns: Like English, Amharic makes a two-way distinction between near (this, these) and far (that, those) demonstrative expressions (pronouns, adjectives, adverbs). Besides number, as in English, Amharic also distinguishes masculine and feminine gender in the singular.

Nouns: Amharic noun can be primary or derived. A noun like XGR "foot, leg" is primary, and a noun like XGr¾ "pedestrian" is a derived noun [32].

2.2.5 Amharic Morphology

Like many other Semitic languages, Amharic has a rich verb morphology which is based on tri consonantal roots with vowel variants describing modifications to, or supplementary detail and variants of the root form [53]. Amharic verbs exhibit nonlinear words formation with combination of consonantal roots with vocalic patterns. This also applies to non verbal nouns and adjectives.

Verbs

Verbs are morphologically the most complex part Of Speech (POS) in Amharic, with many inflectional forms; numerous words with other POS are derived primarily from verbs [60]. In some Amharic research papers and literatures it is indicated that verbs are the most productive classes of words in the language. Subject, gender, number, objects and possession markers,

mood and tense, beneficative, malfactive, transitive, dative, negative, etc. are indicated as bound morphemes on the verb and producing a complex verb morphology [15]. As [50] mentions the number of consonants in the root varies from two to six, with three-consonant structures being the most common. There are three levels of textual representation that can be used: the original words; the stems obtained by application of stemmer to those words; and the roots obtained by elimination of the vowels from those stems. Simple verbs have five verbal stems that are formed by intercalation of vowels with skeleton patterns of the types CVCVC, CVCC etc. These stems are: Perfective, Contingent, Jussive, Gerundive and Infinitive [34].

In Amharic, non-concatenative operations of vocalic intercalation, reduplication accompanied by vowel insertion and radical reduction are the main components of word formation processes [59]. Verbs are created mostly from tri radical consonantal roots which are inflected by a process of merging with vocalic components based on various templates. The verbal stems generated from the roots do not exist as free form. Instead, they undergo further inflections via the conventional concatenation of affixes (prefixes, suffixes, circumfixes).

Simple verb stems are formed from consonantal roots with the infixation of vocalization. As [33] cited in [60] the vocalic elements often mark a grammatical category, either aspect (perfective or imperfective) or mood (imperative-jussive).

Table 2.1: *Simple verb stems of the root word "SBR"*

Stem type	**stem**
Perfective	sbR
imperfective	sBR
imperative	SbR/Sb¶
Gerund	sBé/sB‰
infinitive	mSbR

Subject, gender, number, etc. are also indicated as bound morphemes on the verb, as well as objects and possession markers, mood and tense, beneficative, malfactive, transitive, dative, negative, etc, producing a complex verb morphology [60].

As [59] states verbs in Amharic can be divided into various types or classes based on the pattern of gemination, the number of radicals (of the root), and the quality of vowels inserted between the radicals.

Nouns

Amharic nouns can be simple or derived from verb root, adjectives or other nouns [60]. For example, bTT "house"; mÊT "earth"; XúT "fire" , etc. are simple nouns whereas _Ãq, "question" from _YQ "to ask"; ¿‰t¾ "worker" from oR "root" ,etc. are derived nouns. Derived nouns are derived from verb roots by intercalating different vowels between the radicals, by adding suffixes to the root without vowel intercalation, or by consonant reduction [60]. Affixation is the major process to derive nouns from adjectives and other nouns. For instance the plural form of mZgB is m²GBT/mZgïC where as the plural form of xNbú is xÂBST/xNbîC, the plural form of qS is qúWST and so on. Most Amharic plural nouns are derived by adding a plural marker affix "-åC" or "- wC" to a singular noun form. Amharic nouns and adjectives can be inflected for gender, number, definiteness, and case, although gender is usually neutral [19]. Bjorn [19] also notes that the definite article attaches to the end of a noun, as do conjunctions, while prepositions are mostly prefixed.

Pronouns

Amharic pronouns can be free or bound to other POS. In the accusative and genitive, free personal pronouns take the affixes for nouns [32].

Adjectives

Amharic adjectives can be simplex or derived from nouns, verbal morphemes [32]. qY which is "red", dG which is " generous", etc. are simplex adjectives where as BRt "strong", from BRT which is "be strong" hYl¾ "force full", from hYL "force, energy" are derived adjectives. Like nouns, adjectives are inflected for Number, Case, Gender and Definiteness.

Adverbs

In Amharic there are few number of adverbs. Some of them are derived from adjectives by adding suffix. Adverbial functions are often accomplished with noun phrases, prepositional phrases and subordinate clauses [32].

Prepositions and Conjunctions

Conjunctions and prepositions have similar behaviors, and are often placed in the same class

mSt ሞጽጽ: no affixation, not used as base for derivations, syncategorematic and only occurring with other words [32]. Prepositions are mostly bound morphemes prefixed to nouns. The definite article in Amharic is also a bound morpheme, and attaches to the end of a noun.

2.2.6 Amharic Affixes

Like any other natural languages in Amharic there are a number of prefixes that are coming at the beginning of the main word. These words may or may not change the meaning of the main word. These include §, Sl, x, XS, XNÄ, xN, xS, etc. are causative markers or morphemes.

On the other hand, a suffix is an affix that is attached after a base. The plural markers åC / wC and xT, xC, x, xu, xuN, etc. are examples of suffix. A circumfix is the combination of prefix and suffix that together express some feature. In Amharic the combination of the prefixes T, xl>, xlx, xChu, etc. are example of circufix. An infix is an affix where the placement is defined in terms of some phonological condition(s).

2.2.7 Compound Words in Amharic

Compound words are more similar to affix. However, the main difference between them as [68] cited in [34] state relies in the nature of morphs they combine. Affixation attaches a bound morph onto a free morph whereas compounding combine two freely standing morph to form another word forms. Amharic has compound verbs, nouns and adjectives. Compound verbs are formed by combining the words ሰጠ or ሰደገ with meaningless morphemes such as q_ [60]:q_ xl, q_ xdrg. Saba and Dafydd [60] also state that compound nouns are formed by concatenating two nouns or a noun and an adjective with the linking vowel –*e*–; compound

adjectives are also formed joining noun and adjective. For example, ቤተ ክርስቲያን is compound noun where as XGr qu N is compound adjective. There is no agreed upon spelling standard for compound words and the writing system uses several ways to denote compounds [12]. ¥D b¤T which means "kitchen", b¤t KRSt/ÃN which means "church", TMHRT b¤T which means "school" b¤t XS‰x¤L, wZ xdR, mS¶Ã b¤T, etc. are example of Amharic compound words.

2.2.8 Amharic Punctuations

The writing system of Amharic has a number of punctuation symbols. However, now a day due to different reasons only some punctuations are being used mostly. Each punctuation symbol has specific function. For instance, hulet neteb (:) is used to separate one word from the next like space, arat neteb (::) is also used as end mark of a sentence like that of dot in English, netela serez (፣) is used to list, dereb serez (\) is used to separate one phrase to other phrase, exclamation (!) to give more emphasis, etc.

2.2.9 Amharic Numeric Digits

In addition to alphabets, Amharic language has its own numeric system. It has ten base numeric digits from one to ten and the rest of numbers are formed with a combination of any of these base digits. As [18] cited in [1] the basic digits are derived from Greek letters and some are modified to look like Amharic FEDEL. One of the main difference between Amharic numeric system and universal numeric system is there is no zero symbol in Amharic i.e. counting start from one.

2.3 Natural Language Processing

Natural language processing (NLP) is a subfield of artificial intelligence and linguistics [66]. The idea of natural language processing is to design and build a computer system that will analyze, understand and generate natural human-languages. It studies the problems of automated generation and understanding of natural human languages. Natural language generation systems convert information from computer databases into normal-sounding human language, and natural language understanding systems convert samples of human language into

more formal representations that are easier for computer programs to manipulate [66]. NLP techniques are based on the fact that the content of a document or a query is encoded in natural language. Various steps are involved in natural language processing. To mention some:

2.3.1 Morphological Synthesis

Morphological synthesizers have vital role in NLP systems. They are used to generate surface word forms, which are the ones that are found in everyday communication, from lexical components that could be stored separately in different databases (lexicons) [20].

2.3.2 Morphological Analysis

Morphological analysis deals with componential nature of words, which are composed of morphemes – the smallest units of meaning [41]. Individual words are analyzed into their components, and non word tokens (such as punctuation) are separated from the words. For example, the word preregistration can be morphologically analyzed into three separate morphemes: the prefix “pre”, the root “registra”, and the suffix “tion”. Given a word form, a morphologic analyzer, unlike a stemmer, which only produces a more or less linguistically motivated stem, should return its base form, properties of the base, such as the word class, gender, etc., as well as information on the specific form, such as case, number, tense, mood, or, in the case of a compound, its constituent words [43].

2.3.3 Syntactic Analysis

This level focuses on analyzing the words in a sentence so as to reveal the grammatical structure of the sentence. Linear sequences of words are transformed into structures that show how the words are related to one another. This parsing step converts the flat list of words of the sentence into a structure that defines the units represented by that list [66].

2.3.4 Semantic Analysis

Semantic processing determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence. This includes the semantic disambiguation of words with multiple senses, syntactic disambiguation of words, etc. [41].

2.3.5 Discourse Integration

This NLP activity works on units of text longer than a sentence. That is, it does not interpret multi sentence texts as just concatenated sentences, each of which can be interpreted separately [41].

2.3.6 Pragmatic Analysis

It concerned with the purposeful use of language in situations and utilizes context over and above the contents of the text for understanding [41]. The goal is to explain how extra meaning is read into texts without actually being encoded in them. Inflection is a common phenomenon in natural languages. Its nature and complexity may vary a lot from language to language. Morphological variation of words can also be handled with different kinds of means in different applications of language technology. The morphological variation of keywords in information retrieval (IR) has been solved by stemmers and lemmatizes [17].

2.3.7 Lexical Analysis

Lexical analysis is natural language processing task. It starts with tokenization which is the identification of all the individual words that constitute the input text or document. That is, given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens. Punctuation marks and spaces are usually used to infer the beginning and the end of a token. In addition to tokenization, lexical analysis also incorporates a sort of text cleaning process. It eliminates punctuations, accents, extra spacing, converting into low/upper case [51]. The text cleaning process removes numbers and symbols such as numbers, $, @, %, #,*, etc. which are not important for index terms construction. It also converts abbreviations and acronyms into their full text, and merges hyphenated words. For instance, full-text is treated as a single word “full text” and the abbreviation ዓ.M which is “year” in Amharic is expanded as ዓመተ ምህረት.

CHAPTER THREE

DOCUMENT RETRIEVAL

3.1 Information Retrieval

Information Retrieval (IR) is the field of study that examines how people find information and how tools (such as search engines) can be constructed to help people to access information. It is a means of finding documents of an unstructured nature that satisfies an information need from within large collections (usually stored on computers) [14]. It aims at identifying documents from a large collection which are relevant with respect to some query. Two major tasks for IR are text retrieval and document retrieval [22]. IR has so many applications in different areas such as web, bibliographic system, and digital libraries and so on.

IR system responds to the user's query by selecting documents from a database and ranking them in terms of relevance. A successful IR system is able to filter out extraneous information and return only relevant documents. To do so they are required to handle natural language both in the documents and in the user queries. Conceptually, an IR system is similar to a traditional library: there is a collection of documents and an access method. The simplest access method is linear searching, i.e. one document after another is scanned to see if it matches with the query [43]. The assumption upon which the whole field of IR based is that if the query and the document have a keyword in common, then the document is about the query to some extent [16]. If there are more keywords in common, then the document is more about the query. In that respect, the IR problem is represented by matching the bag of keywords in the users' query with the bag of keywords representing the document. According to [16] this approach suffers from a number of problems which originate from the following linguistic variations:

1. It does not handle cases where different words are used to represent the same meaning or concept in queries and documents. For example, the keyword "film" does not retrieve documents which contain its synonym "movie".
2. It does not distinguish cases where single words have multiple meanings due to semantically variation. For instance, a singer looking for "bands" will be faced with "radio frequency bands".

3. It does not sufficiently deal with the problem of syntactical variation. A document saying "near to the river, air pollution is a major problem" is not about "river pollution".
4. It makes things worse due to morphological variants. Key words appear in different numbers. For instance "wolves" and "wolf" or different cases like "man" and "man's".

Document retrieval (DR) systems focus on the problem of retrieving documents relevant to a user's information need represented as a query. Traditional DR models usually represent documents and queries as a set of keywords. The use of NLP in DR attempts to overcome such shortcomings [8, 9]. NLP techniques are based on the fact that the content of a document or a query is encoded in natural language. They aim to extract accurate linguistic structures that are then used to represent documents and queries, where linguistic structures can vary from noun-phrases to tagged sentences.

Computational linguistic is the study and development of computer systems for performing automatic natural language processing. Modern IR systems use a range of statistical and linguistic tools to maximize the effectiveness of searching textual documents [50]. It is generally believed that applications such as information retrieval, text classification, or document filtering could benefit from the existence and availability of basic NLP tools such as stemmers, morphological analyzers , part-of-speech taggers and so on [53]. Many of these NLP techniques are applied on the IR dataset and on the query. In information retrieval NLP is used to eliminate irrelevant words, to produce various forms of a single word, to produce common form of various forms of related words, to disambiguate words based on their part of speech and so on [17]. The content based manipulation operation such as indexing and retrieval, categorization, classification, filtering, and so on are beneficiary from NLP techniques and tools [7].

3.1.1 Basic Language problems in Information Retrieval

Graphical and Orthographic Variants: Some Amharic words such as Wh,¥HbR, ÷MpWtR, etc. can be spelt in various ways. Since a query may contain a word with a first spelling and documents may contain the same word but with another spelling, this phenomenon can lead to a decrease of recall [42].

Part –Of- Speech: The use of a stop list in order to eliminate empty words (that is words that have little value in reflecting content) may be hazardous because some of those words are ambiguous with content words (for instance "A" with the musical note or in vitamin A) [42]. Part of Speech (POS) tagging is consequently necessary in this case. POS is also needed in order to get the lemma of a word.

Morphology: One of the first problems related to the use of natural language in information retrieval is that of morphologic variation (for instance if there is a singular in the query and a plural in the documents). This refers to the fact that words may occur in inflected forms, or that derivation is used to produce new but related words, or that words are combined into compounds. Morphologic variations can very often be regarded as semantically related and thus equivalent for retrieval purposes [43]. In English, the number of possible inflected and derived forms is relatively small, and variation is mostly restricted to the attachment of suffixes. Compounds which are not yet lexicalized are in most cases written as separate words. Stemming, i.e., the removal of suffixes using a list of possible suffixes, is therefore considered a practical way to map related word forms to a common stem [43].

A large number of retrieval systems use a very simple stemming function like Porter's one [55]. This is the economical way to manage morphological variations to improve the results. The advantage of a simple stemming procedure like Porter's one is that both inflectional and derivational morphology are taken into account [42]. A simple stemming procedure gives the same stem for various forms of the same word while lemmatization keeps them separate. [42] states that the advantage of lemmatization is that it is more accurate. Two type of morphological normalization:

a. Inflected word forms are reduced to their root forms as specified in the dictionary.
b. Normalize verb forms are converted to the root form of corresponding verbs (e. g" implementation" was converted to "implement").

3.1.2 Query

A user query is processed in a similar way to extract query terms which are then matched against the indexing terms. In response to the query the system returns a set of references to

documents in the document collection which are considered relevant to the query. Query length has an effect on the impact of natural language processing. The impact of NLP is larger for longer queries where as short queries lack a lot of the context information that is used in NLP [9]. A user's query is often considered as a pseudo document and is represented as a vector in the reduced term-by-document space [22]. First, the terms in the query are represented by an (m X 1) vector 'q' whose elements are either zero or the frequency of the terms that exist in the database of the reduced vector space. The query vector is represented by the weighted sum of its constituent term vectors. The query vector can then be compared to all existing document vectors and the documents are ranked according to their similarity (nearness) to the query.

3.1.3 Natural Language Processing in Information Retrieval

Morphological information's are discussed under this section as follows are used for indexing, query processing, searching and document ranking [17]. Due to this the input query and the documents are undergone linguistic processing to take advantage of the information provide. Natural language processing has its own application in information retrieval in order to improve the effectiveness and efficiency of information retrieval systems. Some of its importance are listed as follows.

Stop Words Removal: Sometimes, some extremely common words that would appear to be of little value in selecting documents that match with user's need are excluded from the vocabulary entirely. These words are called stop words. Stop words are too frequent to be of any use, e.g. determiners, prepositions, articles and other function words. The general strategy for determining a stop list is to sort the terms by collection frequency (the total number of times each term appears in the document collection), and then to take the most frequent terms, often hand-filtered for their semantic content relative to the domain of the documents being indexed, as a stop list, the members of which are then discarded during indexing [22]. IR applications remove stop words (function words, low-content words, very high frequency words) before processing documents and queries. This usually increases system performance. After removing stop words the remaining word forms are mapped to a common form. Finally, the index terms are weighted according to their frequency and stored in the inverted file.

Stemming: For grammatical reasons, documents are going to use different forms of a word, such as organize, organizes, and organizing. On top of that, there are families of derivationally related words with similar meanings, such as democracy, democratic, and democratization. In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set [22]. Stemming is a NLP technique which is frequently used in IR with the assumption that morphological variants represent similar meaning. It maps several words into one base form that can be used as a term in vector space model. This means that it increases similarities between documents and queries because they have an additional common terms after stemming, but not before. It is a language dependent and should be tailored for each language since languages have a varying degree of differences in their morphological properties. It is applied during indexing and is used to reduce the vocabulary size, and it is used during query processing in order to ensure similar representation as that of the document collection.

One of the stemming methods is suffix stripping. It uses a list of frequent inflectional and derivational suffixes which will be cut off from word forms to produce their stems [43]. The two main Suffix stripping methods are:

(1) Linguistic/dictionary-based stemming: In this method suffixes are removed by dictionary looking. It has higher stemming accuracy, but also higher implementation and processing costs and lower coverage [55].
(2) Porter-style stemming: is a standard tool which achieves normalization by simply chopping off suffixes [55]. It has lower accuracy, but also lower implementation and processing costs and is usually sufficient for IR.

Part-of-Speech Tagging: Part-of-speech tagging is the task of assigning a syntactic category to each word in a text, thereby resolving some ambiguities [27].

Chunking and Shallow Parsing: Chunking and Shallow Parsing aim at separating words in a sentence into basic phrases, e.g. noun phrases or simple verb phrases.

Word Sense Disambiguation: When different parts of language such as word, term, notation, sign, symbol, phrase, sentence, or any other forms used for communication be ambiguous if it

can be interpreted in more than one way. Word sense disambiguation is the task of distinguishing the correct sense or meaning of a word in context [8]. When used for information retrieval, terms are replaced by their senses in the document vector.

On top of that Compounds and Statistical Phrases, Compound Splitting and Head-Modifier Pairs NLP tasks for information processing [74].

3.1.4 Document Indexing

One of the critical aspects of information system that determines its effectiveness is how it represents concepts in items. The transformation from the received item to the searchable data structure is called indexing. A document index is a set of terms which show the contents of the document and helps in distinguishing a given document from other documents in the collection of documents. Indexing is to build a data structure that will allow quick searching of the text and it is the core of every modern information retrieval system [73]. An index of an information retrieval system allows finding the documents matching a particular query without having to look at the documents themselves. The process of creating index can be done manually or automatically.

Web document indexing is an important part of every Search Engine (SE) and its quality has an overwhelming effect on retrieval effectiveness [21]. As [21] points out small index size can lead to poor results and may miss some relevant items. Large index size allows retrieval of many useful documents along with a significant number of irrelevant ones and decreases the search speed and effectiveness of the searched item.

In IR, a document is represented in terms of a set of index terms which are often stem words [36]. Such terms are obtained by text processing and used to represent a document or a request expressing a user's information need. The terms supplied for documents may be formed at file time, when a document enters the data base, or at search time, stimulated by a user's request. The terms and descriptions constructed from user requests, at search time, form one or more queries. The Indexer is one part of search engine that can process the documents and stores

them in a structure that is efficient for searching [25]. As [31] states some of the processing steps in this component are: tokenization, stop word removal, stemming, etc.

Indexing the content of the document is done through three phases described as: keyword extraction, capturing the document linguistic structure and capturing the role of the selected keyword in the sentence where it occurs. Based on the analysis of the documents, several indices are built up:

1. using the information about the lexical unit (the normalized form),
2. using the derivational information and
3. decomposition of information [21].

