ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION STUDIES FOR AFRICA

N-GRAM-BASED AUTOMATIC INDEXING FOR AMHARIC TEXT

By

BETHLEHEM MENGISTU HAILEMARIAM

A thesis submitted to

the School of Graduate Studies of Addis Ababa University

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

Information Science

July 2002

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION STUDIES FOR AFRICA

N-GRAM-BASED AUTOMATIC INDEXING FOR AMHARIC TEXT

By
BETHLEHEM MENGISTU

Name and Signature of Members of the Examining Board

Ato Getachew Jemaneh, Chairman, Examining Board

Ato Workshet Lamenew, Advisor

W/t Saba Amsalu, Advisor

Dr. Abebe G/Tsadik, Advisor

Dr. Fiaz Hussein, External Examiner

ACKNOWLEDGEMENT

First and foremost, to God, who makes everything possible. I would also like to extend my sincere gratitude to the following; Ethiopian Airlines Enterprise that sponsored my studies at this school. my advisors Dr. Abebe G/Tsadik, Wzt. Saba Amsalu, Ato Werkshet Lamenew for their constructive reviews of my work and their support during the research, my family who have been supporting me in so many ways, all SISA staff who have facilitated access to resources at the school, and helped me in any way, fellow classmates for their moral support and also constructive ideas during the research, and all those who have contributed in one way or other to the success of this paper.

DEDICATION

I dedicate this thesis to the greatest god given gift that I have, my family, I love you all. My dedication also goes to the memory of my late sister Melen Mengistu who passed away about a month and a half ago.

Meliti you will always remain in our hearts.

Table of Contents

List of Figures

List of Tables

List of Appendices

List of Appendices

ABSTRACT

This research explored the applicability of the n-gram method for indexing text written in the Amharic language. 100 documents (Amharic news articles written in the Visual Ge'ez font obtained from Walta Information Center) and 24 queries (collected from people who frequently read newspapers) were selected and used for the test. The values of n used were n=2 (bi-grams) and n=3 (tri-grams). For comparison purposes, unstemmed words were also used as index terms.

The Vector Space Model (VSM) was used for document representation and retrieval. Thus, the individual words, bi-grams and tri-grams were identified for the collection. These unique terms were then weighted using the TF/IDF weighting technique used in the VSM. The term vectors were generated from these calculated weights for each type of term, i.e. unstemmed word, bi-gram, and tri-gram. The query terms (words, bi-grams, and tri-grams) were also identified and weighted. A different weighting formula was used for the query terms. The vectors of terms were then formed.

In order to retrieve relevant documents, similarity calculations were performed between each document-query vector pair. The ranked results from this calculation were then used to calculate precision and recall measures that are used in the VSM to test or compare retrieval effectiveness. The relevance information that was used to determine recall and precision was stored in a table. Recall and precision values for the queries for each type of index (word, bi-gram, and tri-gram) were calculated and compared.

The results showed that although word indexes are better in overall indexing performance, bi-grams and tri-grams also have values for indexing comparable to words.

CHAPTER ONE

INTRODUCTION

1.1. Background

The overwhelming volume of information available in this age has made it necessary to develop systems in order to handle/process this information. As more and more information accumulates, access to and organization of the information becomes more difficult. Information processing systems are designed as solutions to such problems. Some examples of information processing systems are database management systems, management information systems, decision support systems, and information retrieval systems. Information retrieval (IR) systems are designed to facilitate access to stored information i.e. facilitate information retrieval. However, according to Salton and McGill (1983), the functions of information retrieval (IR) systems extend to include the representation, storage, and organization of information.

In Salton and McGill (1983), the following elements of a typical IR system have been identified.

a) a set of information items (e.g. text, image, video, audio documents)
b) a set of user requests
c) a mechanism to determine which information items are most likely to meet the requirements of the requests

One can say that retrieval is accomplished when the IR system produces some information item as a result of matching the user query to some stored information item. The matching between queries and information items may not be direct. The information items and the queries are represented in some other form (e.g. a list of terms, vectors, etc.).

Representing information items implies the assignment of appropriate terms that could describe the content of the information item. This task of assigning descriptive terms to information items is called indexing (Salton and McGill, 1983). Terms can be words, groups of words, or slices of words. When indexing is carried out by using computers, it is known as automatic indexing. Automatic indexing is carried out on information stored in the computer.

According to Salton and McGill (1983), the indexing process involves two processes, (a) assigning terms or concepts to each stored information item that can describe the content, and (b) assigning a weight, or value, to each term reflecting its importance for the purposes of content identification.

Although information items can be of many types (text, graphics, pictures, audio, video, etc.) the focus of this research is on text documents. Reference to an information item will thus be made as text document or document from this point on.

There are different approaches to indexing, which differ on how they select indexing terms. Since the extraction of terms requires the analysis of text, the approaches also refer to the type of analysis used on the text to derive the index terms. The major categories are two; linguistic and statistical (Van Rijsbergen, 1975 ; Sparck Jones and Willet, 1997, Salton, 1989). Leung and Kan (1997) put forth the following three categories; the syntactic, the semantic and the statistical. The syntactic and semantic approaches fall in the category of linguistic approaches. Combinations of the approaches may be used.

The statistical method makes use of frequency properties of terms in text in order to derive index terms. The n-gram method, which is the subject of the current research, belongs to this category of techniques.

An n-gram is a sequence of a specified number (n) of characters occurring in a word (Kimbrell, 1988). In Robertson and Willet (1998) an n-gram is defined as a sub-string of length n characters derived from a text string (usually, but not necessarily, a word) containing not less than n characters. The characters in the n-gram retain the same order as in the source text from which the n-gram is derived. For example, the overlapping tri-grams (sequences of three characters) generated from the term COMPUTER would be COM OMP MPU PUT UTE TER. The formation of these tri-grams can be likened as Cohen (1995) indicated, to the sliding of an n-long (3-character long in this case) window over the word COMPUTER moving one character at a time.

The underlying principle in the n-gram method for indexing is that document texts are matched by the number of strings of characters they share, which means that the matching is not necessarily done at word level. Portions of words (the n-grams) can also be considered. N-grams have many applications (as discussed in section 2.3.4), among which is indexing. N-grams have been used as index terms in information retrieval.

1.2. Statement of the Problem

The use of the n-gram method for indexing (or in information retrieval) is not new. It has been explored since three decades ago (e.g. Burnett et al., 1979) Experiments have been conducted in automatic indexing using n-grams as index terms for languages like English, Chinese,

Japanese, Korean, with results that are comparable to conventional word-based indexing (e.g. Lee, Shin, and Ahn, 1996; Nie, Gao, Zhang and Zhou, 2000; Hackett and Oard, 2001).

In conventional word-based automatic indexing, such tools as stemmers, thesauri and stopword lists have been used. The n-gram method has been used as a complement to these tools and also independently of them (e.g. Huffman, 1995; Miller et al., 1999).

Although more than 80 languages are spoken in Ethiopia (Bender, 1976), Amharic is the working language of the Federal Government of Ethiopia. According to a census report by ECSA (Ethiopian Central Statistics Authority) (1998), it is the first language for more than 17 million and second language for over 5 million people. Research work in automatic indexing for Amharic, the working language of the Federal Government of Ethiopia, is still in its infancy. One research in the area of automatic indexing is Nega's (1999) work, conducted to develop a stemmer for the Amharic language. A stemmer is a procedure that reduces the different morphological variants (different forms of the same root word) of the same word to a common form, the stem. Nega reported that there are a very large number of word variants in the Amharic language that result from the complex nature of the morphological structure of the language. Other research works that required the derivation of indexing terms have also made use of stemming (for example Zelalem, 2001 and Saba, 2001) developed for the purpose by the researchers themselves. In addition to stemming, stop-word lists had also been set up and used by the researchers in order to meet the purpose of their research objectives. A stop word list is a list of commonly occurring words that are unlikely to be of use for retrieval purposes (Salton and McGill, 1983; Sparck Jones and Willet, 1997) for example (prepositions, articles, etc.).

To the researcher's knowledge, there is no available standard stop-word list or general stemmer for use in indexing for Amharic text. A language-independent method for automatic indexing that does not make use of such tools (stemmers, stop-word lists, thesauri) can serve as an alternative solution. The focus of this research is on one such indexing method; the n-gram method, as applied to the Amharic language.

The relative simplicity of the method offers another good reason to test the applicability to the Amharic language. Language-dependent elements or procedures in the process of indexing can be avoided. The results from this research can augment further future research in Amharic information retrieval.

The growth in Amharic publications is one reason to consider development of automatic Amharic information retrieval systems. The method used in this research in automatically indexing Amharic text can contribute to the simplification and promotion of automatic information retrieval efforts for Amharic text documents.

1.3. Objectives of the Study

The general and specific objectives of the research are the following

1.3.1. General Objective

The general objective of this research is to explore the possibility of applying the use of n-gram-based indexing for Amharic text retrieval purpose.

1.3.2. Specific Objectives

- to review concepts in automatic indexing

- to review the general n-gram method that has been applied to information retrieval processes in earlier experiments with the purpose of selecting a specific method of n-gram generation and the value(s) for n.
- to review the linguistic features of the Amharic language applicable to the research (e.g. how the division of words is achieved, how the alphabets are represented in the computer).
- to set up the test set (i.e. select documents and queries)
- to apply the n-gram method to the selected Amharic text in order to derive indexing terms.
- to analyze the effectiveness of the indexing terms for the representation of the documents and for the discrimination of a specific document/text from the collection by making use of standard measures of recall and precision.
- to develop and test the prototype system
- to draw conclusions and forward recommendations for further study

1.4. Methodology

The following methods were employed in conducting the research

1.4.1. Literature Review

Extensive literature review was conducted to understand the general n-gram approach to automatic indexing and select a suitable n value as well as a suitable n-gram generation method to be used for the experiment. Evaluation techniques for testing the effectiveness of the method were also determined from this review. Printed materials like books, journal articles, previous related research work as well as electronic materials on the web were consulted for this purpose.

1.4.2. Data Sources for the Experiment

100 Amharic local news articles available in electronic form were used as a source of data for testing. These news articles were obtained from Walta Information Center and were used for research conducted at SISA by Saba (2001). Walta Information Center is a government information center that distributes news for broadcast over television and radio for local consumption. In addition, queries (24 in number) were collected from people who are regular readers of newspapers. The news articles and queries were written in the Visual Ge'ez Amharic font (see section 3.2.6). The relevance judgment for each of the queries was made by a journalist.

1.4.3. Experimentation Method

Owing to its good string manipulation features and user-friendliness the Visual Basic 6.0 programming language was used to develop all the programs that were used to manipulate the files and develop the prototype. In addition, Microsoft Access database tables were used to store the indexing terms (words, bi-grams, tri-grams), their frequency information, and their document references. The document and query vectors and their similarity values were also stored in tables. The prototype system was run on a Dell machine with Microsoft Windows 2000 operating system, 1.70 MHz speed, and 261,136 KB RAM.

As has been indicated above, the n-gram method belongs to the class of statistical approaches. In the statistical approach to indexing, the selection of index terms makes use of statistical analysis of frequencies of terms to score terms (for example the term weighting functions specified in Salton and McGill (1983)) or in other words to give weights to terms. Term frequency information was used in this research as the scoring method and applied on the terms generated from the text. Once representative terms had been derived for the documents,

the effectiveness of the representation had to be tested. This was done by computing similarity values between the query and document representations. A similarity measure; the cosine similarity coefficient (as discussed in chapter two) was used to calculate similarity between the query and the document vectors. The retrieval effectiveness of each type of representation (indexing) was then compared using the conventional recall-precision measures. Depending on results of the output, conclusions and recommendations were made.

1.5. Significance of the Study

In addition to being an academic exercise to fulfill the requirement of the program, this research is believed to produce results that can indicate the possibility for the development of a general Amharic indexing software that does not depend on the use of stemmers, thesauri and stop-word lists.

Since it is not common practice to produce indexes for Amharic documents, the availability of such software if developed may serve to promote the exercise. As a result, information retrieval in Amharic can be made easier. The method can be applied on other Ethiopian languages that make use of the same Ethiopic alphabets, for example Tigrignya, Guragignya, etc.. The method can also be extended to other applications on the Amharic language like the development of an Amharic spellchecker, and text compression for Amharic text since it has been shown in literature that n-grams can be used for other applications (e.g. spell checking), as discussed in chapter two below.

1.6. Scope and Limitation of the Study

Due to the long processing time it takes to generate the n-grams weighted n-gram vectors, and similarity values, the size of the n-grams used for this research has been limited to two and three. Other values of n may also be tried in future research.

In addition, the research has made use of no stop word lists and stemmers for two reasons. The first is, from the outset, the research was also an attempt to explore the possibility of indexing text written in Amharic without making use of stemming, and stop word lists, which are language-dependent tools. The second is more related to the availability of these resources. To the knowledge of the researcher, there are no compiled standard Amharic stop word list and stemmer available for use. Thus, it was not possible to make use of such resources although the research recognizes its value.

The prototype developed in this experiment does not function for text written in a different Amharic font. This results from the fact that there is no standard set of symbols used by all the Amharic fonts as discussed in section (3.2.6.).

Due to the long processing time (several hours) it takes to generate the bi-grams and tri-grams and also compute similarity between the document and query vectors, and considering the time constraint, the experiment was conducted using 100 documents and 24 queries as the test set. Larger size collections must be used in future work.

Text files (.txt extension) instead of the original word files (.doc extension) were used for the experiment. This caused some degradation of the text causing some symbols to change. This has been indicated in chapter three.

1.7. Organization of the Thesis

This thesis is organized in five chapters. Chapter one, the present chapter, is a general introduction to the problem, the justification of the research and methodology used for the research. Chapter two is devoted to literature review. It discusses concepts in automatic indexing and n-grams in two sections. In the first section, concepts that underlie the experiment in this research are discussed in adequate detail. In the second, the n-gram method to automatic indexing and a review of works on n-gram related indexing is presented. In chapter three, the characteristics of the Amharic language that are applicable to the research area are discussed briefly and the model of the prototype system is described. The experimental settings, the process of the experimentation and the findings are presented in chapter four. Finally, in chapter five general conclusions and recommendations are made based on observations and results from the experiment.

CHAPTER TWO

REVIEW OF RELATED LITERATURE

2.1. Introduction

This chapter is divided into two. The first part presents a review of literature about concepts surrounding indexing. It presents what a typical IR system is composed of and further explains how these components function in the system. Concepts like documents, indexes, terms, weights, queries, IR models, and relevance are briefly introduced. One particular model, the Vector Space Model is described in a bit more detail since it is the model selected for the experiment in this research.

The second part introduces the approach adopted in this study, namely, the n-gram approach. Before that the different approaches to automatic indexing are briefly introduced. N-grams are defined and a review of n-gram-based automatic indexing works is presented.

2.2. Concepts Related to Indexing

Automatic indexing systems and experiments are based on a number of theories established and proved through experiments over the years. This section discusses such concepts that underlie the experiment in this research.

2.2.1. Information Retrieval Systems

By definition, an Information Retrieval (IR) system is a system that is capable of storage, retrieval and maintenance of information (Salton and McGill, 1983; Van Rijsbergen, 1975). The function of any IR system is to process a user request for information and retrieve materials that have contents that could potentially satisfy the information need of the user. According to Van Rijsbergen (1975), there are three major concerns of an IR system. They

are; (a) the description of contents of documents in a form suitable for computer processing, (b) the exploitation of relationships between documents to improve the efficiency and effectiveness of retrieval strategies, and (c) the measurement of the effectiveness of retrieval. Of these, according to Salton and McGill (1983), the first (document representation) is the most crucial function. The focus of this research is also on this particular function.

Like any system, an IR system has three basic components; input, process and output. The following figure (Fig. 2.1) is an adaptation of Van Rijsbergen's (1975) three-component-diagram which illustrates a typical IR system.

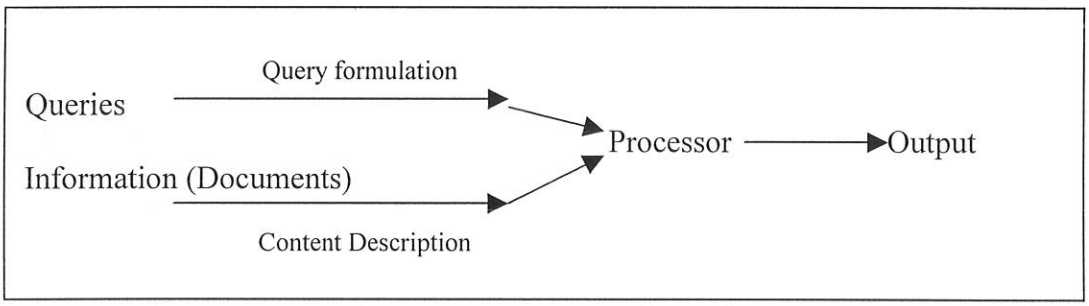

Fig. 2.1 A typical IR system

As depicted in the figure, an IR system takes both documents and queries as input. Two distinct processes are involved at the input side; one is the description (indexing) and storage of documents and the other is the processing of queries (description and formulation). Queries must also be represented by a set of terms just like documents before they can be matched with documents. Query formulation refers to the preparation of the query for input to the matching system. As is discussed in section (2.2.5), below some models of IR require that queries be rewritten in a specified form (e.g. in Boolean systems).

