ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

AMHARIC-ENGLISH BILINGUAL SEARCH ENGINE

By: Mequannint Munye Zeru

A THESIS SUBMITTED TO

THE SCHOOL OF GRADUATE STUDIES OF THE ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

November, 2010

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF COMPUTER AND MATHEMATICAL SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

AMHARIC-ENGLISH BILINGUAL SEARCH ENGINE

By: Mequannint Munye Zeru

ADVISOR:

Solomon Atnafu (PhD)

APPROVED BY

Examining Board:

1. Dr. Solomon Atnafu, Advisor _____
2. _____ _____
3. _____ _____

DEDICATION

To my father: may God bless you and put your soul in heaven, I appreciate your efforts and struggling to show me the route that brings me to the today's fruit. But, unlucky to see it.

To my mother: I appreciate your struggle, patient and hope to bring me here.

To my brother: אחי, do you think that without you I can reach here?

Acknowledgements

This thesis is the harmonized results of many people. Today, I am highly delighted because I have got the opportunity to express my gratitude for all of them.

Every work could be successfully accomplished because of the will and interests of the almighty **God**. So, all my first praises and thanks are reserved to him. Then, I would like to extend my heartfelt and deepest sense of gratitude to my advisor **Dr. Solomon Atnafu** for his continuous and unreserved support, guidance, constructive comments and suggestions. It is with his continuous follow up, encouragement and advice that this thesis came to realization. I appreciate his dedication from the beginning to the end to teach me for the first time how to conduct and write a research.

I would also like to take this opportunity to express my profound gratefulness to my families for their continuous moral and financial support. I love you all. **Tsegaw**, you are really a brother and a role model for all brothers to treat their brothers as you do. I appreciate your inspiration, hope and patience to follow up me throughout my study.

My sincere thanks also go to my friends who are with me from my undergraduate years in Haramaya to my post graduate here in Addis Ababa to the end of this thesis. **Simachew Endale** and **Mehari Bayou**: you have been in my life ever since we started our higher education to this day and no stones unturned with you to get the today's achievement. You were my stimulator to start my study every day. I cherish our many holidays together and hope that we will be able to stay in touch despite the large distances between us.

I would also like to extend my gratitude to Jijiga University for sponsoring me to attend my second degree. I also want to thank Graduate Committee of Addis Ababa University Computer Science Department for giving me the permission to commence this research work. Since this thesis is the cumulative results of the two years learning, I would like to thank all the staff members of the department of Computer Science involved in the process as well as my classmates for working together in different projects and assignments in harmony.

At last but not the least, I would like to thank all those people who have fruitful and valuable assistance and their finger prints in this work.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Appendices

Acronyms and Abbreviations

API	Application Program Interface
ASCII	American Standard Code for Information Interchange
ASP	Active Server Page
BLIR	Bilingual Information Retrieval
CLEF	Cross-Lingual Evaluation Forum
CLIR	Cross-Lingual Information Retrieval
DHTML	Dynamic Hyper Text Markup Language
EBMT	Example Based Machine Translation
HMT	Hybrid Machine Translation
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IR	Information Retrieval
JDBC	Java Database Connectivity
JSP	Java Server Page
MLIR	Multilingual Information Retrieval
MRD	Machine Readable Dictionary
MT	Machine Translation
NER	Named Entity Recognizer
NII	National Institute of Informatics
NLP	Natural Language Processing
NTCIR	NII Test Collection for Information Retrieval
OOV	Out-Of-Vocabulary
POS	Part-Of-Speech
PRP	Probabilistic Ranking Principle
RBMT	Rule Based Machine Translation
SDK	Software Development Kit
SMT	Statistical Machine Translation
TREC	Text REtrieval Conference
URL	Uniform Resource Locator
VSM	Vector Space Model
XML	eXtensible Markup Language

ABSTRACT

As non-English languages have been growing exponentially on the Web with the expansion of multilingual World Wide Web, the number of online non-English speakers who realizes the importance of finding information in different languages is enormously growing. However, the major general purpose search engines such as Google, Yahoo, etc have been lagging behind in providing indexes and search features to handle non-English languages. Hence, documents that are published in non-English languages are more likely to be missed or improperly indexed by major search engines. Amharic, which is the family of Semitic languages and the official working language of the federal government of Ethiopia, is one of these languages with a rapidly growing content on the Web. As a result, the need to develop bilingual search engine that handles the specific characteristics of the users' native language query (Amharic) and retrieves documents in both Amharic and English languages becomes more apparent.

In this research work, we designed a model for an Amharic-English Search Engine and developed a bilingual Web search engine based on the model that enables Web users for finding the information they need in Amharic and English languages. In doing so, we have identified different language dependent query preprocessing components for query translation. We have also developed a bidirectional dictionary-based translation system which incorporates a transliteration component to handle proper names which are often missing in bilingual lexicons. We have used an Amharic search engine and an open source English search engine (Nutch) as our underlying search engines for Web document crawling, indexing, searching, ranking and retrieving.

To evaluate the effectiveness of our Amharic-English bilingual search engine, precision measures were conducted on the top 10 retrieved Web documents. The experimental results showed that the Amharic-English cross-lingual retrieval engine performed 74.12% of its corresponding English monolingual retrieval engine and the English-Amharic cross-lingual retrieval engine performed 78.82% of its corresponding Amharic monolingual retrieval engine. The bilingualism advantage of the system is also evaluated by comparing its results with general purpose search engines. The overall evaluation results of the system are found to be promising.

Key Words: *Bilingual search engines, cross-lingual information retrieval, query preprocessing, query translation, transliteration.*

CHAPTER ONE

INTRODUCTION

1.1 General Background

The rapid growth of online non-English speakers that use the World Wide Web as their major source of information and a means of communication channel, has led to enormous amount of information in different languages and encoding schemes to be online on the Web. This creates new challenges in information search and retrieval since information search and retrieval needs language specific treatment. According to [2, 13], general purpose search engines, such as Google, Yahoo, etc, often ignore the special characteristics of non-English languages, and sometimes they do not even handle diacritics well. However, non-English speaking users use these search engines that do not take into account the structure and the special characteristics of the specific language they use, because they may not have alternatives. This has led to the need to develop language specific search engines instead of general purpose once.

However, developing search engines that support only a specific language and writing scheme do not allow the users to access all relevant documents available on the Web. This is because; there may be relevant documents in response to the user's query that are not in the same language and script of the query language particularly when the query language is not mostly used on the Web.

In recent years, multilingual information retrieval (also known as "translingual" or "cross-lingual" IR) or bilingual IR for the case of two languages has got considerable attention with the increased accessibility of diverse online information on the Web [10]. Multilingual information retrieval (MLIR) systems take the advantage of the multi-lingual knowledge of users and present the results in second or third language of which the user is aware of [11], which is a better option than a null or not desired result. Multilingual information retrieval involves providing a query in one language and searching document collections in one or more different languages. Since the query and the document are written in different languages, either the query or the document should be translated.

According to [9], many bilingual information retrieval (BLIR) systems have used a query translation approach since translating the document needs large bilingual corpus particularly in morphologically rich languages. In such cases, the documents are indexed in their source language(s), the query is translated into each of the document languages, and the retrieval is done in the source languages. An alternative approach of bilingual information retrieval is to do retrieval in the query language by indexing the document translations in the query language and searching using the original query. Both query translation and document translation attempt to map the query and the documents into a common language, but they use different translation strategies. In query translation, a query may be short and non-grammatical with little or no context, whereas in document translation, each document is a large, coherent context with full, grammatical sentences. On the other hand, once a document is translated, any mistakes or deletions in the translation cannot be remedied, whereas translating a query allows for more flexibility in incorporating multiple possible translations using synonyms and related terms. Because of these reasons, we chose the first method for our system. However, in our system, the documents are crawled, indexed, searched, and retrieved for both the query as well as the document languages.

Any Cross Language Information Retrieval (CLIR) system integrates a machine translation system as its core component to translate either the query into the document language or the documents in the query language. The machine translation (MT) system may use different approaches of machine translation such as Dictionary-based, Rule-based, Corpus-based, etc. The selection of the translation approach may depend on the linguistic resources available for the languages, whether the translation is query translation or document translation, the requirements of the translation system and others. Having these MT approaches selection criteria in mind, we have followed dictionary based MT approach for our system.

If the translation is dictionary based, there is one serious problem in every translation system. This problem is the word coverage limitations of dictionaries because of the appearance of Out-Of-Vocabulary (OOV) words in the query. This often occurs because most of the queries contain proper names and borrowed words that do not often present in the bilingual lexicons, i.e. which may be OOV and do not demand the need of dictionary [11].

As a result, the translation system, in bilingual information retrieval, should also integrate a transliteration component to alleviate the problem of OOVs and to translate them into the target language without using dictionary. The OOV problem is more exacerbated for applications involving translation between languages that use different scripts like English and Amharic [8]. So, bilingual applications should also deal with proper names and their transliterations to another language and script.

1.2 Statement of the Problem

Amharic is the second most-spoken Semitic language in the world, after Arabic, and the official working language of the Federal Democratic Republic of Ethiopia (A country with more than 78 million populations). It has been the working language of the government, the military, and of the Ethiopian Orthodox Tewahedo Church throughout modern times. Outside Ethiopia, Amharic is the language of some 2.7 million emigrants (notably in Egypt, Israel, and Sweden) [15, 16]. Thus, it has official status and is spoken by many people as their native and second language. Of these who speak Amharic, a significant number of them (usually the educated class) can understand and speak English as well. The percentage of Ethiopian official working language, Amharic, content on the Web is very less compared to English. However, the contents could be different than contents that are published in English language, as contents in Amharic are released mostly to Ethiopian events. Thus, using only one of the languages do not allow for accessing all the available relevant documents.

There are significant amount of works about Amharic-English information retrieval [1, 3, 4] and a little about Amharic search engines [2, 17]. However, as the knowledge of ours, there is no attempted works about Amharic-English information retrieval on the Web (i.e. bilingual Amharic-English search engine) that accepts user queries according to the user's language preferences and returns the result in both languages. Therefore, Monolingual Amharic speaker Web users have to know English language to use the Web.

1.3 Motivation

With the availability of vast amount of information on the Web, the interest of the people to use search engines to locate information of interest is highly increasing. As the currently available search engines do not handle all the Web documents written in different languages well, there is a need to develop language specific search engines. Amharic is one of the languages which have rapidly growing content on the Web. It has a complex morphology which combines consonantal roots and vowel intercalation. Amharic and English differ substantially in their morphology, syntax and the writing system they use. Therefore, the search engines which are mainly developed for English cannot efficiently be used to retrieve Amharic documents.

Until recently, Ethiopian Web users have to know English language because there are only few alternatives in their native language to use the Web. To alleviate such problems, an Amharic search engine that searches the Web for Amharic documents has already been developed [2, 17]. However, the majority of Web documents are published in English when compared to Amharic language and accessing only Web documents which are written in Amharic may not satisfy user requirements since there may be relevant documents for user queries in English as well.

To address this problem, we are motivated to work on a bilingual (Amharic-English) search engine which pulls up results in both languages for user queries in his/her language of preference. This makes it possible for users to bilingually search the World Wide Web, Images, Videos, Wikipedia, etc. The bilingual search engine for Amharic-English will make it easy for the bilingual Web users to operate and work with the Web, as they are not expected to formulate queries in both languages. Its main advantage will be allowing the Web user to formulate query in one language and get the Web search results in both languages. The people are also able to use the Web in their own native language to query search engines that has contents of their events of interest written in both languages (Amharic and English).

1.4 Objectives

1.4.1 General Objective

The general objective of this research work is to design and develop a generic model for a bilingual search engine that integrates a query translation system for Amharic and English languages.

1.4.2 Specific Objectives

The specific objectives of the work are:

- Studying and analyzing the language specific behaviors of Amharic and English.
- Analyzing the Amharic/English and English/Amharic query translation requirements.
- Designing a bidirectional bilingual (Amharic/English) search engine model.
- Exploring Amharic preprocessing tools to prepare the query for translation.
- Exploring and integrating Amharic-English bilingual dictionary for the purpose of translating queries from one language to the other.
- Developing a bidirectional Amharic/English dictionary based machine translation system for the purpose of query translation.
- Developing and integrating a transliteration system for proper name and borrowed word transliterations which do not often present in the Amharic-English bilingual dictionary but frequently occurs in queries.
- Designing an interface that accept user's query either in English or Amharic language.
- Exploring and adopting Web based search engines for both Amharic and English languages.
- Developing an algorithm that integrates Amharic and English search engine results in order to search both Amharic and English Web documents
- Developing the prototype of our work and demonstrating the effectiveness of the proposed system.

1.5 Scope and Limitations of the Study

There are several Web documents that are written in different languages, scripts and encodings and also there are different kinds of Web documents such as image, audio, video and text. However, this research work considers only text documents on the Web that are written in Amharic and English languages.

A full functional bilingual search engine requires a number of tools to be developed such as Part Of Speech (POS) tagger, Stemmer, Named Entity Recognizer and so on for query preprocessing and query translation. However, some of such tools are not publicly available, some of them do not have full functionalities, and some of them are not yet developed. Most of these Amharic Natural Language Processing (NLP) tools which we used for our query preprocessing and the Amharic search engine are developed for the academic purpose by post graduate students. For the purpose of evaluation and demonstration of our system, we have integrated and used some of such tools with their limited functionalities for which are publicly available and some of the tools such as NER are not integrated at all since they are not yet developed. On the other hand, because of the absence of bilingual Amharic-English Machine Readable Dictionary (MRD), we have used an in-house developed dictionary which has limited number of words.

1.6 Methodology

To achieve the general and specific objectives of the research, we have reviewed different literature reviews and we used open source software and programming language tools.

1.6.1 Literature Review

Different literatures that are considered to be relevant for the research have been reviewed and some of the concepts have been adopted for our work. Since our research work is on bilingual search engines, it touches several numbers of areas like: scripts, word translation between different languages, word transliteration between different languages that use different scripts, bilingual information retrieval systems, and search engines. Almost all of these are a recent research areas and are extensively researched particularly in languages such us Japanese, Chinese, Korean, Indian, and others.

The characteristics of Ethiopic character encodings and their phonetic representations have been studied, and related works have been reviewed to properly design and develop a transliteration system to translate proper names and borrowed words that are not normally present in the Amharic-English bilingual dictionary but frequently occur in queries. For the bilingual information retrieval, there are research works on Amharic-English Information Retrieval then we have studied and used them as our source of knowledge for the query translation in parallel with related works in other languages.

A search engine developed for Amharic language [2, 17] has been reviewed to understand the way in which searching and retrieving Amharic Web documents can be performed. Search engines developed for languages other than Amharic and English as well have been used as a reference. Bilingual and multilingual search engines have been reviewed and some of the important concepts are adopted for our work.

1.6.2 Analysis and Design

To design the model of bilingual Amharic-English search engine, different translation system models, and bilingual and multilingual search engine models that are developed for other languages are studied.