Feature selection is the process of term selection for indexing in text retrieval system. The frequency of word occurrences in an article furnishes a useful measurement of word significance [43]. According to [21] the feature selection algorithms usually perform three steps to select the term for indexing:

- ✓ apply a stemming and stop-word algorithm, i.e. extracting the root of each word and removing the common words (terms) (example and, or, a, the, etc.) from the text of the document;
- ✓ compute the term frequency for all remaining terms in each document and
- ✓ select *N* terms with high term frequency from each document as index vector.

An index of an information retrieval system allows finding the documents matching a particular query without having to look at the documents themselves [43]. This speeds up the search considerably. However, an index has to be built before it can be used. Some of the common index types or structures are inverted file, suffix tree and signature file [73]. The description of each is presented as follow in different literatures.

Inverted File

In order to perform efficient retrieval of document in a database all the stored documents need to be properly indexed using appropriate indexing method. One of the indexing methods that is word oriented and most now a day operational IR systems are based on is the inverted list data structure [9]. This enables fast access to a list of documents that contain a term along with other information (for example, the weight of the term in each document, document name). According to [10] inverted files work as follows;

I. Each document in the collection is assigned a list of attributes which are supposed to represent the document. The most common type of attributes is keywords.

II. The inverted file is then the sorted list of keywords of all documents, where each keyword has links to the documents that contain that keyword. The modern approach is to use the full text of the documents as indexing terms (hence the term full-text retrieval).

III. After the removal of stop words, the remaining word forms are normally conflated, i.e. semantically related word forms are mapped to a common form (the actual index term).

IV. Finally, the index terms are weighted according to their frequency and stored in the inverted file.

Term conflation is usually done for two reasons [43]. First, different morphological forms of a word should be mapped to a common form. The assumption is that the word forms are semantically related. The second reason is that the conflation of terms reduces the size of the index. Suffix stripping uses a list of frequent inflectional and derivational suffixes which will be cut off from word forms to produce their stems. The two most frequently used stemming algorithms in IR are the Lovins stemmer and the Porter stemmer [71, 50].

Suffix Tree

Suffix tree is another string searching technique to index a string of length which has a natural partitioning into m multi character substrings or words. It stores all unique substrings of the text and represents only the m suffixes at word boundaries [13].The boundaries are determined by delimiters. Suffix tree store references to all sistrings of the text in a trie to accelerate string

searching. A sistring (semi-infinite string) is a substring of the text, defined by its starting position and continuing to the right as far as necessary to make the string unique [12]. As Andersen and Nilsson [12] state this data structure has a wide range of applications for approximate string matching, compression schemes and genetic sequences.

3.1.5 Term Weighting

One of the critical pieces of information needed for document ranking in IR models is a term's weight in a document. Various methods for weighting terms have been developed so far. Weighting methods developed for the probabilistic models rely heavily upon better estimation of various probabilities where as weighting methods developed for the vector space model are often based on researchers' experience with systems and large scale experimentation [9]. In any models, there are three significant factors that determine the final term weight formulation. These are:

a. Term frequency (tf): Words that repeat multiple times in a document are considered relevant.
b. Document frequency: Words that appear in many documents are considered common and are not very indicative of document content. A weighting method based on this, called inverse document frequency (or idf) weighting.
c. Document length: when the collections have documents of varying lengths, longer documents tends to score higher since they contain more words and word repetitions. This effect is usually compensated by normalizing for document lengths in the term weighting method [9].

Term weight or frequency is the number of occurrence of term in a document and can be denoted as tft, d, with the subscripts denoting the term and the document in order.

Document frequency (dft) is the number of documents in the collection that contain a term t. Inverse document frequency (idft) of a term i is defined as follow:

$$\text{idft}=\log \frac{ND}{NDi} \quad \ldots\ldots\ldots\ldots\textit{Equation 3.1}$$

Where ND is total number of document in the database.

NDi is number of documents in which term i is occurs.

Thus according to equation 3.1 idft of a rare term is high, whereas the idft of a frequent term is likely to be low.

Tf-idf weighting is a combination of term frequency and inverse document frequency and is used to produce a composite weight for each term in each document [75].

Tf-idf= tft,d ×idft……..…….*Equation 3.2*

Scoring determines whether or not a query is present in a document. If a document mention a query term more often, then it receive higher score. The score between query t and document d is based on the weight of t in d.

3.1.6 Ranking Search Results

Once the searchable data structure has been created, techniques must be defined that correlate the user-entered query statement to the set of items in the database to determine the items to be returned to the user. This process is called searching. The core task of IR system is to process queries against the data that is indexed. Searching in general is concerned with calculating the similarity between a user's search statement and the items in the database. Now a day IR systems have logically stored weighted values for the indexes to an item. Once items are identified relevant to the user's query, it is best to present the most likely relevant items first. This process is called ranking. Usually the rank of retrieved document is computed by the TF*IDF [17].

3.1.7 Documents Database

Documents database which contain all items that have been received, processed and stored by the system. It is the search source for the system. Each query is processed against the total document database. Database management systems can give very precise answers to detailed queries but cannot provide information on the basis of queries that are only vaguely worded. A

DBMS requires precisely stated queries, and the main issue is therefore how to efficiently process the query and provide the requested data.

3.1.8 Information Retrieval Models

Boolean Model

The Boolean retrieval model is a model for information retrieval in which logical operations are involved. It can pose any query which is in the form of a Boolean expression of terms that is terms are combined with the operators and, or, and not [22]. These operations are implemented by using set intersection, set union, and set difference procedures, respectively. This model views each document as just a set of words. Even though Boolean systems allow to formulate very precise queries, they have several shortcomings, e.g., there is no inherent notion of document ranking and it is very hard for a user to form a good search request when the query become more complex [20]. However, according to [20] in Boolean system information are retrieved in either a sorted order (e.g., sort by Title or in time order from the newest to the oldest item). In Boolean model a document either matches or does not match a query.

Vector Space Model (VSM)

The representation of a set of documents as vectors in a common vector space is known as the vector space model and is fundamental to a host of information retrieval (IR) operations including scoring documents on a query, document classification, and document clustering [15]. It is a mathematical model that is used often in IR systems and attempts to determine how similar retrieved documents are to the user's query by constructing an N dimensional token space, where N is the number of tokens in a query.

In VSM users largely use free text queries, that is, just typing one or more words rather than using a precise language with operators for building up query expressions, and the system decides which documents best satisfy the query [22]. In the vector space model text is represented by a vector of terms. It implements full text automatic indexing and relevance ranking. In VSM, documents and queries are modeled as elements of a vector space. This vector space is generated by a set of basis vectors that correspond to the index terms. To assign a

numeric score to a document for a query, the model measures the similarity between the query vector (since query is also just text and can be converted into a vector) and the document vector [9]. Document vector captures the relative importance of the terms in a document where as query vector capture importance of terms in a query statement. Typically, the angle between two vectors is used as a measure of divergence between the vectors, and cosine of the angle is used as the numeric similarity (since cosine has the nice property that it is 1.0 for identical vectors and 0.0 for orthogonal vectors) [9].

If $\vec{D}$ is the document vector and $\vec{Q}$ is the query vector, then the similarity of document $\vec{D}$ to the query $\vec{Q}$ according to [9] can be represented as:

$$\text{Sim}(\vec{D},\vec{Q}) = \sum_{ti \in D,Q} WtiQ.WtiD \quad \text{.....Equation 3.3}$$

Where W_{tiQ} is the value of the i^{th} component in the query vector $\vec{Q}$, and W_{tiD} is the i^{th} component in the document vector $\vec{D}$.

Probability Model

This model is based on the general principle that documents in a collection should be ranked by decreasing probability of their relevance to a query. Since true probabilities are not available to an IR system, probabilistic IR models estimate the probability of relevance of documents for a query [9]. This estimation is the key part of this model and makes this model differ from other models. The probability relevance of a document is denoted by $p(R|D)$ and the document is ranked by using:

$$\log \frac{P(R|D)}{P(\overline{R}|D)} \quad \text{.....Equation 3.4}$$

Where $P(\overline{R}|D)$ is the probability that the document is non-relevant.

Inference Network Model

In this model, document retrieval is modeled as an inference process in an inference network. In this model a document instantiates a term with certain strength, and the credit from multiple terms is accumulated given a query to compute the equivalent of a numeric score for the document [9]. From an operational perspective, the strength of instantiation of a term for a document can be considered as the weight of the term in the document, and document ranking in the simplest form of this model becomes similar to ranking in the vector space model and the probabilistic models described above. The strength of instantiation of a term for a document is not defined by the model, and any formulation can be used.

3.1.9 Evaluation Techniques for IR System

In the preceding sections we have seen many alternatives in designing an information retrieval (IR) system. To know which of these techniques are effective in which application the effectiveness of the IR system should be measured. The standard approach to IR system evaluation revolves around the notion of relevant and no relevant documents. Relevance score is an estimate of the search system on how closely the item satisfies the search statement. Its value is between 0.0 and 1.0 and the highest value 1.0 is interpreted that the system is sure that the item is relevant to user query statement [31]. Relevant items are those documents that contain information that helps the searcher in answering his/her question. Non-relevant items are those items that do not provide any directly useful information. When a user decides to issue a search looking for information on a topic, the total database is logically divided into four segments as shown in Figure 3.1

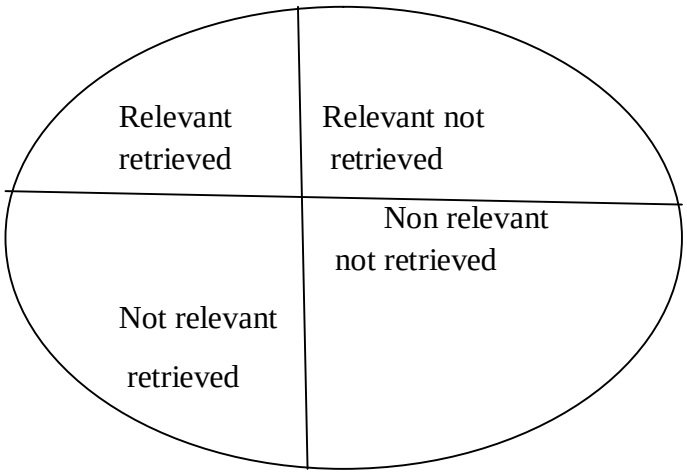


Figure 3.1*: Effects of search on document space*

To assess the effectiveness of an IR system (the quality of its search results) the following techniques are used most frequently [22].

Recall: Recall gauges how well a system processing a particular query is able to retrieve the relevant items that the user is interested in seeing. The ratio of documents retrieved versus the number of available documents relevant to the query, i.e., the fraction returned out of all desirable documents [18].

$$\text{Recall(R)} = \frac{Numberofrelevantitemsretrieved}{Numberofrelevantitems} \quad \text{..... Equation 3.5}$$

Precision: Precision measures one aspect of information retrieval overhead for a user associated with a particular search. If a search has 85 percent precision, then 15 percent of the user effort is overhead reviewing non relevant items. The ratio of the number of relevant documents retrieved versus the total number of documents retrieved, or the useful fraction of what was actually retrieved [18].

$$\text{Precision (P)} = \frac{NumberOf\ \text{Re}levantItems\ \text{Re}trieved}{NumberOf\ \text{Re}trievedItem} \quad \text{……. Equation 3.6}$$

F-measure: The recall and precision methods are often combined as their harmonic mean, known as the F-measure, which can be calculated as:

$$\text{F=}\frac{2RP}{R+P} \quad \text{……..….Equation 3.7}$$

The value of F-measure is range in [0, 1]. F-measure will be 1.0 if the matching between two objects is perfects or relevant [76] and 0 when no relevant documents have been retrieved.

3.2 Document Image Retrieval

Document retrieval is the process of the retrieving relevant documents from huge collection of documents. Electronic document retrieval used to be a task commonly associated with librarians, specialized business and legal analysts, working with proprietary online information services. Like text based information proper organization and access of document image to the archives is critical for efficient use of information. To do so a number of operations must be carried out. Preprocessing of image documents is first done and involves de-skew, noise

removal and layout analysis to remove headlines and pictures or photographs leaving finally the main text body of an article to be processed which is typically of one predominant font type and size [81]. Connected component analysis is then performed to identify character objects. The various steps involved in document image retrieval are data capturing, noise removal, feature extraction, matching and retrieval of documents, which are discussed here.

3.2.1 Document Image Processing Stages

Data Capturing: Data in a paper document are captured by optical scanning and stored in a file of picture elements, called pixels, which are sampled in a grid pattern throughout the document. These pixels may have values: OFF (0) or ON (1) for binary images, 0–255 for gray-scale images, and 3 channels of 0–255 color values for color images [81].

Pixel-level Processing: The next step in document analysis is to perform processing on the captured image to prepare it for further analysis. Such processing includes: Thresholding to reduce a grayscale or color image to a binary image, reduction of noise to reduce extraneous data, segmentation to separate various components in the image and finally thinning or boundary detection to enable easier subsequent detection of pertinent features and objects of interest [20].

Binarization of Document Image: Binarization plays a key role in document processing and usually is performed first in document image processing. Its performance affects quite critically the degree of success in line, word and character segmentation and recognition [21]. The objective of binarization is to automatically choose a threshold that separates the foreground and background information. There are different types of binarization techniques like Otsu, Adaptive, Sauvola, Global threshold based, etc [77].

Noise Reduction: After binarization, document images are usually filtered to reduce noise. Digital capture of images can introduce noise from scanning devices and transmission media. Salt-and-pepper noise (also called impulse noise, speckle noise, or just dirt) is a common form of noise on a binary image [81]. Noise removal is carried out to get rid of any noise or printed text overlapping the extracted images such as signature, logos, machine-print etc. In the preprocessing step the printed text is removed from the image samples. To remove the printed

text from images, variety of methods can be used such as image enhancement procedures based on chain code [21], Support Vector Machine (SVM) [46] to classify each connected component as a part of noise components, signature, a small handwritten text, logo, noise etc. The image has to be enhanced so that the result is more important than the original image for specific application. Filtering is a fundamental operation in image processing and used for image enhancement, noise reduction, edge detection and sharpening [37]. It involves the techniques such as median filtering, average filtering, Gaussina filtering, convolution, etc.

Skew Detection and Correction: A text line is a group of characters, symbols, and words that are adjacent, relatively close to each other, and through which a straight line can be drawn (usually with horizontal or vertical orientation). The dominant orientation of the text lines in a document page determines the skew angle of that page [65]. A document originally has zero skew, where horizontally or vertically printed text lines are parallel to the respective edges of the paper, however the alignment of the image with the horizontal and vertical axis is often broken during copying, scanning or faxing and the document becomes skewed. In literature many skew detection techniques such as Hough transformer, Cross correlation, projection profile, Fourier transform and K Nearest neighbor clustering etc have been implemented [65].

Detection of Text Region: For a fast and efficient word spotting procedure, it is desired to constrain the applied matching process only on certain regions of interest. These regions should correspond to the text regions of the document image.

Segmentation: Segmentation is the process of dividing the document into homogeneous areas of interest and describing each geometrical structure of the document. Before feature-level analysis can take place, segmentation must be performed to detect individual regions objects in the image [70]. Segmentation is dependent on local decisions with regards to shape similarity, as well as global decisions with regards to surrounding context [52]. The two common segmentation operations in scanned document are line and word segmentation.

Line Segmentation: Text line segmentation is a labeling process which consists in assigning the same label to spatially aligned units (such as pixels, connected components or characteristic points) [70]. Text line extraction is generally seen as a preprocessing step for tasks such as document structure extraction, printed character or handwriting recognition [23]. There are two

categories of text line segmentation approaches: searching for separating lines or paths, or searching for aligned physical units. The choice of a segmentation technique depends on the complexity of the text line structure of the document. Lines separation from the text is done by using the measure of the density of white lines in the horizontal projections. Local minima in the horizontal projections are analyzed to localize line separation.

Word Segmentation: Words are the fundamental unit for generating index of a document, enabling very rapid search by query. Word images are used directly in some application without converting in to strings [56]. So the identification of words is useful for many applications. Segmentation of every text line into constituent word images is done by using vertical projection profiles.

3.2.2 Feature Extraction

Feature extraction involves extracting the meaningful information from the document images [46]. So that it reduces the storage required and hence the system becomes faster and effective in document image retrieval. Once the features are extracted, they are stored in the database for future use. To query an image in the database, the feature value of the query image is computed first. Then, similarity measure is computed with each of the feature values of the database images from feature database and the query. As [38, 3, 78, 79, 80] points out different features are used for document image retrievals. Some of them are listed as follows:

1. Width to height word ratio: This is the division of a given word width to height and proved important information about word.

2. Word area density ratio: This feature represents the percentage of black pixels included in the word bounding box.

3. Center of gravity: This feature represents the Euclidean distance from the word's center of gravity to the upper left corner of the bounding box.

4. Vertical projection: Each feature value is the sum of pixels intensities in the corresponding image columns.

5. Horizontal projection: Each feature value is the sum of black pixels in each row of a given word.

6. Upper word profile: Each feature value is the distance from the top of the word's bounding box to the first ink pixel in the corresponding image column.

7. Lower word profile: It is the same as upper word profile, but distance is measured from bottom of image bounding box.

8. Top word shape: It is the distance from the first black pixel to bottom boundary of a word image in each column.

9. Bottom word profile: This is similar to top shape except distance is measured from top boundary of a word.

3.2.3 Word Image Normalization

To extract word objects from document images, connected component analysis is employed to detect connected image pixels and identify word bounding boxes. Since each word object may be in different size, all the above word image features listed under (section 3.2.2) must be normalized so that their maximum range is [0…1] [24,3].This ensures that one word features can be compare with words with different heights.

3.2.4 Similarity Measures

In the case of large document collections, the resulting number of matching documents can be difficult for human to filter out according to the extent to which the document is relevance to the query statement. Accordingly, it is essential for a search engine to rank-order the documents matching a query. To do this, the search engine computes a score with respect to the query at hand for each matching document. Similarity measures play a very significant role in matching. There are many ways to measure text similarity of documents. One way is to analyze the similarity of the documents' contents based on semantics but this needs a large amount of processing time [24]. Another way is to use a statistical method without the need to understand the meaning of documents. Operational IR systems are predominantly based on statistical

measures of overlap between documents and queries, counting the number of words or index terms in common between the two as part of some similarity measure [7]. So similarity measures play a very significant role in information retrieval matching. Different similarity measures can be used to calculate the similarity between the item and the search statement. A characteristic of a similarity formula is that the results of the formula increase as the items become more similar and the value is zero if the items are totally dissimilar [22]. In literature there are a number of matching techniques. Some of them are listed as follows:

Euclidean Distance

The Euclidean distance is an example of a distance measure that can be used to measure the syntactic similarity of two images. The Euclidean distance captures the syntactic differences

between images very accurately [2].

This can be calculated with a Pythagorean formula:

$$\left\|\vec{D}-\vec{Q}\right\| = \sqrt{\sum_{i=1}^{p}(xi-yi)^2} \quad \text{.....Equation 3.8}$$

Where xi and yi are the i[th] values of $\vec{D}$ and $\vec{Q}$ respectively.

The query image will be more similar to the database images if the distance is smaller.

Normalized Cross-correlation (NCC)

The NCC is a similarity measure that is used in image matching to measure the similarity between matching entities in one image and their corresponding entities in the other image [26].

For two variables x and y, which yield the data sets

X={x1,x2,x3,…xn} and y={y1,y2,y3,…..yn} then the normalized cross-correlation between these variables can be calculated as :

$$R(x,y)=\frac{\sum_i (xi-\bar{X})(yi-\bar{y})}{\sqrt{\sum_i (xi-\bar{x})2}\sqrt{\sum_i (yi-\bar{y})2}} \quad \text{.....Equation 3.9}$$

Where $\bar{x}$ and $\bar{y}$ are the sample mean values of x and y, respectively.

The values R range between [-1, 1]. R=-1 when the matching entities are inverses of each other, R=0 when there is no relation between the matching entities and R=1 if the matching entities are exactly the same.

Cosine Distance

The cosine distance is one of the simplest ways of computing similarity between two documents by measuring the normalized projection of one vector over the other [8]. It is a measure of similarity between two vectors of n dimensions by finding the cosine of the angle between them. Once documents and queries are represented in terms of vectors, we can retrieve a document based on the similarity measure which is the normalized inner dot product between the document and the query vectors. Given two vectors q and d, the cosine similarity between images is given by:

$$\text{Sim}(\vec{q},\vec{d}) = \frac{\sum_{i=1}^{n} qi.di}{\sqrt{\sum_{i=1}^{n} qi2 \sum_{i=1}^{n} di2}} \quad \text{.....Equation 3.10}$$

Where $\vec{q}$ and $\vec{d}$ are vector of the two images, and n is the dimension of each image vector.