The component that determines which items are significantly related (i.e., relevant) to a user's need takes both the document and query description as input and does some relevance processing (matching computation). If relevant items are found, references to the items (such as the storage place, title or file name) or the whole document will be made available to the user.

There are different types of IR systems depending on the type of information they process. Before the invention of digital technology, information storage was paper-based. Nowadays we have masses and masses of information in electronic or digital form; digital text, images, audio, video and other multi- media objects. IR systems that process such different types of information can be very different in terms of the devices employed and the techniques and procedures applied. The current research is concerned with text documents, and experiments on automatic indexing and subsequent document retrieval using the index terms extracted from the text documents.

2.2.2. Content Representation and Descriptors

With the ever-increasing volume of text information stored in electronic media, searching for full texts becomes more and more time-consuming and uncontrollable. As a solution, keywords or terms that are considered as appropriate content descriptors are selected and assigned to documents to provide short-form descriptions of the documents. In order to identify and extract such descriptors, the content of the documents must be analyzed. The process of analyzing text and deriving the short form descriptions known as index terms is called "indexing" (Salton and McGill, 1983). The general view to indexing is that it is the selection of 'key' words or phrases or expressions from text that are 'significant' indicators of content and which together sum up the message of the document. (Van Rijsbergen, 1975;

Salton and McGill, 1983; Salton, 1989). As indicated in section 2.1., indexing (representation) is a crucial function of an IR system.

In Salton (1989), a number of distinctions are made in the indexing environment; whether (a) the indexing is manual or automatic, (b) the index terms are controlled or uncontrolled (section 2.2.2.1.), and (c) whether single terms or complex terms (groups of single terms) are used for indexing (section 2.2.2.3). Index terms can also be characterized depending on how exhaustive or specific they are (section 2.2.2.2). Indexes also vary in the importance they have to the content representation, which is measured by a weighting process; term weighting (section 2.2.2.4).

2.2.2.1. Controlled and Uncontrolled Vocabularies for Indexing

Before computer-based IR systems were developed, indexing was done manually. In manual indexing, typically, a set of controlled keywords is pre-defined and used to pick indexing terms from (controlled vocabulary for indexing). Such a list of keywords is called the indexing language (Van Rijsbergen, 1975; Salton and McGill, 1983). Computer-based IR systems usually make use of keywords extracted directly from the documents and queries (uncontrolled, free text indexing). The current research makes use of free-text indexing, deriving the indexing terms directly from the text.

2.2.2.2. Term Exhaustivity and Specificity

The other factors that characterize index terms are exhaustivity and specificity (Salton and McGill, 1983). They refer to how much the index terms cover the subject of the indexed document. Exhaustivity refers to how comprehensive the index terms are in representing all

the concepts in the indexed text. Index terms are exhaustive if they represent all concepts found in the document. Exhaustive indexing thus increases recall (number of retrieved documents) at search time. On the other hand, specificity refers to the degree of specificity of the index terms, i.e. if broad terms (that could mean a number of related concepts) or narrow terms (terms that are very specific and bear no ambiguity) have been used. If the indexing vocabulary is very specific, and if narrowly defined terms are used, a large proportion of non-relevant documents may be rejected from among the number of documents that are retrieved. This implies the concept of 'precision'. If exhaustive indexing vocabulary is used, a large number of documents will be retrieved, implying high 'recall'. (Salton and McGill, 1983; Van Rijsbergen 1975).

Ideally, both exhaustivity and specificity are desired characteristics of indexing vocabularies. However, the two are inversely related. These properties (exhaustivity and specificity) have been considered by Sparck Jones (1972) in association with the frequency properties of terms in text in order to derive weights for the indexing terms. Free text indexing is naturally exhaustive since it uses almost all of the terms from the text.

2.2.2.3. Single and Complex Terms

Indexing terms can be single terms (e.g. individual words) or compound terms (phrases composed of nouns, adjectives, prepositions, etc.). Automatic indexing systems mostly use single terms for indexing (Salton and McGill, 1983; Salton, 1989). The current research experiment also uses single terms.

Complex terms can actually be formed by the combination of single terms that share some kind of relationship. Thesaurus relationship and phrase relationship are the most common

types of relationship (Chen, 2001). Semantic relationships such as synonymy are included in the kind of thesaurus term relationship. In indexing, thesaurus terms are usually grouped together to form a class of terms so that appearance of one member in the class will represent the appearance of the whole set. On the other hand, if two terms share a relationship of modification or specification, they are said to have a phrasal relationship. For example, a single term like 'aircraft' is very general. If it is modified with another term, say 'maintenance', thus forming a phrase 'aircraft maintenance', then these two terms are said to have a phrasal relationship. The use of phrases in IR system helps to increase specificity (ibid.).

2.2.2.4. Term Weighting Factors

Index terms also vary in the importance they have to content representation, which is measured by a weighting process. Term weighting means assigning numeric values to terms in order to determine the importance of the terms for indexing. Term weights help to distinguish terms that are more important for indexing and as a result for retrieval (ideally retrieval of all relevant and rejection of all non-relevant documents for a query) from other, less important terms (Salton and McGill, 1983; Van Rijsbergen, 1975; Salton and Buckley, 1988).

Term weighting, according to Salton and McGill (1983) is a part of the indexing task, which first assigns to each information item terms that describe the content and then assigns numeric values (weights) to each term, to determine its importance for indexing. Moreover, the consideration of term weights can assist in ranking documents in decreasing order of the matching terms at search time (Salton, 1989).

The distributional properties of terms are used in deriving weights in automatic indexing systems. Luhn (1961) was the pioneer in exploring the idea of using frequency properties of terms in the automatic selection of index terms.

Accordingly, the three most used frequency factors in calculating a term weight are:

1. Term frequency (TF): the number of occurrence of a term in a document. Intuitively, the more a term occurs in a document, the more important it is.

2. Document Frequency (DF): the number of documents that contain a certain term. DF has a reverse impact on the importance of a term, that is, the more common a word is in a number of documents, the less important it will be. Intuitively, if a word appears in every document in the collection, it would contribute nothing to the retrieval because all documents would be returned for a query that contains the term. This inverse function of DF is denoted by IDF (Inverse Document Frequency). The most commonly used IDF formulation is log (N/ DF) of Sparck Jones (1972). Sparck Jones established in her experiments that words found in a specific document, but rarely in other documents, were important for use as indexing terms and developed the inverse document frequency as a term score. She also showed that making use of weighted terms resulted in better retrieval than making use of unweighted terms.

3. Document length: the total number of words in a document. This factor is used to eliminate the bias that longer documents tend to be ranked higher for retrieval than shorter documents simply by virtue of their length and because they tend to repeat terms more often. A process called 'document length normalization' is applied to prevent this kind of problem. Normalization factors are introduced in term weighting formulae in order to achieve this (Salton and Buckley, 1988). For example, in the

vector space model (an IR model described in section 2.2.5 below), different length vectors are normalized by dividing the weights of each term by the length of the vector representing the document (ibid.)

The above three factors have been incorporated into various weighting schemes for documents and queries (Salton and Buckley, 1988). Salton and Yang (1973) combined the inverse document frequency of Sparck Jones (1972) with the in-document frequency (TF). This combined term weighting scheme is known as the TF/IDF (Term Frequency/Inverse Document Frequency) weight. They considered the product of TF and IDF. Their formula is as follows

$$W_{ik} = FREQ_{ik} * \log_2(N/DOCFREQ_k) \quad (1)$$

Where W_{ik} denotes the weight of term "k" in document "i"

$FREQ_{ik}$ denotes the frequency of the term "k" in document "i" (TF)

$DOCFREQ_k$ denotes the number of documents that contain the term "k"

Signal-noise ratio that exploited the Shannon's information theory, and term discrimination value that considers the discriminatory value of a term when it is added to a document and removed from a document are other weighting techniques for terms discussed in Salton and McGill (1983).

Robertson and Willet (1996) have explored the adaptation of genetic algorithms employed in the life sciences as term weighting mechanism in ranked output searching systems in which documents retrieved for a query are ranked in order of relevance.

The most widely used weighting scheme disclosed in researches related to automatic indexing is the TF/IDF weighting scheme. This research also makes use of this weighting scheme.

2.2.3. Query

A query can be defined as the verbalized expression of a user's information need (Tague-Sutcliffe, 1992). Queries may be real or artificial. Real queries represent real information needs of a user and artificial queries on the other hand are derived from titles and other parts of document text.

A query is one of the two inputs to an IR system, as shown in figure 2.1. Different models in IR (see section 2.2.5) make use of different formats for queries. In the vector space model, for example, a query stated in the natural language that we use for communication may be used (e.g. 'information retrieval and computers'). In the Boolean model on the other hand, queries must be formulated as keywords combined by Boolean operators (e.g. "information AND retrieval"; "(information AND retrieval) OR computer"). In models like the Boolean model, natural language queries must first be changed to a format required by the model (Salton et al. 1975; Salton, 1989).

Ideally, users of an IR system put forward their information requirements verbally or in written form. They then submit this query to the IR system from which they will be presented with materials of potential interest to them. Queries are also input to an automatic indexing process and are processed in a similar manner to documents. The encoded queries are then matched with the encoded documents. In other words, queries are also mapped into the language of indexing (Salton and McGill, 1983).

Term weighting also applies to query terms. Salton and Buckley (1988) present a number of weighting schemes used for documents and queries. The following weighting formula, which is the ideal formula for queries (ibid.) as established in experiments, is used in this research.

$$W_{iq} = (0.5 + (0.5\ tf_{iq} / \max tf)) \times \log (N/n_i) \qquad (2)$$

where W_{iq} is the weight of term i in query q

tf_{iq} is the frequency term i in query q (which usually is = 1)

max tf is the maximum frequency value of all query terms

N is the total number of documents in the collection (this does not include the queries)

n_i is the number of documents in which the query term is found

2.2.4. Relevance Judgment and Evaluation

Any system has to be evaluated for performance using the appropriate parameters. IR systems are no exception. IR systems can be evaluated with respect to efficiency (operational issues like cost, time factor, etc.) as well as effectiveness (how well the retrieved documents satisfy the user request) (Salton and McGill, 1983). Being a laboratory experiment, this research deals only with the effectiveness aspect.

The effectiveness of an IR system is the ability of the system to retrieve the information items sought by users of the system; in other words, how many of the items that are relevant to the

user's need have been retrieved, and how effectively the non-relevant documents have been rejected. The concept of relevance is important in this respect.

According to Van Rijsbergen (1975), relevance is a central concept in IR which can be broadly defined as the 'aboutness' or 'appropriateness' of a document to a user's query. The purpose of an ideal IR system then is to retrieve all relevant documents while at the same time rejecting as much of the non-relevant ones as possible. Relevance judgement then is important in order to determine how much of the information items are pertinent to the specified query (ies).

2.2.4.1. Relevance Judgment

Relevance judgment can be defined as the determination of which documents in a collection can potentially satisfy a given query (or are relevant). Relevance is a highly subjective notion. It has been a subject of research by itself, as disclosed in literature (e.g. Saracevic, 1975). Due to many aspects of individual differences (such as background knowledge, expectation etc.), the usefulness of the same list of retrieved documents produced for the same query by the same system may result in very different judgments by users; a document that appears very useful to one user may be judged as irrelevant by another. Relevance judgments in experimental systems are usually made by subject experts; people who are experts in the area that a collection represents and thus can determine which query is related to which document for the specified collection (e.g. journalists who can determine the topic of a news article).

2.2.4.2. Evaluation

Relevance information determined by experts is used in the evaluation of an IR system. Recall and precision values are the most commonly used measures of the effectiveness of an IR system. Recall (R) measures the proportion of relevant documents that have been retrieved (recalled) from all the relevant documents in the collection. Precision (P) on the other hand, measures the proportion of relevant documents from those that have been retrieved (Van Rijsbergen, 1975; Salton and McGill, 1983). Both recall and precision, being ratios, give values between 1 and 0. The computation equations of recall and precision values for the retrieval of a number of documents for a query are given by:

$$\text{Recall}(R) = \text{RelRet} / \text{TotRel} \quad (3)$$

$$\text{Precision}(P) = \text{RelRet}/\text{TotRet} \quad (4)$$

where RelRet denotes the number of relevant documents retrieved

TotRel denotes the total number of relevant documents in the collection

TotRet denotes the total number of retrieved documents

R and P can be calculated for each query for a list of documents produced by the IR system. When there are more than one queries, an average recall, R_{avg}, and an average precision, P_{avg}, can be computed using these formulae (Salton and McGill, 1983) :

$$R_{avg} = \text{sum(all recall values)} / \text{total number of queries} \quad (5)$$

$$P_{avg} = \text{sum(all precision values)}/ \text{total number of queries} \quad (6)$$

For systems that produce a ranked list of retrieved documents, parameter TotRet is controllable. Therefore precisions can be computed at fixed recall intervals, say when R= 0, 0.1, 0.2,...,1.0.. (Salton and McGill, 1983). In this case, for each query, there will be more than one pair of R and P, which makes the plotting of recall-precision curves possible for a specific query.

Precision-recall curves can be used to compare the retrieval effectiveness of different indexing techniques or different systems. As revealed by research literature, precision and recall are the most commonly used evaluation measures (e.g. in the TREC experiments in Harman (1995) and also the SMART system in Salton and McGill (1983)) and they have thus been selected for the purpose of this research.

Other, complementary and single valued measures of effectiveness (e.g. Fallout, the E measure) are also disclosed in Salton and McGill(1983).

2.2.5. Retrieval Models

A retrieval model specifies how the contents of a document and a query are represented (indexing) in an IR system, and how the documents and the queries are matched (retrieval) so that relevant items can be retrieved. From among the various IR models that have been proposed over the years three major ones (Boolean, Probabilistic and Vector Space models) are in current use (Sparck Jones and Willet, 1997):

a) Exact Match Models (Boolean Models), which treat a document as a set of terms and a query as a Boolean expression and compare the two during searching; (e.g. using the following form of query "aircraft AND maintenance")

b) Partial Match Models (Best Match Models) consider partial matching of documents to queries and produce ranked lists of matched document–query pair; two types of models fall in this category;

   a. Probabilistic Models, which are based on the estimation of the probability that a document's representation matches a query;
   b. Vector Space Models (VSM), which view documents and queries as vectors in an n-dimensional vector space and use distance as a measure of similarity.

The Exact Match Model, as the name implies, is based on exact matching between query and document terms for retrieval. Although it is the most widely used model for commercial IR systems, and easy to implement, this has been criticized as being too simple in its matching function and being unable to rank the retrieved results according to their importance since the matches are only exact matches and all of the retrieved documents thus have equal value (Sparck Jones and Willet, 1997). Probabilistic Models and Vector Space Models, also called statistical models, on the other hand, can produce a ranked search result based on the documents' relevance to the query (ibid.; Salton, 1989). Other models discussed in Sparck Jones and Willet (1997) are those that consider the social and cognitive contexts in which the other models operate and classified as cognitive models.

This research makes use of the VSM. There are a number of advantages to using the Vector Space Model (VSM) as explained in Salton (1989). Among the advantages are, (a) its relative simplicity for representation of documents and queries (the sets of terms (their weights) are used to form the vectors), and (b) the fact that it can make use of natural language text as a query. The following section will be devoted to explaining the model in a bit more detail.

The Vector Space Model (VSM)

In the VSM of Salton et al.(1975), a collection of documents and queries is pictured as a document space consisting of documents D_i each identified by one or more index terms T_j. The terms may be weighted according to their importance, or unweighted with weights restricted to 0 and 1 to indicate their presence or absence in a document or query. When t indexing terms are available, each document or query can be represented by a vector of size t, where each coefficient indicates the weight of the specific term (Salton, 1989; Salton et al., 1975). The number of terms t can be considered as the dimension of the space in which each document or query can be visualized as a point described in terms of the t coordinates.

The VSM assumes that an available term set is used to identify both stored records and information requests or queries. Both queries and documents can then be represented as term vectors of the form

$$D_i = (a_{i1}, a_{i2}, a_{i3}, a_{i4}, \ldots a_{it}) \quad (7)$$

and

$$Q_j = (q_{j1}, q_{j2}, q_{j3}, q_{j4}, \ldots q_{jt}) \quad (8)$$

Where the coefficients a_{ik} and q_{jk} represent the values or weights of term k in document D_i or query Q_j respectively. In a binary vector representation, a_{ik} or q_{jk} is set to 1 when term k appears in document D_i or in query Q_j, and to 0 when the term is absent from the document or query.