To successfully accomplish our system implementation, programming language softwares, open source software, database management system software, etc have been used. To do this, we have followed the following basic procedural steps:

- **Query preprocessing module development:** Amharic query preprocessing tools such as tokenizer, normalizer, stop words remover, and stemmer are customized to fit for our system and used to develop our Amharic query preprocessing component. Open source English query preprocessing tools such as tokenizer, stop words remover and stemmer are used to develop our English query preprocessing component.

- **Amharic-English bilingual dictionary development:** Two hard copy dictionaries Amharic-English and English-Amharic are typed in Microsoft Access and exported to MySQL database for our bilingual dictionary.

- **Translation module development:** The translation module is developed using Java programming language and we have used the Java API called JDBC to access our bilingual dictionary.

- **Open source search engine customization and configuration:** Nutch an open source search engine is customized, configured and used to crawl, index and search Web documents in both languages, Amharic and English.

- **User Interface design:** Java server page is used to design the user interface of the system and Apache Tomcat is used as our servlet container.

1.6.3 Evaluation

Web documents were collected by crawling from selected websites. Since the query selection has an important impact on the CLIR, queries are carefully prepared and different monolingual and cross language runs has been conducted to evaluate the performance and efficiency of the system. Domain experts have been involved to judge the relevancy of documents to the queries. Finally, the conclusions and recommendations have been driven from the evaluation results.

1.7 Application of Results

Multilingual Information Retrieval typically focuses on the task of retrieving documents in a language or a set of languages that are relevant to a query in another language. However, in many real-life scenarios, such as intelligence work, international business, or tourism, users may not be able to read language of the retrieved document and therefore they may need results in the query language as well. This requires searching and returning documents in both the query as well as the document language.

Even though most Internet users in Ethiopia are not native English speakers, they are forced to use search engines that are mainly designed for English document retrieval. Bilingual search engine will allow monolingual (or Amharic speakers) to access and retrieve the vast online information resources that are available in English and Amharic by using their own mother tongue (native) language queries.

In Tourism domain, the bilingual search engine will play a vital role by allowing tourists or visitors to have query language preferences and by providing Web documents in both Amharic and English languages for both local people and foreigners. In addition to this, queries, in tourism domain, are mostly Out-Of-Dictionary words such as person, place, monument or organization names. These words are not likely to be present in the bilingual dictionaries. Therefore, bilingual search engine for Amharic-English that integrates proper name transliteration will have a vital application in the domain.

International marketing companies can access, or make available valuable Website information that could contribute to worldwide brand recognition and online sales in either of the languages. These companies will enjoy a diversity of visitors by ultimately improving the Website's online position in search engine results. The availability of bilingual search engine will help the companies to develop multilingual Websites that are essential in order for a company to reach its entire target market.

1.8 Organization of the Thesis

The rest of this thesis is organized as follows. Chapter 2 discusses about information retrieval in general and cross language retrieval in particular and some related issues such as translation and transliteration. The Chapter discusses and reviews literatures about information retrieval models, search engines and its components, and multilingual searches. Furthermore, the Chapter also discusses different query and document translation approaches and some concepts of transliteration and its application areas. Chapter 3 discusses some related works about cross language information retrieval and multilingual and bilingual search engines. Chapter 4 presents a detailed discussion on the model of the proposed system and the functions of each of its components.

The implementation details of the system such as: the tools, the algorithms, the techniques and methods used to develop the system are described in Chapter 5. Chapter 6 presents the experimental setups and the experimental results obtained from the proposed system along with their interpretations and the reasons behind each of the results. Finally, Chapter 7 concludes the thesis with the benefits of the research, recommendations, and future research directions.

CHAPTER TWO

REVIEW OF LITERATURES

2.1 Information Retrieval

In recent years, with the expansion of the World Wide Web, people have realized the importance of finding information that is stored on the Web. The availability of large amount of information on the Web, allows the user to possibly find and retrieve useful information that satisfies their needs from such collections. The Web document collection is massive in size and diverse in content, format, purpose, and quality. Therefore, the users need an interface to directly request and able to retrieve documents that are relevant to their needs. This can be achieved by translating this information need to a query which can be processed by the search engine (or IR system). As a result, several IR systems or search engines are used on everyday basis by a wide variety of users.

The meaning of the term information retrieval can be very broad. However, according to [21], Information retrieval (IR) can be defined as finding materials (usually documents) of an unstructured nature (usually text) that satisfies information needs from within large collections (usually stored on computers). Information Retrieval (IR) deals with the representation, storage, organization of, and access to information items [25]. The aim of Information Retrieval (IR) is to find and retrieve documents relevant to a given query, usually where documents and query are in the same language. With further advances in research and technology the goal was extended beyond language barriers to include differences in different languages, which is known as Cross Language Information Retrieval (CLIR). To support these diversities, different information retrieval models that have a better performance than the previous one have been proposed [29].

2.2 Information Retrieval Models

2.2.1 Early Information Retrieval Systems

Early Information Retrieval(IR) systems were based on *Boolean retrieval model* which allowed users to specify their information need using a complex combination of Boolean ANDs, ORs and NOTs [29]. The users can pose any query which is in the form of a Boolean expression of terms.

The model views each document as just a set of words. In Boolean system, there is no inherent notion of document ranking, and hence relevance ranking is often not critical.

However, most everyday users of IR systems expect IR systems to do ranked retrieval. IR systems rank documents by their estimation of the usefulness of a document for a user query. A document is relevant if it is one that the user perceives as containing information of value with respect to their personal information need. Retrieval performance is often measured by parameters such as recall and precision, reflecting the ratio of the relevance of relevant items actually retrieved and of retrieved items actually relevant [2, 29].

- *Precision*: What fractions of the returned results are relevant to the information need?
- *Recall*: What fractions of the relevant documents in the collection were returned by the system?

This can be expressed as:

$$\text{Recall} = \frac{\text{Relevant Documents Retrieved}}{\text{Total Number of Relevant Document}}$$

$$\text{Precision} = \frac{\text{Relevant Documents Retrieved}}{\text{Total Number of Retrieved Documents}}$$

2.2.2 Modern Information Retrieval Systems

Since Web data still consist mostly of text, they need various text retrieval tools (e.g., term weighting, term similarity computation, query expansion) that may be useful in Web IR. On the other hand, as the name implies, Web documents are heavily interconnected. Therefore, they need Link analysis approaches, such as PageRank, HITS, etc. These approaches have created hyperlinks as implicit recommendations about the documents to which they point. As stated in [30], there are several promising Web IR tools which use the combinations of these different retrieval techniques. For instance, as in [30], Google seems to employ a variety of techniques to obtain high performance text retrieval influenced by a universal link analysis score called PageRank. Google improves link analysis by combining it with text retrieval techniques and heavily leverage the implicit human judgment embedded in hyperlinks.

The relevance of the document is determined by the ranking algorithm. The ranking algorithm works on the bases of the IR models. Several models have been used for the ranking process. These includes: vector space model, the probabilistic models, and the inference network model [30].

Vector Space Model

The representation of a set of documents as vectors in a common vector space is known as the vector space model [21]. It is fundamental to a host of information retrieval operations ranging from scoring documents on a query, document classification and document clustering. In the vector space model, text is represented by a vector of terms (words or phrases). The vector model assigns non-binary weights to index terms in queries and in documents. If a term belongs to a text, it gets a non-zero value in the text-vector along the dimension corresponding to the term. These values are used to compute the similarity between each document stored in a system and the user query. In this model, each document will be represented by an *n*-dimensional vector and a user query is similarly represented by an *n*-dimensional vector, where n is the number of terms in the vocabulary. To assign a numeric score to a document for a query, the model measures the similarity between the query vector and the document vector. If $\vec{D}$ is the document vector and $\vec{Q}$ is the query vector, then the similarity of document D to query Q can be given as: [29]

$$Sim\ (\overrightarrow{D},\overrightarrow{Q}) = \sum_{ti\,\varepsilon Q\,,D} WtiQ\ \ .WtiD$$

Where $\mathcal{W}t_{iQ}$ is the value of the i^{th} component in the query vector $\vec{Q}$, and $\mathcal{W}t_{iD}$ is the i^{th} component in the document vector $\vec{D}$. The summation can only be done over the terms that are common in the query and the document since the word that is not present in either the query or the document has a value zero.

Probabilistic Model

Users may issue a query on the collection of documents and an ordered list of documents is returned in response. Using a probabilistic model, the obvious order in which to present

documents to the user is to rank documents by their estimated probability of relevance with respect to the information need. The vector-space model does not take the existing term relationships in a document into account. The probabilistic model takes these term dependencies and relationships into consideration. This family of IR models is based on the general principle that documents in a collection should be ranked by decreasing probability of their relevance to a query. This is often called the probabilistic ranking principle (PRP).

Inference Network Model

Information retrieval is an inference or evidential reasoning process in which we estimate the probability that a user's information need, expressed as one or more queries, is met given a document as evidence. Therefore, network representation can be used to model this technique. The inference network has the ability to perform a ranking given many sources of evidence by performing a combination of confidence. The inference network is used to model documents, the contents of documents, and the query given by a user. The inference network consists of two sub-networks: the first is the document network that is produced during indexing and then become static during retrieval process, and the second is the query network that is produced from the query text during retrieval process [31].

In this model, document retrieval is modeled as an inference process in an inference network. Most techniques used by IR systems can be implemented under this model [29]. In the simplest implementation of this model, a document instantiates a term with certain strength, and the credit from multiple terms is accumulated given a query to compute the equivalent of a numeric score for the document. From an operational perspective, the strength of instantiation of a term for a document can be considered as the weight of the term in the document, and document ranking in the simplest form of this model becomes similar to ranking in the vector space model and the probabilistic models.

2.3 Search Engine

Web search engines provide the way to reach information on the Web. The availability of more information on the Web increases the need of the people to use search engines. As defined in [5], search engine is a software package that collects Web pages on the internet through a robot (spider, crawler) program and stores the information on the pages with appropriate indexing to facilitate quick retrieval of the desired information. Even though most search engines have similar components, the advancements in technology have brought the development of several types of general purpose, language specific and multilingual search engines.

2.3.1 Components of Search Engines

The most important measure for a search engine is the search performance, quality of the results and ability to crawl, and index the Web efficiently. The primary goal is to provide high quality search results over a rapidly growing World Wide Web. Some of the efficient and recommended search engines are Google, Yahoo, MSN, etc, which share some common features and are standardized to some extent. Even though, there are differences in the ways these various search engines work, they all perform three basic tasks:

- They search the Internet or select pieces of the Internet based on important words.
- They keep an index of the words they find, and where they find them.
- They allow users to look for words or combinations of words found in that index.

In order to perform these basic tasks, a Web search engine consists of three major components [2, 17, 28]:

- ✓ A Crawler Component
- ✓ An Indexer Component
- ✓ A Query Handler Component

Crawler

Web crawlers are an essential component to search engines. To find information on the hundreds of millions of Web pages that exist, a typical search engine employs special software robots, called spiders, or crawlers, to build lists of the words found on Websites. A Web crawler is an automated program, which automatically traverses the Web by downloading documents and

following links from page to page. When the crawler is building its lists, the process is called Web crawling [26]. Crawling is the most fragile application [27] since it involves interacting with hundreds of thousands of Web servers and various name servers, which are all beyond the control of the system.

Indexer

Search engine indexing collects, parses, and stores data to facilitate fast and accurate information retrieval [28]. The purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query. Without an index, the search engine would scan every document in the corpus, which would require considerable time and computing power. Even though additional computer storage required for storing the index and considerable increase in the time required for an update to take place, indexing saves the time required during information retrieval. The Indexer component extracts all the words from each page (parsing), and records the URL where each word occurred [27]. The result is a large lookup table that can provide all the URLs that point to pages where a given word occurs. Data collected from each Web page are then added to the search engine index. When you enter a query at a search engine site, your input is checked against the search engine's index of all the Web pages it has analyzed. The best URLs are then returned to you as hits, ranked in order with the best results at the top.

Query Handler

This component interacts with users and answers user queries using the index [2, 17]. The query handler component accepts user queries (keywords) for a particular topic and looks up the index component. The document in the index with the keyword will be selected and displayed to the user. In displaying these pages, ranking mechanism is used.

2.3.2 Multilingual Search Engines

Due to the rapid growth of multilingual content on the Web, today's search engines are trying to solve the problem of language barriers in searching online information. This is to mean that a query written in one language can be used to retrieve documents on the Web that are written in any other language. The proportion of Web documents written in non-English language have been increasing as the number of non-English Web users is dramatically increasing [34, 41]. To

improve the ability of a monolingual speaker to search multilingual content, there is a need to build a system that supports cross-lingual search of different languages written in different languages and scripts. A multilingual search engine automatically translates your request and submits it to search engines in multiple languages. Nowadays, a number of multilingual search engines that support a fast search capability for typical query and achieve a very good performance in the high precision region of the result list have been developed for major languages [34, 36]. However, under resourced languages such as Amharic are not considered still now, because of the unavailability of several tools that can be used for language specific information retrieval and translations between major languages.

2.4 Translation

Machine translation has become a key technology in our globalized society [42]. As a result, machine translation software is available for major language pairs and for major computer platforms, including Web-based machine translation. Machine translation software basically relies upon the availability of extensive linguistic resources [40]. To develop a good quality machine translation system, we need to use large collections of parallel texts aligned at the sentence level, and amounting to at least several millions of words. At the same time, parallel corpora of this size tend to be very rare, especially for under-resourced languages. Due to this, it is difficult to create a new good quality machine translation system for these language pairs.

Machine translation (MT) software is special in the way it strongly depends on data [43, 44, 46]. Rule-based machine translation (RBMT) depends on linguistic data such as morphological dictionaries, bilingual dictionaries, grammars and structural transfer rule files. Corpus-based machine translation (such as statistical machine translation and example-based machine translation) depends, directly or indirectly, on the availability of sentence-aligned parallel text. However, in all cases, one may distinguish three components [43, 44]:

i) *A machine translation engine:* a transformer system which may consist of different modules for the purpose of
    - Separating the text from the format information
    - Tokenizing the text in lexical category and morphological inflection information (morphological analysis).

- Part of speech tagging
- Transforming from the source language to the target language
- Morphological synthesis
- Restoring the formats of text

However, the sophistication of the translation engine depends on the aims of the translation is designed. For instance, a translation system that is designed for query translation in cross language information retrieval may not consider the grammar and structure of the query sentences for several reasons [50].

- First, most IR models are based on bag-of-words models. Therefore, they don't take the syntactic structure and word order of queries into account and hence they are easy to develop.
- Second, queries submitted to IR systems are usually short and therefore not able to describe the user's information need in an unambiguous and precise way. These indicate that complex process used in machine translation for producing a grammatical translation is not fully exploited by current IR models.

ii) *Linguistic data*: is a bilingual dictionary, or bilingual corpora with structural transfer rules that perform grammatical and other transformations between the two languages involved in each direction, and control data for each one of the part - of -speech taggers.

iii) *Tools (optional):* to maintain these data and turn them into a format which is suitable for the engine to use. It includes tools such as compilers to turn linguistic data into a fast and compact form used by the engine and software to learn disambiguation or structural transfer rules.