In the case of information retrieval, the cosine similarity of two documents will range from 0 to 1. Where 0 means the documents are independent, 1 means exactly same and in-between values indicates intermediately similarity.

Dynamic Time Warping

Dynamic time warping (DTW) is a dynamic programming based procedure to align two sequences. It is a well-known technique to find an optimal alignment between two given (time-

dependent) sequences under certain restrictions [45]. The sequences are warped in a nonlinear fashion to match each other. Though it is originally has been used to compare different speech patterns in automatic speech recognition processes, bio-informatics and handwriting recognition [38, 45, 76], two word images can be compared by matching their character features using DTW and it can be used successfully in the fields such as data mining and information retrieval to automatically cope with time deformations and different speeds associated with time-dependent data. The objective of DTW is to compare two sequences X = (x1, x2,…, xN) of length N x N and Y = (y1, y2, . . . , yM) of length M x N. These sequences can be features sequence sampled at equidistant points in time.

Let the word images are represented as sequence of vectors of query terms and document terms. The DTW-cost between these two sequences is calculated using dynamic programming as:

$$D(i, j) = \begin{cases} D(i-1, j-1) \\ D(i, j-1) \\ D(i-1, j) \end{cases} + d(i, j) \quad \text{.....Equation 3.11}$$

Where d (i, j) is the local distance cost in aligning the i^{th} element of term with j^{th} element of the document and is computed using a simple squared Euclidean distance:

$$d(i, j) = \sqrt{\sum_{i=1}^{p} (Fik - Gjk)^2} \quad \text{.....Equation 3.12}$$

Where Fik is the k^{th} features of Fi and Gjk is k^{th} features of Gj

3.3 Related Works

In this section some related works that are more related to our work and become a base for our work are presented. Few researches in Amharic document images retrieval without OCR have done so far. So in this section we point out research works have been done in Amharic and other languages that are related to us, summarize the approaches they follows and the results obtained from the researches. The main previous contributions in the area include the work done by

Anand [11], Zhang [39], Million [47], Abrham[1] and Mesfine[40] in which they investigated some word image features for information retrieval and word image matching techniques.

3.3.1 English and Other Non-Amharic Document Images Retrieval

Anand's 2007 article about efficient document image indexing and a retrieval scheme for searching in large document image databases is one of the most cited articles in the literature of the area. According to this article, word image matching is used to retrieve document images in response to a word image query. Anand et al [11] argue that searching using direct matching of images is inefficient due to the complexity of matching and thus impractical for large databases and proposed directly hashing word image representations to solve that problem. The working principle of their system is automatic word segmentation followed by feature extraction at word level and finally indexing. According to this work reports word retrieval is done very efficiently by using an approximate nearest neighbor retrieval technique called locality sensitive hashing (LSH). Word images are hashed into multiple tables with features computed at word level. Content-sensitive hash functions are used to hash words such that the probability of grouping similar words in the same index of the hash table is high. The sub-linear time content-sensitive hashing scheme makes the search very fast without degrading the accuracy. An experiment is conducted to evaluate the performance of their proposed system and techniques on Indian language books. The results show the approach used to search large document image collection is practical. In the proposed retrieval technique the index is built by hashing word level features of document images. The features are hashed using content sensitive hash functions; such that the probability of finding words with similar content in the same bucket is high. The same content sensitive hash functions are used to query similar words during the search. As their conceptual diagram shows scanned document pass through different image processing for indexing purpose. The textual word query is first converted to an image and then features are extracted from these images and then search is carried out to retrieve relevant word images. To represent word image they employee a combination of scalar, profile, structural and transform domain feature extraction methods. An accuracy of 90% recall and precision are obtained.

Zhang et al [39] have done another important work on document images retrieval on the web. They employee word image matching to retrieve document image in digital library. In this work

some image preprocessing are first carried out off-line to extract word objects from document images stored in the digital library. Then, each word object is represented by a string of features codes. As a result, each document image is represented by a series of feature code strings of its words, which are stored in a feature code file. Query word specified by the user at client machine is sent to the server and the server convert user text into feature code. An inexact string matching technique is applied to match the query word with the words in the documents and then term frequency of the query term is used to rank retrieved documents. As their report indicates the results of the experiment shows that proposed system is efficient and promising for document image retrieval, and has potential applications to digital libraries.

Lu et al [62] report document image retrieval through word shape coding. This work presents a document retrieval technique that is capable of searching document images without optical character recognition (OCR). The technique applied is word image annotation and its applications to the document image retrieval by either query keywords or a query document image. They annotate word images by a set of topological character shape features including character ascenders/descenders, character holes, and character water reservoirs. According to the report of this paper the word annotation technique presented in this paper has the following three advantages relative to some earlier works.

1. It is much faster because it does not require the time-consuming for connected component labeling.

2. The character shape features in use are more tolerant to the document skew and the variations in text fonts and text styles.

3. Its collision rate is much lower because of the distinguish ability of the three character shape features in use.

The proposed word image annotation technique apply document image preprocessing, word shape feature extraction and word image representation operations. During document image processing document images are first smoothed to suppress noise by a simple mean filter within a 3 x3 window. The filtered document images are then binarized. After that, words and text

lines are located through the analysis of the horizontal and vertical document projection profiles.

At the time of word shape feature extraction three character shape features which are character ascenders/descenders, character holes and character reservoirs are used. Character ascenders and descanders are simply located based on the observation that they lie above the x line and below the base line of the text, respectively. Then, Character holes and character reservoirs are detected through the analysis of character white run scanning vertically (or horizontally) from top to bottom (or from left to right).

During word images representation each word is represented as a linear sequence of codes rather than representing each and every character in a word. To deal with character segmentation error, they annotate word images by using five shape features including character ascenders or descenders, character holes, and leftward and rightward character reservoirs. The five shape features in use are annotated by two types of codes according to their vertical alignment. The first type has been used when the five shape features have no vertically aligned shape features (such as the hole of 'o' and the rightward reservoir of 'c'). The second has been used when the five shape features have vertically aligned features (such as 'e' whose hole lies right above its rightward reservoir). Based on the word shape coding they describe document image and it is retrieved by either query key words or a query document image based on their content similarity. Experimental results show that the proposed word image annotation technique is fast, robust, and capable of retrieving imaged documents effectively.

In [63], Simone et al present a system that performs DIR in Digital Libraries without OCR. The system allows users to retrieve digitized pages on the basis of layout similarities and to make textual searches on the documents. The system integrates a font-independent word indexing and a layout-based document retrieval tool into a unique framework to perform word indexing at level with layout based retrieval components. Their system has three main components: indexing, query formulation and similarity computation without subsequent ranking of the indexed documents. The similarity between the indexed documents and their query is computed at the feature level. The approach used during indexing is recognition free. This approach has been exploited for word indexing, keyword spotting, graphical items, layout and hand writing

retrieval. The queries to the system are given in the form of text and layout of the interested document. If it is text, it is converted into image by using Latex package. During indexing, pages are first processed by a layout analysis tools that extracts homogeneous regions. Textual regions are subsequently analyzed so as to extract the words which are encoded with appropriate character labels. At the same time the layout is encoded in order to obtain a page-level representation of the documents. Then, the pages can be retrieved by taking into account both textual and layout queries. The query and indexed words are encoded similarly. Analogously, a query page is represented in the same way of indexed pages that can be ranked according to their layout similarity.

Zagoris et al [71] present a document images retrieval system that can locate words in document image archives. In this paper, the word spotting technique that can perform word matching directly in the document images by passing character recognition and using word images as queries. They describe both the query and the words occurring in the document images with features, which may then be matched in order to identify query term occurrences. The main strong side of this paper is high noise tolerance and is language independent. The document images are analyzed in order to locate the word boundaries inside them offline. Then, a set of features that can capture the word shape and discard detailed differences due to noise or font differences are calculated from these words and the results are stored in a database. On the other hand, the user enters a query word and then the proposed system creates an image of it and extracts the same set of features. Consequently, these features are used in order to find similar words through a matching procedure. Finally, the documents that contain these similar words are presented to the user. The offline activities involves Preprocessing operation such as noise removal and binarization; word segmentation which follows preprocessing and used to detected word limits ,filter noise and punctuation marks. The word segmentation is done by using connected component analysis. Feature extraction is another offline operation that has done in this paper. Seven features are used to represent each word in the document. These features are capable of capturing the word similarities and discarding the small differences due to remaining noise or different style of fonts. These features are width to height ratio, word area density, center of gravity, vertical projection, Top-bottom projection, Upper grid and down grid features. User's created query is processed exactly in the same way as the document. Then, matching

between query word image and that of document word image is based on these feature vectors. According to the report of the experiment their system performs 87.8% mean precision and 99.26% mean recall.

3.3.2 Amharic Document Images Retrieval

Jawahar et al [47] proposed a technique for searching Amharic, English and Hindi document images from a database without using OCR. The retrieval from scanned documents is performed by matching the features of word images. In this research work 16 word features are used to describe various properties of a word images. These features are upper and lower profiles, projection, density, ink-to-background, Normalized moments (M00, M01, CM01), statically moments (mean, media, skew). To group similar word images in document image together for the purpose of indexing DTW is used. It is used for indexing and retrieving documents online. It aligns and compares sets of features extracted from two images. Words that are common in all documents are considered as stop words and removed from index file. The results of search are returned into user in ranked manner based on the relevance of the query. The researchers extends a method extends a method to cross-lingual retrieval by transliteration among Indian languages and a table look-up translation among other languages. In this research work the index file is constructed after removing only affixes that is found at the beginning and end of a word. Their system is evaluated with real life document images and accuracy close to 95 % is obtained for both precision and recall to all languages.

Mesfin [40] has attempted to design Amharic document images retrieval system that provides response to user query. To identify word images from document collection segmentation technique is employed. In the study, he has used the word shape analysis or parallel bar vertical technique in order to extract feature. To measure the similarity between a query and the document image, Euclidean similarity measurement with a parallel bar vertical line feature extraction is used and the result of mean average precision is 82% and suffix detection and removal is performed. His report says that searching information from 483 pages takes about 22 seconds.

In [1], Abreham dealt with searching from Amharic document images corpus without explicit recognition. His work aims at introducing an index structure using inverted index file in order to speed up searching in Amharic document images. In this work an attempt has been made to identify index terms after stop words are removed and variants words are detected. The inverse document frequency and the cutoff (threshold) values are calculated to remove the stop word images. If the cutoff value is greater than the inverse document frequency, then the word image is considered as stop word, otherwise indexed. The inverted index structure is used for indexing word images. Cosine similarity measurement is used to compare the degree of similarity between two word images. At the time of building index file and searching, the system handles either suffix or prefix attached to the index term or query term to reduce to its stem. The system removes high frequency words that do not have discriminating power among document images in Amharic corpus using inverse document frequency. The system displays the search result in ranked order based on their relevance weight to the query. Abreham reports that the effectiveness of the system shows F-measure value of 41.59% and searching time of the system has improved after building file by 26.6%. The prototype of his system is designed in Java. The query provided to the system is limited to single query and also the approach for indexing is not sufficient. However, searching can be performed by a phrase and linguistic analysis must be done on document and query side.

In addition to these a number of document images retrieval systems have been reported in different languages by using different techniques.

CHAPTER FOUR

PROPOSED SYSTEM ARCHITECTURE

4.1 Introduction

This chapter presents the proposed ADIRS based on exact word images matching by using word image features. Figure 4.1 illustrates the overall system framework and its workflows. Some image preprocessing are first carried out off-line to extract word image features from document images. Then, each word object is represented by its feature values. Two feature values about word shape information and one feature values about word projection are used to describe word image. The extracted feature values from each word image are used for the purposes of indexing and searching. All matching word objects those are recognized according to a threshold value are used for indexing. On the other hand, Amharic word synthesizer is integrated in this architecture. By providing an Amharic root word this tool produces possible surface words. Then, surface words are converted into images and finally feature code for each word image is generated containing a set of word image descriptions such as root word code, word width, word height and feature value 1, feature value2,... feature value n. The description of each word image is stored in a database and used while indexing and searching. On the client side, the user is prompted to input a set of query words through an interface. Once the request is submitted the proposed system converts user input into image and computes feature values of each query word. Feature matching algorithm is applied to compare the feature values of the query word with the feature values in feature corpus. The occurrence frequency of the query word in each retrieved document image is calculated for ranking purpose. The system performs all preprocessing and matching procedure online when the user inputs query word/s. Finally, a temporary table that stores all the matching documents and the normalized occurrence frequency of the query words will be

returned to the user for display. As shown in Figure 4.1, ADIRS architecture can be broken into four major components. These are:

- user query processor;
- Amharic word synthesizer and words features database construction;
- document image processor and
- document image indexing, ranking and retrieving.

In this architecture (Figure 4.1) Amharic word image's feature file construction ,document image processing and indexing operations are carried out offline where as searching is done online. Each of the system components has subcomponents and performs a number of operations. The descriptions of the components are described in the following sections.

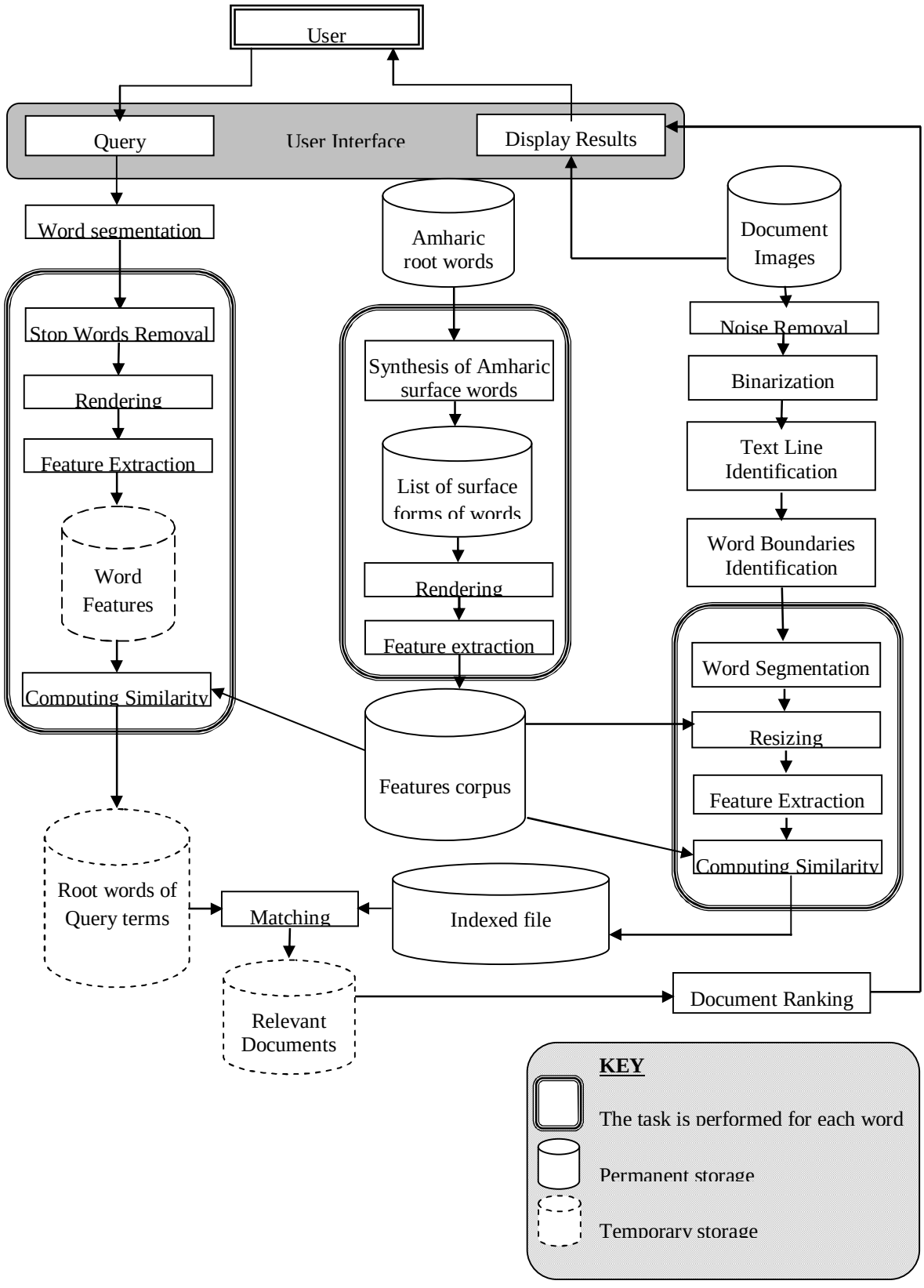


Figure 4.1: *Overall system architecture*

4.2 Amharic Word Synthesizer and Word Image's Feature Corpus Preparation

As mentioned earlier under chapter 2, Amharic is a morphologically rich language. It is possible to create many words by attaching different affixes to a stem. In our system we use Amharic word synthesizer that is produced as a research thesis for master graduation at Addis Ababa University by Kibur Lisanu [34]. This system can produce more than 1000 words for a given Amharic root word. To build our feature corpus database for the purposes of index creation and searching the outputs of the synthesizer are converted into word images. Finally, features that represent each word image are computed and stored in a database. The presence of the word synthesizer plays crucial role for languages those are highly morphological. Since Amharic belongs to such language it is beneficial to use the synthesizer while indexing and searching. The words formation from Amharic language can be prefixation, suffixation, nfixation,circumfixation and concatenation. If the word is formed by attaching prefix or/and suffix, it is possible to remove such affix to get the root word for the sake of indexing. However, it is very difficult to compare two words those are under similar word group with different word formation structures other than at the beginning or/and end of a word to return root for index file. Due to this it is very difficult to group similar words together for selecting terms that could be the representative of Amharic documents. This was one of the main problems for previous research works in the area of ADIR. Having such database addresses this problem well. Figure 4.2 shows how the integrated Amharic word synthesizer generates surface words from a given root word and also how features are extracted from those word images.

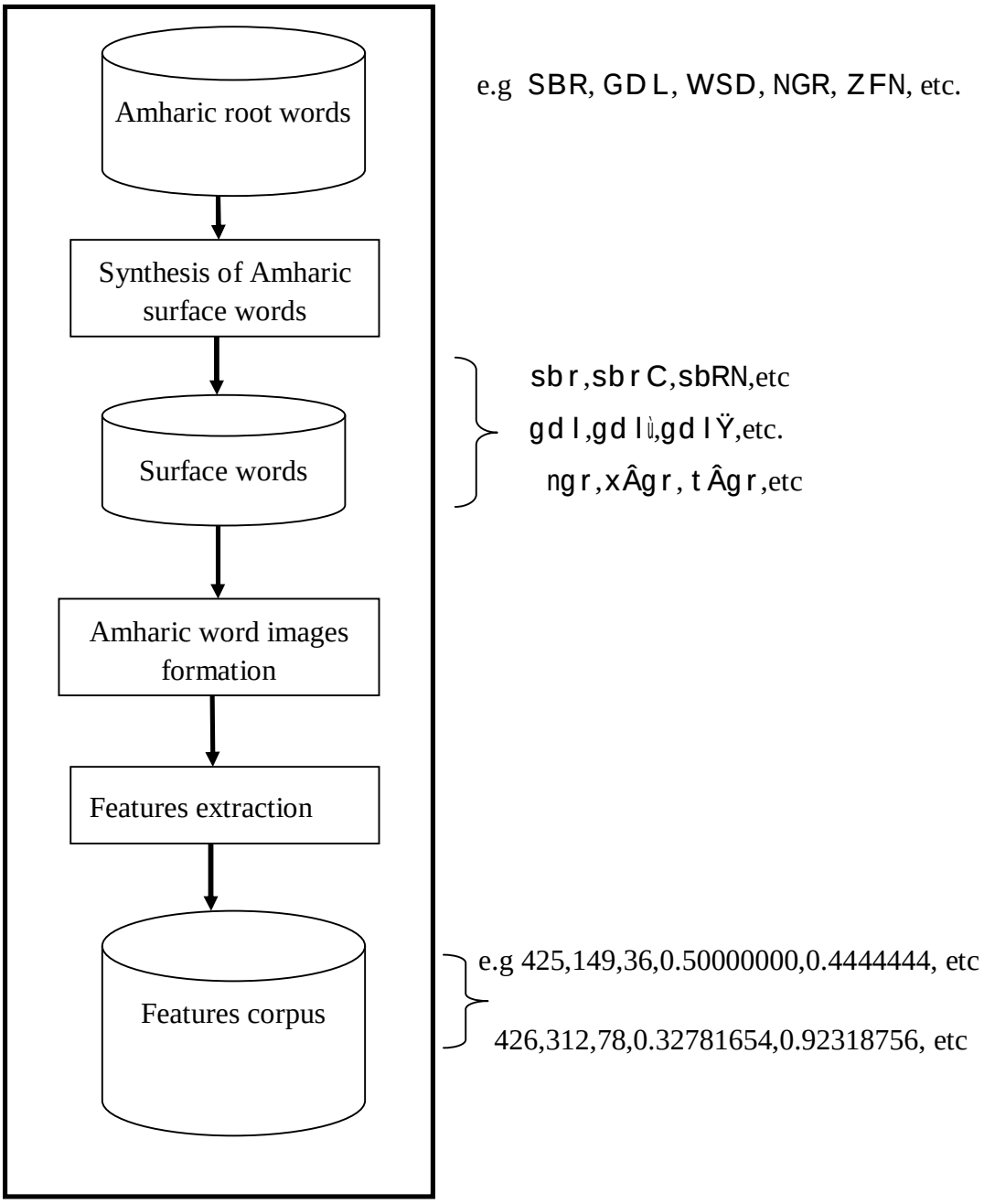


***Figure 4.2**: Amharic words formation and feature extraction*

The feature values of words' images of the output of the synthesizer are used during indexing. Also it is used during query processing in order to ensure similar representation as that of the document collection.