Searching and retrieval of documents in the VSM is based on the computation of similarity between the vectors (Salton et al., 1975 ; Salton, 1989). Once the documents and queries have been given a vector representation, it is possible to compute a similarity coefficient $s(D_i, D_j)$

for any two vectors (document-document or query-document), say D_i and D_j, which reflects the degree of similarity in the corresponding terms and term weights. A number of similarity computation formulae have been devised by Salton et al.(1975). A typical similarity coefficient is an inverse function of the angle between the corresponding vector pairs (the cosine similarity coefficient); when the term assignment for two vectors is identical, the angle between the two vectors will be zero, producing a maximum similarity measure (i.e. cosine(0) = 1).

The most widely used similarity coefficient computations are the inner product, Dice's coefficient, cosine coefficient, and Jaccard's coefficient (Salton, 1989, Van Rijsbergen, 1975). Of these, the most commonly used in the VSM is the cosine coefficient. It has thus been selected for this research.

Cosine coefficient - $\Sigma x_i.y_i \ / \sqrt{(\Sigma x_i^2 \ \Sigma y_i^2)}$ for i = 1 to t (9)

In the formula above, x_i denotes the weight of term "i" in document X, y_i denotes the weight of term "i" in document Y, $\Sigma x_i.y_i$ denotes the sum of the products of each corresponding coefficient in the vectors.

Computed similarity values between document and query vectors can be arranged in decreasing order to form a ranked list of document-query similarity in decreasing order of similarity. When a ranked list is used, a threshold value can be set so that document-query pairs in the list with values below the threshold value can be removed from consideration for retrieval. The underlying principle behind the use of ranked lists and threshold values is that the smaller the similarity value gets, the lesser the similarity between the query and the

document, i.e. the lesser the likelihood that the document will satisfy the query, so that document-query pairs below the threshold value can be ignored.

Threshold values are usually set based on observation (e.g. Frieder, Chowdhury, Grossman and McCabe (2000)). Thresholding makes retrieval easier in environments where there is a voluminous collection of documents to consider for perusal by limiting the number of documents retrieved.

2.3. Approaches to Automatic Indexing

Approaches to automatic indexing can be categorized based on the methods they employ for extracting the index terms. Van Rijsbergen (1975) classifies the approaches into two; linguistic and statistical. Leung and Kan (1997) recognize the semantic, the syntactic and the statistical. The semantic and syntactic approaches belong to the linguistic category. They make use of linguistic knowledge. Combinations of the approaches may be used. For example Sparck Jones and Willet(1997) cite Fagan (1989), Salton et al.(1990), Sparck Jones and Tait (1984) as works that have used both syntactic and statistical techniques.

2.3.1. The Semantic Approach

In this approach, the meanings conveyed by words, phrases, and sentences in documents are identified. Terms that reflect these meanings are then assigned to the documents as index terms. A knowledge base and a thesaurus are often required in this approach; for example, Chen (2001).

Some researches that made use of this approach quoted in Leung and Kan(1997) used the following; thesauri, NLP (natural language processing) techniques like parsing, formalized languages developed for the purpose of indexing, frame-based knowledge representation

document, i.e. the lesser the likelihood that the document will satisfy the query, so that document-query pairs below the threshold value can be ignored.

Threshold values are usually set based on observation (e.g. Frieder, Chowdhury, Grossman and McCabe (2000)). Thresholding makes retrieval easier in environments where there is a voluminous collection of documents to consider for perusal by limiting the number of documents retrieved.

2.3. Approaches to Automatic Indexing

Approaches to automatic indexing can be categorized based on the methods they employ for extracting the index terms. Van Rijsbergen (1975) classifies the approaches into two; linguistic and statistical. Leung and Kan (1997) recognize the semantic, the syntactic and the statistical. The semantic and syntactic approaches belong to the linguistic category. They make use of linguistic knowledge. Combinations of the approaches may be used. For example Sparck Jones and Willet(1997) cite Fagan (1989), Salton et al.(1990), Sparck Jones and Tait (1984) as works that have used both syntactic and statistical techniques.

2.3.1. The Semantic Approach

In this approach, the meanings conveyed by words, phrases, and sentences in documents are identified. Terms that reflect these meanings are then assigned to the documents as index terms. A knowledge base and a thesaurus are often required in this approach; for example, Chen (2001).

Some researches that made use of this approach quoted in Leung and Kan(1997) used the following; thesauri, NLP (natural language processing) techniques like parsing, formalized languages developed for the purpose of indexing, frame-based knowledge representation

languages, and dictionaries containing some lexical knowledge and concepts to support the indexing process. Strzalkowski (1994), has made use of several; NLP (natural language processing) techniques; a part of speech tagger, a morphological stemmer, and a fast syntactic parser had been used.

2.3.2. The Syntactic Approach

In this approach, terms that are believed to reflect the content of the document (content-bearing words) are used as index terms or as clues for selecting suitable index terms. Here also knowledge of whether a word is content-bearing for the context or not is required. A list of words that are non-content bearing (known as stop-word list or negative dictionary in literature) must also be used in order to eliminate the non-content bearing function words such as articles and prepositions. Research works using this approach, quoted in Leung and Kan (1997) made use of syntactic analysis of text (also in Strzalkowski (1994), identification of nominal constructions, use of the layout structure of text, development of syntactical rules, use of the knowledge of linguistic regularities to recognize important phrases from text, use of a sub-language grammar to extract information contained in text for indexing.

2.3.3. The Statistical Approach

In this approach, the index term assignment is mainly based on the exploitation of occurrence properties of words or portions of words (e.g. stems, n-grams) found in documents, and statistical calculations with the word or word fragment frequencies as explained in Luhn (1961). Some of the techniques in this approach as quoted in Leung and Kan (1997) are; n-gram analysis to select index terms from the text for abstracting purposes (e.g. Cohen (1995)), corpus-based statistical algorithms to generate back-of-book indexes, use of statistical information about repeated phrases in a document for conceptual phrase indexing, use of

statistical methods to develop word descriptor relations and phrase-descriptor relations to assign index terms, use of the frequency of a term in a particular document (TF) and that of a term in the complete document collection (DF) to determine whether the term is a good index term, and use of discrimination value of words for index term selection.

The n-gram approach, which is the subject of this research, belongs to this class of approaches.

2.3.4. The N-gram Approach to Automatic Indexing

The use of n-grams for indexing is not new. It has been pursued starting from a few decades ago (e.g. Burnett et al., 1979; Willet, 1979). Indexing by n-grams has also been applied on many languages with results comparable to word-based indexing as discussed in the review below. This research explores n-gram-based indexing for text written in the Amharic language. In the subsequent sections, a definition of n-grams is provided, features of n-grams are discussed and a review of literature on n-gram-based indexing is also presented.

2.3.4.1. N-grams

N-grams are n units or grams where grams can be words (Heja, 2001; Galescu and Ringger, 1999, Zhao 2000), phonemes (e.g. Wechsler and Schäuble, 1995), characters (Jaruskulchai, 1998), morphemes (Lee, Shin, Ahn, 1996), syllables (Hackett and Oard, 2001) etc. The n-grams are formed by considering n adjacent or non-adjacent units extracted from the source. For example, in the sentence "this is the house that jack built", the word bi-grams (2-words) can be "this is", "is the", "the house", etc. In the word 'house' the character bi-grams (2-characters) can be 'ho','ou','us','se' and so on.

The value of n can vary from 1 to many (usually not larger than 7 or 8). N-grams that are too long become almost equal to words (or sentences in the case of word n-grams) and hence will fail to capture similarity between different (in the morphological sense) but similar words. On the other hand, n-grams that are too short (e.g. uni-grams) will tend to find similarities between words that are due to factors (e.g. distribution of alphabets) other than semantic relatedness (Zhong Gu and Daniel Berleant, 2000).

2.3.4.1.1. Types of N-grams

There are different types of n-grams differing in method of formation and size (Robertson and Willet, 1998). With respect to size, we can have bi-grams (2-grams, sometimes written as di-grams in literature), tri-grams (3), tetra-grams (4), penta-grams (5), etc. Two examples of the different methods of forming n-grams are the adjacent and non-adjacent methods depending on whether the constituent characters of the resulting n-gram are found adjacent to each other or not in the source word, sentence etc. For example with the term COMPUTER, we can form the character tri-grams from non-adjacent characters: COM, COP, COU, COT, etc. The process results in more number of tri-grams. The most common type of n-gram formation method for information retrieval, also used in the current research, disclosed in literature is the adjacent method.

Other types of n-grams are also mentioned in Robertson and Willet (1998); binary n-grams and positional n-grams. Binary n-grams register the absence or presence of a particular n-gram. Positional n-grams further go on to register the position of the n-gram.

2.3.4.1.2. Applications of N-grams

N-grams have been investigated for tasks related to IR at least as early as 1979(Burnett et al.). Since then, they have been investigated in such tasks as language identification (Cavnar and Trenkle, 1994), spelling error detection and correction (Zamora, 1980; Salton 1989), document (text) categorization (Huffman 1995; Gustavsson 1996; Cavnar and Trenkle, 1994), robust handling of noisy (misspelled, OCR'ed, etc.) texts (Cavnar and Gillies, 1994), topic highlighting (Cohen, 1995), document space visualization (Huffman 1995), ADI (Adaptive Information Filtering) (Tauritz and Sprinkhuizen-Kuyper, 2000), text compression (Cooper et al. , 1982 ; Wisniewski, 1987) and other information retrieval related applications.

2.3.4.2. N-gram-Based Indexing

The n-gram-based indexing method uses n-grams as the index terms. The extraction of n-grams is merely a technical issue, requiring only the recognition of characters in an alphabet. No linguistic knowledge is required. For this reason, the technique of indexing by n-grams is labeled a 'language-independent' technique (e.g. Cohen, 1995; Huffman, 1995; Miller et al., 1999)

As revealed in research literature, it is character n-grams that are mostly considered for indexing although some researchers have also used syllable n-grams (e.g. Hackett and Oard, 2001) and word n-grams (Héja, 2002). The current research also considers character n-grams. In forming the n-grams, the slicing process starts from the first character of the word and combines the n consecutive characters to form an n-gram. Then it goes on to the second character and repeats the grouping, goes on to the third character and forms the group and so on up till the last n characters that will form the final n-gram. As a result, all possible overlapping n-grams are obtained.

For example, for the word RESEARCH, if n is set to 2, the following bi-grams can be generated:

RE ES SE EA AR RC CH

If n is set to 3, the following tri-grams can be generated:

RES ESE SEA EAR ARC RCH

Padding characters are usually used in forming the n-grams. If n is the size of the n-gram then n-1 padding characters are used both at the beginning and the end of a word before the n-grams are formed. The use of padding characters is justified as follows (Robertson and Willet, 1998);

a) each of the distinct characters that makes up the n-grams must occur an equal number of times (say, twice in the bi-grams or three times in the tri-grams.)

b) padding also results in better distinct representation of words that have identical n-grams; for example consider the words ADA and DAD that both have the unique bi-grams AD and DA. Both words will have the same bi-gram profile (set of bi-grams). If padding were to be used, however, the profiles would be distinct.

I.e.

ADA would have the bi-gram profile (*A AD DA A*)

DAD would have the bi-gram profile (*D DA AD A*)

As disclosed in research literature, bi-grams and tri-grams (n=2 and n=3 respectively) have been widely used as indexing terms. The choice of n for n-grams is mostly based on

considerations of storage (space efficiency) and processing time (time efficiency). When padding characters are used in forming the n-grams, the number of n-grams generated increases substantially as n increases.

For example consider padding the word RESEARCH with *;

RESEARCH

the resulting bi-grams are:-

R RE ES SE EA AR RC CH H

and the resulting tri-grams for the padded word **RESEARCH** are: -

R *RE RES ESE SEA EAR ARC RCH CH* H

Bi-grams and tri-grams are used as index terms both for document and query text. Documents and queries are then matched using these sets of index terms. Similarity values can be calculated to quantify the degree of the matching.

Review of Related Works

In this section, a review of automatic indexing research using n-grams as index terms is made.

The n-gram technique can supplement stop-listing and stemming. But it has also been used without these tools. Cohen (1995) used n-grams extracted from text as indexing terms. The objective of his research was to extract what he termed 'highlights' from text that could serve as abstracts. His approach made use of no stop-word lists, stemmers or other language- and domain-specific components. He demonstrated that the technique could be applied to different languages (Georgian, German, Japanese, Russian, Spanish) and domains with only slight

adaptations with respect to preprocessing of the text. This refers to the filtering done on the text to remove numbers, punctuation marks, and other extraneous characters that will not be considered for indexing or the identification of individual words (word separation methods) and characters for processing. The values used for n were 5 and 4. The results obtained from the indexing of the different language texts suggested the language-independence of the technique.

Mayfield and McNamee (1998) compared the use of n-grams versus words for indexing in an ad-hoc experiment for TREC-7 (Text Retrieval Conference – 7) for English text. They used 5-grams and words as terms for indexing. They also compared different weighting schemes; TF and OKAPI. The TF (term frequency) and Okapi BM 25 were used to weight the 5-grams and words. These weighted terms (both the 5- grams and the unstemmed words) were then used and in order to determine their comparative effectiveness recall and precision measures were drawn for each group (word and n-gram) for each type of weighting (TF and Okapi BM 25). Results showed that 5-grams using TF weighting did about as well as words using Okapi BM 25 term weighting.

Crowder and Nicholas (1996,1995) made use of n-grams to create descriptions of data (metadata) for a large and dynamic corpus of data. Data were distributed over several (in the count of thousands) of physical locations. As an alternative method to conventional information retrieval techniques (which they reported did not seem to scale), they proposed the use of what they termed mediated agent architecture.

The mediated agent architecture consisted of local server agents managing local corpora and communicating via agent servers and brokers. Local corpora were to be managed as automatically generated, effective metadata. N-gram-based text profiles were created for the data, and these were used by the agents to locate information. Telltale, an n-gram-based information retrieval system, was used to build and manage the n-gram profiles (the lists of unique n-grams and their frequencies for each document). The means (averages) of the n-gram profiles called centroids were used to characterize the collection of n-gram profiles. N was varied from 2 to 3.

N-grams were used because they satisfied the researchers' criteria set for metadata; conciseness, effectiveness in representation, abstractability (centroids of n-grams could also have their own centroids), interchangeability (queries could also be represented by their n-gram profiles) and the quality of being generated automatically. Precision and recall were calculated to measure retrieval effectiveness.

One important observation that Crowder and Nicholas (1996,1995) made in the experiments was that the choice of n seemed related to heterogeneity or homogeneity of the corpus. For a homogeneous corpus, larger n seemed to give better retrieval effectiveness, measured as the number of documents retrieved in response to a certain fixed set of queries. For a heterogeneous corpus, by contrast, the number of documents retrieved in response to a fixed set of queries did not seem to depend on n.

In an IR testing for Oriental languages, Lee, Shin, and Ahn (1996) tested n-gram-based indexing for Korean text. Earlier indexing efforts that made used of word-based and

morpheme-based indexing (reported in ibid.) were compared with results that were obtained from this n-gram based-method.

As reported in their research, the application of word-based indexing to Korean text is made difficult by the fact that sometimes simple nouns are written separately and sometimes as compound nouns, so that the stemming effort (as is common in conventional word-based indexing) reduces a compound noun not to a stem but to another noun. (The process of separating the compound nouns into single, simple nouns is called segmentation and is prevalent in information retrieval research in oriental languages.) As a solution to this problem, the morpheme-based indexing was tested. This approach analyses each word (morphological analysis) to reduce it to the smallest meaningful unit (morpheme). In the experiment, word-based indexing with stemming and n-gram indexing were compared.

The vector space model was used for representing documents and queries. 1-gram, 2-gram, 3-gram, 4-gram, and 5-grams were used. The inner product (dot product) similarity measure was used to determine similarity between documents and queries. The TF/IDF weighting scheme was used. The conclusion drawn from this experiment was that n-gram based indexing has results comparable to word-based indexing and that it could also perform as effectively as morpheme-based indexing with the added advantage of not having to use dictionaries and linguistic knowledge.

A number of experiments and research have been done in Chinese information retrieval using n-grams.

In an experiment for Chinese IR in TREC-6 Leong and Zhou (1998) considered a combination of uni-grams and bi-grams for indexing and reported good performance. In this research bi-grams were used and compared with words. It was established that bi-gram indexes handled unknown words (words not in the dictionary of indexing) and abbreviations in a better way than word indexes. The rationale is that better string matching is achieved if we consider similar portions of words (strings of lesser size than the full word) instead of absolute string matching between whole words. If a bigger portion of two words match then it is very likely that the words are similar in content. The vector space model for representation or encoding, the TF/IDF weighting scheme for weighting index terms and the inner product (dot product) for similarity measure calculation precision and recall for retrieval effectiveness measure were used in this experiment.