Machine translation is one of the toughest problems in natural language processing [40]. It is generally accepted however that machine translation between close or related languages is simpler than full-fledged translation between languages that differ substantially in morphological and syntactic structure.

2.4.1 Translation and Information Retrieval

Information retrieval is traditionally based on matching the words of a query with the words of document collections. Since the query and the document collection are in different languages, this kind of direct matching is impossible in Cross-Lingual Information Retrieval (CLIR). Therefore, translation is needed to translate either the query into the language of the documents or the documents into the language of the query. Obviously, translating the whole document collection is more demanding, as it requires more scarce resources like full-fledged machine translation system which is not available for a number of languages in developing countries. Hence query translation techniques become more feasible and common in development and implementation of CLIR system, particularly for minor languages such as Amharic-English CLIR system.

Query Translation

The basic methods in query translation are Statistical Machine Translation (SMT), corpus-based translation and knowledge-based or dictionary-based translation. Since established MT system and/or parallel corpora are not available for under-resourced languages such as Amharic, it is advisable to use a query translation which is based on machine-readable dictionary to work on bilingual or cross-lingual information retrieval.

Document Translation

Translating document collections is less practical, because of its demanding cost to prepare large bilingual corpora and other translation tools. Moreover, writing a topic in another language and then asking the search engine to automatically translate the document to be fetched into the language of the query before retrieval, degrades retrieval effectiveness compared to a monolingual search in which requests and documents are written in the same language. Due to this, document translation reduces retrieval performance and is not commonly used specially for minor languages. In addition to this, users typically prefer to give isolated words, or at best, short phrases (three-four words) to an IR system. Therefore, the question here should be how to best translate a set of isolated words into the target language to retrieve a document in a language different from the query language rather than the sentence based translation.

2.4.2 Approaches of Machine Translation

Machine translation can use a method based on linguistic rules. The words in the source language are translated into the most suitable words of the target language in a linguistic way. It is often argued that the success of machine translation requires a better understanding of natural language problems first. The methods of machine translation require extensive lexicons with morphological, syntactic, and semantic information, and large sets of rules. Machine translation systems differ in sophistication, and there are several basic approaches to translation.

2.4.2.1 Rule-based Approach

Rule-based systems were the first 'real' kind of machine translation system. Rather than simply translating word to word, rules are developed that allow for words to be placed in different places, to have different meaning depending on context, etc. In this approach, the original text is first analyzed morphologically and syntactically in order to obtain a syntactic representation. This representation can then be refined to a more abstract level putting emphasis on the parts relevant for translation and ignoring other types of information. The transfer process then converts this final representation (still in the original language) to a representation of the same level of abstraction in the target language. Translation is carried out by repeating pattern matching and transformation of tree or graph structures that represent the syntax or semantics of a sentence. A rule based system is an effective way to implement a machine translation system because of its extensibility and maintainability. However, it is disadvantageous in processing efficiency.

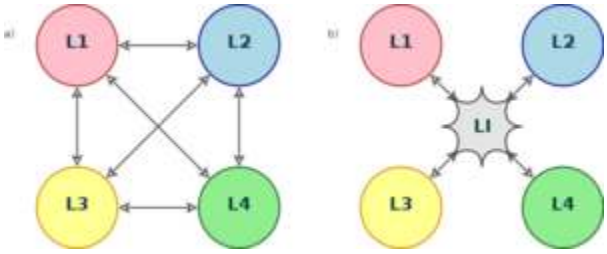
The rule-based machine translation paradigm includes transfer-based machine translation, interlingual machine translation and dictionary-based machine translation paradigms.

Transfer-based machine translation

Transfer-based systems are another approach to rule-based machine transfer, influenced by the Interlingua idea. Instead of using a whole language, an intermediate representation of equivalent pieces is used. This still uses language-pair-specific translation, but the amounts of language-specific rules are reduced to a minimum. There are two kinds of transfer-based translation:

- ***Shallow transfer (syntactic)***, where words are translated based on combinations of word types.
- ***Deep transfer (semantic),*** which uses a representation of the meaning of each word as a basis for how it should be translated.

In Transfer-based machine translation, it is necessary to have an intermediate representation that captures the "meaning" of the original sentence in order to generate the correct translation. It has some dependence on the language pair involved. The way in which transfer-based machine translation systems work varies substantially, but in general they follow the same pattern: they apply sets of linguistic rules which are defined as correspondences between the structure of the source language and that of the target language. The first stage involves analyzing the input text for morphology, syntax, and sometimes semantics to create an internal representation. Then the translation is generated from this representation using both bilingual dictionaries and grammatical rules. Figure 2.1 shows the translation graph required for transfer-based machine translation.



a) b)

Figure 2.1: **a) Translation graph required for transfer-based machine translation**

b) Translation graph required when using a bridge language (Source: [39]).

Interlingual machine translation

Interlingua systems are an extension of rule-based systems that use an intermediate language instead of direct translation [39, 45]. Systems based on Interlingua can then more readily translate between various combinations of languages. In this approach, the source language (the text to be translated) is transformed into an Interlingua, an abstract language-independent representation. The target language is then generated from the interlingua. The inter-lingua

approach rests on the assumption that all languages share a common underlying representation. In interlingua-based MT, the intermediate representation must be independent of the languages in question, which distinguishes it from the transfer-based approach. Within the rule-based machine translation paradigm, the interlingual approach is an alternative to the transfer-based approach. One of the advantages of this approach is that no transfer component has to be created for each language pair. However, the definition of an Interlingua is difficult and may be even impossible for a wider domain. Figure 2.2 shows translation graphs for intrelingua-based machine translation using two interlinguas.

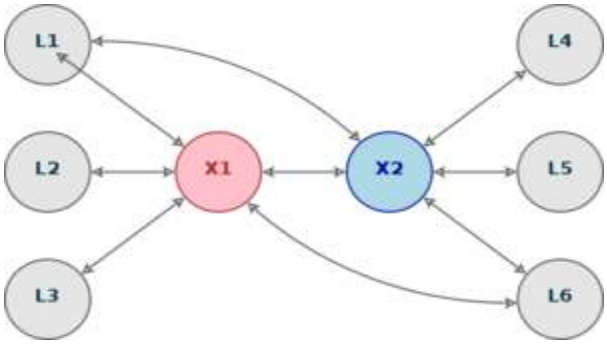


Figure 2.2: **Translation graph using two interlinguas** *(Source: [39]).*

In general, the translation process in transfer-based and interlingual MT involves three subtasks, namely, analysis, transfer and generation, as depicted in Figure 2.3. Analysis deals with the transformation of the source utterances into a predefined format of internal representation, through morphological processing, part-of-speech (POS) tagging, syntactic parsing, semantic analysis, etc. Transfer works to convert the representation of the source language into that of the target language; except for the inter-lingua approach, there used to be different representations for different languages. Generation, or synthesis, is concerned with the derivation of target utterances from the representation, observing necessary syntactic, semantic and pragmatic constraints.

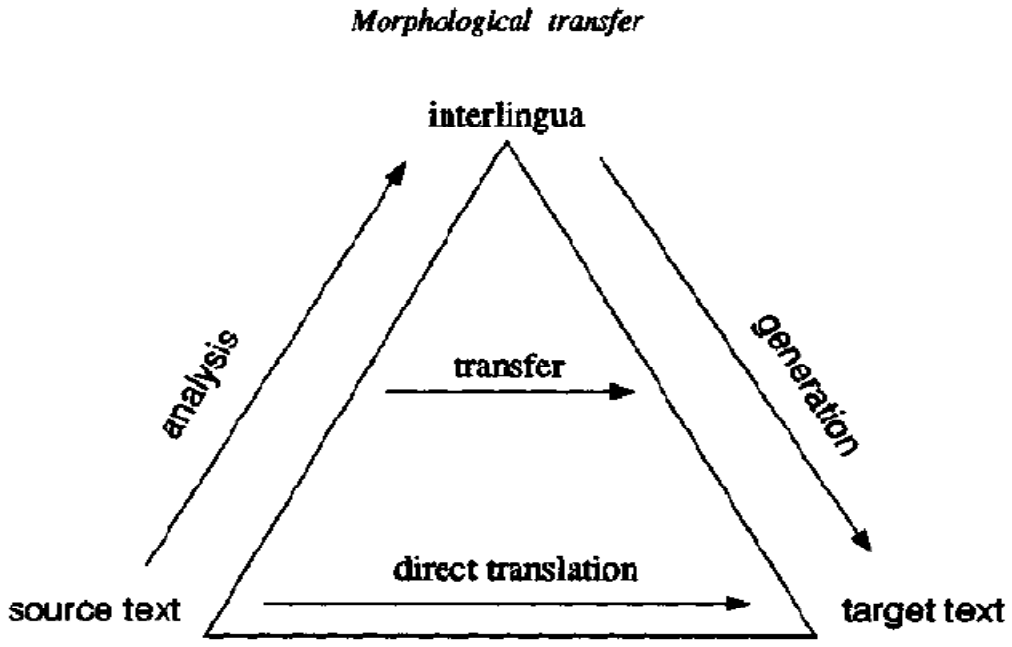


Figure 2.3: **Transfer and Interlingua pyramid diagram** (Source: [40]).

Dictionary-based machine translation

Machine translation can use a method based on dictionary entries, which means that the words will be translated as they are by a dictionary, usually without much correlation of meaning between them. Dictionary lookups may be done with or without morphological analysis. This approach is the least sophisticated and is suitable for the translation of long lists of phrases and for languages which do not have more linguistic resources.

2.4.2.2 Statistical Approach

Statistical Machine Translation is an approach to MT that is characterized by the use of machine learning methods. By examining many samples of human-produced translation, SMT algorithms automatically learn how to translate [33]. Statistical machine translation (SMT) is the most basic and recent and more complicated form of word translation, where statistical weights are used to decide the most likely translation of a word [47]. Modern SMT systems are phrase-based rather than word-based, and assemble translations using the overlap in phrases. Statistical machine translation (SMT) is a machine translation paradigm where translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual text

corpora. Where such corpora are available, impressive results can be achieved translating texts of a similar kind, but such corpora are still very rare.

2.4.2.3 Example-based Approach

Example-based machine translation (EBMT) approach is often characterized by its use of a bilingual corpus as its main knowledge base, at run-time [46]. The basis for EBMT is the existence of a large volume of translated texts (i.e., parallel bilingual texts), which have been translated by professionals with not only language proficiency but also specialist expertise [32]. Example-based machine translation systems are trained from bilingual parallel corpora, and translate using the results of previous translations which contain sentence pairs. Sentence pairs contain sentences in one language with their translations into another. In general, EBMT systems use a large corpus of example pairs of previously translated sentences in order to find close matches and translations of words and phrases in context.

2.4.2.4 Hybrid Machine Translation

This approach for machine translation is combination of statistical and rule based or example based approaches and some elementary grammatical analysis. Hybrid machine translation (HMT) leverages the strengths of both statistical and rule-based or example based translation methodologies. The hybrid approach to MT improves translation of large volumes of speech and text compiled from a variety of sources and assists linguists, translators, and analysts in achieving greater productivity more quickly and in a more cost effective manner.

Form this subtopic discussion, we have learnt that which translation approaches are suitable for which types of language and which types of translation approaches are suitable for which types of translation systems. Based on the discussion, minor languages such Amharic, which have a scarcity of translation tools, should use machine readable dictionary; and translation systems for cross language information retrieval do not need high sophistication of translation system and hence simple translation approaches such as direct word-to-word translation systems are sufficient to provide query translations.

2.5 Transliteration

Transliteration, as defined in [22, 23, 24], is the process of converting terms written in one language into their approximate spelling or phonetic equivalents in another language. It converts a word or phrase in the closest corresponding letters or characters of a different alphabet or language so that the pronunciation is as close as possible to the original word or phrase. Transliteration is defined for a pair of languages, a source language and a target language. The two languages may differ in their script systems and phonetic inventories. Transliteration can also be viewed as the process of replacing words in the source language with their approximate phonetic or spelling equivalents in the target language. Thus, transliteration is meant to preserve the sounds of the syllables in words. Transliteration is more phonetic than orthographic. Even though transliteration between languages that use similar alphabets and sound systems is relatively simple, transliterating names from Amharic into English is a non-trivial task, mainly due to the differences in their sound and writing systems. The method of transliteration is also depends on the characteristics of the source and target languages (in this case English and Amharic) and on the larger purpose the transliterator is meant to serve.

Transliteration is helpful in situations where one does not know the script of a language but knows to speak and understand the language. Transliteration techniques, especially name transliterations, are useful in several areas including machine translation and information retrieval since proper names are commonly observed in important query words. Translation of proper names is generally recognized as a significant problem in many multi-lingual text and speech processing applications. Even when large bilingual lexicons used for machine translation (MT) and cross-lingual information retrieval (CLIR) which provides significant coverage of the words encountered in the text, a significant portion of the tokens not covered by such lexicons are proper names [12]. For CLIR applications in particular, proper names and technical terms are particularly important, as they carry some of the more distinctive information in a query. In IR systems where users provide very short queries (2-3 words), their importance grows even further. Therefore, to handle the problems related to proper names an efficient transliteration system is required.

Transliteration has acquired a growing interest recently, particularly in the field of Machine Translation (MT). It handles those terms where no translation would suffice or even exist. Some of the terms (words) that need transliteration in MT systems are:

- Person names, although many of them have homographs that can be translated.
- Foreign words, which retain the sound patterns of their original language with no semantic translation involved.
- Names of countries may often be subject to transliteration instead of translation.

There are several situations where transliteration is especially useful, such as the following:

- When a user views names that are entered in a world-wide database, it is extremely helpful to view and refer to the names in the user's native script.
- When the user performs searching and indexing tasks, transliteration can retrieve information in a different script.
- When a service engineer is sent a program dump that is filled with characters from foreign scripts, it is much easier to diagnose the problem when the text is transliterated and the service engineer can recognize the characters.

CHAPTER THREE

RELATED WORK

While users searching for Web information on a specific topic or in a language other than English, they often find it difficult to search for useful and high-quality information using general-purpose search engines. In response to this problem, many domain-specific or language-specific search engines that are capable of searching in English and in the users' native language have been built. This facilitates for the users to have more efficient searching over a collection of Web documents in their own domain or native language.

In this chapter, we have reviewed and discussed some related works on cross lingual information retrieval from the point of view of the translation approaches used, the information retrieval model or tool used, the language pairs used, etc. We discussed by focusing on Cross-Lingual Information Retrieval (CLIR), multilingual/bilingual search engines, Amharic-English cross-lingual information retrieval, and information retrieval for non-English language query in general purpose search engines.

3.1 Cross-Lingual Information Retrieval

Cross-language information retrieval has been studied widely in different languages, such as English, Chinese, Spanish, Arabic, etc. A number of research works have been reported and the evaluation results of most of the systems have been satisfactory and have almost similar performance with systems for monolingual retrieval. In this subtopic, we have discussed two of cross lingual information retrieval systems which are not basically Web based.