4.3 Document Image Processing

This operation is done offline and involves scanning and storing documents as images in a database, noise removal, binarization, line segmentation, word segmentation and removal of unnecessary objects. Document images database is a collection of scanned Amharic document images. This database holds dataset as image for testing the performance of the system and formulated algorithms. Document representing terms are taken out from this database. Moreover, searching document images is done from this database. These documents must pass through some phases for indexing purposes. Figure 4.3 clearly shows the step by step procedures to process the document images for the purpose of indexing.

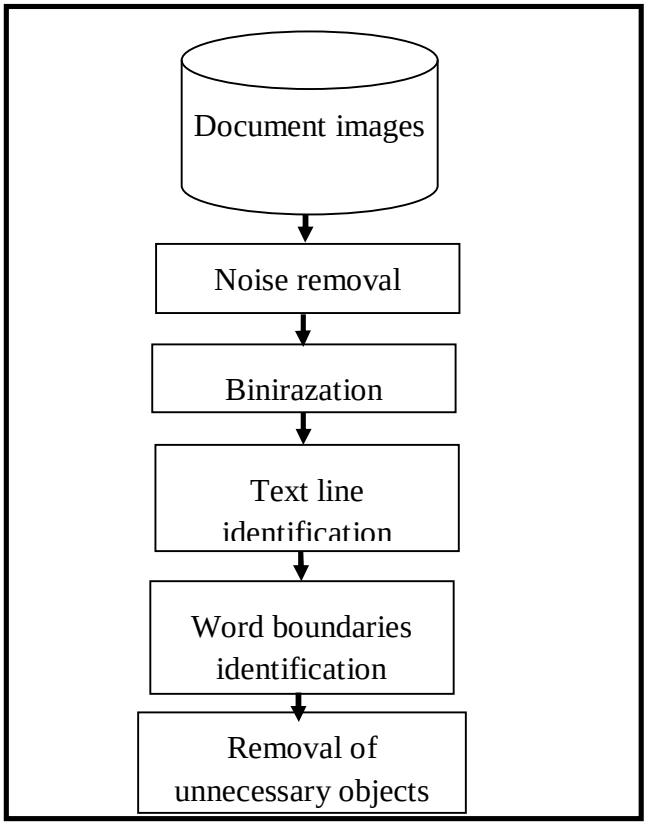


Figure 4.3*: Document images processing*

A connected component analysis algorithm is first applied to detect all the connected components. Those with too small area are considered as noise and thereby removed. Even though, skewing is unavoidable error during scanning documents we assume that all document

images are normal. Since our dataset are full of text rather than images, pictures, graphs and tables, we are not taking into account detection of text region. However, this information has a big impact on indexing and retrieval.

4.3.1 Noise Removal

For some noisy documents we used matlab noise removal techniques that are linear and median filter by adjusting impulse parameter according to the nature of the noise. Median filtering is a cornerstone of modern image processing, and is used extensively in smoothing and de-noising applications.

4.3.2 Binarization

The document images file may have different pixel values. But for our purpose we need only two values –either 0 or 1. For that purpose, we convert the gray scale images into binary. All pixels values presented in the documents are converted into either black (pixel value 0) or white (pixel value 1). This is the second step in our document image processing next to scanning for noisy documents. Each pixel is set as per Listing 4.1. Listing 4.1 is used to differentiate text objects from background.

```
1. Input: grayscale document image.
2. Output: binirized document image.
3. if (pixel_intensity > ð) then
4. Pixel_intensity=0; // which is an object on the back
   ground
5. Else
6. Pixel_intensity=1; //which is background
7. End if
```

Listing 4.1*: Binarization algorithm*

4.3.3 Line Segmentation

After passing the above two operations (noise removal and binarization) the next important phase is line segmentation which is essential for word segmentation. The line segmentation operation involves top and bottom text lines identifying activities.

Listing 4.2 is an algorithm that is implemented for identifying the upper boundary of a given text line.

```
1. Input: Binarized Amharic document image, initial top row,
   height and width of the document.
2. Output: upper boundary of a text line.
3. Initialization: initial top row =1;
4. For i=1 to height_of_document do
     a. For j=1 to width_of_document do
          i. If image(i,j)==0  then // if pixel value is black
         ii. top_row=i;
        iii. Else
         iv. Continue ;
          v. End if
     b. End for
5. End for
```

Listing 4.2*: Top row identification of a text line*

Listing 4.3 is implemented for identifying the lower boundary of a given text line.

```
1. Input: Binarized Amharic document, top row, height and
   width of the document.
2. Output: Lower boundary of a text line.
3. Initialize: bottom row from top row
4. For  p=top_row to height_document
      a. For j=1 to width_of_document
           i. Search row with all pixel values are 1(white)
          ii. If  step i is true then
         iii. Take p-1 as the bottom row of a text line
          iv. End if
      b. End for
5. End for
```

Listing 4.3*: Searching bottom row of a text line*

4.3.4 Word Segmentation

Word segmentation process is the next phase after text line segmentation process. The upper and lower boundaries of a word image are found from text line boundaries. So word segmentation performs word start and word end identification operations from left and right end of the word image, respectively. It has been performed by using Listing 4.4 and Listing 4.5. Listing 4.4 is an algorithm that is important for identifying the starting point of a word.

```
1. Input: Binarized document image, initial column, top
   and bottom row of a text line.
2. Output: the starting point of a word.
3. Initialize: word start at top left corner of text line.
4. For   p= 1 to width_of_document do
      a. For  j= top_row to bottom_row do
           i. Search the location of the first black pixel
          ii. If step i is true then
         iii. Take p as word start
          iv. End if
      b. End for
5. End for
```

Listing 4.4*: Finding the starting point of a word*

To find the word end point Listing 4.5 is implemented.

```
1. Input: Binirazed document image, word start point, top and bottom
   row of a text line.
2. Output: location of word end point.
3. Initialize: word start as word end
4. For i=1 to width_of_document  do
      a. For j=top_row to bottom_row  do
           i. Calculating blank space between two consecutive
              different pixel values along columns.
          ii. If length_of_blank_space > height_of_text_ line/3 then
         iii. Take i-1 as word end
          iv. End if
5. End for
```

Listing 4.5*: Finding the end point of a word*

4.4 Document Indexing

Once the system stores all the necessary information about the document in index database, upon which searching can be conducted in this database. This index database contains document representing terms, terms frequency in a given document and links to the original document. Therefore, inverted indexing is used as indexing technique in this work. All the pre-processing operations (segmentations, removal of unnecessary objects, etc.) are applied on the document before indexing process has begun. As much as possible we are trying to reduce the size of index database file without affecting the content of document representative words. Hence, the index database holds only list of Amharic root words code with terms frequency and documents ID.

For example, the document might contain variants of the root word "**ዘፍን**" which is "sing", such as **ዘፈነ**, **ዘፈንኩ** ,**ዘፈናችሁ**, **ዘፈንሽ**, **ዘፈንክ**, **ዘፈኑ**, **ዘፋፈነ**, **አዘፈናቸው**, **ዘፍነዋል** , **ይዘፍናሉ**, **ትዘፍናለች**, etc. when they are used originally in the document. However, the index database holds the code of the root word **ዝፍን** as index term. Thus, by doing so it is possible to reduce the index file size and making search fast and accurate.

The index database is constructed by comparing segmented word features with Amharic word image lexicon features. After performing cropping, normalization, feature extraction, stop-words removal and stemming, the next step is to weight the terms according to their importance in representing a document. Not all terms are equally important in reflecting the content of the document. Thus, some terms that appear high frequently in a document images are selected to represent document. The weighting function used in this work is relying on the distribution patterns of the terms within a document. The selection is done by thresholding. Instead of making or selecting index terms by using Part Of Speech (POS), the system simply assign weight and use high frequency terms that are not stop words. Nouns, verbs, adjectives and adverbs are looked up in feature corpus database. Each word object is represented by a string known as its feature values, based on which the index database of the corresponding document is constructed. Common Amharic words such as connectors (**ይህ**, **ያ**, **እና**, YhN, XNJ, MKnÃtM, SlzH, bzH MKnÃT, SlçnM, etc.) are frequently presented in documents yet have no inherent meaning. Consequently, such words have been safely removed from documents index without losing contents. Figure 4.4 indicates the working principles of creating index files for each document image.

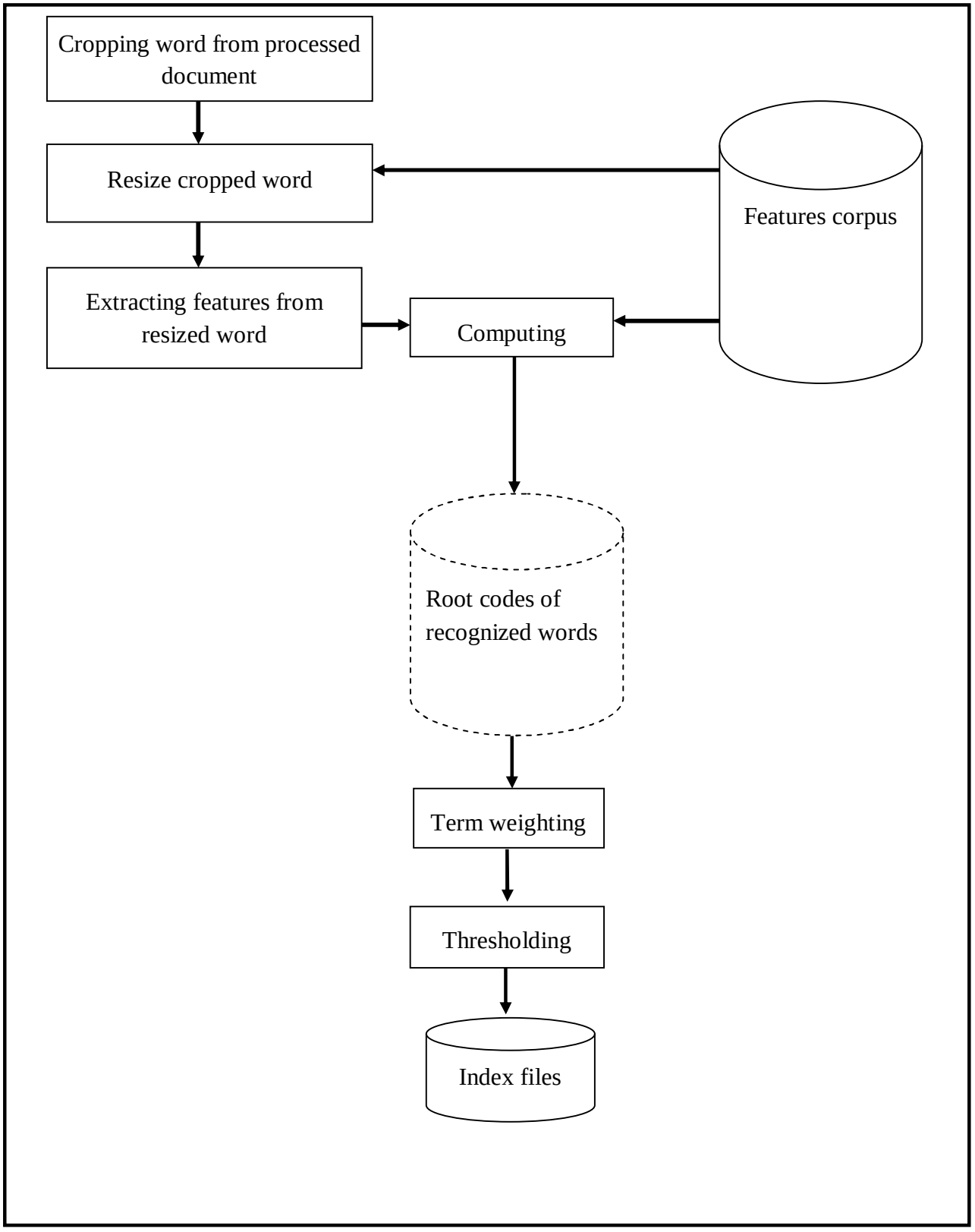


***Figure 4.4**: Index database construction work flows*

4.4.1 Cropping Word Image

For the purpose of indexing all words in a document have been cropped turn by turn. To do so the limits of each word image (top, bottom, left and right) are very essential. However, all words with in a given text line have not identical top and bottom boundaries. So that it is needed to get top and bottom for each word within the same line. These have been done by using Listing 4.6 and Listing 4.7, respectively.

To find the upper bound of each word in a given text line Listing 4.6 is implemented.

```
1. Input: Document image, top row, bottom row, word start and word end.
2. Output: Top boundary of a word.
3. Initialization: top boundary as top left corner of the word image.
4. For j= word_start to word_end do.
        a. For p= top_row to bottom_row do
             i. If(image(p,j)== 0) then     // which is black
            ii. Put p^th value in vector;
           iii. Exit;
            iv. Else
             v. Continue searching black pixel value;
            vi. End if
        b. End for
5. End for
6. Take the minimum value of the vector as word top boundary
```

Listing 4.6*: Searching top line of a word algorithm*

To find the bottom boundary of each word in a given text line Listing 4.7 is implemented.

```
1. Input: processed Document image, word start, word end, top-row,
   bottom row.
2. Output: Bottom boundary of a word image.
3. Initialization: Bottom left corner of a given word.
4. For j=word_start to word_end do
        a. For p= bottom_row to top_row  do
             i. If image(p,j)==0  then   // which is black
            ii. put p^th value in a vector;
           iii. Exit;
            iv. Else
             v. Continue searching pixel value with black;
            vi. End if
        b. End for
5. End for
6. Take the highest value of the vector as bottom boundary of a
   given word.
```

Listing 4.7*: Searching bottom line of a word algorithm*

After the top and bottom of each word image in a document are found each word is segmented form the document by using the limits of the word.

4.4.2 Word Image Resizing

The contents of the document may be formatted using different font sizes, types and styles. We have attempted to handle such variations by resizing each segmented word by the width of each template word. We have done it by applying the following approach. Listing 4.8 is important to resize segmented document word into each template word image size.

```
1. Input: cropped word, template features.
2. Output: normalized word image.
3. Cropping word using the limits of the word.
4. Calculating width of segmented word.
5. For i=1 to total _number_of_template_word_feature do
      a. Get the width of i^th template word;
      b. Calculate resizing parameter as
         factor=width_of_template_word/width_of_segmented_word;
      c. Resize by using imresize (segmented word, factor);
6. End for
```

Listing 4.8*: Word image resizing algorithm*

4.4.3 Features Extraction

After segmentation and resizing of a word image have been done, the feature vector of each word has been computed to recognize the shape of the word for the purpose of indexing and searching. In literature since many researchers [11, 39, 40, 62, 63, 71] agree on word level

approach rather than character level approach for document image retrieval we have used word as the smallest unit for indexing and searching. In this research work vertical projection (number of black pixel in each column), upper bound (distance from top of word rectangle to the first black pixel) and lower bound (distance from bottom of the rectangle to the first black pixel) of word image have been used. So far we have seen how to resize one word to other to have equal width for the sake of comparison. However, on top of width similar words have different height in different font size. So that to handle such problem each feature value is normalized by using the height of the given word.

Vertical projection: To compute the vertical projection feature values of a given word image, Listing 4.9 has been used. This algorithm counts the total number of black pixel in each column starting from word start to word end points and normalize by the height of a given word.

```
1. Input: word image (img).
2. Output: vertical feature values.
3. Initialization: feature vector=[].
4. For j=1 to maximum_ width_ of_ word _image do
        a. Count=0;
        b. For i=1 to maximum_height_of_word_ image do
                i.  If img(i,j)== 0  then
                ii. Count=count+1;
                iii. End if
        c. End for
5. Normalize the result of count using the height of the word.
6. Put in a vector.
7. End for
```

Listing 4.9*: Extracting feature values of a word using vertical project*

Upper bound profile: To compute the upper bound feature values of a given word image, Listing 4.10 has been used. This algorithm is used to identify only the upper shape of a word image.

```
1. Input: word image (img).
2. Output: upper bound feature values.
3. Initialization: feature vector=[].
4. For j=1 to maximum_width_of_word_image do
        a. Count=0;
        b. For i=1 to maximum _height_ of_ word_ image do
                  i. If img(i,j)== 1 then
                 ii. Count=count+1;
                iii. End if
        c.  End for
5. Normalize the result of count using the height of the word
6. Put in a vector
7. End for
```

Listing 4.10*: Extracting feature values using upper bound profile*

Lower bound profile: To compute the lower bound feature values of a given word image, Listing 4.11 has been used.

```
1. Input: word image (img).
2. Output: lower bound feature values.
3. Initialization: feature vector= [].
4. For j=1 to maximum_width_of_word_image do
        a. Count=0;
        b. For p= maximum_height to 1 do
                i.  If img(p,j)== 1  then  // which is background
                ii. Count=count+1;
                iii.  End if
        c. End for
5. Normalize the result of count using the height of the word
6. Put in a vector
7. End for
```

Listing 4.11*: Extracting feature values using lower bound profile*

To create an index file that contains full information about each document image such as Document Identification, root word and occurrence of instances of a given word Listing 4.12 is implemented.

```
1. Input: processed document image, template feature values.
2. Output: index file.
3. For i=1 to total_number_of_words_in_document do
     a. Segment word(i)
     b. For j=1 to maximum_length_of_template_features do
          i. Resize segmented  word image  by using Listing 4.8;
         ii. Calculate feature values by using one of feature
             extraction algorithms;
        iii. Compute similarity between resized word image and
             currently selected template word feature values by using
             equation 3.10;
         iv. If similarity >=ð  then
          v. Add the code of the root  and document ID to “temporary
             file”;
         vi. End if
4. End for
5.  Read “temporary file”
6. For p=1 to length of temporary file do
     a. Count frequency of identical  root word codes  in a given
        document id
     b.  Write  root code, term frequency and document id in “index
        file” database
7. End for
```

Listing 4.12*: Index database construction*

4.5 Query Processor

The user interacts with the proposed ADIRS using a graphical user interface. The interface facilitates the process of specifying a query and provides a good visualization of the results. This component lets the user enter queries in Amharic and retrieve the relevant documents for the given query. It has a text box that lets the user enter his/her query in Amharic and search (**ፈልግ**) button. The results of search will be returned back to the user. Figure 4.5 depicts how the user's input to the system is processed to get the required document images from a database.