In the TREC-5 Chinese track experiment, Tong, Zhai, Mili'c-Frayling, and Evans (1997) explored lexical term indexing (a linguistic approach) and character n-gram indexing (a purely statistical approach) for Chinese text. Linguistic units (words, compound words, and phrases), single Chinese characters, and overlapping character bi-grams were used for the experiment. The CLARIT retrieval system was used for the experiment. Preprocessing of text was done on the Chinese text to perform word segmentation. The preprocessed text (broken down into individual words that would be considered as the indexing terms by the system) was then input to the system for indexing. In the alternative n-gram indexing approach, each sentence in the text was first converted into overlapping n-grams that were separated by spaces. Text in this form was then processed by the CLARIT indexer. Similar to the document text, queries were converted automatically into overlapping n-grams. The resulting character n-grams and terms were weighted based on the section of the topic they originate from (terms in the title a weight of 2, terms in the description a weight of 4, and terms in the

narrative a weight of 1.) This was a weighting scheme that was designed by the team. The precision recall metric was used to measure the indexing performance using the two types of indexes. The conclusions drawn were that single character indexing method was significantly worse than other indexing methods indicating that Chinese characters had low discrimination value since they tended to be too general and ambiguous. On the other hand, the retrieval performance based on overlapping character bi-gram indexing was found to be comparable to that based on lexical term indexing, suggesting that the character bi-grams were useful features in capturing the patterns of Chinese texts, even though a large number of bi-grams are not meaningful linguistic units.

Cavnar and Gillies (1994), described a Digital Libraries Initiative at the Environmental Research Institute of Michigan aimed at the conversion and subsequent perusal of large numbers of paper and microfilm documents in English into electronic form using scanning coupled with optical character recognition OCR (optical character recognition is a technique used to scan text or other paper-based document content into a computer and make the electronic version editable). Any OCR process produces some degree of error in recognition. It follows then that retrieval of such documents is also made difficult because there will be errors that prevent perfect matches.

The n-gram technique was considered by Cavnar and Gillies (1994) as a possible solution for document representation. N-gram matching was used as a technique of inexact matching that would enable access to these products of large-scale document conversion. The natural redundancy present in overlapping n-grams provided a means for matching most sequences of characters in terms. By contrast a conventional word-based matching would miss

misrecognized versions of the same word. In the case of n-grams, even misrecognized forms of the same word share many common bi-grams or tri-grams. For example, if the word INFORMATION, were misrecognized by an OCR system as INFORNATION, the two words would still be matched by the many bi-or tri-grams they shared).

Another advantage that Cavnar and Gillies saw in using n-grams was that n-gram analysis essentially provided word stemming for free. For example, RETRIEVAL and RETRIEVING share 7 out of 10 bi-grams. As a result they can be considered equivalent for retrieval purposes. The fact that both aspects, retrievability and word-stemming, were completely automatic, offered another attraction for the method.

In a technical report of their research findings, Natrajan, Powell, and French (1997) described how they applied n-grams in order to overcome problems of transliteration in indexes produced for Hindi text. They also compared the word-based approach to the n-gram based approach to indexing. They demonstrated that n-grams were highly resistant to garble. Some queries were purposely garbled to a certain extent (up to 25%) to test the system.

Different transliterations of the same Hindi word produced different spellings. In order to find a match between stored records and queries the different transliterations of the words presented a problem. Word-based matching would extract exact matches while n-gram-based methods can produce a large number of potentially matching strings. In cases where misspellings or differences in spellings due to transliteration have to be handled, word-based techniques fail n-gram-based methods do not.

Using n-grams as indexes, the researchers could easily retrieve stored text in spite of differences in transliteration and the garbling. For example in the report, for garbling percentages up to 20%, the correct song (the document) could be located on average over 80% of the time. The vector space model was used for representation of documents, n-grams ranging in size from 1 to 6 were used as indexing terms and TF/IDF weighting scheme was used for weighting the index terms. In their experiment 3-grams were found to be the best for indexes. For similarity measure between query and document (song) vectors, they applied Jaccard's coefficient (one type of similarity measure).

The finding of the research was that n-grams performed at least as well as the word-based approach for all queries and outperformed the word-based approach for queries with a larger transliteration difference.

In a spell-checking experiment Natrajan, Powell, and French (1997) used the same logic as they did for indexing and retrieval. The term that was to be spell-checked was considered as a query term and the dictionary that was to be used for the checking was considered as the document collection. The word was encoded using 1 up to 7-grams of the word. These grams were then similarity checked with the dictionary words. The fragments (n-grams) were weighted simply by using TF (Term Frequency weighting technique). Given a misspelled word, the system returned the top few normalized similarity measures over all the words in the dictionary.

Ekmekçioglu, Lynch and Willett (1996) compared stemming with n-grams for conflation in Turkish texts. As reported in (ibid.), Turkish is a language in which words are a combination of several morphemes and suffixes, the result being that high morphological variance of

words is exhibited in text. Stop word lists were used in the experiment. The values of n used for the n-grams were 2 and 3. A dictionary was created containing all of the word types and the words in the dictionary were represented by their constituent bi-grams and tri-grams. Queries were also represented in a similar manner. The query words were some 50 words selected from the dictionary. The query and text bi-gram and tri-gram representations were then compared for similarity using the overlap similarity measure. The findings were that the use of stop-word lists and a stemming algorithm could bring about substantial reductions in the numbers of word variants encountered in searches of Turkish text databases. Moreover, n-gram matching for conflating such variants had comparable performance with stemming.

In a system that they developed, HAIRCUT (Hopkins Automated Information Retriever for Combing Unstructured Text) McNamee (2001) and his group made use of overlapping character n-grams and simple words in indexing and retrieval of text written in Japanese. They explored the applicability of longer n-grams (6-grams) to unsegmented Japanese text. (Segmentation problems in text analysis had been discussed for Chinese earlier. Like Chinese texts, Japanese texts need segmentation as a process of tokenization when individual words must be derived.) Their research was inspired by an earlier use of long n-grams (e.g. 6-grams) for English that had yielded an effective form of linguistic term normalization. They participated in monolingual Japanese and English retrieval and in cross-language retrieval using each. In a desire to make use of language neutral techniques, they avoided the use of stemming and multi-word phrases.

Their findings were that 6-grams performed comparably with English words and that 2-grams and 3-grams performed equally well in Japanese text. Moreover, uninformed methods (methods that do not make use of linguistic elements) of segmentation and tokenization could

be effective. Recall and Precision measures were used to measure retrieval effectiveness. To weight terms the TF/IDF scheme was used.

CHAPTER THREE

DESIGN AND DEVELOPMENT OF THE PROTOTYPE SYSTEM

3.1. Introduction

This chapter discusses the design and development of the prototype indexing system. It begins by introducing the Amharic writing system. The Amharic alphabets, numerals and punctuation marks are described. The model used for the prototype is also described along with the algorithms used.

3.2. The Amharic Writing System

As cited by Lo (no year), the Blackwell Encyclopedia of Writing Systems defines a writing system as "a set of visible or tactile signs used to represent units of language in a systematic way". There are different classifications for writing systems depending on the way they represent the underlying language. Some examples of the classes are proto-writing, logographic, logophonetic, syllabic, consonantal alphabetic, syllabic alphabetic, alphabetic (ibid.). However, not every script neatly fits into each type. Some scripts share characteristics that belong to different classes. The Ethiopic script used by the Amharic language can be an example of one such script as will be indicated in later sections.

3.2.1. Brief History

The present Amharic writing system was adopted from the Ge'ez writing system. Ge'ez, which belongs to the class of Semitic languages, was the language of literature in Ethiopia in earlier times (Bender et al., 1976). The ancient Sabaean script is in turn attributed as the source of the Ge'ez script. However, as Bender explains, the number of symbols in the original Sabaean script and their shapes, have changed as they descended into Ge'ez and then later on into Amharic. Moreover, some new symbols have been added to Amharic. Amharic

did not discriminate in adopting the Ge'ez fidel; it took all of the symbols (Baye, 1997) and added some of its own. Although Sabaean is not used currently, Ge'ez is still used especially as a language of liturgy (mass) in the Ethiopian Orthodox and Catholic churches and in church literature.

The Sabaean alphabet is said to have had twenty-nine symbols. All twenty-nine had been in use in Ethiopia. When Ge'ez became the spoken and written language, it took over only twenty-four of the twenty-nine symbols (Getachew, 1967). In Ge'ez, two new symbols were created to represent sounds of Greek and Latin loan words, ጰ /p'/ and ፐ /p/ (e.g. ጳጳስ and ፖሊስ Baye(1997)) . When Ge'ez was abandoned as the spoken language and other languages like Tigrignya and Amharic came into being, additional symbols were added to the script (Bender et al., 1976). The new symbols added in the Amharic script are, ሸ(š), ዠ(ž), ቸ(c), ጀ(j), ኘ(n), ጨ(c'), ቨ(v), and ኸ(hɛ).

One of the results in the development from Ge'ez is redundancy in the number of symbols with the same pronunciation. For example, the three different symbols ሀ, ሐ, and ኀ (all with the same pronunciation; h) are used interchangeably in text written in Amharic although they gave different meanings to words in the Ge'ez language. Likewise, both ሠ and ሰ have the same pronunciation (/s/), both ፀ and ጸ (/s'/), and both ዐ and አ (/a/). This redundancy has been recognized in literature (e.g. Getachew, 1967) as a problem of the language.

3.2.2. The Alphabets

The Ethiopic writing system, which the Amharic language uses, consists of a core of thirty-three characters (ፊደል, fidel) each of which occurs in one basic form and in six other forms

all known as orders. The seven orders (the first basic order and the other six orders) of the Ethiopic script represent the different sounds of a consonant-vowel combination (a characterization known as syllabic). The non-basic forms are derived from the basic forms by more-or-less regular modifications (Bender et al., 1976; Hudson, 2001). The 33 core characters then yield 231 distinct symbols. In addition to the 231 characters, there are others that contain special features usually representing labialization, e.g. k (kwe from k – ke). (Ibid.) Refer to Appendix 1 for a complete list of the symbols.

Each symbol represents a consonant together with its vowel. The vowels are fused to the consonant form in the form of diacritic markings. The diacritic markings are strokes attached to the base characters to change their order (e.g. the first order 'le' ለ is transformed into the second order symbol lu "ሉ" by attaching "ˉ" to it; the same first order is transformed into the fifth order symbol lai (as in laid) "ሌ" by attaching "ˌ" to it; and so on). This means that Ethiopic does not use independent symbols for vowels in representing a syllable. This is a characterization known as syllabic (Bender et al., 1976). However, nowadays there is a debate as to whether Ethiopic is actually syllabic or alphabetic (Baye 1997; Hudson 2001). Alphabetic writing systems present the consonants and the vowels separately (Hudson, 2001). Examples of alphabetic writing systems given by Hudson are English and Greek. Ethiopic differs from Greek, in that consonant and vowel are attached (ibid.). Baye (1997) argues that Ethiopic is alphabetic on the grounds that each symbol can be broken down into consonant and vowel phonemes which can be independently represented by separate symbols (e.g. bi ቢ (as in "big") can be broken down into b ˈበ and i ኢ). In fact, he describes the Ethiopic script in terms of 27 consonant and 7 vowel phonemes. From the above, it can be said that Ethiopic is a syllabic-alphabetic script.

3.2.3. Punctuation

There are a number of symbols for punctuation in Amharic. According to Beletu (1982) (as quoted in Zelalem (2001)) , there are about 17 punctuation marks. Only some of them are commonly used and have representations in Amharic software. The following are the most commonly used both in handwritten and computer written text.

Two square dots arranged like a colon (፡ hulet netib) are word delimiters. The equivalent of a full stop is four dots arranged in a square pattern (። arat netib). Some others include equivalents for the comma (፣ netela serez) and the semi-colon (፤ dirib serez), as well as a number of the borrowed symbols (?, ! , " , ", ', /, \, etc.). The word delimiter (two dots) is mostly used in handwritten text. It is more and more becoming the practice to exclude the word separator punctuation (two dots) from computer written text. The space is being used as word separator instead. The sentence delimiter (full stop) however, continues to be used. Other punctuation as comma, colon, or semi-colon are used where appropriate.

3.2.4. Numbers

Numbers in Amharic consist of single characters for one to ten, for multiples of ten (twenty to ninety), hundred, and thousand. (see Appendix 2 for the list). According to Bender et al. (1976), these characters are derived from Greek letters, and some were modified to look like Amharic fidel. Each of the symbols has a horizontal stroke above and below. There is no symbol for zero in the Amharic script. Thus, arithmetical computations using the symbols are very difficult, if ever done. As a result, people generally use the Hindu-Arabic numerals. Ethiopic numbers are used mostly in writing dates and page numbers in text.

3.2.5. Some Characteristics of the Amharic Writing System

The following characteristics of the Amharic writing system have been discussed in literature (e.g. Getachew, 1967; Bender, 1976) as problems of the language.

3.2.5.1. Formation of Compound Nouns

Bender and Ferguson (1976) indicate that compound nouns are sometimes written as two separate words and sometimes as a single word. E.g. "ማዕድ ቤት" and "ማድቤት" which literally mean "dining room". "ብርድ ልብስ" and "ብርድልብስ" which mean "blanket", "ብረድስት" and "ብረት ድስት" which both mean "metal cooking pot", "ቤተመቅደስ" and "ቤተ መቅደስ" which are the same words for "temple" and so on.

3.2.5.2. Problems of Transliteration

When foreign terms are transliterated in Amharic, different spellings may be used; as varied as the number of possible pronunciations (Bender, 1970). E.g. the word "Oxford" may be transliterated as አክስፈርድ (oxferd) or አክስፎርድ (oxford). Such different transliterations usually result in text from use of loan words (words borrowed from other languages and that do not have their own translation in Amharic). Some examples of such loan words are:

ቴሌቭሽን	ቴሌቪሽን	television
እስፖርት	ስፖርት	sport
ዳይሬክተር	ዲሬክተር	director
አውሮፕላን	አይሮፕላን	airplane
ብስክሌት	ቢስክሌት	bicycle
ሻምፒዮን	ሻምፒዎን	champion
ኤሌክትሪክ	ኤሌትሪክ	electricity

ሬፑብሊክ ሪፐብሊክ republic

In all the above cases, the use of n-grams can handle the problem of variation by recognizing words that share a big proportion of strings as equivalent.

For example ኤሌክትሪክ and ኤሌትሪክ share five characters.

3.2.5.3. Different Symbols with the Same Sound

The different symbols with the same pronunciation also pose a problem in making words appear different (not in meaning, but in spelling.) Although in the Ge'ez language, these different symbols give each word different meanings, in the Amharic language they have been used interchangeably (Getachew 1967, Bender et al. 1976). The class of symbols with the same sound falls into two

1. the same sound for the first and fourth order alphabets.
2. different alphabets that share the same sound.

The following belong to the class of the first type. Letters in the same row share the same sound.

1st order	4th order
ሀ	ሃ
ሐ	ሓ
ነ	ኃ
አ	ኣ
ዐ	ዓ

The following is a list of the alphabets that fall in the second class. All alphabets in the same column have the same sound

h

ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	
	ኈ	ኊ	ኋ	ኌ	ኍ	ኯ

s

ስ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ

a

አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ

ts

ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ

E.g. in the two words

እሣት and **እሳት** , the symbols (letters) "**ሣ**" and "**ሳ**" have the same sound and do not change the meaning of the word, which is "fire". Likewise, the two words "**ጸሐይ**" and "**ፀሐይ**", "**ጸሀይ**" and "**ፀሀይ**" all mean the same, "sun" although they are written differently (indicating alternate use of **ጸ** and **ፀ**, and **ሀ** and **ሐ** respectively).

Uniform substitutions may be made for similar sound letters in words to group words by shared strings since such substitutions do not make any changes in meaning in the Amharic language, unlike the Ge'ez in which they have significance for the meaning (e.g. ሰረቀ and ሠረቀ meaning "he stole" and "it penetrated, or rose (as in the rise of the sun)" respectively in Ge'ez, and both meaning either of the two in Amharic). In the processing of text in the experiment, such different symbols have been recognized as equal and they are treated in the preprocessing of the text (refer section 3.4.2.).

3.2.5.4. Different Ways of Writing the Same Word

Different ways of writing (spelling) the same Amharic word are also exercised. For example, "ብራቢሮ" and "ቢራቢሮ" meaning "butterfly", "ኢትዮጵያ" and "ኢትዮጲያ" meaning "Ethiopia".