A work of Mohammed et al. [35], this paper evaluates the use of a Machine Translation based approach for query translation in an Arabic-English Cross-Language Information Retrieval (CLIR) system. Arabic is a Central Semitic language, thus related to and classified along with other Semitic languages such as Hebrew and the Neo-Aramaic languages. It is spoken by a large number of people as a first and second language and it is the largest member in Semitic language family in terms of speakers [35]. Modern Standard Arabic is widely taught in schools,

universities, and used in workplaces, government and the media. Amharic shares a number of common linguistic properties with Arabic for which active research has been carried out except it is written from left to right unlike Arabic. As Arabic is a relatively widely researched Semitic language and has a number of common properties that share with Amharic, some of the computational linguistic researches [48, 49] conducted on Amharic language nowadays, recommended to customize and use the tools developed for this language.

The work of Mohammed et al. [35] used ALKAFI, Arabic-English Machine Translation system, which is a commercial system developed by CIMOS Corporation and is the first Arabic to English machine translation system to evaluate the effectiveness of Machine Translation based system for CLIR. The system is experimented on two standard TREC collections and topics using the three query types (title, description, and narrative considering them as short, medium, and long query types respectively) to determine the effects of query length on the performance of the machine translation based method for CLIR. The results showed that the machine translation achieved 61.8%, 64.7%, and 60.2% for title, description, and narrative fields, respectively.

Another work on Cross-Lingual Information Retrieval is the work of P. L. Nikesh et al. [36], this paper describes about an English-Malayalam Cross-Lingual Information Retrieval system. The system retrieves Malayalam documents in response to query given in English or Malayalam language. It supports both cross-lingual (English-Malayalam) and monolingual (Malayalam) Information retrieval. Due to the absence of full-fledged online bilingual dictionary and/or parallel corpora to build the statistical lexicon for the language, the authors used a bilingual dictionary developed in house for translation. For document ranking and retrieval, a system developed based on the vector space model (VSM) has been used.

India is a multilingual country which has 22 official languages. Malayalam is one of the most prominent regional languages of Indian subcontinent. It is spoken by more than 37 million people and is the native language of Kerala state in India [36]. As Malayalam is under resourced language like Amharic, the query translation approaches that the authors follow and the Information Retrieval (IR) tool used are also important to develop CLIR for Amharic language.

3.2 Multilingual Search Engines

Even though 'traditional' cross lingual information retrieval techniques produced satisfactory results as we have discussed in Section 3.1, they cannot be employed directly in Web applications. As stated in [34], there are several factors that make multilingual Web retrieval different from traditional CLIR:

- Web pages are more unstructured and are very diverse in terms of document content and document format (such as HTML or ASP).
- Traditional CLIR usually focuses on effectiveness, measured in recall and precision. However, Web retrieval also concerned with efficiency to end users (i.e. response time and query length).

As most cross lingual information retrieval researchers have used standard text collections like news articles for their test set, they have not encountered with the problems related to multilingual Web retrieval. As a result, they have got better performance results. However, at present the navigation in this multilingual information space is far from the ideal scenario and the study to integrate CLIR techniques into a multilingual Web retrieval system has arisen. In this subsection we have reviewed and discussed two multilingual search engines and one cross lingual Web portal.

Joanne Capstick et al. [20], have developed MULINEX which is a fully implemented multilingual search engine and it is available in German, French and English languages. The system was developed by a MULINEX consortium which consists of five European companies, who aim to improve their competitiveness in the internet market through the development and application of advanced language technology for providing improved user-friendly Web search and navigation services.

MULINEX is a multilingual Internet search engine that supports selective information access, navigation and browsing in a multilingual environment. The goal of MULINEX development is to find techniques for the effective retrieval of multilingual documents from the Internet. It facilitates multilingual information access with navigation and browsing, enabling effective multilingual searching on the Internet by providing translation of queries, customized summaries, and thematic classification of documents.

MULINEX incorporates Web spiders, concept-based indexing, relevance feedback, translation disambiguation, document categorization, and summarization functionalities. It also translates retrieved documents into the users' language such that the users can read them. In this system, queries are morphologically analyzed and then translated by making use of multilingual dictionaries. Since the retrieval performance of automatically translated queries is poorer than monolingual information retrieval, there is an (optional) step of user interaction, where the user can select terms from the translated query and add his own translation. Summaries and result of foreign language documents have been translated on demand by making use of the LOGOS commercial machine translation system.

Another work on multilingual search engine is the work of Wen-hui Zhang et al. [18]. This work is a multilingual Chinese-English search engine developed by a project of Chinese Academy of Sciences which has been accomplished in June 1998. The project provides a solution to the multilingual search problems. The work was conducted with the intention to develop systems that can efficiently search, index and retrieve multilingual information for Chinese (mother tongue) and English information. The search engine provides Chinese and English indexing, retrieval and searching using English and Chinese language. In this research work the authors presented the concepts, technologies, algorithms and detailed measures to achieve the goal of providing highly relevant, and up-to-date multilingual information search, index, and retrieve.

The system has uniform query interfaces for both Chinese and English languages to allow the user to conduct searching. The character encoding detection module determines the language and sends request to the database supporting the corresponding language. A set of information retrieval functions (such as: Boolean query (and, or, not), phrase query, fuzzy query, proximity query, stemming, language parsing, and spelling error override) that natively support Chinese and English searches are developed. The system consists of four parts: the information gatherer, the search engine, the query server and the scheduler. Each part has its own several functionalities in the information retrieval process. In this work, the procedure of retrieving information is as follows.

i. User submits a query, with search options such as preferred categories, to the query server.
ii. The encoding detection module is invoked to detect the language of the query.

iii. The query is reformulated by the query server and passed to appropriate search engines.
iv. The search engines work in parallel and return ranked results to the query server. Each result is hypertext-linked to the Web site which contains the expected document.
v. The query server reorganizes the results and returns them to the user.

In this system, the query server accepts user's query and invokes related search engine, organizes the retrieved results provided by search engine and returns the document only in the language of the user's query. There is no concept of query translation or cross lingual information retrieval, rather the user types the query in his/her own language preferences and gets the result in the language of the query.

From this paper we have learnt that we can retrieve Web documents in the language of the query from several search engines by detecting the language of the query. As a result, in our work, to search, index and retrieve both English and Amharic Web documents, we can use one of the well known general purpose search engine for English Web documents and Amharic Search Engine [2, 17] for Amharic Web documents.

Another work on Multilingual Web Retrieval with an Experiment in English–Chinese Business Intelligence is developed by Jialun Qin et al. [34]. This research work is on an English–Chinese Web portal, named ECBizPort. Their work deals with developing and evaluating a multilingual English–Chinese Web portal that incorporates various CLIR techniques for use in the business domain. The system architecture of the Web portal consists of five major components: Web spider and indexer, pre-translation query expansion, query translation, post-translation query expansion, and document retrieval.

As the translation system is the core component in every cross lingual information retrieval systems, the authors exhaustively compared and made convincing decisions to use the best query translation approach that can be better applicable for Web retrievals. Among the three translation approaches (machine translation based, corpus based and dictionary based), which have been used in different translation systems, the authors used the dictionary-based approach after pointing out several reasons. They tried to combine the dictionary based approach with phrasal translation, co-occurrence analysis, pre- and post-translation query expansion for translation disambiguation.

According to the paper [34], the dictionary based approach of query translation is the most promising for Web applications for two reasons: First, machine-readable dictionaries that are used in dictionary-based translation approach are more widely available and easier to use than the parallel corpora required by the corpus-based approach. The limited availability of existing parallel corpora cannot meet the requirements of practical retrieval systems in today's diverse and fast-growing Web environment. Second, the dictionary-based CLIR approach is more flexible, easier to develop, and easier to control when compared with Machine Translation based CLIR which has little space for users to modify it for their specific purposes, or it is too costly.

The English and Chinese collections for the Web portal was built using a digital library development tool called AI Lab SpidersRUs toolkit. The toolkit is capable of building collections in different languages and encodings. The toolkit is designed by the same research group and has components which can support functionalities like document fetching, document indexing, collection repository management, and document retrieval.

The researchers conducted an experiment to measure the effectiveness and efficiency of Web portal system following TREC evaluation procedures. They evaluated their system by business/IT domain experts who are fluent in both English and Chinese language, using a set of queries in both English and Chinese. The results showed that co-occurrence-based phrasal translation achieved a 74.6% improvement in precision over simple word-by-word translation. When both pre- and post-translation query expansion was used together, the performance achievement improved slightly to 78.0% over the baseline word-by-word translation approach.

The authors strongly recommended to use dictionary based approaches for Web based applications especially for under resourced languages. It has also shown the effectiveness of the approach as queries are usually short and IR models are based on bag-of-words.

3.3 Amharic-English Information Retrieval

Much effort has not been done on Amharic-English bilingual information retrieval. The three series research works done by Atelach Alemu have been discussed in this subtopic. Atelech, in her three consecutive research works entitled with Dictionary-based Amharic - English Information Retrieval [4], Amharic-English Information Retrieval [3], and Amharic-English

Information Retrieval with Pseudo Relevance Feedback [1] discussed cross lingual information retrieval between Amharic and English languages. For all the research works the authors used Amharic-English machine readable dictionaries and an online Amharic-English dictionary for the query terms translation with some additional enhancements for query translation, indexing and searching from one research to the next. The Amharic topic set used in all the experiments was constructed by manually translating the English topics. As the experimental results showed, progressive performance achievements were observed form the first research to the next one and the challenges related to the issues were discussed.

In the first research [4], for words that have more than one translation, all possible translations were taken and manual disambiguation was performed. Out-of-vocabulary words (such as: proper names and borrowed words) were manually added in a separate dictionary. Two different ways of the same basic dictionary based approach, which basically differ in the way less informative words can be identified and removed (called stop word removal) from the query, were used and experimented. A retrieval engine which can support Boolean and Vector Space Model were used for information retrieval. According to the report on the paper, an average precision of 0.4009 and 0.3615 have been achieved by the two methods.

In the second research [3], morphological analysis and part of speech tagging were used for query translation process and Out-Of-Dictionary terms were handled using fuzzy matching. For document indexing and searching, Lucene, an open source retrieval engine, was used. Four different experiments which substantially differ from one another in terms of query expansion, fuzzy matching, and usage of the title and description fields in the topic sets were conducted. As the author reported on the paper, she have got better retrieval performance from the experimental results for Amharic when compared to runs in the previous research.

In the third research [1], for the translation of query terms, they gave precedence for matching bigrams over unigrams of query terms in the dictionaries and Out-Of-Dictionary Amharic query terms were taken to be possible named entities. These Out-Of-Dictionary terms were further filtered through restricted fuzzy matching based on edit distance against automatically extracted English proper names. The Lemur toolkit for language modeling and information retrieval was used for indexing and retrieval. After the first top ranked documents were retrieved the highest

weighted terms were taken to expand the query using the method of Pseudo Relevance Feedback. This paper reported that very limited experiments were conducted and low precision indexing and retrieval performance was observed. The experiments were conducted only to show effects related to the issues like: short queries vs. long queries, the use of Pseudo Relevance Feedback, and the effect of taking the first translation given versus maximally expanding query terms with all translations given in dictionaries.

As the researches of Atelach were conducted from scratch due to the unavailability of bilingual resources for Amharic, the experimental results end up with low precision. Since cross lingual information retrieval requires a number of tools such as stemmer, part of speech tagger, named entity recognizer, morphological analyzer, etc, the paper reported that the effectiveness and efficiency of CLIR system is also highly dependent on these tools and impossible to develop with in short periods of time.

The Amharic-English cross lingual information retrieval researches done by the researcher were not information retrieval on the Web. However, as we have discussed in Section 3.2 cross lingual information retrieval on the Web has additional overheads that should be considered compared to traditional CLIR due to several factors exist on the Web documents.

Atelach [1, 3] also mentioned the importance of handling out of dictionary words. These words are usually proper names and borrowed words which do not often present in the bilingual dictionary but frequently happen in queries. The researcher tried to handle these terms by manually adding on the dictionary and a method of fuzzy matching. However, this practice leads to the buildup of huge bilingual dictionary which reduces the performance of the translation system especially for Web CLIR. In most recent researches, these terms have been handled by using a transliteration system that can be integrated in the CLIR.

3.4 General Purpose Search Engines

Most Web users begin their Web activities by submitting a query to a search engine such as Google, MSN, AltaVista or Yahoo etc. General purpose search engines on the Web are the most popular tools to search for, locate, and retrieve information, and their use has been growing at

exponential rates. Although English is still the dominant language on the Web, information in other languages is steadily gaining prominence. However, users often find it difficult to search for useful and high-quality information on the Web using general-purpose search engines when they search for information in a language other than English. A search engine that can handle these problems of language differences is becoming highly desirable and so researched in some of the major languages.

There have been several research works done to experiment the effectiveness and performance of general purpose search engines' of handling non-English queries. The works tried to compare the results with language specific search engines and arrived in similar conclusions. The work in [37], explores the characteristics of the Chinese language and how queries in this language are handled by different search engines. The authors compared the results by entering queries in two major search engines (Google and AlltheWeb) and two search engines developed for Chinese (Sohu and Baidu). The results showed that the performance of the two major search engines was not equivalent with that of the search engines developed for Chinese.

A research work in [13], was also conducted to test general (English oriented) and local (non-English oriented) search engines on handling queries in Russian, French, Hungarian, and Hebrew. The authors ran queries in all four languages in the local and the general search engines, and found that in most cases the latter ignored the special characteristics of the language of the queries and did not properly handle diacritics. Based on the results the authors recommended that morphological variations among languages must be considered by the developers of search engines, and users should be made aware of what they miss when they use the general search engine to find information in languages other than English.

Another research [38] that used three general search engines (AlltheWeb, AltaVista, and Google) and three Arabic engines (Al bahhar, Ayna, and Morfixa) was conducted over Arabic HTML documents using Arabic queries. The queries were constructed to highlight the special characteristics of Arabic, especially the occurrence of prefixes, which are very common in Arabic words. The results of the searches showed that general search engine search features and its indexing algorithms did not handle Arabic queries well, and did not provide any mechanisms to address Arabic prefixes. The findings highlight the importance of making users aware of what

they miss by using the general engines, underscoring the need to modify these engines to better handle Arabic queries.

In general, general purpose search engines can accept queries of several languages and returns Web documents in the language of the query. However, they do have performance and efficiency problems as they are mainly designed for English language. As they do for other languages, general purpose search engine can also accept Amharic queries and return Amharic Web documents [2, 17], but they do not provide effective and efficient search functionalities with a high-quality collection of Amharic Web documents because of their language specific nature.

In our work, we have used both general purpose and language specific search engines to efficiently access Web documents. One of the well known general purpose open source search engine has been used to access English Web documents for queries which are written in English or which are translated from Amharic to English, and an Amharic search engine that can well support Amharic language encodings has been used to access Amharic documents. As a result, the users of our system can have simultaneous access of Web documents which are written in both English and Amharic languages.

CHAPTER FOUR

THE ARCHITECTURE OF AMHARIC-ENGLISH BILINGUAL SEARCH ENGINE

As it can be surmised from its name, our bilingual search engine is designed to pull up results in not one, but two languages by accepting queries in either of the languages (Amharic and English). In this chapter, we present the general architectural design of the proposed Amharic-English bilingual search engine as depicted in Figure 4.1. The main components of the system together with their subcomponents or modules, and the interaction between each of the main components and their subsystems have been described.