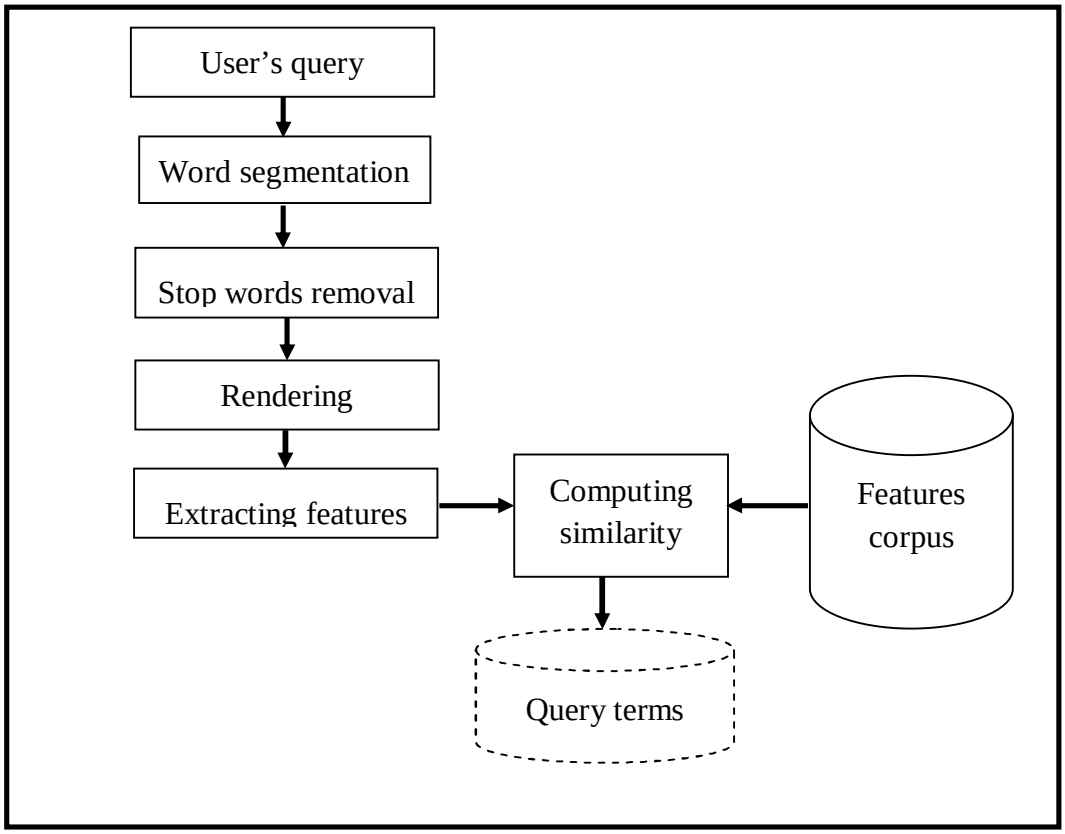


Figure 4.5*: Query processor*

The query processing component depicted in Figure 4.5 above handles most of the Amharic language unique features. It starts by accepting user query and applying words segmentation. The character streams will be break in to words by taking white space. All words of a query do not have equal value for retrieving purpose. Thus, such words could be removed from the query terms by comparing each word in the query with a list of common stop words developed for this

particular purpose. The list of common Amharic words such as **ሆነ**, **ሁለ**, **ነበር**, **ነው**, **ውስጥ**, **የውስጥ**, **በውስጥ**, **ይህ**, **ያ**, **እነዚህ**, **ሆኖም**, **ይሁን**, **እንጅ**, etc. are not important for searching purpose so that removed from query terms. Most of these and the like stop words belong to articles, conjunctions, prepositions, etc. word group. The remaining words will be converted into image and stemmed to root form if they have morphological variants to reduce them to their common form and producing query terms. The stemming is based on the assumption that words with the same stem are semantically related and have the same meaning to the user's query. For example **ቤት**, **ቤቱ**, **ቤታችን**, **ቤቶች**, **ቤቴ**, **ቤቶቻችን**, **ቤታቸው**, **ቤትሽ**, **ቤትህ**, etc. are stemmed into **ቤት**. Image preprocessing operations such as segmentations and features extractions are involved on the query. Like document images processing the same algorithms for word segmentation and word feature extraction are applied on the query. A set of features are extracted from word images in a query and matched against set of pre-stored template images features. The final results of all these processes are used to search documents in an index database.

4.6 Searching Techniques

Searching is done for two purposes. One is on features corpus to get the root code that represents a particular segmented word from the document for indexing purpose. This is done by word image feature to word image feature matching. On the other hand, the system tries to search the final processed query word(s) from index database that contains the representation of each document in the database. Searching is done by matching the query word image features with template word image feature in the feature corpus database. The word feature similarity measure discussed under section 3.2.4 (cosine similarity) is used to match the query image against all target images. If there are exact matches, information of the corresponding documents that contain this query word will be retrieved and stored in a temporary table for subsequent ranking. Otherwise, the system generates a message that indicates the absence of relevance documents to user's need. Since we have done searching operation with the assumption that the query and template words have the same specifications we remove cropping and resizing operations from our searching diagram. Figure 4.6 shows how the query is compared with template word images and index database to return relevant documents according to their relevance order.

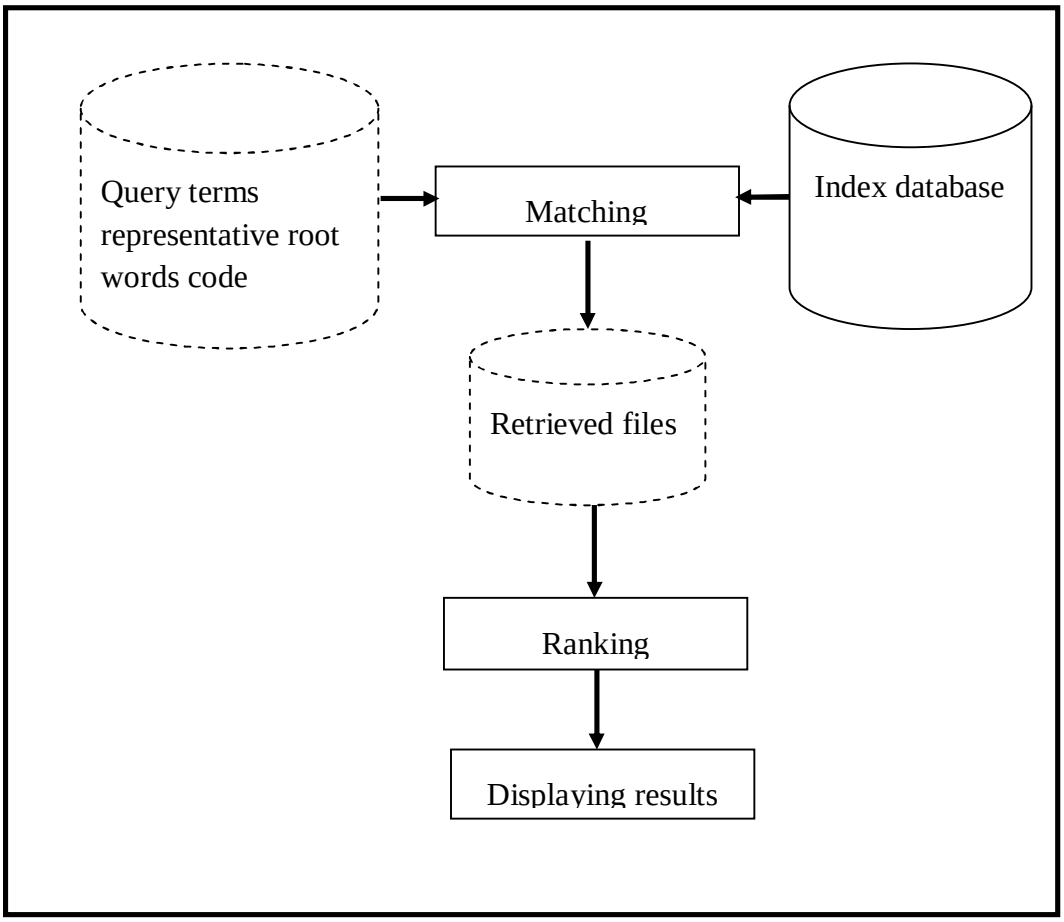


Figure 4.6*: Searching flow diagram*

```
1. Input: processed query, template feature and index file database.
2. Output: document image.
3. For p=1 to total_number_of_query _words do
      a. Calculate feature values using one of feature extraction
         algorithms;
      b. Write feature values in “some file”;
4. End for
5. Read “some file”.
6. For i=1 to  total _number_ of_ query_ feature _values do
      a. For j=1 to total _number_of_template_features do
            i. Compute similarity between selected query word feature
               and selected template feature  using equation 4.1;
           ii. If similarity >=ð then
          iii. Write root code to temporary query terms database.
           iv. end if
      b. end for
7. end for
8. Read temporary query terms database.
9. for i=1 to total_number_of_returned_query_root_words  do
      a. for j=1 to length _of_ index_ file do
            i. Search returned query root word in index database and
               put in a temporary database.
      b. End  for
10. End for
11. Rank the contents of the vector using equation 4.2.
12. Display ranked results to the user.
```

Listing 4.13*: Query processing and searching algorithm*

4.7 Matching

Matching is one of the important components next to query formulation and document characterization. It measures the strength of resemblance between feature of query terms and each template feature in feature corpus at the time of query processing and also at the time of index construction. In this study rather than segmenting word into characters and matching words based on consecutive characters, words are considered as the smallest unit for matching. Resizing and aligning two candidate word images with respect to each other and then compares their feature values have been applied in this research work. The similarity between two terms is computed by applying one of the matching techniques that is cosine. It is expressed numerically. Based on this numerical value index database has been constructed. Furthermore, once the user specify his/her query, the Amharic Document Image Retrieval System retrieves document images which are best approximate to the user query from the database. The vector space model is employed as retrieval model and the degree of similarity between two vectors that is query and document is measured by taking into account the cosine/dot product similarity measure. For two vectors $\vec{a}$ and $\vec{b}$, cosine similarity measure can be calculates as :

$$\text{Sim}(\vec{a},\vec{b}) = \frac{\sum_{i=0}^{n-1}(ai.bi)}{\sqrt{\sum_{i=0}^{n-1}\text{ai}\text{\^{}}2}.\sqrt{\sum_{i=0}^{n-1}\text{bi}\text{\^{}}2}} \quad \text{.....Equation 4.1}$$

Where $\vec{a}$ =(a1,a2,a3,a4,…) is the document vector of the query image a and $\vec{b}$ =(b1,b2,b3,…) is the document vector of the image b. n is the length of features.

If the similarity score of the two word images is greater than a certain specified value, then we assign them in the same word class.

```
1. Input: feature values of two word images.
2. Output: numerical Similarity value between two word
   images.
3. For i=1 to width_of_feature_value do
      a. Compute cosine similarity by using equation  4.1
      b. If similarity>  ð   then
      c. The two words are similar in content;
      d. Else
      e. The words are dissimilar;
      f. End if
4. End for
```

Listing 4.14: Cosine similarity measure

4.8 Ranking

The system attempts to rank the retrieved documents using some measure of relevance. The ranking algorithm we are using in this work is term frequency. It can be calculated and normalized as follow:

$$\text{Tf(t,d)}=\frac{n(t,d)}{\sum_k n(k,d)} \quad \text{.....Equation 4.2}$$

Where n(t,d) is the number of times term t occurs in document d and $\sum_k n(k,d)$ is the total number of words in the document.

First the input is processed. Then, it will be looked up in an index table stored in the database. If the query is constructed from more than a word, then the results of search in index table will be

merged. Then, the system will compute a numeric score to every document by counting terms in every document and rank documents by this score.

CHAPTER FIVE

EXPERMENTS

5.1 Introduction

This chapter describes an experiment with the retrieval prototype. The experiment aims at evaluating the retrieval effectiveness of our approach. To evaluate our proposed system model an experimental retrieval system is implemented. Four groups of real-life document images in different font type (VG 2000 main, Visual Geez Unicode, Geez-1), style (normal, bold), size (8, 10,12, 16, 20) and image qualities (clean and noisy)are used to test the validity of our method. A set of query word images have been also used to test the ability of retrieving relevant documents.

5.2 Resources

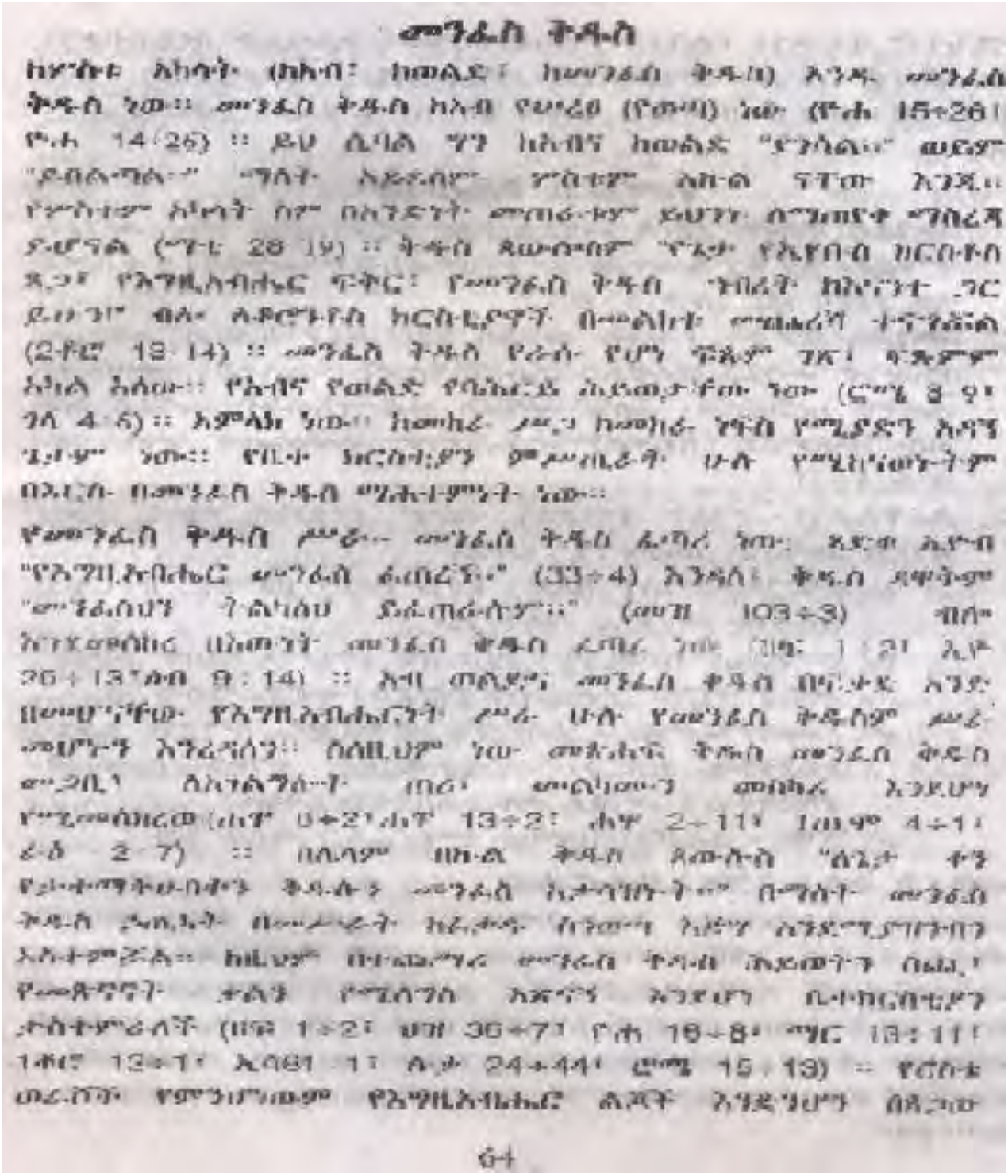
5.2.1 Experimental Set Up

The experiment has been done on two laptops with specifications of processor of Pentium(R) dual-core CPU 2.0 GHz, 2 GB RAM & processor of Intel(R) Celeron(R) CPU 2.20 GHz, 2 GB RAM and desktop computers with specifications processor of Intel(R) core 2 duo CPU 3.0 GHz, 2 GB RAM. All programming languages listed under section 1.7.4 have been installed on these machines to develop the prototype of our system.

5.2.2 Data Sets

In this study**,** the experiment is conducted by using four groups of Amharic document images collection (printouts, magazines, newspapers and books) written in different font type, size and style and set of queries. All document images are a grayscale format (see appendix II). On the other hand 32,020 template word images are presented in the lexicon for indexing and searching purposes. In our model the Amharic word synthesizer is used to build most of these words. The images of these words are grouped in clusters containing similar words. This study has used a database that comprises scanned Amharic documents. The system has been tested on those Amharic words database and scanned documents.

5.3 Binarization



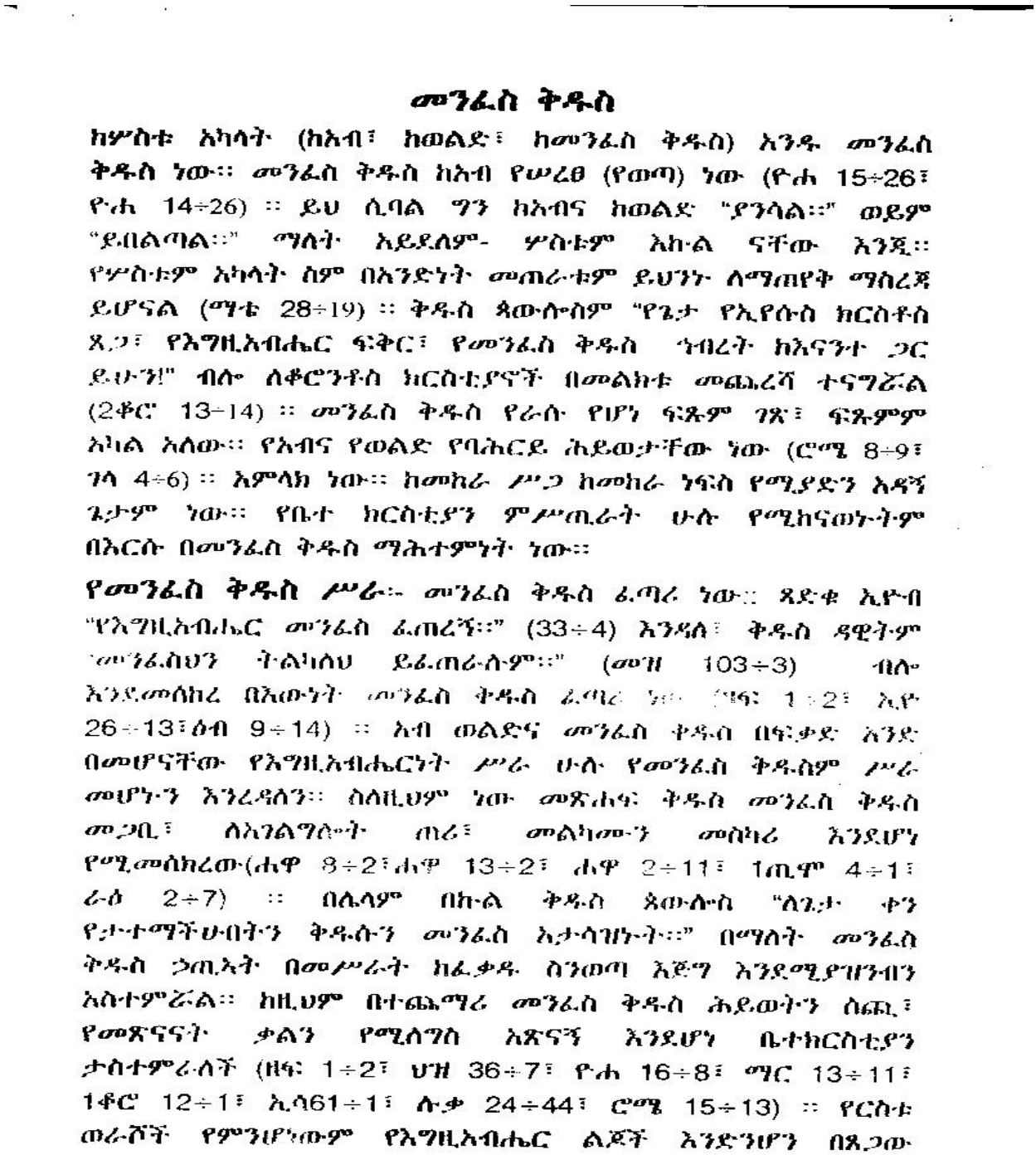
መንፈስ ቅዱስ

ከሦስቱ አካላት (ከአብ፣ ከወልድ፣ ከመንፈስ ቅዱስ) አንዱ መንፈስ ቅዱስ ነው። መንፈስ ቅዱስ ከአብ የሠረፀ (የመጣ) ነው (ዮሐ 15፡26፣ ዮሐ 14፡26)። ይህ ሲባል ግን ከአብና ከወልድ "ያንሳል" ወይም "ይበልጣል" ማለት አይደለም ምስጢሩ ለክብ ፍጥረት እንጂ። የምስጢሩ አነሳስ በዮ በአንድነት መጠራታቸው ይህንኑ ስንመለከቅ ግልፅ ይሆናል (ማቴ 28 19)። ቅዱስ ጳውሎስም "ጥንት የእየበብ ከርቤዎስ ጸጋ፣ የእግዚአብሔር ጥቅር፣ የመንፈስ ቅዱስ ኅብረት ከእናንተ ጋር ይሁን" ብሎ ለቆሮንቶስ ክርስቲያኖች በመልዕክቱ መጨረሻ ተናግሯል (2ቆሮ 13 14)። መንፈስ ቅዱስ የራሱ የሆነ ጥልቅ ገጽ፣ ባሕርይ አለው። የአብና የወልድ የባሕርይ ስያሜታቸው ነው (ሮሜ 8 9፣ ገላ 4 6)። አምላክ ነው። ከመከራ ያሥጋ ከመከራ ነፃነት የሚያድን አማኝ ኢያሣም ነው። የቤተ ክርስቲያን ምሥረቤት ሁሉ የሚከናወኑትም በእርሱ በመንፈስ ቅዱስ ሚሕተምነት ነው።