3.2.6. Amharic Fonts

As reported in Zelalem (2001), Amharic alphabets do not have a representation in the ASCII (American Standard Code for Information Interchange) code table. As a result, font developers have tried to develop their own keyboard driver programs that make use of the existing English keyboard (ASCII codes) for writing Amharic. The English keyboard buttons are used in various combinations to produce Amharic characters.

Different Amharic fonts have been produced over the years (e.g. Alpas, Brana I, Brana II, Power Ge'ez, Geez, Agafari, Alxethiopian, Visual Ge'ez, ...) but they all use the existing symbol sets differently so that an Amharic text written in Ge'ez font cannot be read in another one of the fonts. The need for standardization has been felt and as a result an association has been established in order to undertake the task.

ECoSA (Ethiopian Computer Standards Association) is a professional association established in 1998 to solve problems that result from the disparity in the available different Amharic Software. In order to solve the problem, ECoSA is currently working on standardization issues on Ethiopic including:

- character definition
- keyboard layout
- character encoding
- transliteration

The above-mentioned standardization projects are sponsored by the Ethiopian Quality and Standards Authority. Each one of the projects is handled by a sub-committee consisting of members from various professions (linguists, software developers, etc) from various governmental and non governmental organizations (EcoSA Newsletter, 2000).

The Amharic text used for this research is written in the Visual Ge'ez font. The symbol representation of the Visual Ge'ez Amharic alphabets is attached as Appendix 3. In the Visual Ge'ez font, as in most Amharic fonts, some Amharic characters (those with diacritic markings) extend to more than one byte in their internal representation. The base character one byte and the diacritic marking, another byte. e.g. ቧ which is a composition of ቨ and ◌ , which is internally represented as the two symbols 'b@'. Consideration has been made of this fact in the bi-gram and tri-gram formation in the experiment for this research. A list of the diacritic markings for the Visual Ge'ez font is attached as Appendix 3.

3.3. The Model

The model in the figure below shows the general functions of the prototype indexing system NAATI (N-gram-based Automatic Amharic Text Indexer) and the flow of processes. It depicts the elements of a typical IR system shown in figure 2.1 in chapter two. The primary purpose of the system is to create spaces of terms (i.e. spaces of words , bi-grams, and tri-grams). To this end, the document and query files stored in their respective directories are read and individual words are extracted. From the unique words (the space of words) the bi-grams and tri-grams are generated. Each word is read and the respective bi-grams and tri-grams generated and stored in tables. This defines the spaces of unique bi-grams and tri-grams respectively. Once the space has been defined, then these available terms are used for indexing.

Weights are calculated for each type of term (word, bi-gram, tri-gram) and vectors of weights are generated for each document and query. For retrieval, similarity values are computed for each document-query vector pair. The list of similarity values is then maintained. Effectiveness of the indexing for the three terms is compared by calculating precision-recall values for each type of index and comparing the values.

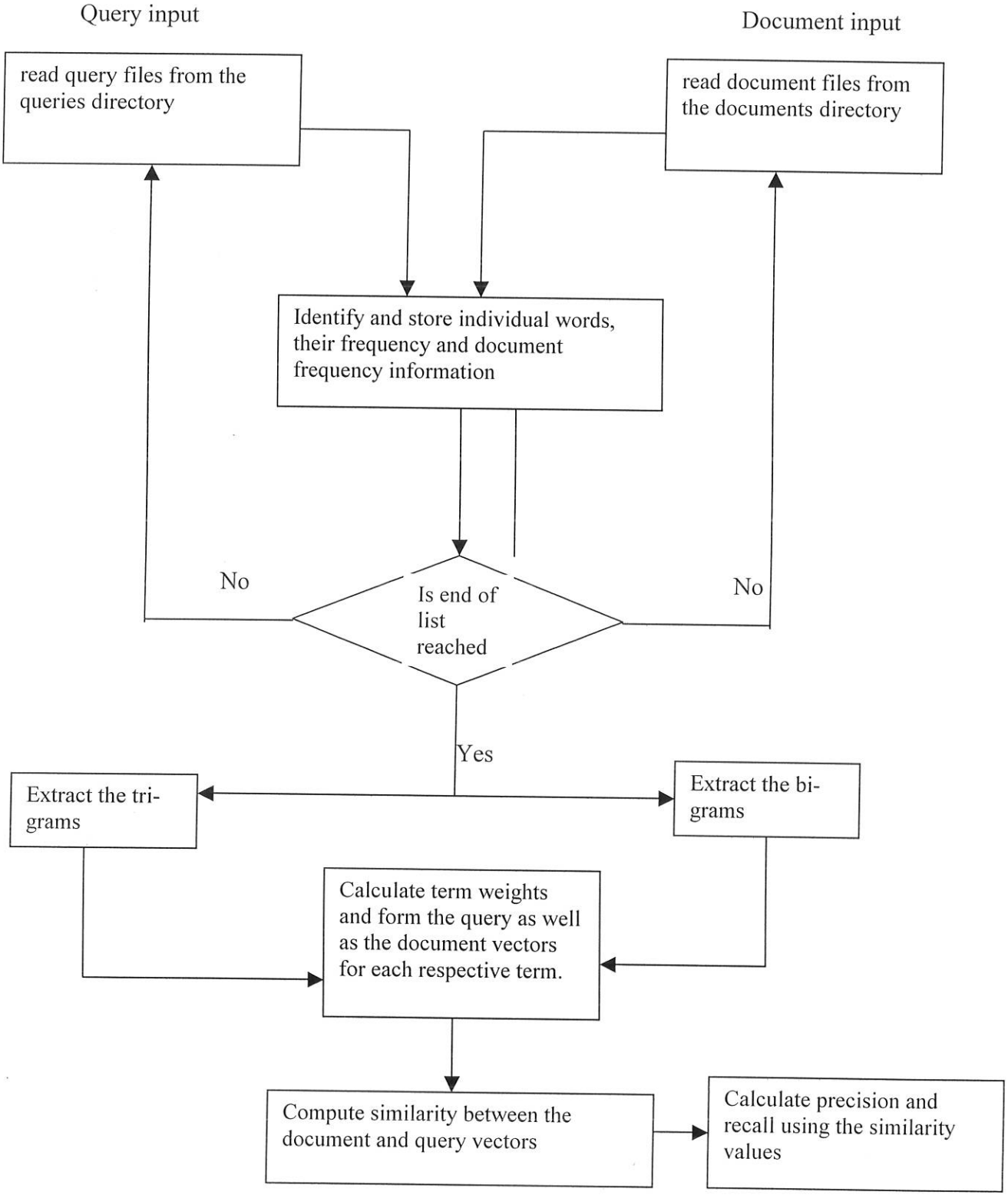

Fig.3.1 The general model for the prototype

3.4. Document Preprocessing

Document preprocessing in general prepares the text for the process that extracts the index terms. It refers to the removal of extraneous characters from the text that are believed to contribute nothing in any way to the content description and that may facilitate processing of the text. Examples of extraneous characters are punctuation marks, control characters (line feed, carriage return, tab, etc.). Preprocessing may also consider changing of cases (upper to lower or vice versa), removal of numbers, etc..

In the Visual Ge'ez font, as is also common with other Amharic fonts, upper case and lower case of the same alphabet represent two different symbols (orders) (Amharic fidel). For example, 'B' is the character used to represent "ሰ " (sixth order) whereas "b" is used to represent "ሀ" (first order), 'K' is used to represent "ኽ" (sixth order) whereas "k" represents "ከ" (first order). Therefore, no case conversion was done as part of text preprocessing.

3.4.1. Removal of Extraneous Characters

The numbers, punctuation marks and control characters in the text of each file were not considered for indexing because they do not contribute anything to content description except for some significant values (e.g. in referring to software versions like Windows-2000, or G8 economic summit, etc.). Words containing numbers like (2nd i.e. 2ኛ or ዋኢ.ማ/1993/01) were excluded at the time of preprocessing. One advantage of using plain text files is that they do not contain much formatting characters as in other, say Word files, although they may cause some symbols to change (see Appendix 6). The standard control characters like carriage return, form feed, line feed were filtered.

3.4.2. Changing Characters to their Common Form

In section (3.2.5.3.) of the previous chapter, a discussion of the different symbols in the Amharic writing system with the same sound was made. These different symbols must be considered as equivalent because they do not cause changes in meaning. As a result, in the experiment, all different symbols of the same sound were converted to one common form in order to exploit this equivalence. Zelalem's (2001) algorithm was used to achieve this. Thus, for example, if the character was one of ሐ ኀ ኃ ሐ ኻ or ሃ (all of them with a similar sound , h) then it was converted to ሀ. By the same token, all orders of ሠ (with the sound s) were changed to their equivalent respective orders of ሰ, all orders of ዐ (with the sound a) were changed to their equivalent respective orders of አ, all orders of ፀ (with the sound tse) were changed to their equivalent respective orders of ጸ. for those orders that use diacritic markings, the base characters are changed and the diacritic markings are attached.

The algorithm is as follows (for each of the seven orders);

1. read the character
2. if the character is any of

   ሐ ኀ ኃ ሐ ኻ ሃ or any other order thereof then

   change it to ሀ

   else if it is ሠ or any other order thereof

   change it to ሰ

   else if it is ፀ or any other order thereof

   change it to ጸ
3. if the character that follows is a diacritic marking, attach it to the changed base

character.

3.5. Word Identification

The word delimiters discussed in section (3.2.3.) are considered in identifying individual words in the texts of the articles. The word identification algorithm of Zelalem (2001) was adapted for the purpose. The algorithm is as follows: -

1. read a character from the text
2. if the character is any one of the delimiters

   then the characters that you have read so far make

   up a word (i.e. the word variable now contains a complete word)

   look up the word and its source file in the table

   if it exists then increment its frequency

   otherwise add it to the table as new with a frequency count of 1

   Refresh the word variable

   Else

   Append the character to the word variable
3. If end of file (no more characters to read) is reached

   then exit
4. go to step 1

Microsoft Access database tables are used to store frequency information about a term. Thus, the list of words along with frequency in each document (TF) and occurrence frequency throughout the collection (DF) is maintained.

The TF and DF data are used to calculate weights for each term by using the following weighting formula discussed in section 2.2.2.4;

$$W = TF * \log(N/DF)$$

The document vectors are formed from these weights of terms.

3.6. Bi-gram and Tri-gram Generation

As discussed in section (2.3.4.1), bi-grams are sequences of two characters extracted from adjacent characters in a word and tri-grams are sequences of three characters extracted from adjacent characters in a word. The bi-grams and tri-grams for the automatic indexing of the texts are extracted from the unique words identified in the previous step (section 3.5).

Information about the frequency of a bi-gram and tri-gram in each document (in-document frequency, TF) as well as DF information (DF denotes the number of documents found in which the term is found) is maintained in Microsoft Access tables separately for the bi-grams and tri-grams.

The in-document frequency (TF) and document frequency (DF) are used in calculating weights for each term (bi-gram and tri-gram). The weights are then used in forming the document bi-gram and tri-gram vectors.

The algorithms used to generate the bi-grams and tri-grams are similar. They are presented below.

Bi-gram formation algorithm

1. Read a word from the words file
2. Pad the word with a single padding character on the left and right sides

3. initialize the bi-gram variable
4. initialize the bi-gram size counter
5. Read the next character of the padded word
6. if the character read is not any one of the diacritic markings, then read the next character
    a. if the next character is a diacritic marking, then merge the two characters and concatenate the resulting merged string to the variable that holds the bi-gram ,
        i. increment the bi-gram size counter by one
    b. else concatenate the character to the variable that holds the bi-gram
        i. increment the bi-gram size counter by one
7. if the bi-gram size counter = 2 then /* we have a bi-gram
    a. check if the bigram already exists,
        i. if it exists, increment its count
        ii. else write the bi-gram to the bi-gram table with a frequency value 1
    b. initialize the bi-gram variable to " "
    c. initialize the bi-gram size counter
8. else read the next character in the padded word (step 5)
9. if the end of file is reached, then exit the routine, else
10. goto step 1

The ')' character, which translates into the question mark in the Amharic symbols table was selected as the padding character for the simple reason that it is one of the symbols not used as a diacritic marking. There is no restriction on the choice of the padding character in the n-gram method.

The algorithm for the tri-grams is different in the number of padding characters it uses and the count of the tri-gram size (which is = 3)

1. Read a word from the words file
2. Pad the word with the two padding characters on the left and right sides
3. initialize the tri-gram variable
4. initialize the tri-gram size counter
5. Read the next character of the padded word
6. if the character read is not any one of the diacritic markings, then read the next character
    a. if the next character is a diacritic marking, then merge the two characters and concatenate the resulting merged string to the variable that holds the tri-gram,
        i. increment the tri-gram size counter by one
    b. else concatenate the character to the variable that holds the tri-gram
        i. increment the tri-gram size counter by one
7. if the tri-gram size counter = 3 then /* we have a tri-gram
    a. check if the tri-gram already exists,
        i. if it exists, increment its count
        ii. else write the tri-gram to the tri-gram table with a frequency value
    b. initialize the tri-gram variable to " "
    c. initialize the tri-gram size counter
8. else read the next character in the padded word (step 5)
9. if the end of file is reached, then exit the routine, else
10. Goto step 1

CHAPTER FOUR

TESTING AND ANALYSIS

4.1. Introduction

This chapter reports on the experiment conducted using the prototype designed in the previous chapter, and the findings from the experiment. It describes the test environment. The experiment was based on the automatic indexing concepts discussed in chapter two.

4.2. The Test Set

According to Oddy (1981), a test collection in IR consists of a static collection of document descriptions (e.g. abstracts, titles)), queries, and relevance judgments. In a setup for a laboratory experiment, the numbers of documents and queries are usually small (reasonable) (e.g. 200, 780, even 82 in Salton and Lesk, 1968). The use of a reasonable size collection for laboratory tests is justified, as Oddy explains from the point of view of the labor and time required to set up the test collection with complete relevance judgments.

To the researcher's knowledge, there is no standard established test collection for Amharic information retrieval testing. Experiments in Amharic IR therefore usually make use of sets of documents and queries set up by the researchers themselves.

For the purpose of this research, the test set consisted of 100 short news articles (obtained from Walta Information Center and used in a previous research) and 24 queries in electronic form. A sample of the text documents is attached as Appendix 4. The queries used in the experiment are attached as Appendix 5. The average document length was 179 words. The

average query length, on the other hand was 5 words. The queries were collected from people who frequently read newspapers and watch current affairs on public media.

The relevance judgments are the lists of documents that have been judged by subject experts to be relevant to each query. The relevance judgments for the test set were prepared by a journalist (subject expert) who identified how many of the documents (articles) are relevant (could be answers to) to each of the queries. They were stored in a Microsoft Access database table

The articles and queries were stored as text files in separate directories from which they were read sequentially in order to generate the vectors of terms (words, bi-grams, tri-grams).

Using the algorithms described in the previous chapter, words, bi-grams and tri-grams were generated and stored in tables along with their frequency information. The following tables depict samples of the contents of the tables.

Table 4.1. Sample Words table in the database

word	docfreq
ኮሚሽኑ	3
ከረድኤት	2
ድርጅቶች	15
ጋር	25
የእርዳታ	4
ስምምነት	5
ትፈራረመ	1
ባህርዳር	2
ጥር	26
የአማራ	4
ክልል	25

Table 4.2. Sample Bi-grams table in the database

Bigram	frequency	Filename
ሽኑ	3	0205933.txt
ኑ?	5	0205933.txt
?ከ	4	0205933.txt
ከረ	1	0205933.txt
ርድ	1	0205933.txt
ድኤ	1	0205933.txt
ኤት	1	0205933.txt
ት?	9	0205933.txt
?ድ	1	0205933.txt

Table 4.3. Sample Tri-grams table from the database

trigram	frequency	filename
??ከ	4	0205933.txt
?ከሚ	3	0205933.txt
ከሚሽ	5	0205933.txt
ሚሽኑ	3	0205933.txt
ሽኑ?	3	0205933.txt
ኑ??	5	0205933.txt
??ከ	4	0205933.txt
?ከረ	1	0205933.txt
ከረድ	1	0205933.txt
ረድኤ	1	0205933.txt
ድኤት	1	0205933.txt

In the tables, for example, the word " **ከረድኤት** " is broken down into the following bi-grams and tri-grams respectively.

?ከ ከረ ረድ ድኤ ኤት ት?

??ከ ?ከረ ከረድ ረድኤ ድኤት ኤት? ት??

As can be observed, the number of tri-grams generated for a word is more than the number of bi-grams. As the value of n gets higher, so will the number of n-grams generated.

For the test set used in this research, the following table presents a statistics of the counts of words, bi-grams, and tri-grams.