4.1 Components of the System

Most of the bilingual/multilingual search engines have similar basic components. These components can be summarized as Query server, Retrieval and Indexing engine, and Information Gatherer/Crawler. The ***Query server*** component provides a translation of a query or query terms in such a way that the translated query can be used to retrieve documents in the corresponding target language using bilingual resources(such as: bilingual dictionary, parallel corpus, etc). ***Retrieval and Indexing engine*** receives search requests from the query server and is responsible for searching and indexing documents collected by the information gatherer, ranking the documents based on the relevance of the document to the received query, and retrieval of the appropriate document. ***Information gatherer/crawler*** is responsible to download documents and revisit pages to update information.

Even though, the general basic architectures of the bilingual systems are nearly similar, they substantially differ in their internal functionalities depending on the languages used, the algorithms employed and the aims of the bilingual search engines were designed. As a result, the subcomponents of each component have various structures and functionalities.

As many of the bilingual search engines, our proposed Amharic-English bilingual search engine also shares some of the common components and as a result it consists of components such as

Query Preprocessing, Query Translation, Amharic Search Engine, and English Search Engine as shown in Figure 4.1.

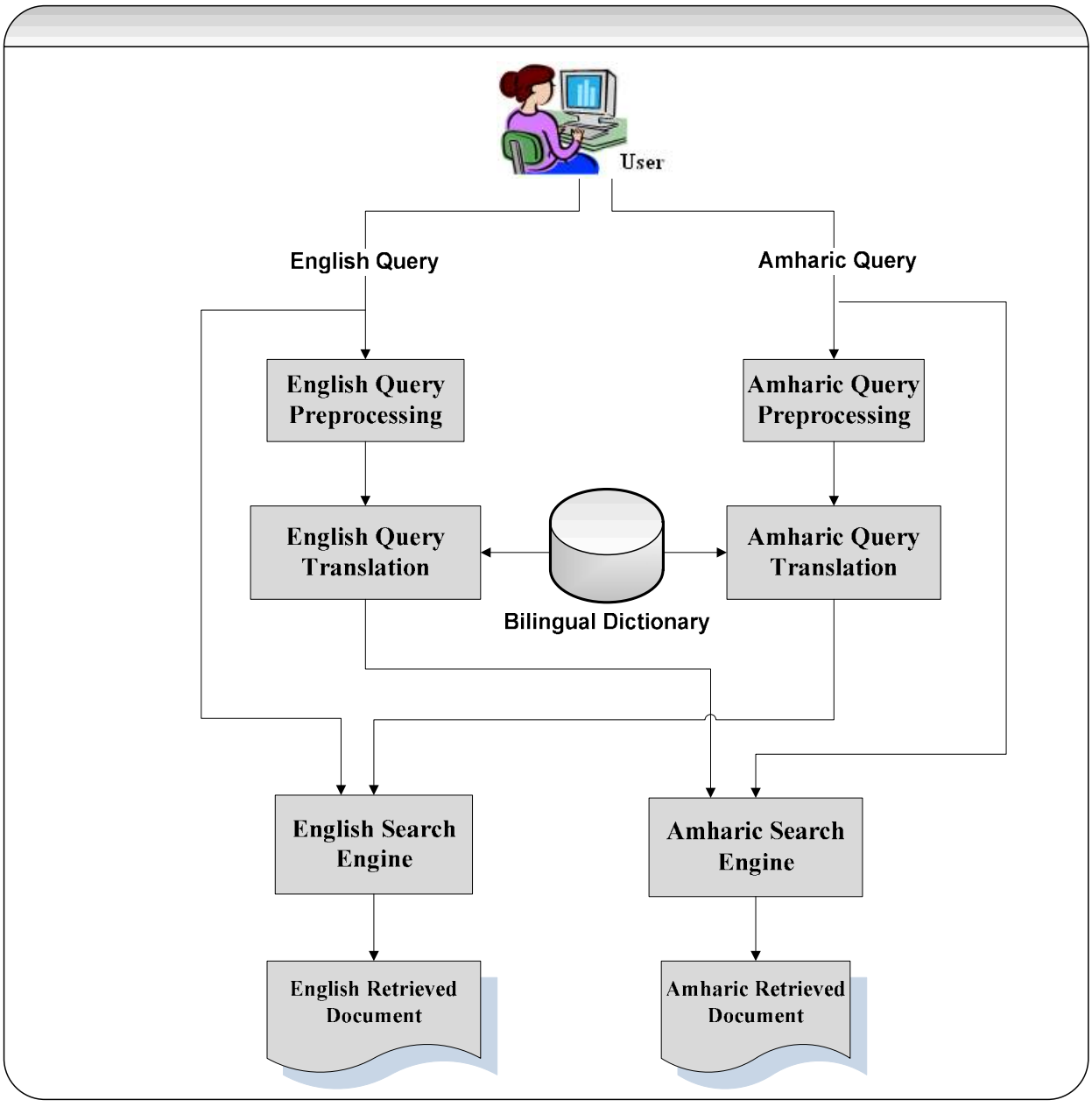


Figure 4.1: **The Model of the Amharic-English Bilingual Search Engine**

Since our bilingual search engine is capable of accepting queries in both Amharic as well as English languages and query preprocessing is a language specific task, the query preprocessing process is done in two different query preprocessing modules (Amharic query preprocessing and English query preprocessing modules). The **Amharic query preprocessing module** is

responsible for tokenizing the Amharic queries into words, normalization of different characters of the Amharic scripts which have the same sound and expanding short forms, eliminating stop-words (less informative words), and stemming inflectional and some derivational Amharic morphemes. The **English query preprocessing module** is responsible for tokenizing the English queries into words, eliminating English stop-words (less informative words), and stemming inflectional and some derivational English morphemes. The query translation process also uses two components which are responsible for lexical transfer or dictionary lookup in the bilingual Amharic-English dictionary and transliterating the out of dictionary words assuming that they are proper names or borrowed words. These components are **Amharic Query translation** and **English Query translation**. The **Amharic search engine** component is responsible for crawling, indexing, and ranking of the Amharic Web documents. The **English search engine** component is responsible for crawling, indexing, and ranking of English Web documents.

4.2 Query Preprocessing

The user interface of the system accepts either Amharic or English query which is posed by the users. The query must pass through the query preprocessing stage before it has been translated for cross language retrieval and at the same time it has to be directly submitted to the appropriate search engine for monolingual retrieval. There are two query preprocessing modules in our system to analyze the queries based on the query language. These are Amharic query preprocessing and English query preprocessing modules.

4.2.1 Amharic Query Preprocessing

To analyze the Amharic query, the Amharic query preprocessing component uses different techniques and produces bag-of-Amharic words of the query in their normal forms. This component consists of subcomponents such as tokenization, normalization, stop-words removal, and stemming. The output of this component is a set of Amharic bag of words which is used as an input for the Amharic query translation component. The detailed architecture of this component is shown in Figure 4.2 and the details of its subcomponents have been discussed below.

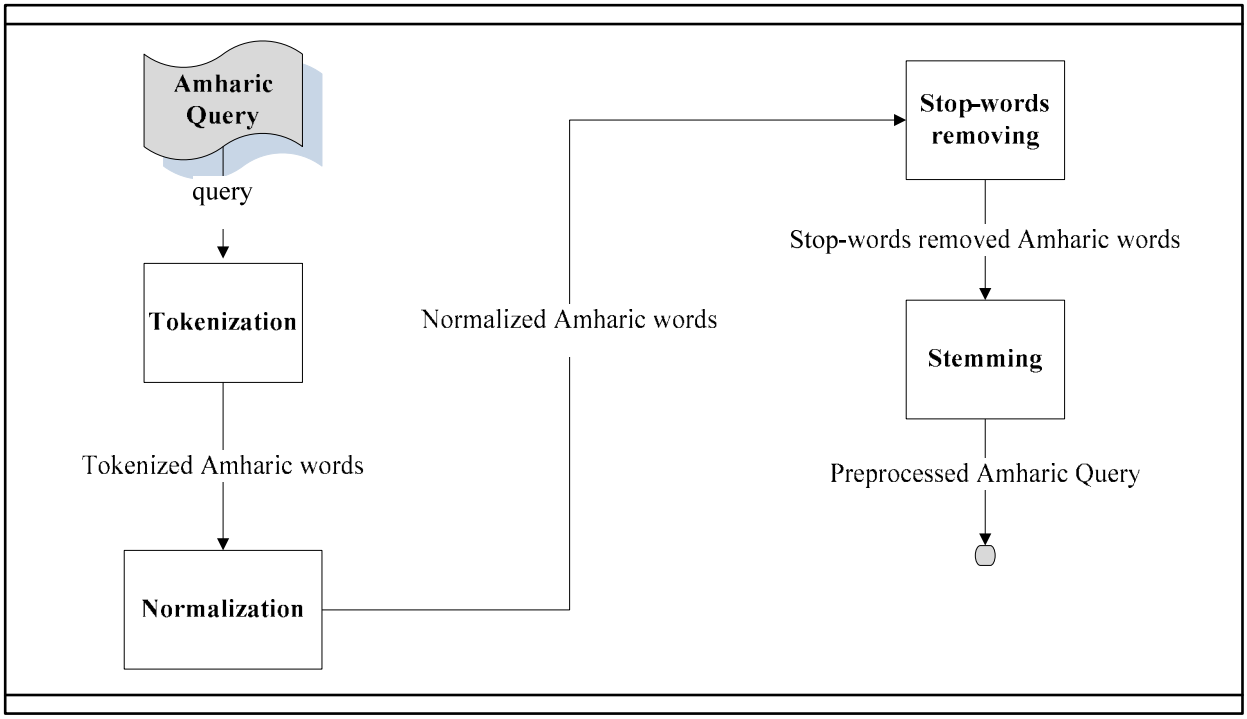


Figure 4.2: **Amharic Query Preprocessing Component**

Tokenization: it is the process of demarcating, classifying, and forming tokens from an input stream of characters. Tokenization could be used to split the query phrase into an instance of a sequence of characters which can be grouped together as a useful semantic unit for processing. During tokenization, we chop on whitespaces, throw away punctuation characters and choose the correct token to use. Tokenization requires the language of the document to be known since it is language specific.

The first step in the query preprocessing component is tokenizing the Amharic query into words using white spaces and Amharic punctuation marks. For example, in the statement "**ኢህአዴግ፣ መኢአድ፣ ኢዴፓና ቅንጅት በምርጫ** 2002 **ሥነ-ምግባር አዋጅ ላይ ተወያዩ።**", unlike human beings, a computer cannot spontaneously understand that the sentence has 10 words; rather it understands the statement only as it has a sequence of 51 characters. Therefore, to convert them into meaningful tokens, they have to be tokenized using certain Amharic language tokenization criteria as:

ኢህአዴግ	መኢአድ	ኢዴፓና	ቅንጅት	በምርጫ	2002	ሥነምግባር	አዋጅ	ላይ	ተወያዩ

Our tokenization component is responsible for handling such activities.

Normalization: Amharic has some redundant symbols with the same sound in its alphabet. For example, *ኦ, ኡ, ዐ and ዓ, ሰ and ሠ, and ጸ and ፀ* have the same sound. These characters can be used interchangeably without any meaning difference in the language. So we can replace them by one character and normalize the representation of words in different forms to one common form. The other functionality of this component is expanding short words. In Amharic, it is common to shorten some words using the forward slash ('/') and English full stop (**ነጥብ** (.)) like ወ/ሮ፣ዓ.ም፣ወ/ሚካኤል፣ etc. These words will be expanded to their normal forms for dictionary lookup. The normalization subcomponent of the Amharic query preprocessing component handles these two activities.

Stop-words removing: stop-word is the name given to words which are filtered out prior to or after processing of natural language text. Most search engines and query translation systems do not consider extremely common words in order to save disk space or to speed up search results.

In our system, when a posed query is analyzed, words which have less content (stop words) should be removed from the query words before translation. As a result, after the query is tokenized into words and the characters and short words are normalized, the next step in our query preprocessing component is identifying and removing the Amharic stop words such as፡ *ነው፣ሆነ፣ወደ፣ናቸው፣ውስጥ፣etc*.

Stemming: it is the process of reducing inflected (or sometimes derived) words into their stem or citation forms. Most search engines and translation systems use stemmer to compare the root forms of the search terms to the documents in their database. As Amharic language is morphologically complex, words are inflected with prefixes, suffixes and infixes. The need of stemming in the query processing is to reduce the size of the dictionary by avoiding the entry of different variants of a word in machine readable dictionaries. This increases the performance of the translation system which in turn has a positive impact on the effectiveness of the cross lingual IR systems to retrieve documents in another language.

In our Amharic query preprocessing component, after the Amharic stop words are removed, the stemming process has been performed on the remaining Amharic words**.** For example, the Amharic words *"በሉ" ፣"በላን"፣"በላች" ፣"በላሁ" ፣"በላችሁ"፣ "ሰሰበሳ*", etc can be reduced to

their citation word *"ስላ"*. This helps the query translation component to handle morphological variations and to find matches in the dictionaries for as many of the query words as possible.

4.2.2 English Query Preprocessing

Like Amharic query preprocessing component, the English query preprocessing component also use different techniques to analyze the English query. As a result, it has three components like tokenization, stop words removing, and stemming. The detailed architecture of this component is depicted in Figure 4.3 and each subcomponent is described below.

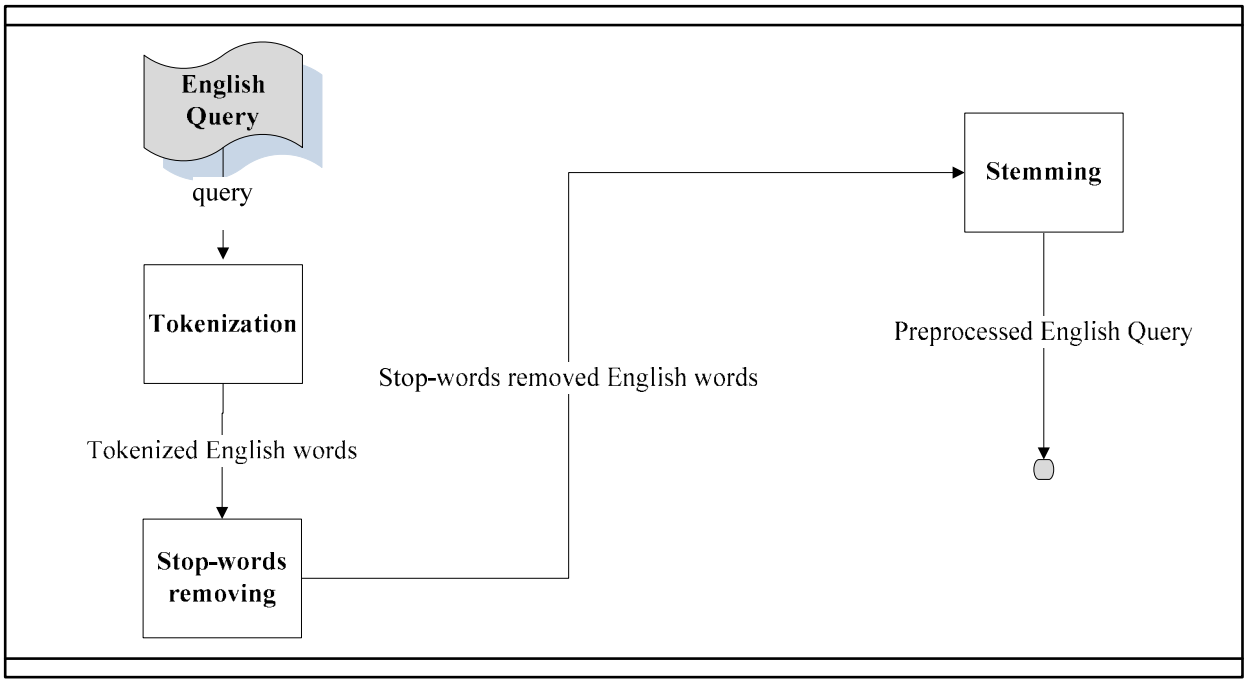


Figure 4.3: **English Query Preprocessing Component**

Tokenization: this component splits the English query phrase into English words using white spaces and some English punctuation marks as word delimiters.