የመንፈስ ቅዱስ ሥራ መንፈስ ቅዱስ ፈጣሪ ነው፤ ጸድቀ አምላ "የእግዚአብሔር መንፈስ ፈጠረኝ" (33፡4) እንዳለ፣ ቅዱስ ጳውሎስም "መንፈሱህን ትልካለህ ይፈጠራሉ" (መዝ 103፡3) ብሎ አስተምሯል በአጠቃላይ መንፈስ ቅዱስ ፈጣሪ ነው (ዘፍ 1 1፣ ኢዮ 26፡13 መክ 9 14)። እጣ ጠልያን መንፈስ ቅዱስ በቅዱሳኑ አንድ በመሆናቸው የእግዚአብሔርንት ሥራ ሁሉ የመንፈስ ቅዱስም ሥራ መሆኑን አንፈሳለን። ስለዚህም ነው መጽሐፍ ቅዱስ መንፈስ ቅዱስ ሙታንን ስለማስነሣት ብርታ መስጠት እንደሆነ የሚመሰክረው (ሐዋ 8፡29፣ ሐዋ 13፡2፣ ሐዋ 2-11፣ 1ጢሞ 4፡14፣ ፊል 2 7)። በሌላም በኩል ቅዱስ ጳውሎስ "ሰጌታ ቀን የታመንማትሆኑበትን ቅዱሱን መንፈስ አታሳዝኑት" በማለት መንፈስ ቅዱስ ሥጠኑት በመሥራት ከፈጥሩት ሰንድማ አስቸጋሪ እንደሆነ አስተምሯል። ከዚህም በተጨማሪ መንፈስ ቅዱስ ሕጎችንት ሰጪ፣ የመጽናናት ምልክት የሚሰጥ አገናኝ እንደሆነ ቤተክርስቲያን ታስተምራለች (ዘፍ 1፡2፣ ዘፀ 36፡71 ዮሐ 16፡8፣ ማር 13፡11፣ 1ቆሮ 12፡1፣ ኢሳ61 1፣ ሉቃ 24፡44፣ ሮሜ 15 19)። የሮስቱ መፈሸት የምንሠጠውም የእግዚአብሔር ልጆች እንድንሆን በጸጋው

Figure 5.1*: An image sample from church book*

The following image is the binarized results of the above image (Figure 5.1).



መንፈስ ቅዱስ

ከሦስቱ አካላት (ከአብ፣ ከወልድ፣ ከመንፈስ ቅዱስ) አንዱ መንፈስ ቅዱስ ነው። መንፈስ ቅዱስ ከአብ የሠረፀ (የወጣ) ነው (ዮሐ 15÷26፣ ዮሐ 14÷26) ። ይህ ሲባል ግን ከአብና ከወልድ "ያንሳል።" ወይም "ይበልጣል።" ማለት አይደለም- ሦስቱም እኩል ናቸው አንጂ። የሦስቱም አካላት ስም በአንድነት መጠራቱም ይህንኑ ለማጠየቅ ማስረጃ ይሆናል (ማቴ 28÷19) ። ቅዱስ ጳውሎስም "የጌታ የኢየሱስ ክርስቶስ ጸጋ፣ የእግዚአብሔር ፍቅር፣ የመንፈስ ቅዱስ ኅብረት ከእናንተ ጋር ይሁን!" ብሎ ለቆሮንቶስ ክርስቲያኖች በመልክቱ መጨረሻ ተናግሯል (2ቆሮ 13÷14) ። መንፈስ ቅዱስ የራሱ የሆነ ፍጹም ገጽ፣ ፍጹምም አካል አለው። የአብና የወልድ የባሕርይ ሕይወታቸው ነው (ሮሜ 8÷9፣ ገላ 4÷6) ። አምላክ ነው። ከመከራ ሥጋ ከመከራ ነፍስ የሚያድን አዳኝ ጌታም ነው። የቤተ ክርስቲያን ምሥጢራት ሁሉ የሚከናወኑትም በእርሱ በመንፈስ ቅዱስ ማሕተምነት ነው።

የመንፈስ ቅዱስ ሥራ፡- መንፈስ ቅዱስ ፈጣሪ ነው። ጻድቁ ኢዮብ "የእግዚአብሔር መንፈስ ፈጠረኝ።" (33÷4) እንዳለ፣ ቅዱስ ዳዊትም "መንፈስህን ትልካለህ ይፈጠራሉም።" (መዝ 103÷3) ብሎ እንደመሰከረ በእውነት መንፈስ ቅዱስ ፈጣሪ ነው (ዘፍ 1÷2፣ ኢዮ 26÷13፣ ዕብ 9÷14) ። አብ ወልድና መንፈስ ቅዱስ በፍቃድ አንድ በመሆናቸው የእግዚአብሔርነት ሥራ ሁሉ የመንፈስ ቅዱስም ሥራ መሆኑን እንረዳለን። ስለዚህም ነው መጽሐፍ ቅዱስ መንፈስ ቅዱስ መጋቢ፣ ለአገልግሎት ጠሪ፣ መልካሙን መስካሪ እንደሆነ የሚመሰክረው(ሐዋ 8÷2፣ሐዋ 13÷2፣ ሐዋ 2÷11፣ 1ጢሞ 4÷1፣ ራዕ 2÷7) ። በሌላም በኩል ቅዱስ ጳውሎስ "ለጌታ ቀን የታተማችሁበትን ቅዱሱን መንፈስ አታሳዝኑት።" በማለት መንፈስ ቅዱስ ኃጢአት በመሥራት ከፈቃዱ ስንወጣ እጅግ እንደሚያዘንብን አስተምሯል። ከዚህም በተጨማሪ መንፈስ ቅዱስ ሕይወትን ስጪ፣ የመጽናናት ቃልን የሚለግስ አጽናኝ እንደሆነ ቤተክርስቲያን ታስተምራለች (ዘፍ 1÷2፣ ሀዝ 36÷7፣ ዮሐ 16÷8፣ ማር 13÷11፣ 1ቆሮ 12÷1፣ ኢሳ61÷1፣ ሉቃ 24÷44፣ ሮሜ 15÷13) ። የርስቱ መራሾች የምንሆነውም የእግዚአብሔር ልጆች እንድንሆን በጸጋው

64

Figure 5.2*: Binirized and noise removed document*

5.4 Noise Removal

One goal in image restoration is to remove the noise from the image in such a way that the original image is discernible. Noise removal is a very important preprocessing to facilitate other subsequent operations. Thresholding techniques are used to remove the majority of noise from input image. We used knowledge-based thresholding to remove isolated pixels or components having very small number of pixels in the image so that it would not be confused for word segmentation. The filtering function we have been using for noisy documents in this work removes all connected components/objects that have fewer than the specified number of pixels from binary image. In the following tables (Table 5.1&Table 5.2) the 1st, 2nd, 3rd and 4th columns show word starts, word ends, top rows and bottom rows of each word, respectively, before and after noise removal.

Table 5.1*: Word limits in a document before noise removal*

320	324	2833	2833
366	374	2833	2833
388	390	2833	2833
392	396	2833	2833
461	470	2833	2833
482	491	2833	2833
566	575	2833	2833
590	599	2833	2833
636	644	2833	2833
659	667	2833	2833
691	693	2833	2833
695	698	2833	2833
722	727	2833	2833
748	754	2833	2833
765	770	2833	2833
791	796	2833	2833
805	841	2833	2833
881	890	2833	2833
939	948	2833	2833
991	999	2833	2833
1073	1076	2833	2833
1078	1081	2833	2833
1106	1114	2833	2833
1127	1135	2833	2833
1172	1181	2833	2833
1211	1221	2833	2833
1268	1276	2833	2833
1302	1312	2833	2833
1324	1332	2833	2833
1341	1350	2833	2833

Table 5.2*: Word limits after noise removal*

Word start	word end	top row	bottom row
189	568	130	192
606	1001	130	192
1033	1188	130	192
1221	1452	130	192
1484	1693	130	192
1727	1819	130	192
189	567	298	363
605	1000	298	363
1033	1187	298	363
1221	1486	298	363
1518	1698	298	363
1732	1804	298	363
1840	1930	298	363
1968	2123	298	363
188	405	384	445
438	669	384	445
702	910	384	445
943	1060	384	445
1083	1092	384	445
1132	1290	384	445
1322	1612	384	445
1644	1835	384	445
1868	2021	384	445
2053	2268	384	445
188	425	466	528
459	715	466	528

5.5 Segmentation

5.5.1 Line Segmentation

Line segmentation is one of the important phases in our system. In line segmentation the input image of the document is segmented into lines of text. It is achieved by using projection profile that is horizontal projection. The efficiency of our proposed line segmentation algorithm has been evaluated on our data set. From the accuracy of our line segmentation algorithm 100 % performance is obtained from well noise removed documents.

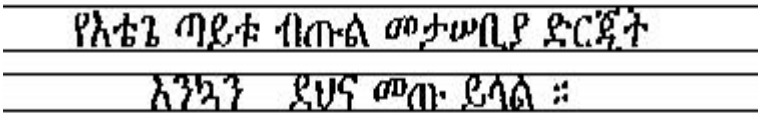


Figure 5.3*: Text lines detection demonstration*

5.5.2 Word Segmentation

In our system word is the basic unit. So word image segmentation is another important operation and this has been done by using projection profile that is vertical projection. For clean documents 100% performance is obtained from our word segmentation algorithm.

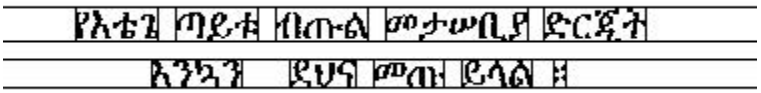


Figure 5.4*: Words segmentation demonstration*

5.6 Removing Objects

Removing some unnecessary objects from the documents has been carried out using their size.

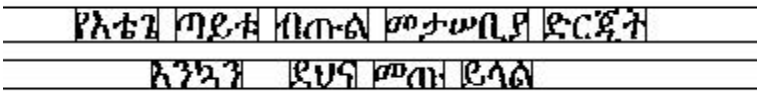


Figure 5.5*: Removing some unnecessary objects demonstration*

As shown in Figure 5.5 some objects that are not important for indexing purpose are removed from the document by using connected components analysis.

5.7 Word Image Resizing

One of the main problems in document image searching or retrieval is that font type, styles and size sensitivity. To address this problem we try to resize each word in a document to every template word during indexing. To demonstrate our resizing approach and its efficiency let us see the input and output of our resizing algorithm in Figure 5.6 and Figure 5.7.

Figure 5.6*:* Word image to be resized

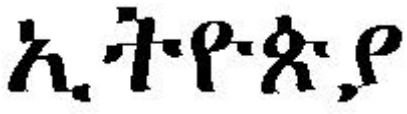


Figure 5.7*: Sample data for resizing purpose*

The height and width of the word image in Figure 5.6 is 78 pixel and 389 pixels, respectively, before resizing. On the other hand the height and width of word image of Figure 6.7 is 46 and 195 pixel, respectively. After resizing by width the height and width of the resized word image has been 40 and 195 pixels, respectively (see Figure 5.9). After resizing by height the height and width of the resized word image has been 46 and 230 pixels, respectively (see Figure 5.10). Form these we can conclude our resizing approach tries to handle font size variations.

Figure 5.8*: Font 20 cropped original word image*

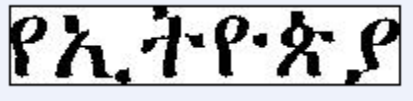


Figure 5.9*: Font 20 word image Resized by width*

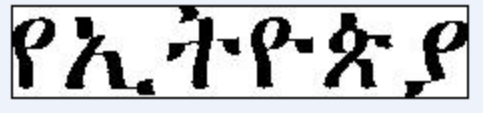


Figure 5.10*: Word image resized by height*

The height and width of the word image in Figure 5.11 is 48 pixel and 235 pixels, respectively before resizing. On the other hand the height and width of word image of Figure 5.12 (sample data) is 46 and 195 pixel, respectively. After resizing by width the height and width of the resized word image has been 40 and 195 pixels, respectively, (see Figure 5.13). After resizing by height the height and width of the resized word image has been 47 and 227 pixels, respectively (see Figure 5.12). Form these we can conclude our resizing algorithm tries to handle font style variations.

Figure 5.11*:Bold original word image*

Figure 5.12*: Bold word image resized by height*

Figure 5.13*: Word image resized by width*

From these pictures we can say that the object is closely related to the database word object but not exactly.

5.8 Stop Words Removal

In these experiments, stop words are removed after the template words features lookup. Since those words are not important for document representation, we try to remove them from index file. On the query side, first the numbers of Amharic words in the given query are reduced by using feature corpus file if it holds stop words. Each word image feature is matched against the entries in the feature corpus that is produced by Amharic word synthesizer. Our system has a good performance in removing stop words from our dataset.

5.9 Results

In order to evaluate the experimental results we have used precision, recall and fault measures. Precision measures how well the system discards irrelevant results while retrieving where as recall measures how good is the system at finding relevant documents. If A is the set of relevant document images for a given query, and B is the set of retrieved document image then:

$$\text{Precision (P)} = \frac{|A \cap B|}{|B|} * 100\% \ldots\ldots\ldots\ldots\ldots\ldots \textit{Equation 5.2}$$

$$\text{Recall(R)} = \frac{|A \cap B|}{A} * 100\% \ldots\ldots\ldots\ldots\ldots\ldots\ldots \textit{Equation 5.3}$$

$$\text{Fault (F)} = \frac{2RP}{R+P} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textit{Equation 5.4}$$

Our system was tested on Amharic documents image files from various sources covering different fields and for diverse queries. Table 5.3 shows the results recorded from 9 selected queries in document images database.

Table 5.3*: Test results from vertical projection, upper and lower word profiles.*

Query word Image	**Lower word profile**			**Upper word profile**			**Vertical projection**		
	Threshold >=0.86			**Threshold>0.83**			**Threshold>=0.90**		
	R	P	F	R	P	F	R	P	F
yXS‰x¤LÂ FLs«¤M WgþÃ	75	100	85.7	100	100	100	100	75	85.7
y›IM êNÅ	100	100	100	50	50	50	100	75	85.7
yñv¤L >L¥T	100	75	85.7	75	75	75	100	100	100
G_MÂ z¤¥	100	100	100	75	75	75	100	100	100
÷Nì ¸nþyM	100	100	100	100	50	66.7	100	100	100
_ÂTÂ MRMR	100	100	100	75	75	75	100	60	75
mNfS QÇS	100	100	100	100	100	100	100	100	100
yxþT×eÃ xyR mNgD	100	75	85.7	100	100	100	100	100	100
yZÂB ý¦ x«ÝqM	75	75	75	50	50	50	100	60	75
Average	94.4	91.7	92.5	72	66.7	76.9	100	85.6	82.9

5.10 Discussions

In this section analysis of the experimental results and comparison between our system and previous research works are given. Table 5.3 depicts search results recorded for some queries. This table shows different experimental results achieved using 3 word features which are vertical projection, upper and lower word profile. The vertical projection of the word image is the total number of black pixels in each column of a word starting left border to right border of a word image. The numbers of features are exactly the same as width of a word image. Using this feature the similarity between two identical word images is greater than or equal to 0.90. Upper and lower word profiles are the upper and lower word shape descriptors, respectively. Like vertical projection, the numbers of feature values are the same as word width. The similarity between two identical word images is greater than or equal to 0.83 for upper bound profile and 0.86 for lower bound profile. The degradation in recalling (R) performance comes due to various reasons such as:

- The quality of image is poor. Some documents are degraded so that characters are not normal or they are broken and could not be recognized.
- The presences of a given word image in various ways .For instance, the words ሙዚ, ሙ ሄ, ሙኗ, ሚዚ ,ሚሄ, ሚኗ , ¥HbR, ¥ÞbR, ¥ጎbR, etc. Since our database does not hold all various forms of a word some document words are not recognized.

On the other hand, degradation (P) in precession comes due to some wrongly recognition of word images and the existence of some query words in non-relevant documents.

As a conclusion the experimental results show that vertical projection has higher recognition power than lower word profile which has higher recognition power than upper bound profile.

It is possible to observe the Statistics results from Table5.3 and compare these results to some previous works. The overall system performance is better than Abrahm's and Mesfin's works this is due to the integration of Amharic synthesizer and resizing each document word image into each template word image to handle font variations.

CHAPTER SIX

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

Document images have become a popular information sources in our modern society. Due to that information retrieval from document images database is one of the important research areas in managing document images. In this paper, Amharic Document Image Retrieval System has been developed by using Natural Language Processing concepts without character recognition. The proposed document images retrieval techniques depend heavily on the performance of the Amharic words synthesizer. The proposed system model browses a large set of scanned Amharic documents without any prior knowledge on their contents and also presents a new approach for indexing document images. The proposed method has been designed for documents with different fonts.

The main contributions of this paper are: First, integrating Amharic words synthesizer to our system. It allows tackling problems in index file creation and query processing for morphological rich languages. Second, our system trying to handle variations in font types, sizes and styles by resizing to candidate word size. Third, it enables users searching using multiple words. This enables users to be free for writing their query and also increase system performance.

In literature even though there are many word recognition features, our system reorganize word image using three features (vertical projection, upper and lower bound profiles of the word image). These features are selected after doing empirical experiments. Finally, cosine similarity is used to construct index file and to retrieve relevant documents by computing the similarity between the query and index file.

In this work several experiments have been carried out. Experimental results show that the proposed strategies achieve a promising performance for searching relevant documents. Even though convenient and promising results have been achieved on system accuracy, our approach could not improve indexing and retrieval time. Taking into account the low quality of the documents considered in this experiment we believe that the proposed approach is very

promising for the historical document image indexing and retrieval if advance noise removal and image restoration techniques are applied. However, a lot work is needed to improve the time required for building index file.

6.2 Recommendations

The results of this research indicate the need for an efficient Amharic word synthesizer tool for developing efficient Amharic document images search engine. Therefore, much more have to be done on Amharic words generator to improve the efficiency of Amharic Document Image Search Engine for all situations (different font sizes, styles, and types, noisy and very old documents). On top of that the following are some of the areas identified in this research for future work.

- Words with similar meaning might appear in different formats. Integrating a system that can produce synonyms of a given Amharic word might increase the terms frequency in a document. This leads to increase the efficiency of the system.

- If a computational linguistic tool such as Part-Of-Speech tagger for Amharic is used in this system, the performance of the system will be increased and also the size of index terms can be decreased.

- Words (proper names and its inherited words), numbers, some symbols and phrases that are not found in the word image lexicon are not recognized and are not found in index file. Since it is not possible to list all proper nouns and numerical data, it is better to apply some techniques such as prefix and suffix removal before matching.

- Punctuation marks are not removed at the time of index construction. So this affects the performance of the system. Thus removing them might increase the performance of the system to some extent.

- Documents with diverse contents are not applied in this research.

- Using dynamic time warping (DTW) might make system performance more accurate.

- A lot should have been done to improve the speed of indexing and searching.

- In this research work only one feature value is used for matching at a time. Using a combination of two or more feature values might increase the performance of the system in recognizing word image.

REFERENCES

[1] Abreham Gebresadik (2010). *Searching in Amharic Document Image Corpus*: Master's thesis, Department of Information Science, Addis Ababa University, Ethiopia.