Table 4.4. Count of generated terms (words, bi-grams, tri-grams)

Term	Total Number of terms	Number of unique terms
Word	12,232	5,877
Bi-gram	38,166	4,496
Tri-gram	52757	12,308

4.3. Vector Representation of Documents and Queries

As described in chapter two (section 2.2.5.), the vector space model of information retrieval uses vectors to represent documents and queries. The coefficients of the vectors are weights of terms. The representation of documents based on the vector space model in this experiment made use of weights of the three types of terms; words, bi-grams, and tri-grams. The vectors were stored as columns of a table in which the first column represents the list of terms and subsequent columns represent the documents and queries. Each row contained information about the weight of each term in each document/query. In other words, we have a term document matrix.

Term weight calculations

Different term weighting schemes were used for the document terms and the query terms. In Salton and Buckley (1988), as discussed in chapter two, a number of different weighting schemes both for documents and queries are presented. For the document terms, the following formula was used.

$$Wt = TF * \log (N/DF)$$

where TF is the frequency of each term in the respective document.

N is the total number of documents in the collection

DF is the number of documents that contain the term.

For the query terms, the following weighting formula was used

$$W_{iq} = (0.5 + (0.5\ tf_{iq} / \max tf)) \times \log (N/n_i)$$

where W_{iq} is the weight of term i in query q

tf_{iq} is the frequency term i in query q

max tf is the maximum frequency value of all query terms

N is the total number of documents in the collection (this does not include the queries)

n_i is the number of documents in which the query term is found

The following are sample vectors of the documents and queries, where the vectors are the columns.

Table 4.5. Sample document and query vectors for the words

term	doc1	doc2	doc3	doc4	doc5	query42	query43
ኮሚሽኑ	5.058894	0	0	0	0	0	0
ከረድኤት	5.643856	0	0	0	0	0	0
ድርጅቶች	5.473931	0	2.736966	0	0	0	2.736966
ጋር	6	0	0	0	0	0	0
የእርዳታ	9.287712	0	0	0	0	0	0
ስምምነት	12.96578	0	0	0	0	0	0
ተፈራረመ	13.28771	0	0	0	0	0	0
ባህርዳር	5.643856	0	0	0	0	0	0
ጥር	1.943416	0	0	0	0	0	0
የአማራ	4.643856	0	0	0	0	0	0
ክልል	2	0	0	0	0	0	0
አዲጋ	3.836501	0	0	0	0	0	0

Table 4.6. Sample document and query vectors for the bi-grams

term	doc1	doc2	doc3	doc4	doc5	doc6	query4	query7
?ኮ	7.773666	1.943416	0	1.943416	0	0	0	0
ኮሚ	13.68483	0	0	0	0	0	0	0
ሚሽ	19.18251	0	0	0	0	0	0	0
ሽኑ	13.93157	0	0	0	0	0	0	0
ኑ?	2.075187	0.415038	0.830075	1.245113	0	0.830075	0.230576	0
?ከ	0.672491	0.504368	1.008737	0.672491	0	0	0	0
ክረ	3.184425	0	0	0	0	3.184425	0	0
ርድ	1.68966	0	0	0	0	0	0	0
ድኤ	5.058894	0	0	0	0	0	0	0
ኤት	5.058894	0	0	0	0	0	0	0

Table 4.7. Sample document and query vectors for the tri-grams

term	doc1	Doc2	doc3	doc4	doc5	query4	query7
??ኮ	10.94786	2.736966	0	2.736966	0	0	0
?ኮሚ	10.93157	0	0	0	0	0	0
ኮሚሽ	23.21928	0	0	0	0	0	0
ሚሽኑ	16.93157	0	0	0	0	0	0
ሽኑ?	15.17668	0	0	0	0	0	0
ኑ??	4.444843	0.888969	1.777937	2.666906	0	0.555605	0
??ከ	3.243865	2.432899	4.865797	4.054831	0	0	0
?ከረ	6.643856	0	0	0	0	0	0
ክረድ	6.643856	0	0	0	0	0	0
ረድኤ	6.643856	0	0	0	0	0	0
ድኤት	6.643856	0	0	0	0	0	0
ኤት?	6.643856	0	0	0	0	0	0
ት??	5.395159	16.78494	11.38978	7.793007	5.994621	0	0
??ድ	1.556393	1.556393	4.66918	1.556393	0	0	0
?ድር	2.184425	2.184425	4.368849	0	0	0	0

As can be observed from the above tables, more bi-grams are shared by documents than words and tri-grams. This indicates that when documents are compared by their constituent words, they tend to be less similar than if they were to be compared by their constituent bi-grams and tri-grams.

4.4. Similarity Computations

The ultimate goal of any IR system is retrieval of the relevant documents to a set of queries by performing some matching between the document and query representations (Van Rijsbergen, 1975 ; Salton and McGill, 1983). The document and query vectors generated in the previous step were compared to produce a list of similarity values for each document-query pair. The following cosine correlation formula discussed under the vector space model in section (2.2.5) was used to compute the document-query vector similarity.

$$\Sigma x_i.y_i \quad / \sqrt{(\Sigma x_i^2 \, \Sigma y_i^2)} \text{ for } i = 1 \text{ to } t$$

The computed similarity values were then stored in tables to be used later for Recall-Precision calculation. Sample entries in the tables are displayed below.

Table 4.8. Sample entries of similarity computed for the document-query pairs

Doc	query	simval
doc99	query5	0.004258151
doc100	query5	0.000741811
doc7	query7	0.047805032
doc10	query7	0.171096882
doc26	query12	0.094131097
doc41	query12	0.168223522
doc36	query40	0.034921595
doc51	query40	0.044431263

In the table, **simval** indicates similarity value. Sorting the above table by decreasing value of simval for each query gives a ranked list of document query similarity. For the purpose of calculating precision and recall values, a threshold value of similarity was set to 0.01. Any document-query similarity value below this threshold is not considered. This value was set based on the researcher's observation that the document and query having similarity value below 0.01 are not related at all.

4.5. Evaluation, Precision Recall Values

As discussed in section (2.2.4.2.), recall and precision are the most widely used evaluation parameters for an IR system. These two parameters were used in this research. The similarity values derived from the previous step for the document-query pairs for each type of indexing term (words, bi-grams, tri-grams) were used to derive the recall and precision values. Precision (P) and Recall(R) values can be derived for each query or averaged for a number of queries (Salton and McGill, 1983 ; Salton and Lesk, 1968). The formulae specified in section (2.2.4.2.) are used to calculate the recall and precision values for each query.

In order to calculate the precision and recall ratios, relevance information should also be available. For the queries used in this experiment, this information was stored in a database table as shown in part in the following table. This table was used in computing the recall-precision values for all types of index terms (words, bi-grams, tri-grams).

Table 4.9. Sample of the relevance information table.

Query	Reldoc
query4	01039315.txt
query4	0103931.txt
query12	0103937.txt
query12	01049312.txt
query12	0105937.txt
query12	01059311.txt
query43	0107932.txt

If the query-document pair from the similarity table has an entry in this table then the document is relevant, otherwise, it is not.

Precision and recall values can be plotted on a graph by calculating average precision values for controlled recall values ranging, say, from 0.0 to 1.0 at intervals of 0.1 (Salton and

McGill, 1983). This results from the controllability of the number of retrieved documents factor (TotRet). In a ranked list of document-query similarity, recall and precision may be calculated initially when one document has been retrieved, then the first two then the first three, and so on. As an example the following precision and recall values calculated for 'query12' when the first 1 document, then the first 2 documents then the first 3 documents, and so on up till the tenth document are retrieved is displayed.

Table 4.10. Precision and recall calculated at each subsequent retrieval

query	docsret	precision	recall
query12	1	1.000	0.143
query12	2	1.000	0.286
query12	3	1.000	0.429
query12	4	1.000	0.571
query12	5	1.000	0.714
query12	6	0.833	0.714
query12	7	0.857	0.857
query12	8	0.750	0.857
query12	9	0.667	0.857
query12	10	0.600	0.857

Docsret indicates the number of documents retrieved in the order of decreasing similarity value.

In order to compare retrieval performance of different techniques, or different systems, the recall and precision values calculated in this method may be plotted on a single graph where one of the axes denotes recall and the other denotes precision as in the following.

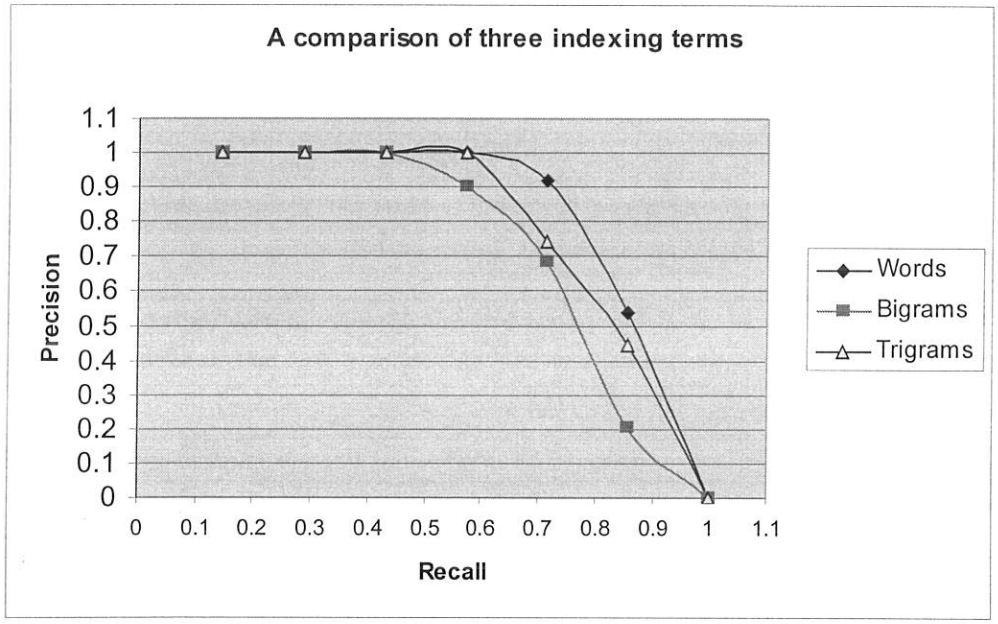

Fig. 4.1. Recall-precision plot for query12 using three types of index terms (words, bi-grams and tri-grams).

In a precision-recall plot like the above, the curve closest to the upper right hand corner of the graph (where recall and precision are maximized) indicates the best performance (Salton and McGill, 1983). In the graph, results obtained for indexing using three different types of terms (words, bi-grams and tri-grams) for query12 are depicted. The following data were used to plot the graph.

Table 4.11. Average recall and precision values for the different terms for a specific query (query12).

Bi-grams		Words		Tri-grams	
precision	Recall	precision	recall	precision	recall
1	0.142857	1	0.142857	1	0.142857
1	0.285714	1	0.285714	1	0.285714
1	0.428571	1	0.428571	0.201973	0.428571
0.816667	0.571429	1	0.571429	0.083406	0.571429
0.682044	0.714286	0.916667	0.714286	0.093553	0.714286
0.156561	0.857143	0.522554	0.857143	0	1
0	1	0	1		

As suggested in Tague (1981), for each specific recall value, the precision values were averaged and a precision value of 0 was assigned to a corresponding recall value of 1 if one did not exist.

Of the three indexing methods, the word-based retrieval is shown to be more effective than the bi-gram and the tri-gram-based retrieval. However, the bi-gram and tri-gram indexes still have comparable performance for indexing to the word indexes. Another observation that can be made is that for the same recall value, precision values for bi-grams and tri-grams are lower than precision values for the word-based indexing. This is due to the larger number of documents retrieved using bi-grams and tri-grams than in the case of words.

For the collection of queries and retrieved documents the following plots show recall and precision values for all queries for each type of term.

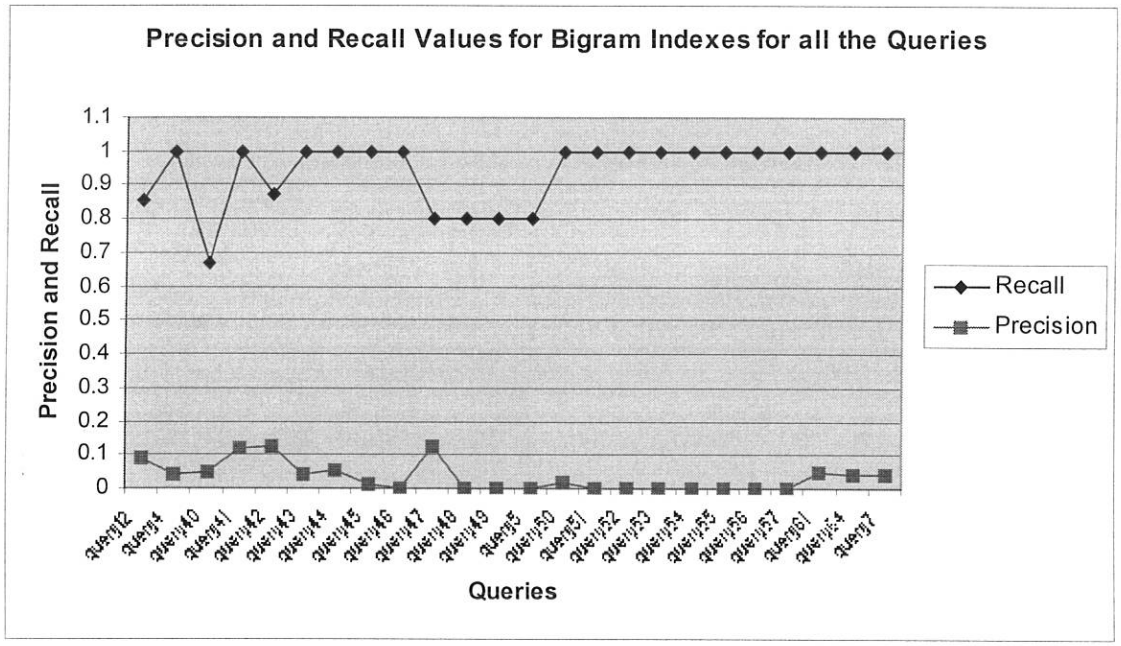

Fig. 4.2. Precision and recall plot for each of the queries for bi-gram indexes.

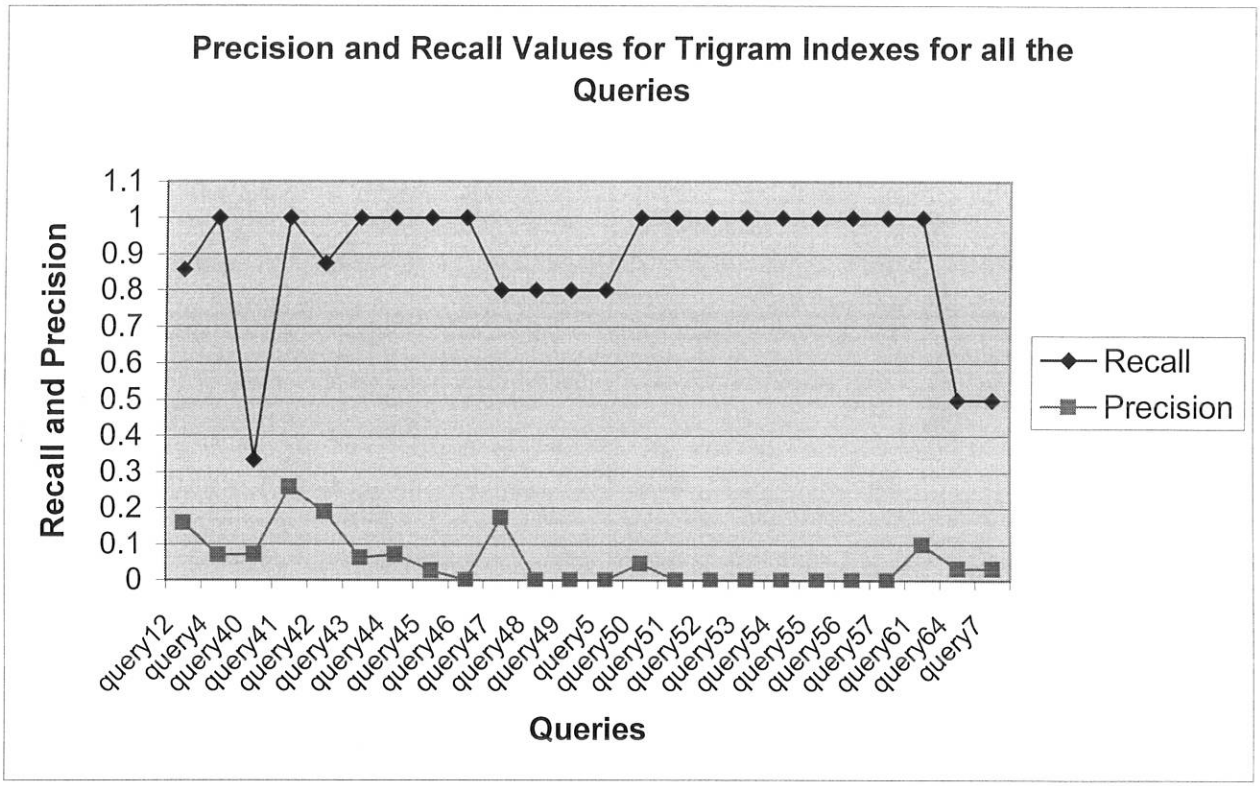

Fig. 4.3. Precision and recall plot for each of the queries for tri-gram indexes.