Stop words removing: as we have discussed in Amharic stop words removing, there are some terms that appear very frequently on the collection of English language documents and most of which are not relevant for the retrieval task. In English language, these include articles, conjunctions, prepositions, etc (for example, the words "a", "an", "are", "be", "for" . . .) are referred as English stop words. The techniques used to remove these non-content bearing words

are differ from one system to another depending on the language of the query. So, this subcomponent is responsible for removing these words from the English query sentence.

Stemming: English language has different inflectional, derivational and compound word morphology. Compound words such as teapot, starlight, etc are made from two simple words, which can stand alone, but joined together to form a single word. However, morphemes which cannot stand alone are said to be suffixes and prefixes which collectively known as infixes. The suffixes and prefixes form inflectional and derivational morphology for the language. Past tenses of regular verbs and plural forms of nouns are examples of inflectional morphology which inflects or alters the word by adding the suffixes. In derivational morphology, new words are formed without reference to the internal grammar of a sentence. Table 4.1 shows examples of different morphologies of English language.

Table 4.1: **Inflectional and derivational morphology of English language**

Derivational morphology				**Inflectional morphology**
prefix	**Base word**	**suffix**	**Complex word**	
un	like	ly	unlikely	liked
dis	infect	ion	disinfection	infected
re	vision	ist	revisionist	visions

In our English query preprocessing component, English words such as: “charger”, “charging”, “charged”, “charges” are reduced to their base word “charge” using this subcomponent. The output of this component is used by the English query translation module to find matches in the dictionary.

4.3 Query Translation

Cross Language Information Retrieval (CLIR) has to deal with a lot of challenges like out of vocabulary words (OOV), translation, and word sense disambiguation. The query translation component is the core of the CLIR system that solves the problem of language barriers [34]. It is responsible for translating search queries in the source language into the target language (in our case from Amharic to English and vice versa).

Among the several translation approaches discussed in Chapter two, our proposed query translation component is based on dictionary look up. As we have discussed in the preceding two Chapters, there are different reasons that have made this approach an appropriate one for Web based applications especially for under resourced languages. We can see these reasons from two angles:

a) From the advantages of machine readable dictionary translation approach points of view [34]:
   - Machine-readable dictionaries that are used in dictionary-based translation approach are more widely available and easier to use than the parallel corpora required by the corpus-based approach.
   - The limited availability of existing parallel corpora cannot meet the requirements of practical retrieval systems in today's diverse and fast-growing Web environment.
   - The dictionary-based CLIR approach is more flexible, easier to develop, and easier to control when compared with Machine Translation based CLIR which has little space for users to modify it for their specific purposes.
b) From the current IR models points of view [50]:
   - Most IR models are based on bag-of-words models. Therefore, they don't take the syntactic structure and word order of queries into account and hence they are easy to develop.
   - Queries submitted to IR systems are usually short. Therefore, they don't able to describe the user's information need in an unambiguous and precise way. These indicate that complex process used in machine translation for producing a grammatical translation is not fully exploited by current IR models.

This means that a simpler translation approach may suffice to implement the translation process. Having these ideas and the scarcity of linguistic resource for Amharic in mind, we have encouraged developing our query translation system based on a word-by-word translation method by looking up the general-purpose Amharic-English bilingual dictionary.

4.3.1 Amharic Query Translation

Our Amharic query translation component consists of two main subcomponents which are a lexical transfer which uses the Amharic-English dictionary as a knowledge base, and a transliteration subcomponent which uses character mappings to transliterate from Amharic word to English word. The architecture of the Amharic query translation component is shown in Figure 4.4. The details of each subcomponent have been discussed below.

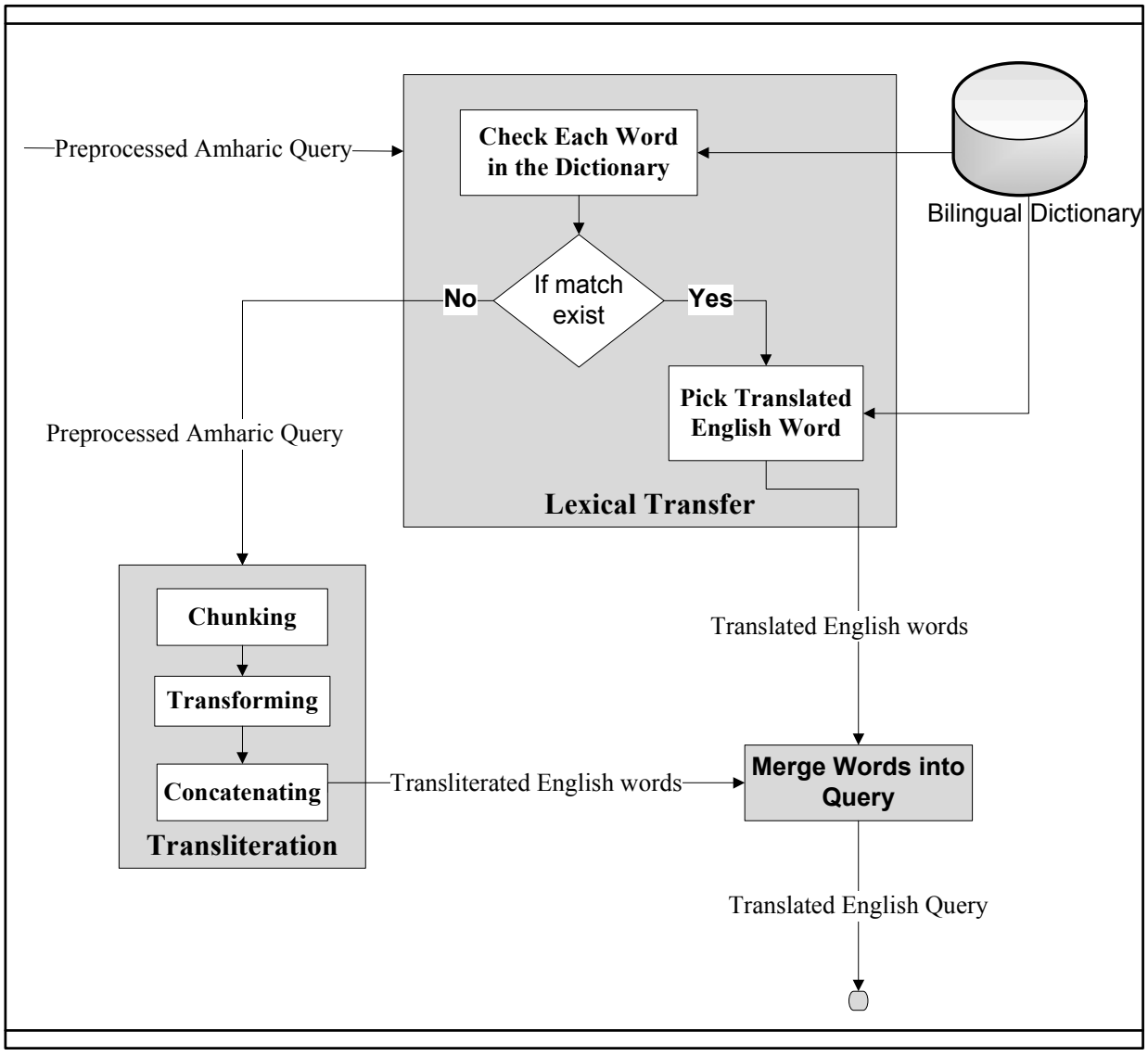


Figure 4.4: **Amharic Query Translation Component**

Lexical Transfer: the Amharic query translation process in this subcomponent is performed by getting Amharic word lists of the Amharic query from the Amharic query preprocessing component, looking up the English sense sets of each term in the general-purpose Amharic-English bilingual dictionary, and selecting the possible English translation senses from the dictionary. Generally, in this subcomponent, the Amharic words received from the query preprocessing component are automatically translated into English words only by using the bilingual dictionary.

Transliteration: one of the problems in dictionary based translation approach is coverage of the words of the language. Particularly, proper names and borrowed words are not usually covered in the bilingual dictionary. As many of Amharic loanwords and proper names, and their transliteration have the same pronunciation or are nearly identical when written in English alphabet, a transliteration system is the best choice to transform from one language character script to another.

In our transliteration subcomponent, all the words that are not found in the bilingual dictionary will pass through this module. In CLIR, the three tasks: name identification, name translation, and name searching are required to handle proper names in the query. However, because of the absence of named entity recognition system for Amharic, instead of identifying and transliterating the proper names, all the words which are not found in the bilingual dictionary will be subjected to transliteration assuming that they are either proper names or borrowed words for both of which a transliteration system is very important.

The transliteration module works by segmenting each stemmed Amharic words into Amharic phoneme (or characters), transforming each character into its corresponding English characters based on the convention for the transcription of Ethiopic script into ASCII, and concatenating each translated English phoneme (or character) into a single English word.

4.3.2 English Query Translation

Like the Amharic query translation component, this component has also two subcomponents which are the lexical transfer and transliteration component. The basic algorithmic differences of the two components lay on the way the transliteration component works and the input and output languages they receive and produce. In our English query translation component, the lexical

transfer subcomponent uses the Amharic-English dictionary as a knowledge base like its corresponding Amharic lexical transfer module does. However, the transliteration subcomponent uses the transliteration database as its knowledge base instead of direct character mappings. The architecture of the English query translation component is shown in Figure 4.5. The details of each subcomponent have been discussed below

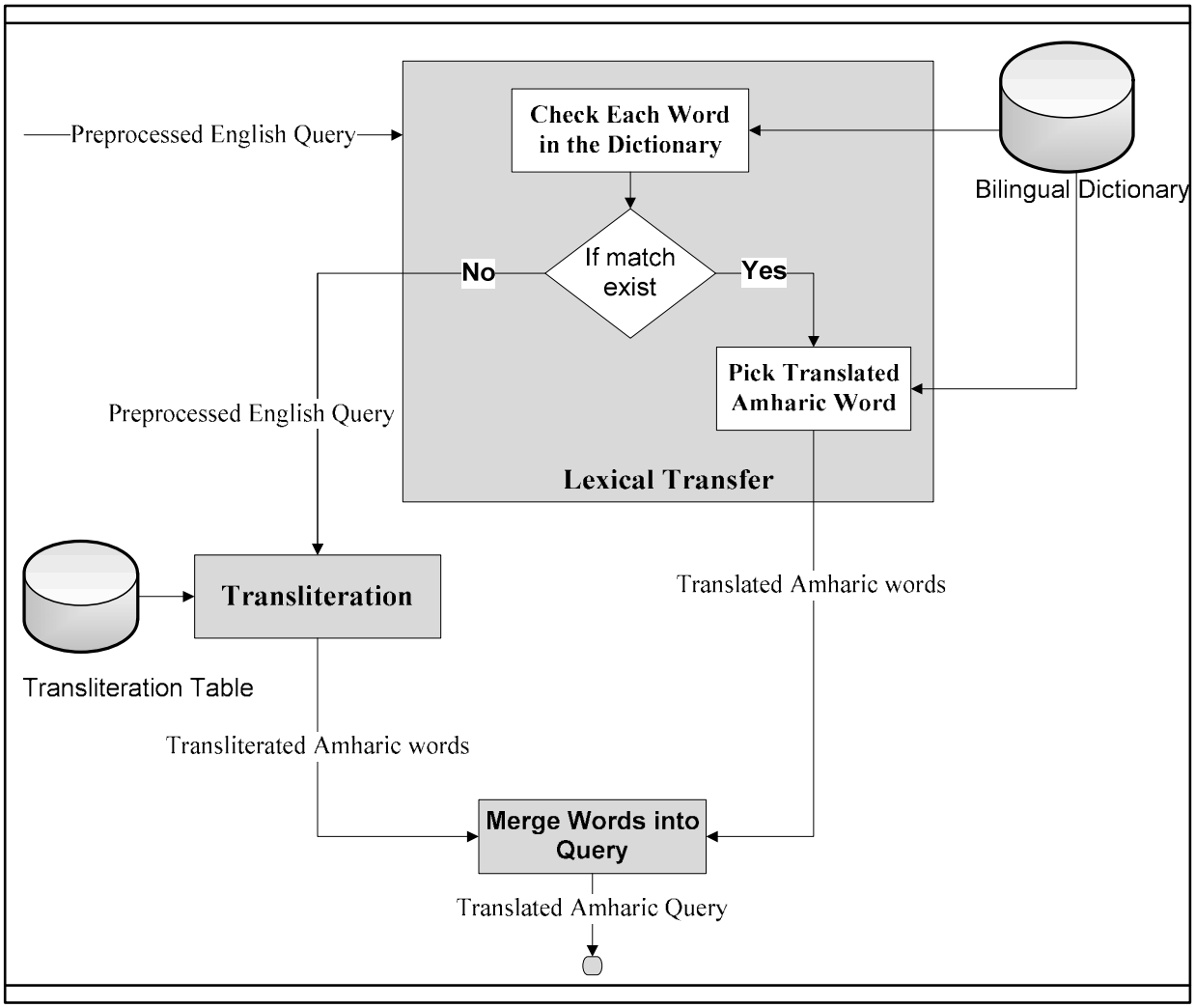


Figure 4.5: **English Query Translation Component**

4.4 English Search Engine

The results of our bilingual search engine are highly dependent on the underlying English and Amharic search engines used to collect Web documents from the Web. Each search engine performs the crawling, indexing and ranking according to its own mechanism. In this subtopic, we have offered the architecture of general Web search engines, as described in [51], which have been used to crawl, index and rank the English Web documents in our Amharic-English bilingual search engine. Figure 4.6 shows the general Web search engine architecture [51]. Based on the figure we have given a high level overview of how the whole system and the different components work. The general Web search engine architecture consists of the components such as: crawler module, crawl control module, indexer module, ranking module, page repository, collection analysis module and the query engine.