[2] Jukka Perki, Antti Tuominen and perti Myllmaki (n.d.). *Image Similarity: From Syntax to Weak Semantics using Multimodal Features with Application to Multimedia Retrieval.* Helsinki Institute for Information Technology, Department of Computer Science, university of Helsinki, Finland.

[3] Toni M. Rath, Victor Lavrenko and R. Manmatha (n.d). *A Statistical Approach to Retrieving Historical Manuscript Images without Recognition.*Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, MA 01002.

[4] Amharic Ethiopia Language.

Available on http://www.free-press-release.com/news/00907/1248234344.html,accessed on August 23, 2010.

[5] A. F. Smeaton (1992). *Progress in the application of natural language processing to information retrieval tasks.* The Computer Journal, Special Issue, 36(3):268–278.

[6] A. F. Smeaton (1995). *Natural Language Processing and Information Retrieval*. Tutorial notes presented at the Second European School in Information Retrieval (ESSIR 95), Glasgow, Scotland.

[7] Alen F.Smeaton (1997). *Using NLP or NLP resources for information retrieval tasks*. Dublin, city Universtiy, Glasnevin, Dublin 9, Ireland.

[8] Alexander Gelbukh (2007). *Advance in natural language processing and application*, Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexco.

[9] Amit Singhal (2001). *Modern Information Retrieval*: A Brief Overview**.** Google, Inc.

[10] Anurag Bhardwaj, Srirangaraja Setlur and Venu Govindar (n.d). *Keyword Spotting Techniques for Sanskrit Documents*. Center for Unified Biometrics and Sensors, Department of Computer Science and Engineering, University at Buffalo, Amherst NY – 14228.

[11] Anand Kumar, C.V.Jawarhar and R.Manmatha (2007). *Efficient Search in Document Image Collections*. Center for Visual Information Technology, International Institute of Information Technology, Hyderabad, India – 500032, Part I, LNCS 4843, pp. 586–595.

[12] Arne Andersson and Stefan Nilsson (1995). *Efficient Implementation of Suffix Trees*. Department of Computer Science, Lund University, Box 118, S-221 00 Lund, Sweden. Software practice and experience, VOL. 25(2), pp.129–141.

[13] Arne Andersson, N. Jesper Larsson and Kurt Swanson (n.d). *Suffix Trees on Words*. Department of Computer Science, Lund University, Box 118, S-221 00 LUND, Sweden.

[14] Atelach Alemu and Laser Asker (2007). *An Amharic stemmer: Reducing words to their citation forms*. Department of computer and Systems Science,Stockholm/KTH, Swden**.**

[15] Atelach Alemu and Lars Asker (2006). *Amharic-English Information Retrieval*. Department of Computer and Systems Sciences, Stockholm University/KTH.

[16] A.T.Arampatzis,T.Tsoris and C.H.A.Koster (1998). *Phrase-based Information Retrieval*; Information processing and management. Computer Science Institute Nijmegen, Faculity of Mathematics and Informatics, Catholic University of Nijmegen, Toeroonivedl, Netherland, vol.34,no.6 pp.693-707.

[17] Barbel Ripplinger (n.d). *The use of NLP techniques in CLIR*. IAI, Martine luther-str.14,66111 saarbrucken, Germenay.

[18] Bender et al (1976). The *Ethiopian writing System: Language in Ethiopia*: Oxford University press, London.

[19] Bjorn Gamback ,Fredrik Olsson, Atlach Alemu and Lars Asker (2009).*Methods for Amharic Part-of-Speech Tagging*. Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 104–111, Athens, Greece.

[20] Bronislave Frid, Liza Logounova, Alexander Michailov,Olega Nusinzon and Leonid Zeltser (1997). *High precision Information Retrieval with Natural Language Processing Techniques*. A project report submitted in partial fulfillment of the requirements for the course CSE 401.

[21] Byurhan Hyusein and Ahmed Patel (2003). *Web Document Indexing and Retrieval*. Computer Networks and Distributed Systems Research Group, Department of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland.

[22] Christopher D.Manning (2008). *Introduction to Information Retrieval*. Cambridge,

University press,USA.

[23] Ed Greengrass (2000). *Information Retrieval*; Survey.

[24] Chew Lim Tan et al (n.d). *Text Retrieval from Document Images based on Word Shape Analysis***.** School of Computing, National University of Singapore 3 Science Drive 2, Singapore 117543.

[25] D. Doermann (1998). *The Indexing and Retrieval of Document Images*: Survey, language and media processing laboratory center for automation research, University of Maryland. Computer vision and understanding, vol.70, No.3, pp.287-298, article no. IV980692.

[26] Misganu Debella Gilo and Andreas Kääb(2011). *Sub-pixel precision image matching for measuring surface displacements on mass movements using normalized cross-correlation.* Remote sensing environment, *115, pp.130-42.*

[27] Daniel Jurafsky and James H. Martin (2006). *Speech and Language Processing*: An introduction to natural language processing, computational linguistics, and speech recognition.

[28] Daniel Yacob (2005). *Developments towards an Electronic Amharic corpus*. Geez Frontier Foundation,7802 Solomon Seal Dr,Springfield, VA 22152,USA.

[29] Daniel Yacob (2006). *Application of the Double Metaphone Algorithm to Amharic Orthography*, International Conference of Ethiopian Studies XV.

[30] G. Salton (1989). *Automatic text processing*;the transformation,analysis and retrieval of informationby computer, Addison-Wesley.

[31] Gerald J.Kowalski and Mark T.Maybury (2002).*Information storage and retrieval systems theory and implementation*. Second Edition ,Kluwer academic publishers, New york, Boston, Dordrecht, Iondon, Moscow.

[32] Tolemariam Fufa (1964) .*A Typology of Verbal Derivation in Ethiopian Afro-Asiatic Languages.* LOT, Janskerkhof 13 , 3512 BL Utrecht, Netherland.

[33] Girmay Berhane (1992). *Word Formation in Amharic*. Journal of Ethiopian Languages and Literature. No. 2. pp. 50–75.

[34] Kibur Lisanu (2002). *Design and development of automatic Morphological for Amharic verb forms*: Master's Thesis, Department of Information Science, Addis Ababa University, Ethiopia.

[35] K.Alesandrini (1992): Survive Information Overload. Homewood, IL, USA: Business One Irwin.

[36] Karen Sparck Jones (1997). *What is the role of NLP in text retrieval*? Computer Laboratory, University of Cambridge.

[37] Huiyu zhou, Jiahua Wu and Jianguo and Zhang (2010). *Digital image processing-part one.*ASP

[38] Khurram Khurshid,Caludie and Nicole Vincent (n.d). *A novel approach for Word Spotting using Merge-Split Edit Distance*. Laboratoire CRIP5 – SIP, University Paris Descartes, 45 rue des Saints-Peres, 75006, Paris, France.

[39] Li Zhang (n.d). *A System of Web-based Document Image Retrieval for Digital Library Using Word Matching*. Department of Computer Science, School of Computing National University of Singapore, Kent Ridge, Singapore 117543.

[40] Mesfin Worku (2009). Amharic *Document Image Retrieval without Explicit Recognition*. Master's Thesis, Department of Information Science, Addis Ababa University, Ethiopia.

[41] Liddy, E. D (n.d). *In Encyclopedia of Library and Information Science*, 2nd Edition. Marcel Decker, Inc.

[42] Laude de Loupy et al (n.d). *Linguistic resources for Information Retrieval***.** Sinwqua,51-54,rue Ledru Rollin,94200,Ivry-sur-Seine,France.

[43] Michael Piotrowski (1998). *NLP-Supported Full-Text Retrieval*. Master's Thesis, Friedrich Alexander University, Institution of ermanistik, Abteilung for computer linguistic ,Erlangen, Nürnberg,.

[44] L.Zehang,Y.Lue and C.L.Tan (n.d). *A Web-based System for Document Images from Digital Labrary*. National university of Singapore,kent Ridge.

[45] Muller M. (2007). *Information retrieval for music and motion*.XVI 318 p.136 illu,

[46] Manesh B. Kokare and M.S.Shirdhonkar (2010). *Document Image Retrieval*: An Overview. International Journal of Computer Applications (0975 – 8887), Volume 1 , No. 7.

[47] C.V.Jawahar, Million Meshesha and A.Balasubramanian (n.d). *Searching in Document Images.* Center for Visual Information Technology,International Institute of Information Technology, Gachibowli, Hyderabad-500 019, India.

[48] Million Meshesa (2000). *A general Approach to Optical Character Recognation (OCR) of Amharic Texts*: Master's Thesis, Department of Information Science , Addis Ababa University, Ethiopia.

[49] Mark T. Maybury (1997). Intelligent *Multimedia Information Retrieval,* AAAI press/The MIT.

[50] Nega Alemayehu and Peter Willett (n.d). *The effectiveness of stemming for information retrieval in Amharic*. Department of Information Studies, University of Shef.eld, UK.

[51] Nuno Filipe Escudeiro and Alípio Mário Jorge (2008). *Satisfying Information needs on the Web*: a Survey of Web Information Retrieval. Revista de Estudos Politécnicos, Polytechnical Studies Review, Vol VI, No 9.

[52] Kurt Alfred Kluever (2008). Independent Study Report Character Segmentation and Classification. Golisano College of Computing and Information Sciences, Rochester Institute of Technology, Department of Computer Science.

[53] Lars Asker,Atlach Alemu,Bjorn Gamback and Magnus Sahlgren (n.d). *Applying Machine Learning to Amharic Text Classification*. Stockholm University1 and Swedish Institute of Computer Science.

[54] P.Ingwersen (2002**).** *Information Retrieval Interaction.* Royal School of Library and Information Science, Department of Information Studies, Birketingeting 6,Dk 2300 Copenhagen S Denmark.

[55] M.F Porter (1980). *An algorithm for suffix striping*. Computer Laboratory, Cambridge, UK. Vol. 14, no.3, pp.130-7.

[56] Peter Jackson and Isbelle Moulinier (1984). *Natural Language Processing for Online Applications Text retrieval, Extraction and Categorization*. Welverhampton WV1 ISB, United Kingdom.

[57] R.Suresh et al (2010). *Intelligent Layout Based Retrieval from Document Images*. Proceedings of the Int. Conf. on Information Science and Applications ICISA, Chenna, India.

[58] Saba Amsalu and Sissay Fissaha Adafre (n.d). *Machine translation for Amharic*: where we are. University of Amsterdam, Informatics Institution.

[59] Saba Amsalu and Girma A.Demeke (2006). *Non-concatinative Finite-State Morphotactics of Amharic Simple Verbs.* ELRC Working Papers Vol. 2, number 3.

[60] Saba Amsalu and Dafydd Gibbon (n.d). *Finite State Morphology of Amharic*. university at Bielefeld, university at Strasse 25 D-33501, Faculty of linguistic and Literaturwissenschaft, Germany.

[61] Sebsibe H/Mariam, S P Kishore, Alan W Black, Rohit Kumar and Rajeev Salgan (n.d). *Unit selection voice for Amharic using festvox*. Language Technologies Research Center, International Institute of Information Technology, Hyderabad; Language Technologies Institute, Carnegie Mellon. University and Institute for Software Research International, Carnegie Mellon University. 5th ISCA Speech Synthesis Workshop , Pittsburgh.

[62] Shijian Lu (2008). *Document Image Retrieval through Word Shape Coding*. IEEE transactions on pattern analysis and machine intelligence, VOL. 30, NO. 11.

[63] Simone Marinai, Emanuele Marino and Giovanni et al (n.d). *Exploring Digital Libraries with Document Image Retrieval.* Department of Informatics, University of di Firenze, Via S.Marta, 3 - 50139 Firenze , Italy.

[64] Sisay Fissaha Adafre (n.d). *Formal Analysis of Some Aspects of Amharic Noun Phrases*. Institute for Logic, Language and Computation, University of Amsterdam, Nieuwe Achtergracht 166, 1018 WV Amsterdam, the Netherlands.

[65] R. J. Ramteke and Imran Khan Pathan etal (2011). Skew Angle Estimation of Urdu Document Images: A Moment Based Approach. International Journal of Machine Learning and Computing, Vol.1, No. 1, April 2011.

[66] Tamanna Sehgal, Gaurav Gupta and Virinder Kumar (2008). *NLP and its Applications.* Chitkara and RIMT Institute of Engineering & Technology, Proceedings of 2nd

National Conference on Challenges & Opportunities in Information Technology (COIT-2008) RIMT-IET, Mandi Gobindgarh.

[67] Tessema Mindaye Meron Sahlemariam Teshome Kassie (n.d)**.** *The Need for Amharic WordNet.* Computer Science Department, Addis Ababa University, Ethiopia.

[68] Computational Morphology.

Available at http://www.univie.ac.at/~harald/handbook.html. accessed on February, 2011.

[69] Y.lue and C.L.Tan (2002). *Word Spotting in Chinese Document Image Without Layout Analysis*. National University of Singapore, Kent Ridge.

[70] Yue Lu and Chew LimTan, Senior Member, IEEE (2004). *Information Retrieval in Document Image Databases*. IEEE transactionson knowledge and data engineering, vol. 16, No.11.

[71] Konstantinos Zagoris, Kavallieratou Ergina and Nikos Papamarkos (2010). *A Document Image Retrieval System*. Engineering Applications of Artificial Intelligence, doi:10.1016 or j.engappai.2010.03.002.

[72] Yaregal Assabie and Josef Bigun (2009). *HMM-Based Handwritten Amharic Word Recognition with Feature Concatenation*, 10th International Conference on Document Analysis and Recognition, School of Information Science, Computer and Electrical Engineering Halmstad University, Halmstad, Sweden.

[73] Masliana Binti Wahid (2005). *Development of Indexer Considering Tag Weighting For XML Document*. Bachelor thesis, department of information technology, Mara University of technology, Shaha Alam.

[74] Thorsten Brants (n.d). *Natural Language Processing in Information Retrieval*. Google Inc.

[75] Christophor D.Manning, Prabhakar Raghavan and Hinrich (2009*). An Introduction to Information Retrieval*. Cambridge university press, Cambridge, England.

[76] Eugen Barbu, Pierre H´eroux etal (n.d). *Clustering document images using a bag of symbols representation.*CNRS FRE 2645-University de Rouen,UFR des Sciences et Techniques,76821 Mont-Saint-Aignan cedex , France.

[77] Samir Malakar, Dheeraj Mohanta , Ram Sarka and Mita Nasipuri (n.d). *A Novel Noise-removal Technique for Document Images*. MCKV Institute of Engineering, Howrah. Jadavpur University, Department of MCA, IT and CSE, Kolkata, India.

[78] Toni M. Rath, Victor Lavrenko and R. Manmatha (n.d). *A Statistical Approach to Retrieving Historical Manuscript Images without Recognition.* Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, MA 01002.

[79] Toni M. Rath and R. Manmatha (n.d). *Word Image Matching Using Dynamic Time Warping.* Multi-Media Indexing and Retrieval Group, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, MA 01003.

[80] C. V. Jawahar, A.Balasubramanian, Million Meshesha (n.d). *Word-Level Access to Document Image Datasets***.** Center for Visual Information Technology, International Institute of Information Technology, Gachibowli, Hyderabad - 500 019.

[81] Rangachar Kasturi,Lawrence O'Gorman and Venu Gonvindaraju (2002). *Document image analysis: A primer.*Sadhana vol.27,part I,pp.3-22,India.

APPENDIX I: The Amharic Character and their Roman counter parts

	Ge'ez ä	ka'eb u	Salis ï	Rab'e a	Hamis é	Sadis i	Sab'e o
h	h	hù	£	¦	ÿ	H	ç
l	l	lù	lþ	§	l¤	L	lÖ
h	¼	¼ù	¼þ	ˆ	¼¤	Þ	‡
m	m	Ñ	¸	¥	»	M	ä
s	¿	¿ù	œþ	œ	œ¤	o	*f*
r	r	„	¶	‰	Ê	R	é
s	s	sù	sþ	ú	s¤	S	î
s	¹	¹ù	¹þ	š	¹¤	>	ë
q	q	qï	qï	Ý	qï	Q	ö
b	b	bù	bþ	Æ	b¤	B	ï
v	v	vù	vþ	Š	v¤	V	‹
t	t	tï	tï	¬	tï	T	è
c	c	cï	cï	Ò	cï	C	Ó

h	%	%‹	%›	‚	%¦	‹	‘
n	n	n¡	n¦	Â	n¤	N	ñ
n	ß	ß¡	ß¦	¾	ß¤	Ÿ	®
x	x	x¡	x¦	¨	x¤	X	å
k	k	k¡	k¦	‹	k¤	K	÷
x	¤	¤‹	¤=	¥	¤?	¦	§
w	w	ý	ê¦	ê	ê¦	W	ã
z	z	z¡	z¦	²	z¤	Z	ø
z	¢	¢¦	¢¦	Ï	¢¦	i	Î
y	y	†	‘	Ã	ü	Y	×
d	d	Ç	Ä¦	Ä	Á	D	ì
j	j	°	©¦	©	Ë	J	í
g	g	g¡	g¦	U	g¤	G	–
t	«	«¡	«¦	È	«¤	_	õ
c	=	Œ	À	Å	~	u	ô
p	’	’¡	’¦	Ô	’¤	e	Õ

s	™	™ù	™þ	Ú	™¤	A	Û
s	{	{ù	Éþ	É	É¤	I	ò
f	f	û	ð	Í	Ø	F	æ
p	p	p	p	ᵴ	P	P	±

Appendix II: sample data from different sources

በዓለም ዋንጫ ሰበብ ሥራ እንዳይስተጓጐል ማሳሰቢያ ተሰጠ

ሰሞኑን በተጀመረው የዓለም ዋንጫ እግር ኳስ ውድድር ሳቢያ ስራ እንዳይበደል የሥራ ኃላፊዎች ሠራተኞችን በሥራ ሰዓት
መቆጣጠሪያ ፊርማ ብቻ ሳይሆን በሥራ ገበታም ሊቆጣጠሯቸው እንደሚገባ የፌዴራል ሲቪል ሰርቪስ ኮሚሽን አሳሰቡ።

ኮሚሽነሩ አቶ ዓለማየሁ ኃይለማርያም ትናንት እንዳስታወቁት፤ በተለይ መንግሥት የጣለባቸውን ኃላፊነት በመተው በሥራ ሰዓት
የእግር ኳስ ጨዋታ በቴሌቪዥን በሚከታተሉ ሠራተኞች ላይ ኃላፊዎች ጥብቅ ቁጥጥር ሊያደርጉ ይገባል።