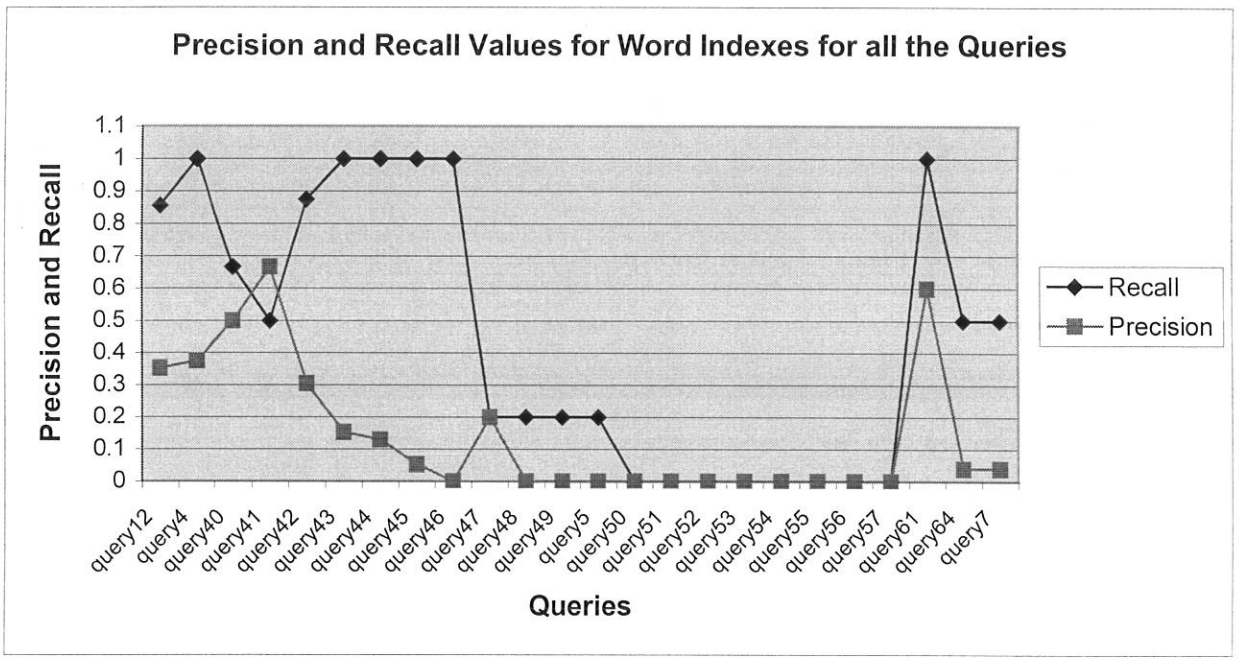

Fig. 4.4. Precision and recall plot for each of the queries for word indexes.

The data used to plot the graphs are attached as Appendix 7.

For the collection of queries and retrieved documents averaged recall and precision values for all three types of terms (words, bi-grams, tri-grams) are compared in the following bar graph. The average values were calculated over the number of queries.

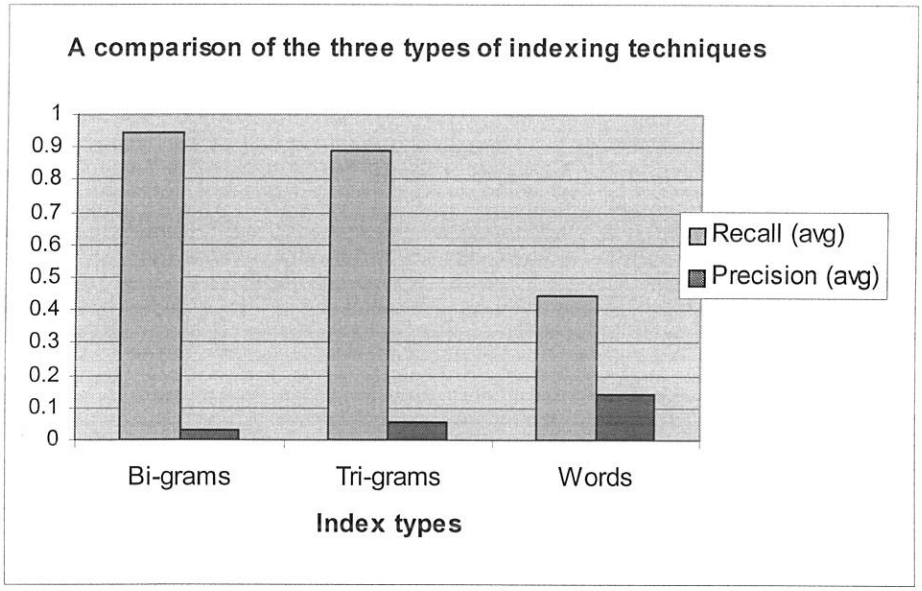

Fig. 4.5. A comparison of the three indexing techniques used in the experiment

The following data was used to generate the bar graph.

Table 4.12. Values used to plot Precision-recall bar graph for the three types of terms

Index term type	Recall (avg)	Precision (avg)
Bi-grams	0.941617067	0.033901066
Tri-grams	0.886061508	0.053769685
Words	0.445783733	0.142327845

As depicted by the graph, recall decreases as we proceed from bi-grams to tri-grams to words and inversely precision increases as we proceed from bi-grams to tri-grams to words.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

This research has presented an n-gram-based method of generating index terms for Amharic text. The n-gram-based method, as discussed in chapter two, is a statistical method of deriving index terms.

The objective of the research was to test the applicability of the n-gram-based indexing method on Amharic text. To this end, 100 Amharic news articles and 24 queries written in the Visual Ge'ez font were selected to build a test set.

Three kinds of terms were considered for indexing. They were, words, bi-grams, and tri-grams. The number of bi-grams and tri-grams generated for indexing is large as discussed in section 3.10. The exponential growth in the number of the n-grams generated from words for higher values of n has implications for processing time and storage. This means that for a large corpus of text, ways of optimizing storage and processing time must be considered in order to work with n-grams.

As briefly mentioned in section (1.6), the test files were saved in .txt format. This caused some degradation (although not significant) in the content causing some symbols to change (refer Appendix 6). However, this has offered itself as an opportunity to test the garble-resistance (as mentioned in section 2.3.4) of the n-gram method.

The comparison of the three techniques for indexing has shown that bi-grams and tri-grams have a comparable performance for indexing as full-word indexes for Amharic. As discussed in chapter two, indexing terms may be characterized as exhaustive or specific. When indexing

exhaustivity is desired, bi-gram and tri-gram (n-gram) indexes might be used instead of word indexes because many documents are retrieved. For precision however, word indexes perform better than bi-grams or tri-grams. This is shown (as depicted in fig. 3.3. and fig. 3.4.) by the low precision and high recall values that result from the large number of retrieved documents.

Although the n-gram method has calls for consideration of storage requirement and processing time, it still offers one method of indexing that is divorced from language- and domain–specific lists (e.g. stopword lists, thesauri, etc.) and rules, which makes the method attractive.

5.2. Recommendations

As part of further research, the following are the researcher's recommendations.

- The size of the collection used in this research is small. Larger collections must be set up and used in order to refine retrieval results. The larger the collection size, the finer the results.
- Although the n-gram method has the shortcoming that it is demanding of storage and processing time, higher values of n should still be tested
- Combinations of different n-grams had also been used for indexing. The same could be tried for Amharic (for example, combinations of both the bi-grams and the tri-grams).
- There is no standard stop word list for use in the Amharic language. The use of a stop word list enhances retrieval performance by removing words from text that do not have any contribution to the description of content of text (for example articles, prepositions, etc.). If a standard stop word list were to be developed, it could be used in such IR researches in Amharic.

- Although in researches related to automatic indexing the TF/IDF weighting scheme is one of the most frequently used, comparisons have been made between systems that use the same type of term but different weighting techniques. One possible area of research could be a comparison of n-gram indexing for a specific n using different weighting techniques. Examples of other techniques are, weighting using term discrimination value, probabilistic term weighting that makes use of probability, signal-noise ratio and OKAPI term weighting..
- Similarity computation techniques other than the cosine coefficient (e.g. Inner product, Jaccard, Dice coefficients) may also be tested for the same type of term.
- Since the n-gram method is a conflation technique, it can be compared to stemming (which is another conflation technique) for Amharic or other Ethiopian languages.
- The type of n-grams used for this experiment is overlapping n-grams. Non-overlapping n-grams may be tested in further research.
- The prototype system is not interactive, it made use of static documents and queries from which the retrieved set information in the form of similarity values was stored in tables for all three types of terms. It could be improved to process queries submitted online and also cater for expansion of the document collection.

REFERENCES

ባዬ ይማምና ቲም፤ 1997፤ "ፊደል እንደገና" የኢትዮጵያ የቋንቋዎችና የሥነ ዕሑፍ መዕሔት፤ ቁጥር 7 (1-32)

Bender, Marvin, L.(1970).Problems of Transliteration into Amharic.*Journal of the Language Association of Eastern Africa*,1(2),112-115.

Bender, M. L., Sydney W. Head, and Roger Cowley .(1976). *The Ethiopian Writing System.* In Bender et al (Eds.) Language in Ethiopia. London: Oxford University Press.

Burnett, J.E., Cooper, D., Lynch, M.F., Willet, P. (1979). Document Retrieval Experiments Using Indexing Vocabularies of Varying Size. I Variety Generation Symbols Assigned to the Fronts of Index Terms. *Journal of Documentation*, 35,197-206.

Cavnar, W.B., Gillies, A.M. (1994) Data Retrieval and the Realities of Document Conversion. at URL http://www.csdl.tamu.edu/DL94/position/cavnar.html

Cavnar, William B. and John M. Trenkle. (1994). N-Gram-Based Text Categorization. at URL http://citeseer.nj.nec.com/68861.html

Chen, Hongbiao. (2001). Looking for Better Chinese Indexes: A Corpus-based Approach to Base NP Detection and Indexing. at URL http://www.in2in.com/hongbiao/biao_e.htm

Cohen, Jonathan D. (1995). Highlights: Language- and Domain-Independent Automatic Indexing Terms for Abstracting. *Journal of the American Society for Information Science*. 46, 162-174.

Cooper, D., and M.F. Lynch. (1982).Text Compression Using Variable- to Fixed-Length Encodings.*Journal of the American Society for Information Science*, 33(1), 18-31.

Crowder G., and Nicholas, C. (1996). Using Statistical Properties of Text to Create Metadata. at URL http://www.computer.org/conferences/meta96/crowder/onefile.html.

Crowder G., and Nicholas, C. (1995) .An Approach to Large Scale Distributed Information Systems Using Statistical Properties of Text to Guide Agent Search. at URL http://www.cs.umbc.edu/cikm/iia/submitted/viewing/crowder.ps

Ekmekçioglu, C.F., Lynch M. F. and Willett P. (1996). Stemming and N-gram Matching for Term Conflation in Turkish Texts. at URL http://informationr.net/ir/2-2/paper13.html

ECoSA Newsletter. (2000), vol. 1, 1.

Ethiopia: Central Statistical Authority(ECSA). 1998. *The 1994 Population and Housing Census of Ethiopia: Results at Country Level.* Vol. 1 Statistical Report 44. Addis Ababa.

Frieder, O., Chowdhury, A., Grossman, D. and McCabe, M.C. (2000). On the Integration of Structured Data and Text: A Review of the SIRE Architecture. at URL http://www.ercim.org/publication/ws-proceedings/DelNoe01/10_Frieder.pdf

Galescu, Lucian, Eric K. Ringger. (1999).Augmenting Words with Linguistic Information for N-gram Language Models. at URL www.cs.rochester.edu/research/cisd/pubs/ 1999/galescu-ringger-eurospeech99.pdf

Getachew Haile.(1967). The Problems of Amharic Writing System.unpublished

Gu, Zhong and Daniel Berleant .(2000). Hash Table Sizes for Storing N-grams for Text Processing. at URL http://class.ee.iastate.edu/berleant/home/me/cv/papers/hashTablesConcise.pdf.

Gustavsson, Jonas. (1996). Text Categorization Using Acquaintance. at URL http://www.student.nada.kth.se/~f92-jgu/C-uppsats/cup.html

Hackett, Paul G. and Douglas W. Oard. (2001). Comparison of Word-Based and Syllable-Based Retrieval for Tibetan. at URL http://www.clis2.umd.edu/dlrg/filter/papers/iral00b.doc.

Harman, D.K. (1995). The TREC Conferences. In R. Kuhlen and M. Rittberger (Eds.) , *Hypertext – Information Retrieval- Multimedia: Synergieeffekte Elektronischer Informationssysteme, Proceedings of HIM'95,* pp. 9-28. [Reprinted in Sparck Jones, K. and Peter Willet (Eds.), *Readings in Information Retrieval.* San Francisco, California: Morgan Kaufmann Publishers Inc., 1997]

Héja, G. (2001). Analysis Of Medical Documents Using N-Grams. at URL http://www.mit.bme.hu/events/minisy2002/heja_gergely.pdf

Hudson, G. (2001). Aspects of the History of Ethiopic Writing, *IES Bulletin,*25,1-10.

Huffman, S. (1995). Acquaintance:Language-Independent Document Categorization by N-Grams. at URL citeseer.nj.nec.com/huffman95acquaintance.html

Jaruskulchai, Chuleerat.(1998). An Automatic Indexing for Thai Text Retrieval. at URL http://www.cs.sci.ku.ac.th/~chulee/PHD-Thesis/

Kimbrell, R.E.(1988).Searching for Text? Send an N-gram!.*Byte*,13(5),297-312.

Lee, J.H., Shin, J. H., Ahn, J. S. (1996). An Effective Indexing Method for Korean Text Retrieval. In Myaeng, S. H. (Ed.) *Proceedings of the Workshop on Information Retrieval with Oriental Languages*.Taejon, Korea: Korea Research and Development Information Center.79-84.

Leong, M., Zhou, H. (1998). Preliminary Qualitative Analysis of Segmented vs. Bigram Indexing in Chinese. at URL trec.nist.gov/pubs/trec6/papers/iss.ps.gz

Leung, Chi-Hong and Kan, Wing-Kay. (1997). A Statistical Learning Approach to Automatic Indexing of Controlled Index Terms. *Journal of the American Society for Information Science,* 48, 55-66.

Lo, Lawrence K. (no year). Types of Writing Systems, at URL http://www.ancientscripts.com/ws_types.html

Luhn, H.P. (1961). The Automatic Derivation of Information Retrieval Encodements from Machine-Readable Texts. In A. Kent (Ed.), *Information Retrieval and Machine Translation,* Vol. 3, Pt 2., pp. 1021-1028. New York: Interscience Publication. [Reprinted in Sparck Jones, K. and Peter Willet (Eds.), Readings in Information Retrieval. San Francisco, California: Morgan Kaufmann Publishers Inc., 1997]

Mayfield, J., McNamee, P. (1998). Indexing Using Both N-Grams and Words. at URL http://trec.nist.gov/pubs/trec7/t7_proceedings.html

McNamee, P. (2001). Experiments in the Retrieval of Unsegmented Japanese Text at the NTCIR-2 Workshop. at URL http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings2/paul.pdf

Miller, E., Dan Shen, Junli Liu, Charles Nicholas, and Ting Chen.(1998). Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System at URL http://jodi.ecs.soton.ac.uk/Articles/v01/i05/Miller/

Natrajan, A., Powell, A. L., French, J.C. (1997). Using N-grams to Process Hindi Queries with Transliteration Variations Technical Report No. CS-97-17 at URL http://www.cs.virginia.edu/~an4m/papers/TechRep-CS-97-17.pdf

Nega Alemayehu. (1999). *Development of a Stemming Algorithm for Amharic Language Text Retrieval.* Ph.D. Thesis. University of Sheffield. (unpublished).

Nie, J., Gao, J., Zhang, J., Zhou, M. (2000). On the Use of Words and N-grams for Chinese Information Retrieval. at URL http://research.microsoft.com/china/papers/Words_NGrams_Chinese_Learning.pdf

Oddy, Robert N. (1981). *Laboratory Tests: Automatic Systems.* in Sparck Jones (Ed.) Information Retrieval Experiment. London:Butterworths

Robertson, A.M., Willet, P. (1996). An Upperbound to the Performance of Ranked-Output Searching: Optimal Weighting of Query Terms Using a Genetic Algorithm. *Journal of Documentation*, 52, 405-420.