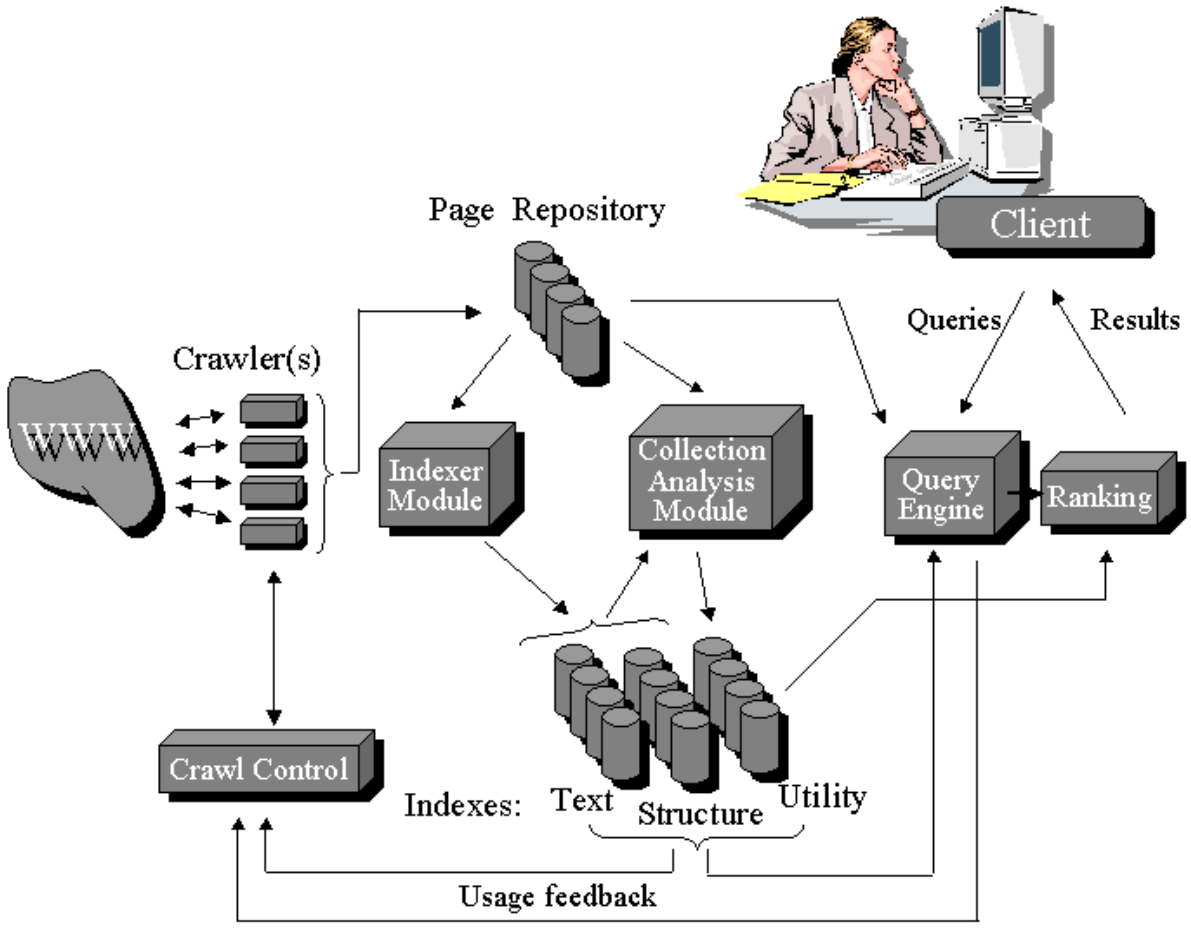


Figure 4.6: **General Search Engine Architecture** (Source: [51])

Every search engine relies on a *crawler module* to browse the Web and to fetch Web documents. Crawlers are relatively simple, small, automated programs or scripts that crawl through Web pages similar to how a human user follows links to reach different pages. A set of starting URLs, called seed URLs whose pages to be retrieved from the Web, will be given to the crawlers. The crawlers then extract URLs appearing in the retrieved pages and give this information to the *crawler control module*. The crawler control module is responsible for determining what links to be visited next and feeding this links back to the crawler, i.e. it directs the crawling operation. The crawler follows links that are considered worthwhile and deposits a copy of the page being crawled into a *page repository*. This process continues until no more local resources, such as storage, are available in the repositories.

The indexer module reads and extracts all the word-occurrences from each page in the repository and associates a list of URLs where each word occurred. A large “lookup table" that can provide all the URLs that point to pages where a given word occurs will be generated. The lexicon lists all the terms found in all the pages that are covered in the crawling process. The lexicon also stores some statistic such as the number of pages the word is found on, the position of the word in document, an approximation of font size and capitalization of the word, etc which are useful for page ranking during querying. Due to the size and rapid rate of change of the Web, indexing poses special difficulties. As a result, it performs some special, less common kinds of indexes such as *structure index* (as shown in the Figure 4.6) which reflects the links between pages.

The collection analysis module is responsible for creating a variety of other indexes. *Utility indexes,* which indexes on values (such as the number of images in each page, or the size of pages), is generated by collection analysis module. The collection analysis module may use the text and structure indexes when creating utility indexes.

During a crawling and indexing run, search engines must store the pages they retrieve from the Web in page repository. On the initial crawl when the repositories are empty, the crawling process is carried out in full. For subsequent crawls, an update strategy must be chosen to update indexes where any of the content of the currently indexed pages has changed. Search engines sometimes maintain a cache of the pages they have visited beyond the time required to build the index. This cache allows them to serve out result pages very quickly, in addition to providing basic search facilities.

The *query engine module* is responsible for receiving and processing search requests from users. The engine relies heavily on the indexes, and sometimes on the page repository to decide which pages are most relevant to the query. Due to the Web's size and the fact that users typically only enter one or two keywords, result sets are usually very large. So, the pages that are thought to be most relevant to the query are listed before those that are less relevant and should be returned in a ranked fashion although relevance can be based on different factors. Hence, the *ranking module* has the task of sorting the results such that results near the top are the most likely to be what the user is looking for.

4.5 Amharic Search Engine

A search engine that indexes and searches only Amharic language Web documents is first designed by Tesema Mindaye [2] and further enhanced by Hassen Redwan [17]. The search engine has the capability of identifying and analyzing Amharic language Web documents and handling the unique characteristics of the language that affect the retrieval of the documents of the language from the Web. As many of other search engines, the Amharic search engine has three main components: the crawler, indexer and searcher component. The detail architectural design of the Amharic search engine, as in [17], is shown schematically in Figure 6.7.

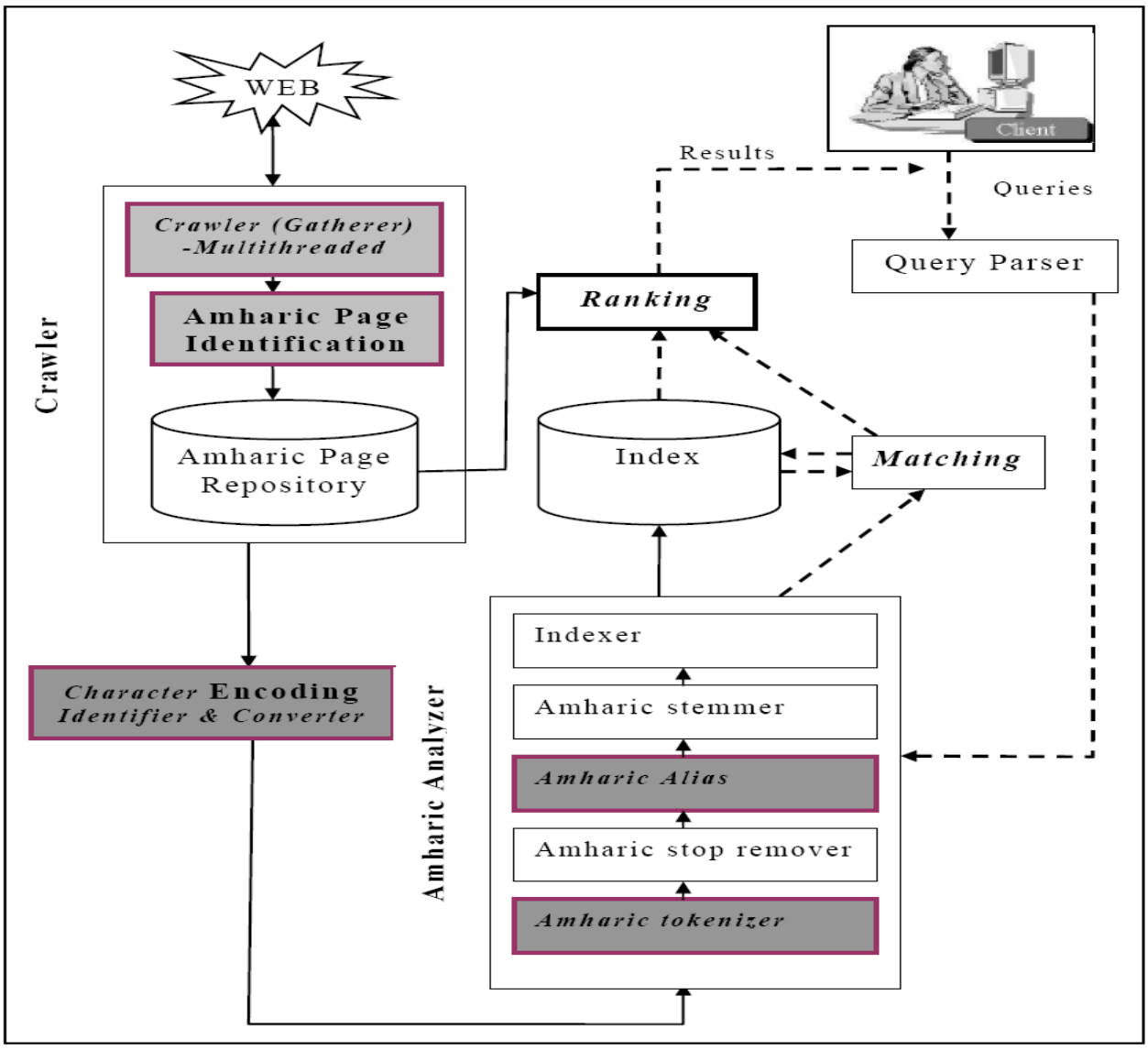


Figure 4.7: **Amharic Search Engine Architecture** (Source: [17])

The *Crawler* component is responsible for collecting Amharic language documents from the Web. The crawler begins its crawling by getting starting URLs, called seed URLs, which have Amharic Web content pages from the frontier. It uses http protocol and multiple threads to fetch the Web page at that URL. The collected pages from each URL will be given to the *Amharic page identification module* to be identified that whether they are Amharic language contents or not. If the pages are Amharic language contents, they will be stored in the *Amharic page repository* for further processing otherwise they will be discarded.

The *character encoding identifier and converter module* parses the pages that are stored in Amharic page repository and extracts the associated texts and links of the page. Its major responsibility is identifying and converting the character encodings of the Amharic Web pages. If the encoding of the page is non-Unicode, it will be converted to Unicode representation.

The *Indexer* component builds indexes from documents that it gets from the Crawler. It is composed of *Amharic analyzer* that considers the typical characteristics of the Amharic language and *indexing module* which stores the indexed terms. The *Amharic analyzer* is responsible for extracting terms from the given Amharic text for indexing purpose. It tokenizes, analyzes Amharic aliases, removes stopwords, and applies stemming to the words before they are indexed.

The *query engine* component provides an interface for the users to enter Amharic search queries. It is responsible for parsing the Amharic queries and it uses the Amharic analyzer to analyze the queries. The *Ranking module* ranks the search results by using the index and Amharic Page Repository. It accesses the index to match the parsed and analyzed query terms with the terms that were indexed.

In our bilingual Amharic-English search engine we will use these search engines discussed in Section 4.5 and 4.6 for crawling, indexing, and ranking English and Amharic Web documents respectively. The search engines will have a common query dispatcher which provides the query to each search engine based on the language of the query terms. The result pages returned by each search engine will be collected and displayed to the user on the systems user interface.

CHAPTER FIVE

IMPLEMENTATION OF AMHARIC-ENGLISH BILINGUAL SEARCH ENGINE

In this Chapter, we have described the implementation details of the bilingual search engine for Amharic-English. The development environment and the tools used to develop the system have been briefly discussed. The tools used for query preprocessing, the customizations made on the existing query preprocessing tools to fit for our system, the new algorithms developed for query translation from Amharic to English and vice versa, and the mechanisms and conventions used for transliteration have been described. The techniques used to dispatch the queries and to fetch the results from the two search engines have been discussed. The tools used, the mechanisms of the tools use to crawl and index web documents, and the configurations made on the tools for crawling and indexing both Amharic and English web documents have been described.

5.1 Development Environment

Developing a full-fledged bilingual search engine requires a lot of resources. High performance computer hardware, network bandwidth, and linguistic tools are among the basic resources that are required to determine the successful development and implementation of the search engines. Crawling the web, storing the indexes, searching and retrieving web documents are network bandwidth, memory storage and high performance hardware intensive tasks. Preprocessing and translation of user queries between languages require the existence of linguistic tools for each language and automated linguistic data between the language pair.

In this subtopic, we have described the development environment used in developing the bilingual search engine. Even though crawl databases should be stored and indexed in a variety of several machines, our system is developed and tested in a single laptop computer for the demonstration, testing and evaluation purpose. The computer has Intel(R) Pentium(R) Dual CPU with 2.00 GHz each processor, 3.00GB of RAM, 320GB of hard disk, and Microsoft XP Professional Service Pack 2 operating system. We used 6MB/sec network bandwidth which is shared by thousands of Addis Ababa University community members.

5.2 Development Tools

During the development of the bilingual search engine, softwares which are developed for academic purposes, open source software, programming language software, and bilingual dictionaries are used. The following software tools are configured, customized and used for developing the prototype of the system.

JDK1.6.0_18 with NetBeans IDE 6.8:

This is the latest version of the java SDK (Software Development Kit) for developing and writing java programs. The software is installed in a Windows environment and different components of our system is developed using java programming language. The choice of the language is made because of the following reasons:

- ✓ The capability of the language for web applications
- ✓ Our experience to the language
- ✓ The platform independent nature of the language
- ✓ The availability of open source components of the language on the web

Nutch-1.0:

It is the latest version of an open source java implementation of a Web search engine. Nutch is built on top of Lucene, which is an API for text indexing and searching and released by the Apache Software Foundation. In addition to Lucene components, Nutch has web crawler for collecting documents from the web. For our bilingual search engine, we have configured and used the software for crawling and indexing English web documents by providing seed URLs.

The choice of this software is made because of the following reasons:

- ✓ Because it is open source
- ✓ The capability of the software to be used at global, local, and even personal scale
- ✓ The easiness and cost effectiveness of the software to deploy a world class search engine
- ✓ The flexibility, scalability, and efficiency of the software

Apache Tomcat 6.0:

Apache Tomcat is an open source software implementation of the Java Servlet and Java Server Pages technologies and released under the Apache License version 2. It is a web container which allows running Servlet and Java Server Pages based web applications. It also provides per default an HTTP connector on port 8080. We used the latest version of the software for the Servlet and Java Server Page container for our system.