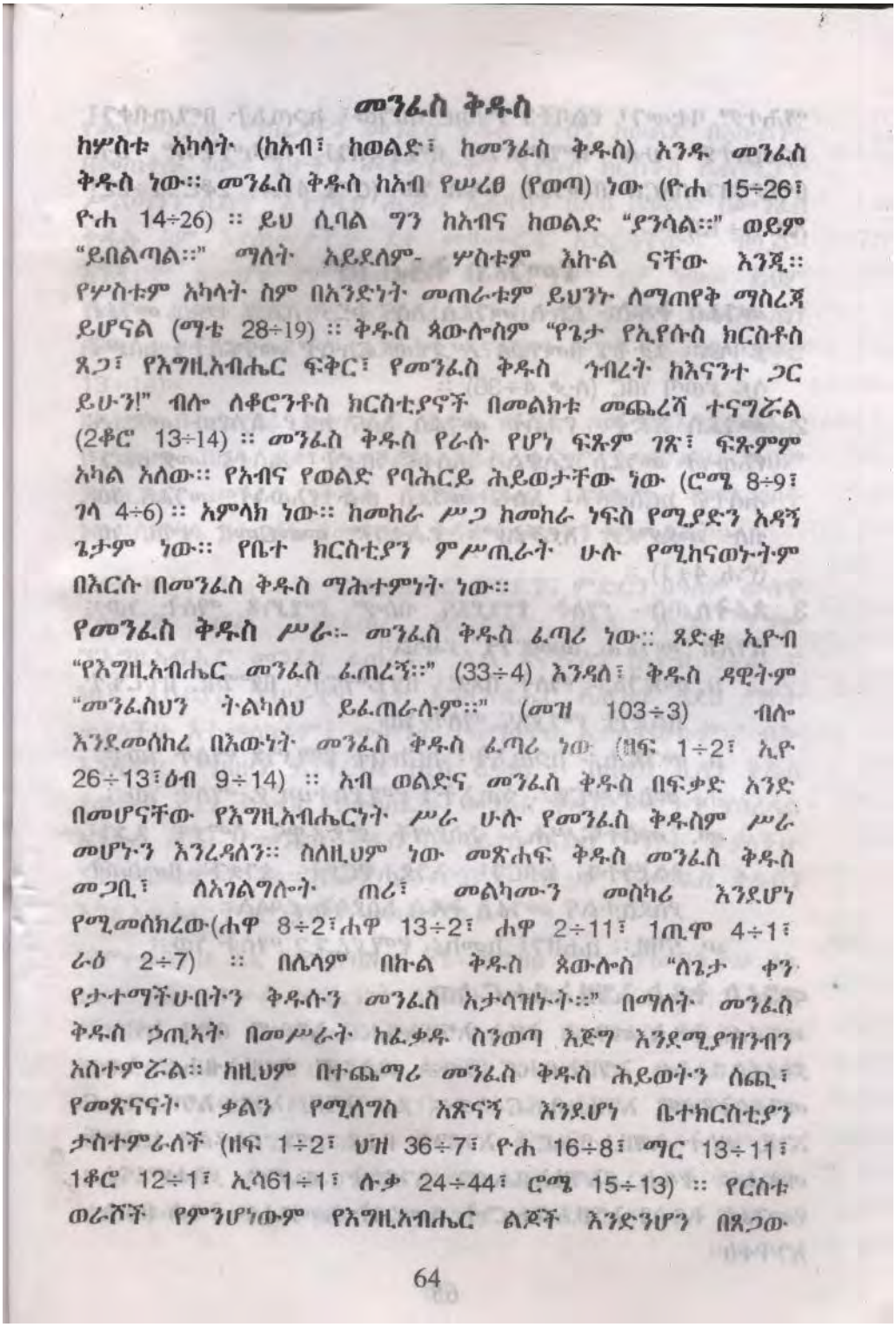
አንዳንድ የሥራ ኃላፊዎችም ቢሯቸው ባለ ቴሌቪዥን ጨዋታ እየተከታተሉ ሥራን መደበል እንደሌለባቸው አሳስበው፤ እያንዳንዱ
የመንግሥት ሠራተኛ የሕዝብ አገልጋይ መሆኑን ተገንዝቦ በአማራጭ ምሽት ላይ ጨዋታው ሲደገም መመልከት እንደሚችል
ጠቁመዋል።

ከዚህ አኳያ መንግስት ለልማት ትኩረት ሰጥቶ በሚንቀሳቀስበት በአሁኑ ወቅት ሥራን እየበደሉ ጨዋታን መከታተሉ በቀጣዮቹ
ቀናት መቀጠል እንደሌለበት ኮሚሽነሩ አሳስበዋል።

ይህ በእንዲህ እንዳለ እግር ኳሱን ለመከታተል ሥራቸው ላይ በማይገኙ ኃላፊዎች ምክንያት ባለጉዳዮች ጉዳያቸውን ለማስፈጸም
እየተጉላሉ መሆኑን አመልክተዋል።

ከአዲስ ዘመን ግንቦት 30 ቀን 1994

Data sample from News paper



መንፈስ ቅዱስ

ከሦስቱ አካላት (ከአብ፣ ከወልድ፣ ከመንፈስ ቅዱስ) አንዱ መንፈስ ቅዱስ ነው። መንፈስ ቅዱስ ከአብ የሠረፀ (የወጣ) ነው (ዮሐ 15÷26፣ ዮሐ 14÷26) ። ይህ ሲባል ግን ከአብና ከወልድ "ያንሳል።" ወይም "ይበልጣል።" ማለት አይደለም- ሦስቱም እኩል ናቸው እንጂ። የሦስቱም አካላት ስም በአንድነት መጠራቱም ይህንኑ ለማጠየቅ ማስረጃ ይሆናል (ማቴ 28÷19) ። ቅዱስ ጳውሎስም "የጌታ የኢየሱስ ክርስቶስ ጸጋ፣ የእግዚአብሔር ፍቅር፣ የመንፈስ ቅዱስ ኅብረት ከእናንተ ጋር ይሁን!" ብሎ ለቆሮንቶስ ክርስቲያኖች በመልክቱ መጨረሻ ተናግሯል (2ቆሮ 13÷14) ። መንፈስ ቅዱስ የራሱ የሆነ ፍጹም ገጽ፣ ፍጹምም አካል አለው። የአብና የወልድ የባሕርይ ሕይወታቸው ነው (ሮሜ 8÷9፣ ገላ 4÷6) ። አምላክ ነው። ከመከራ ሥጋ ከመከራ ነፍስ የሚያድን አዳኝ ጌታም ነው። የቤተ ክርስቲያን ምሥጢራት ሁሉ የሚከናወኑትም በእርሱ በመንፈስ ቅዱስ ማሕተምነት ነው።

የመንፈስ ቅዱስ ሥራ፦ መንፈስ ቅዱስ ፈጣሪ ነው። ጻድቁ ኢዮብ "የእግዚአብሔር መንፈስ ፈጠረኝ።" (33÷4) እንዳለ፣ ቅዱስ ዳዊትም "መንፈስህን ትልካለህ ይፈጠራሉም።" (መዝ 103÷3) ብሎ እንደመሰከረ በእውነት መንፈስ ቅዱስ ፈጣሪ ነው (ዘፍ 1÷2፣ ኢዮ 26÷13፣ ዕብ 9÷14) ። አብ ወልድና መንፈስ ቅዱስ በፍቃድ አንድ በመሆናቸው የእግዚአብሔርነት ሥራ ሁሉ የመንፈስ ቅዱስም ሥራ መሆኑን እንረዳለን። ስለዚህም ነው መጽሐፍ ቅዱስ መንፈስ ቅዱስ መጋቢ፣ ለአገልግሎት ጠሪ፣ መልካሙን መስካሪ እንደሆነ የሚመሰክረው(ሐዋ 8÷2፣ሐዋ 13÷2፣ ሐዋ 2÷11፣ 1ጢሞ 4÷1፣ ራዕ 2÷7) ። በሌላም በኩል ቅዱስ ጳውሎስ "ለጌታ ቀን የታተማችሁበትን ቅዱሱን መንፈስ አታሳዝኑት።" በማለት መንፈስ ቅዱስ ኃጢአት በመሥራት ከፈቃዱ ስንወጣ እጅግ እንደሚያዝንብን አስተምሯል። ከዚህም በተጨማሪ መንፈስ ቅዱስ ሕይወትን ሰጪ፣ የመጽናናት ቃልን የሚለግስ አጽናኝ እንደሆነ ቤተክርስቲያን ታስተምራለች (ዘፍ 1÷2፣ ህዝ 36÷7፣ ዮሐ 16÷8፣ ማር 13÷11፣ 1ቆሮ 12÷1፣ ኢሳ61÷1፣ ሉቃ 24÷44፣ ሮሜ 15÷13) ። የርስቱ ወራሾች የምንሆነውም የእግዚአብሔር ልጆች እንድንሆን በጸጋው

64

Data samples from book

አዲስ አበባን እኔ ደላላውና ቢሮክራሲው ተረባርበን ለውጠናታል። የቢሮክራሲው በዚህም፣ መወደስና መሞገስ እንጂ መታማት የለብንም። ውጤቱን እንጂ አፈጻጸሙን ረሱት። ስለአፈጻጸም የሚጨነቁ ሰዎች ያናድዱኛል። ቢሮክራሲው ከደላሎች ጋር ተባብሮ መስራቱ የታለመውን እቅድ እንዲመታ አድርጎታል። ደላሎች ለልማት ዋነኛው እንዝርቶች መሆናቸውን መገንዘብ በራሱ ያስመሰግናል። ደረቅ ቢሮክራሲ፣ ደረቅ ደንብ አክባሪ ለሃገርም፣ ለሕዝብም፣ ለመንግሥትም አይበጅም። በልቶ የሚያበላ፣ ተለውጦ የሚለውጥ የሆነ እንደሆነ እድገት አለ። ለዝርዝር ሕጎች የሚንቀጠቀጥና የሚያረብድ ቢሮክራሲ ግን መንግሥቱን ያስወቅሳል።

..

ሕዝብ ምን ተሰራ እንጂ እንዴት ተሰራ አይልም። እንዴት ተሰራ የግለሰቦች ጭንቀት ነው። የሕዝብን አጀንዳ ይዞ የግለሰብ ስሞታ ለማዳመጥ ደግሞ ጊዜ የለም። ስለዚህ ስለጉቦ መጥፎነት እየተረከ፣ ስለ ዝርዝር ሕግ እየተነተነ፣ ስለ መግባባት እያጣጣለ ቁጭ ሲል፣ የሾመው መንግሥት ተናዶ ይዘረዋጠዋል። ያንጊዜ መግቢያና መውጪያውን ሲያጣ የናቀውን ደላላ ..አምበርብር ምን ይሻላል?.. ቢለኝ ዋጋ የለውም። ..ሲሾም ያልበላ ሲሻር ይቆጨዋል.. የሚባለውን ተረት የሚንቅ ከተረቶች ለመማር የማይፈልግ መሆን አለበት። እንደ መጽሐፉም የሚለው እኮ ..የተሰጠህን አብዛበት.. ነው። አንድ ጌታ ለእሽክሮቹ ብር ሰጥቶ በዓመቱ ቢጠይቃቸው ገንዘቡን ሳይነካ ያቆየውን እንዴት እንደነቀፈው አልሰማችሁምን።

..

በአዲስ አበባ ግንባታ ብዙ ለፍቼበታለሁ። ከሊዝ ቦርድ እስከ ክፍለ ከተማ ድረስ ..ቀናውን መንገድ.. በመረዳታችንና በማወቃችን፣ ግንባታው እንደምታዩት ውጤቱ ምስክር ሆኗል። ..ቦታው እንዴት ተገኘ..፣ ..የሊዝ ቦርድ አባላት ሳይናሩ በማን ፊርማ ተፈቀደ..፣ ..ግምቱ ምነው አነሰ?.. የሚሉ ሥራ ፈቶች ያናድዱኛል። ስለዝርዝር ሕጎች ወደፊት ይወሰናል የሚለውን ታሳቢ በማድረግ መንቀሳቀስ አስፈላጊ መሆኑን ባለመገንዘባቸው ያሳዝናሉ። የባሰ የሚያሳዝኑኝ ባለማወቃቸው ብቻ ሳይሆን ለማወቅ አለመፈለጋቸው ነው። ..ወይ አውቀው አያውቁ፣ ወይ ሰው አይጠይቁ.. አለ አዝማሪ።

..

ዋናው ነገር ሕገመንግሥቱን ማክበር ብቻ ነው። ጠቅላይ አዲተር ባለፈው ሳምንት ያን ያህል ውርጅብኝ የወረደበት ሕገመንግሥቱን አላከበረም ተብሎ እንጂ ዝርዝር ሕጎች አልፈጸምህም፣ ተብሎ አይደለም። አምበርብር ሕገመንግሥት ያከብራል። አንጉቱን ደፍቶ፣ የሚረግጠውን ለክቶ ይራመዳል። ዝርዝር ደንቦችን ከቢሮክራሲው ጋር እየተመካከረ፣ እየቋጠረ ይፈታዋል።

..

ባለፉት ዓመታት አዲስ አበባን ለመቀየር ያደረግነው ጥረት ፍሬ ሰጥቷል። ግን መቀበል የማይፈልጉ ባለሃብቶች አልጠፉም። አንድ ቀን ከእትዬ አስቴር ግሮሰሪ ያገኘሁት፣ ዐዋቂና ባለገንዘብ ያልሁት ሰው፣ የተናገረውን ሰምቼ አፈርሁለት። ..የአዲስ አበባ ሰፋፊ ቦታ የሚሰጠው በደላላና በትውውቅ ሆኗል.. ሲል ሰማሁት። ነገሩ ሰምና ወርቅ መሆኑ ገባኝና በትህትና ጠያየቅሁት።

..

የራሴ ታሪክ ነው ብሎ ነገረኝ። ..በሊዝ ቦርድ ለሕንፃ መስሪያ ቦታ ተፈቀደልኝ። አጥሬ ካስቀመጥሁት በኋላ ልቀቅ ተብዬ በማስገደጃ ተይዥ አስለቀቀኝ። ገንዘቡን መልሱ፣ ብል እምቢ አሉኝ..... አለኝ። ሲናገር እንባ ይተናነቀዋል። ነገሩ ከነከነኝ። የት አካባቢ እንደሆነ ጠየቅሁት። ነገረኝ። ....አወቅሁት። የማውቀውን ነገር ብነግረው ያንቀኛል ወይም ያዋርደኛል ብዬ አመዛዝኜ ተውኩት። ማመዛዘንንም ..የደላላ የግል ችሎታ አድርጋችሁ ልትቆጥሩት ትችላላችሁ።

..

የሚለውን ቦታ ለከተማዋ ልማት ሲባል በደላላዊ የረቀቀ ዘዴና በቢሮክራሲው የረቀቀ ትብብርና ችሎታ ለሌላ በማዘዋወሩ ሂደት፣ ደላላው አምበርብር ምንተስኖት እጁ ያለበት መሆኑን አላወቀም። በርግጥ ወዳጄ ነበር። ግን ቢዝነስና ወዳጅነት የሰማይና የመሬት ያህል አይገናኙም። አንዳንድ ሰዎች ዶግሞ ወዳጅ ሲያደርጉ ገንዘብ መክፈል እንዳለባቸው ይረሱታል። አንዳንዶቹ ደላላ እንዲህ አይነቱን ትላልቅ ጉዳይ እንደሚፈጽም የማመን አቅም ነስቷቸዋል። በዚያ ቦታ ማለዋወጥ፣ ከታች እስከ ላይ ..የሚንቦገቦግ.. ትልቅ ገንዘብ መዘራቱን አላወቀም። ይኸ ወዳጄ ለዝርዝር ሕግ መጨነቅ እንደማይገባ ብነግረው አይገባውም።

..

..ጋሼ አጥፍተዋል፣ ለምን ለሰው አያማክሩም። ያንን የመሰለ ቦታ መልቀቅ አይገባዎትም ነበር። እኔ ወዳጅዎ ሆኜ.. ሰው እኮ ካፖርቱ ማለቁ አይታይም። አጥፍተዋል.. አልኩት። አላመንችም። ደላላው የሌላውን ብሶት እንዲሰማ እንጂ እንዲያልል የተፈጠረ ከማይመስላቸው ውስጥ የሚመደብ ነው። የእሱ ትምህርትና ገንዘብ ምን ሰራለት። እውቀቱና ፈቃደኛነቱ ..ከደላላ በታች መሆኑ ቦታውን በማስወሰዱ ተረጋግጠበት። ቢሮክራሲን ቢያማርሩት ዋጋ የለውም። መተጣጠፍ፣ ብልሀ መሆን፣ ደፋር መሆን፣ ህሊናን መርሳት፣ ለዝርዝር ሕግ አለመጨነቅ፣ እላይ ድረስ ወዳጅ መያዝ፣ ደላላን ማማከር። ይህ ነው የዕድገት ምስጢሩ። ማማረር ማማት ዋጋ የለውም። ወይም እውቀት ብቻ ዋጋ የለውም። ዳገት በአቋራጭ መንገድ ካልታለፈ አድክሞ እንደሚጎዳ አለማወቅ ደካማነት ነው።

..

..ሚሊዮን ብር ከፈልሁ እስካሁን ቦታውን አልሰጡኝም ገንዘቤም አልተመለሰም..... አለኝ አንዱ ደሃ ውስኪውን እየጠጣ። ገንዘብ ስለከፈለ ብቻ ሁሉም ነገር ይከናወንለት መስሎታል። እኔ አምበርብር ውስኪ ባልጠጣ በነጭ አረቄዬ ከሱ በላይ መስራቴን ባለማወቁ በሆዴ ሳቅሁበት። ..መቼ ነበር የከፈሉት?.. አልሁት። ..ከዓመት በፊት.. ሆዴ እንደገና ሳቀበት። ውጭ ሃገር ድረስ ሄዶ መማሩ ለዚህ ብልጠት እንኳ አልረዳውም። ደላላው ከሸገር ሳይወጣ፣ ከማንጠግቦሽ ሳይለይ የተማሩትን ይመራቸዋል። ደላላው አምበርብር አንድ ወር ባልሞላ ጊዜ ቦታውን ለመረከብ የሚያስችል አሰራር እንዳለው ማን በነገረው። እኔ የያዝኩት ጉዳይ መመላለስ አይበዛበትም፣ የብዙ የኮሚቴ አባል ፊርማ አያስፈልግም። ዋናው ሰው እጅ ካደረጉ የሌሎቹ በስልክ ያልቃል ወይም አያስፈልግም። እንደ ካዛንቺስ ያለ ጮማ ቦታ እጅ ውስጥ ማስገባት ቀላል መሆኑን ብነግረውም አይገባውም። በጥብጠው ቢግቷቸው ከማይገባቸው ሰዎች ጋር መኖር ያስችግራል። ..

Data samples from print out

፫) የቤት ባለቤቶች ቤቶችን የያዙ ሰዎች የጋራ መጠቀሚያ ተከራዮች ይህንን አዋጅ የኮንዶሚኒየም ማሳወቂያና መግለጫው" መተዳደሪያ ደንቡንና ውስጠ ደንቡን ማክበራቸው" ማረጋገጥ፤

፬) የቤት ባለቤቶችን የጋራ ጥቅም በመወከል አስፈላጊ የሆኑ ሌሎች ተግባራትን መፈፀም፤

፲፪. የማህበር አባልነት

ማንኛውሀ የኮንዶሚኒየም ባለቤት የቤት ባለቤቶች ማህበር አባል መሆን አለበት።

፲፫. የማህበሩ ሥልጣንና ተግባር

የቤት ባለቤቶች ማህበር የሚከተሉት ሥልጣንና ተግባራት ይኖሩታል።

፩) የሕን ማሳወቂያና መግለጫ፤ መተዳደሪያ ደንብና ውስጠ ደንብ ማውጣትና ማሻሻል፤

፪) በጀት መወሰንና ማሻሻል፤

፫) የጋራ መጠቀሚያዎችን አጠቃቀም መወሰን፤

፬) ¾Ò¾Ò^ መጠቀሚያዎችን ማከራየት፤
በዋስትና ማስያዝና ማስተላለፍ፤

፭) ቅጣት፤መዋጮና የአገልግሎት ክፍያዎችን መወሰን፤

፮) ሠራተኞች መቅጠር፤ማስተዳደርና ማሰናበት፤

፯) የንB[ƒ ባለቤት መሆን፤ በዋስትና ማስያዝና ማስተላለፍ፤

፰) M መዋዋል፤ መክሰስና መከሰስ&

¾፲፬. የማህበሩ ጠቅላላ ስብሰባ ሥልጣንና ተግባር

የቤት ባለቤቶች ማህበር ጠቅላላ ስብሰባ የሚከተሉት ሥልጣንና ተግባር ይኖሩታል።

፩) የኮንዶሚኒየም ማሳወቂያና መግለጫ እንዲሁም ማሻሻያ ማጽደቅ፤

፪) ¾መተዳደሪያ ደንብና ውስጠ ደንብ መግለጫ እንዲሁም ማሻሻያዎቹን ማጽደቅ፤

፫) የዳይሬክተሮች ቦርድ አባላትን መምረጥና መሻር፤

፬) የማህበሩን የሥራ ክንውን፤ የሂሳብ ምርመራ ሪፖርቶች መስማትና ውሳኔ መስጠት፤

፭) በዚህ አዋጅ መሠረት የቤት ባለቤቶች ማህበር ከሌላ የቤት ባለቤቶች ማህበር ጋር እንዲዋሀድ ወይም ኮንዶሚኒየም በዚህ አዋጅ መሠረት መተዳደሩ እንዲያበቃ መወሰን፤

፮) ዓመታዊ የሥራ ዕቅድና በጀት ማጽደቅ፤

፯) በዳይሬክተሮች ቦርድ በሚቀርብለት ማናቸውሀ ጉዳይ ላይ ውሳኔ መስጠት።

፲፭. ¾uየቤት ባለቤቶች ማህበር ስብሰባ ¾u

፩) የቤት ባለቤቶች ማህበር ዓመታዊ ጠቅላላ ስብሰባ ይኖረዋል።

፪) ¾uየቤት ባለቤቶች ማህበር የዳይሬክተሮች ቦርድ በሚያደርገ ጥሪ መሠረት አስቸካY ስብሰባዎችን ያካሂዳል።

Data samples from Magazine