Robertson, Alexander M. and Peter Willet. (1998). Applications of N-grams in Textual Information Systems. *Journal of Documentation*, 54, 48-69.

Saba Amsalu Tessera. (2001). *The Application of Information Retrieval Techniques to Amharic Document s on the Web*. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa. (unpublished).

Salton, G., and Lesk, M.E. (1968). Computer Evaluation of Indexing and Text Processing. *Journal of the Association for Computing Machinery*, 15, 8-36. [Reprinted in Sparck Jones, K. and Peter Willet (Eds.), *Readings in Information Retrieval*. San Francisco, California: Morgan Kaufmann Publishers Inc., 1997]

Salton, G. and Michael J. McGill. (1983). *Introduction to Modern Information Retrieval*, New York: McGraw-Hill Book Company.

Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Reading, Massachusetts: Addison-Wesley Publishing Company.

Salton, G., Wong, A., and Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18, 613-620. [Reprinted in Sparck Jones, K. and Peter Willet (Eds.), Readings in Information Retrieval. San Francisco, California: Morgan Kaufmann Publishers Inc., 1997]

Salton, G., Buckley, C. (1988). Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24, 513-523. [Reprinted in Sparck Jones, K. and Peter Willet (Eds.), Readings in Information Retrieval. San Francisco, California: Morgan Kaufmann Publishers Inc., 1997]

Salton, G., Yang, C. S. (1973). On the Specification of Term Values in Automatic Indexing. *Journal of Documentation*, 29, 351-372.

Saracevic, T. (1975). Relevance: a Review of and a Framework for Thinking on the Notion in Information Science. *Journal of the American Society for Information Science, 26, 321-343.* [Reprinted in Sparck Jones, K. and Peter Willet (Eds.), Readings in Information Retrieval. San Francisco, California: Morgan Kaufmann Publishers Inc., 1997]

Sparck Jones, K. (1972) A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation*, 28, 11-21.

Sparck Jones, K., Willet, P. (1997) (Ed.) *Readings in Information Retrieval*. San Francisco, California: Morgan Kaufmann Publishers Inc.

Strzalkowski, T. (1994). Robust Text Processing in Automated Information Retrieval. In *Proceedings of the 4th Conference on Applied Natural Language Processing, pp. 168-173.* [Reprinted in Sparck Jones, K. and Peter Willet (Eds.), *Readings in Information Retrieval*. San Francisco, California: Morgan Kaufmann Publishers Inc., 1997]

Tague, Jean M. (1981). *The Pragmatics of Information Retrieval Experimentatin.* Sparck Jones, K. (ed.), *Information Retrieval Experiment*, London: Butterworths

Tague-Sutcliffe, J. (1992) The Pragmatics of Information Retrieval Experimentation Revisited. *Information Processing and Management*, 28,467-490. [Reprinted in Sparck Jones, K. and Peter Willet (Eds.), Readings in Information Retrieval. San Francisco, California: Morgan Kaufmann Publishers Inc., 1997].

Tauritz, Daniel R., Ida G. Sprinkhuizen-Kuyper. (2000). Adaptive Information Filtering: Evolutionary Computation and N-gram Representation. at URL http://www.cs.unimaas.nl/~kuyper/papers/tauritz-etal00.ps.gz

Tong, X., Zhai, C., Mili'c-Frayling, N., Evans, D. A. (1997) Experiments on Chinese Text Indexing --CLARIT TREC-5 Chinese Track Report In the TREC-5 Chinese track experiment. at URL http://trec.nist.gov/pubs/trec5/papers/CLARIT-Chinese.ps.gz

Van Rijsbergen, C. J. (1975). *Information Retrieval*. London: Butterworths.

Wechsler, M. and Peter Schäuble. (1995). Speech Retrieval Based on Automatic Indexing. At URL http://citeseer.nj.nec.com/wechsler95speech.html

Wiesniewski, J.L. (1987). Effective Text Compression with Simultaneous Digram and Trigram Encoding. *Journal of Information Science*, 13, 159-164.

Zamora, A. (1980). Automatic Detection and Correction of Spelling Errors in a Large Data Base. *Journal of the American Society for Information Science, 31, 51-57.*

Zelalem Sintayehu. (2001). *Automatic Classification of Amharic News Items: The Case of Ethiopian News Agency*. (Masters Thesis). School of Information Studies for Africa. Addis Ababa University. Addis Ababa. (unpublished).

Zhao, Jie.(2000). Network and N-gram Decoding in Speech Recognition at URL http://www.isip.msstate.edu/publications/books/msstate_theses/2000/decoding/thesis_v1.pdf

APPENDICES:

Appendix 1. The Amharic character set (Bender *et al.*, 1976).

Order							Labialized				
1st	2nd	3rd	4th	5th	6th	7th					
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ			ሏ		
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ					
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ			ሟ		
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ					
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ			ሯ		
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ			ሷ		
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ			ሿ		
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቊ	ቋ	ቌ	ቍ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ			ቧ		
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ			ቷ		
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ			ቿ		
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኊ	ኋ	ኌ	ኍ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ			ኗ		
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ			ኟ		
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ					
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	ኲ	ኳ	ኴ	ኵ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ					
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ			ዟ		
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ					
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ					
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጒ	ጓ	ጔ	ጕ
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ			ዷ		
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ					
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ			ጧ		
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ			ጯ		
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ			ጿ		
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ			ፏ		
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ					

ᗅ	ᗆ	ᗇ	ᗈ	ᗉ	ᗊ	ᗋ

Appendix 2: Amharic numbers

፲	፳	፴	፵	፶	፷	፸	፹	፺	፻
10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**

፩	፪	፫	፬	፭	፮	፯	፰	፱
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

Appendix 3: List showing the symbols used in the Visual Ge'ez font for the Amharic fidel

ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
h	h#	£	¦	ÿ	H	ç
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
l	l#	l!	§	l@	L	lÖ
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
/	/#	/!	ˆ	/@	?	‡
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
m	Ñ	¸	¥	»	M	ä
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
\	\#	œ!	œ	œ@	|	*f*
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
r	„	¶	‰	Ê	R	é
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
s	s#	s!	ú	s@	S	î
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ
¹	¹#	¹!	š	¹@	>	ë
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
q	q$	qE	”	³	Q	ö
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
b	b#	b!	Æ	b@	B	ï
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
t	t$	tE	¬	t&	T	è
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ
c	c$	cE	Ò	c&	C	Ó
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
x	x#	x!	¨	x@	X	å
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
n	n#	n!	Â	n@	N	ñ
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ
ß	ß#	ß!	¾	ß@	"	®
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
k	k#	k!	µ	k@	K	÷
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ
,	,#	,!	–	,@	<	—
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ
w	ý	ê!	ê	ê&	W	ã
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
;	;#	›!	›	›@	:	â
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
z	z#	z!	²	z@	Z	ø
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ
¢	¢$	¢E	Ï	¢&	™	Î
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
y	†	‘	Ã	ü	Y	×
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
d	Ç	d!	Ä	Á	D	ì

ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ
j	°	©!	©	Ë	J	í
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
g	g#	g!	U	g@	G	¯
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
-	-#	-!	È	-@	_	õ
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ
=	=#	À	Å	~	+	ô
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
’	’#	’!	Ô	’@	ù	Õ
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ
]	]#	É!	É	É@	}	ò
ጸ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ
[	[	[!	Ú	[@	A	Û
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
f	û	ð	Í	Ø	F	æ
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ
p	p$	pE	-	p&	P	±
ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ
v	v#	v!	Š	v@	V	‹
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ
%	Ð	–	—	,	“	a
ቘ	ኗ	ኟ	ኧ	ኣ	ኞ	
¡	•	…	Þ	à	O	
ጫ	ጯ	ጿ	ፇ	ጓ	ጔ	ጧ
þ	ª	Ÿ	Ì	Ù	`	Æ*

The diacritic markings,

!	@	#	$	&	*	E

Appendix 4: Sample of text (news article) used for the indexing

```
yx!¥tLf PéjKèC k:QD bðT Xyt-Âqq$ nW

xRÆ MN+ HÄR 11/419931/4êx!¥1/4 yx!T×ùÃ ¥^b‰êE t/DîÂ L¥T fND b81 n_B 1 ¸l!yN
BR wÀ ÃSjm‰cW yN]#H m-_ W` PéjKèC ktÃz$cW yg!z@ sl@Ä bðT Xy-Âqq$ mçn# ys"N
åä øN W`1/2 ¥:DNÂ x!nR(c)! mM¶Ã gl]ÝÝ

ymM¶ÃW `$ð xè h#s@N xyl sän#N XNdgl]#T xMÂ ktjm"TÂ bzNDé ymjm¶Ã "B ›mT m=rš
m-ÂqQ ynbrÆcW x‰T yW` PéjKèC bxµÆb!W ^BrtsB yg#LbT DUF b"B ›mt$ mjm¶Ã t-
ÂqêLÝÝ

bmÀW sn@ l¥-ÂqQ ytÃz#T z-" mlSt3/4Â _LQ W` g#DÙìC q$Íé1/2 h#lT Æ*NÆ*ãC
mzRUT b^Brtsb# TBBRÂ bmM¶ÃW ÆlÑÃãC KTTL 90 bmè yGMÆ¬ dr(c) $Y XNd¸gß#Â bmÀW
yµtET Ñl# bÑl# t-ÂQqW xgLGlÖT XNd¸s-# xS¬WqêLÝÝ

PéjKèc$ Ñl# bÑl# xgLGlÖT mS-T s!jM" 67 ¹! 500 yg-"N nê¶ yN]#H W` t-"¸
XNd¸ÃdRg#Â 24 bmè ynbrWN yøn#N yN]#H m-_ W` >ÍN wd 26 bmè XNd¸ÃúDg# mGlÉcWN
êL¬ x!NæR">N ¥:kL zGÆ*LÝÝ
```

The above text in Visual Ge'ez font:

የኢማተልፊ ፕሮጀክቶች ከዕቅድ በፊት እየተጠናቀቁ ነው

አርባ ምንጭ ህዳር 11ሐ419931ሐ4ዋኢማ1ሐ4 የኢትዮጵያ ማኅበራዊ ተሐድሶና ልማት ፈንድ በ81 ነጥብ 1 ሚሊየን ብር ወጪ ያስጀመራቸው የንፁህ መጠጥ ውኃ ፕሮጀክቶች ከተያዘላቸው የጊዜ ሰሌዳ በፊት እየጠ ናቀቁ መሆኑ የሰኝን አሞ ዞን ውኃ1ሐ2 ማዕድንና ኢነር-ቸ?¸ መምሪያ ገለፀ::

የመምሪያው ኃላፊ አቶ ሁሴን አየለ ሰሞኑን እንደገለፁት አምና ከተጀመኝትና በዘንድሮ የመጀመሪያ ኝብ ዓመት መጨረሻ መጠናቀቅ የነበረባቸው አራት የውኃ ፕሮጀክቶች በአካባቢው ኅብረተሰብ የጉልበት ድጋፍ በኝብ ዓመቱ መጀመሪያ ተጠናቀዋል::

በመጪው ሰኔ ለማጠናቀቅ የተያዙት ዘጠኝ መለስተ3ሐ4ና ጥልቅ ውኃ ጉድጓዶች ቁፋሮ1ሐ2 ሁለት ባንባዎች መዘርጋት በኅብረተሰቡ ትብብርና በመምሪያው ባለሙያዎች ክትትል 90 በመቶ የግምባታ ደረ-ቸ? ላይ እንደሚገኙና በመጪው የካቲት ሙሉ በሙሉ ተጠናቅቀው አገልግሎት እንደሚሰጡ አስታውቀዋል::

ፕሮጀክቶቹ ሙሉ በሙሉ አገልግሎት መስጠት ሲጀምኝ 67 ሺ 500 የገጠኝን ነዋሪ የንፁህ ውኃ ተጠኝሚ እንደሚያደርጉና 24 በመቶ የነበረውን የዞኑን የንፁህ መጠጥ ውኃ ሽፋን ወደ 26 በመቶ እንደሚያሳድጉ መግለፃቸውን ዋልታ ኢንፎርኝሽን ማዕከል ዘግቧል::

Appendix 5: The queries used in the experiment

1. ኤች አይ ቪ ኤድስን ለመግታት እየተደረጉ ያሉ እንቅስቃሴዎች	query12
2. ንጹህ የመጠጥ ውኃን ለማዳረስ የሚከናወኑ የውኃ ፕሮጀክቶች	query4 *
3. የሰብአዊ መብት ረገጣ	query40
4. የኢትዮጵያና ኤርትራ የድንበር ግጭት	query41 **
5. የከፍተ3ሐ4 ትምህርት ተ□ት	query42
6. መንግስታዊ ያልሆኑ ድርጅቶች ወይም ግብረ ሰናይ ድርጅቶች የሚያደርጉት የልማት እንቅስቃሴ	query43
7. የውጭ መንግስት ዕር□	query44
8. የግብርና ክፍለ ኢኮኖሚውና የኤክስቴንሽን መርሀ ግብር	query45
9. የተወካዮች ምክር ቤት አመታዊ ጉባኤ	query46
10. ህገወጥ የንግድ እንቅስቃሴዎች	query47
11. የአገር ውስጥ ገቢ ባለስልጣን የስራ ሒደት	query48
12. ግብር ስለማይከፍሉ ነጋዴዎች	query49
13. ሙስናና ምዝበራ በሀገሪቱ ላይ እያስከተሉ ያለው ቀውስ	query5
14. ብሔራዊ ቅርሶች	query50
15. ስለአድዋ ድል ታሪክ	query51
16. በኢትዮጵያ በተደጋጋሚ ስለሚከሰተው ድርቅ	query52
17. በኢትዮጵያ ከገጠር ወደ ከተማ ስለሚደረግ የሕዝብ ፍልሰት	query53
18. በአዲስ አበባ ስላለ የትራፊክ ችግር	query54
19. በአዲስ አበባ ስላለ የንፅህና ችግር	query55
20. በአዲስ አበባ ስላለ የሕዝብ ብዛት	query56
21. ስለኢትዮጵያና ኤርትራ ወቅታዊ የድንበር ግጭት ሁኔታ	query57 **
22. ሴቶችን አስገድዶ መድፈር እና ተመሳሳይ ጥቃቶች	query61
23. የፍርድ ቤቶች የስራ እንቅስቃሴ	query64
24. የመጠጥ ውኃ ፕሮጀክቶች	query7 *

* , ** - duplicates

Appendix 6: The characters that were changed

Original character	Changed to
ጀ	-ቸ?
፣	1ሐ2
ሚ	ኝ
ሩ	ኝ
ኖ	3ሐ4
ቃ	ኝ
/	1ሐ4
ኸ	-ተመ?
ኾ	-ረ?

Appendix 7: The precision and recall data used to plot the graphs in figures 4.2, 4.3 and 4.4 respectively.

Bi-grams		
query	Recall	Precision
query12	0.857143	0.090909
query4	1	0.044118
query40	0.666667	0.04878
query41	1	0.121212
query42	0.875	0.125
query43	1	0.044444
query44	1	0.052632
query45	1	0.014493
query46	1	0
query47	0.8	0.125
query48	0.8	0
query49	0.8	0
query5	0.8	0
query50	1	0.017857
query51	1	0
query52	1	0
query53	1	0
query54	1	0
query55	1	0
query56	1	0
query57	1	0
query61	1	0.04918
query64	1	0.04
query7	1	0.04

Trigrams		
query	Recall	Precision
query12	0.857143	0.157895
query4	1	0.069767
query40	0.333333	0.071429
query41	1	0.258065
query42	0.875	0.189189
query43	1	0.0625
query44	1	0.071429
query45	1	0.026316
query46	1	0
query47	0.8	0.173913
query48	0.8	0
query49	0.8	0
query5	0.8	0
query50	1	0.045455
query51	1	0
query52	1	0
query53	1	0
query54	1	0
query55	1	0
query56	1	0
query57	1	0
query61	1	0.1
query64	0.5	0.032258
query7	0.5	0.032258

Words		
query	Recall	Precision
query12	0.857143	0.352941
query4	1	0.375
query40	0.666667	0.5
query41	0.5	0.666667
query42	0.875	0.304348
query43	1	0.153846
query44	1	0.130435
query45	1	0.052632
query46	1	0
query47	0.2	0.2
query48	0.2	0
query49	0.2	0
query5	0.2	0
query61	1	0.6
query64	0.5	0.04
query7	0.5	0.04

DECLARATION

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

Bethlehem Mengistu Hailemariam

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

Dr. Abebe G/Tsadik

Ato Werkshet Lamenew

Wzt. Saba Amsalu