Cygwin 1.7.5:

It is an open source collection of tools that allows Unix or Linux applications to be compiled and run on a Windows operating system from within a Linux-like interface. It provides a Unix-like environment and software tool set to Window users. We used the latest version of the software to compile and run the crawler using command line instructions in the Windows XP environment.

MySQL Server 5.0:

MySQL Database Server is a reliable and easy to use relational database management system. It is a very fast, multi-threaded, multi-user, and robust Structured Query Language (SQL) database server.

We have used the latest version of the software to build our two-way bilingual Amharic-English dictionary. We have chosen this software because it is open source and easy to use. The Amharic words and their corresponding English translations are typed in Microsoft Access database to use the easier user interface advantage of the Ms Access. The tables in the Ms Access are converted into utf-8 encoding and exported to the MySQL server database which is created to support utf-8 encoding and non-English character sets in order to support Amharic characters. The database is exported to the MySQL server for better manipulation of the data using Java programming language.

Bilingual Dictionaries:

Two hard copy dictionaries (Amharic-English and English-Amharic dictionaries) are used to develop machine readable dictionary for the query translation. The words from the dictionaries are typed in a database system and used for developing two-way dictionary which allows us to translate the query from Amharic-English and vice versa. The dictionary database is accessed using JDBC(Java DataBase Connectivity) of the java API.

5.3 User Interface Implementation

The user interface of our system is responsible for accepting user queries, dispatching the queries to the appropriate query preprocessing module based on the language, and displaying the results to the user in an appealing manner in the users' browser window. It provides the user with options of selecting the language (Amharic or English), entering the query based on the selected language, submitting the query, and viewing the results of the request. As a result, the user can type either in Amharic or in English language and get the results in both languages in the same window of different frames.

The user interface of our system is developed using a JavaServer Page (JSP) which is part of the servlet standard. JavaServer Pages (JSP) is a server-side programming technology that enables the creation of dynamic web pages and applications. It allows a web browser to make requests to the java application container called web server and to display dynamic contents to the viewer. We have chosen this programming language because of several reasons. JSPs have the advantage of:

- Embedding Java codes into HTML, XML, DHTML, or other document types.
- Displaying dynamic content to the viewer
- Reducing Web development and maintenance costs
- Creating Web applications easier and faster
- Making Java APIs or frameworks available to Web designers
- Moving Java code outside of existing JSP pages
- Reusing Java code in multiple Web applications, and
- Using the advantage of the Java motto of *"write once, run anywhere"*

5.4 Query Preprocessing

The query preprocessing step is the first step in every IR system. The process requires a special attention when the IR system is cross language IR, because the preprocessed query should be first translated to other language before it has been feed to the search engines. When the translation is based on dictionary lookups, the preprocessed query terms should have a match in the bilingual dictionary together with its appropriate translation. This step will have different

stages and may implement different techniques depending on the characteristics of the language. Most of the steps in the query preprocessing are language specific and needs knowledge about the language to develop the tools required. In this subtopic, the implementation details of each of the components required for both Amharic and English query preprocessing before they have been translated from Amharic to English or vice versa has been described.

5.4.1 Tokenization

In query preprocessing tokenization means splitting the query sentence or phrase into individual terms or words based on some criteria of the language such as word delimiters. In our work, the user can enter either Amharic or English languages, so there is a need to tokenize both Amharic and English language queries based on the languages characteristics.

Amharic query tokenization

To implement the Amharic query tokenizer, we have predefined Amharic word delimiters such as white space and Amharic punctuation marks (like: **ነጠላ ሰረዝ(፣), ድርብ ሰረዝ(፤), የጥያቄ ምልክት(?),** etc) to split individual words.

English query tokenization

Like Amharic query tokenizer, the English query tokenizer is also developed using white spaces and English punctuation marks (such as: comma (,), semi colon (;), colon (:), etc) as a criteria for word delimiters to split the query statement into individual bags of words.

5.4.2 Stop Words Removing

As discussed in Chapter Four Section 4.2, stop words has no significance content to search information from the web or from any collected documents. As a result, they have to be removed from the query sentence or phrase before they have submitted to the translation component. Since stop words removing is a language dependent task, it needs a standard list of stop words for each language.

In this work, even though Amharic does not have standard stop word lists, we have used a set of common Amharic stop words lists that are used by [2, 17, 55] as shown in Appendix I. As it has been stated in these research works, the stop words are collected from Amharic literatures as well as researchers from the Ethiopian Language Research Institute. For English stop words, there is a

standard stop word lists which is available on the Web although it can be modified based on the systems' requirements. For our system, we have used a list of English stop words as in Appendix II.

5.4.3 Stemming

Stemming is an important analysis step in a number of areas such as natural language processing (NLP), information retrieval (IR), machine translation (MT) and text classification [53]. Several research works showed the importance of stemming from two different perspectives. The first one is from the positive impact of stemming in the dictionary based cross language IR, where there is a need in the translation step, to look up terms in a machine readable dictionary (MRD). In this regard, a research work conducted in [53] showed that in cross language information retrieval (CLIR) stemming reduces the size of term entries in the dictionary which in turn increases the effectiveness of the translation system. Words in machine readable dictionaries are usually entered in their citation forms so stemming is required to find term matches in the dictionary during query translation. The second one is from the application of stemming during indexing to reduce the vocabulary size in the index, and it is used during query processing in order to ensure similar representation as that of the document collection. In this regard, a research work in [52] showed the importance of stemming words in the query and retrieved text documents to facilitate searching database of Amharic texts and to increase the effectiveness of IR systems.

Amharic Stemming:

In this research work, after the stop words are removed from the tokenized Amharic query, a stemmer that stripped off the prefixes and suffixes are performed. The stemming algorithm developed by Alemayehu and Willett [52] and latter customized by Tessema[2] is adopted for our work. This stemming algorithm is an aggressive one which stems both inflectional and derivational morphemes. The algorithm also changes the last character of each stem word to *sadis*(the sixth order of Amharic characters). However, the dictionary used in this work consists of an entry for words and their derivational variants. So, query terms that do not have *sadis* ending characters couldn't have matches in the dictionary. Table 5.1 shows the results of the stemmer.

Table 5.1: **Previous stemmer result**

Query term	Stemmer result
ምርጫ	ምርጭ
በላ	በል
ሰበካ	ሰበክ
ስዳጋ	ስዳግ

In addition to this, the algorithm also changes the last character of every proper name to *sadis* which give inappropriate results to our transliteration component which leads inappropriate transliteration of proper names of which don't have *sadis* endings. Table 5.2 shows the results of the stemmer and its impact on the transliteration component.

Table 5.2: **Results of the stemmer on proper names and its impact on the transliteration system**

Proper name	Stemmer result	Transliteration result	Actual transliteration
ፈለቀ	ፈለቅ	felek	feleke
ግርማ	ግርም	girm	girma
ታሪኩ	ታሪክ	tarik	tariku
መቀለ	መቀል	mekel	mekele
ደሴ	ደስ	des	desie

To alleviate this problem, the original algorithm is modified to fit for our system. The stemmer module in our query preprocessing component only strips off prefixes and suffixes and leave the results as it is without changing the last character to *sadis*. In doing so, the proper names are also transliterated in their normal forms.

English Stemming:

Like Amharic words, dictionary entries of English words are usually the base words and their derivations. Therefore, to translate the query terms from English to Amharic, using a machine readable dictionary, preprocessing the query and stemming the suffixes and prefixes of the inflected words are highly advisable. In our work, Porter Stemmer algorithm of the Lucene API

is used together with our EnglishAnalyzer module to produce the stems of English query terms. The Porter stemming algorithm (or 'Porter stemmer') is usually used to remove the commoner morphological and inflectional endings from English words.

5.4.4 Normalization

As we have discussed in Chapter Four, Section 4.2, normalization, in our work, means replacing one character with another if the two characters have the same sound. The implementation of this module is done to handle two basic functionalities as we have described below.

The first functionality is replacing characters with one common and frequently used character if the characters have the same sound. For example, the word "ሠርፀ"can be replaced by "ሰርጸ" with the assumption that the Amharic characters ሰ and ጸ are commonly used in literatures than the characters ሠ and ፀ and they do have the same sound respectively. For this work, characters which have different symbols for the same sound and which are identified by different researchers (Tessema Mindaye [2] and Seid Muhie[54]) are considered.

The second functionality of this component is expanding short words which are shortened by the forward slash ('/') into one or two words. For example, ት/ት is expanded to ትምህርት and ሃ/ስላሴ is expanded to ሃይለ ስላሴ. For this work, lists of common Amharic short words (Appendx III) which are collected from the work of Melese Tamiru [55] are considered. After the Amharic query sentence is tokenized, each word is checked from the list and if it is found, it will be expanded to its corresponding expanded form.

5.5 Query Translation

In any cross language information retrieval (CLIR) system, query translation is the core component which needs a lot of efforts has to be exerted. If the query translation is made using a bilingual dictionary, there may be two fundamental research tasks to be considered [56].

1) How to improve the coverage of the bilingual dictionary and
2) How to select the correct translation of the query among all the translations provided by the dictionary. The first task is the major focus of this research.

Even though dictionary-based query translation is one of the conventional and commonly used approaches in CLIR, the appearance of Out-Of-Vocabulary (OOV) terms is one of the main

difficulties arising with this approach. These OOV terms are often proper names, borrowed words or newly created words which can't be usually found in bilingual dictionary [56]. This problem significantly limits the retrieval performance of the CLIR system. In this Section, we have described the implementation details of designing our Machine Readable Dictionary (MRD) for Amharic-English and vice versa, the lexical transfer between the two languages, and the solution proposed to improve the coverage limitations of the bilingual dictionary.

5.5.1 Dictionary Design

A dictionary-based query translation system requires a Machine Readable Dictionary (MRD) to translate a query from one language to another language. However, during this research work we are unable to get an MRD for Amharic-English. As a result, we have been forced to develop our own bilingual MRD for our query translation system. To accomplish this task, we have used two hard copy dictionaries (Amharic-English [58] and English-Amharic [57]) for our translated word database entry. The following are the steps involved during the development of the dictionary.

Step 1: Word selection

To do this task, the words in the two dictionaries are carefully selected in order to exclude:

1) **Stop words** such as yours, ourselves, about, above, across, after, again, etc from English-Amharic dictionary and **ሁሉም, ኋላ, ሁኔታ, ሆነ, ሲሳ, ማለት**, etc from Amharic-English dictionary.
2) **Inflectional morphologies of root words** such as past tense and past perfect tense form of regular verbs, plural form of nouns, "-ing" form of verbs, etc from English-Amharic dictionary and words which include prefixes and suffixes(such as words which include **ከ, የ, ስ**, etc as a prefix and **ሳች, ሳችሁ, ምች, ም, ች, ን, ቱ**, etc as a suffix) from Amharic-English dictionary.
3) Words which are written in characters of **ሐ, ኀ, ሠ, ፀ**, etc and their derivatives from both dictionaries for Amharic words. Instead of entering them directly in the MRD, we replaced the characters with other characters which have the same sound as the normalization module does.
4) Words which have the same pronunciations in both languages such as: sport, lottery, hotel, film, piano, kilo, etc.

Step 2: Development environment

After the words have been carefully selected, they are typed into Microsoft Access database management system. The main reason why this database management system is selected for words and their translation entry is to use the advantage of the easier user interface of Ms Access particularly for Amharic character entries.

After all the required words are entered into the database, the Ms Access database is migrated into MySQL Server database management system in which a database with the same data types and which can support utf-8 encodings and Amharic character sets has been created. MySQL Server is open source software, cross platform, fast to access, and can support Amharic character to be entered and accessed by Java DataBase Connectivity (JDBC) for manipulation by using java programming language. These advantages of MySQL Server are the main reasons for the migration of the database from Ms Access to MySQL Server.

Step 3: Accessing the database

The database in the MySQL Server is created considering its support of Amharic character sets. The JDBC connection that allows the Java programs to communicate with the database is also established considering its support of Amharic characters. After this setup is performed, we are able to request the database using both Amharic and English query and we have got the response in any of the languages based on the query language.

5.5.2 Lexical Transfer

After the query preprocessing step is accomplished, each of the remaining stemmed bags of words in the query will be checked for its presence in the dictionary. If the match is found, the corresponding translation will be retrieved and given to the dispatcher to provide them for the appropriate search engine. However, there may be two other possibilities: either the word may have more than one translation or the word may not be totally found in the dictionary. We have discussed the techniques used to solve these problems below.

If a given word has more than one translation, it will create the problem of translation disambiguation which refers to finding the most appropriate translation from several choices in the dictionary. As stated in [56], there are several approaches to solve this problem. Among them, one approach is to select the most likely translation, usually the first one offered by a

dictionary. Another solution is to use all possible translations in the query with the OR operator. Even though the second approach introduces noise into the query which leads to the retrieval of many irrelevant documents, we have used this approach as it is likely to include the correct translation. This means that one single Amharic term could in our case give rise to many possible alternative English term translations. At the query level, this means each query was initially maximally expanded and all will be considered for web document searching.

In the other hand, if the word is not found in the dictionary, it will be given to the transliteration module assuming that it is either a proper name or borrowed word that will not be often found in the dictionary and usually have the same pronunciation in both languages. The algorithm used for lexical transfer is shown in Figure 5.1.

```
Input: Amharic/English bags of words
Output: Amharic/English bags of words

For each stemmed word in a query
        Check presence of the word in the dictionary
                If the word is present

                        While next record is not equal to end of database

                            Retrieve the corresponding translation

                        End while loop

                Else

                    Call the transliteration module

                End if

        End for loop
```

Figure 5.1: **Lexical transfer algorithm**

For our work, the Java programming language API(Application Program Interface) called JDBC(Java DataBase Connectivity) is used for accessing the database and lexical transfer between the two languages. JDBC defines how a java programmer can access the database in tabular format from Java code using a set of standard interfaces and classes written in the Java programming language.

5.5.3 Transliteration

As we have discussed in Chapter Four, some of the reasons why dictionary-based translation has been commonly used in cross-language information retrieval is because bilingual dictionaries are widely available, dictionary-based approaches are easy to implement, and the efficiency of word translation with a dictionary is high. However, due to the vocabulary limitation of dictionaries, the translations of some words in a query cannot be often found in a dictionary.

Proper names, such as personal names and place names, are a major source of Out Of Vocabulary (OOV) terms because many dictionaries do not include such terms. It is common for proper names to be translated word-by-word based on phonetic pronunciations. The process of pronouncing a word (proper name, borrowed word, etc) in one language in a similar manner with in another language is called transliteration [12, 56]. As a result, transliteration has become one of the solutions for the problem of coverage limitation of bilingual dictionaries during query translation.

In this subtopic, we have described the Amharic-English and back transliteration between the two languages. The main reason of using transliteration is to handle OOV terms that do not present in Amharic-English dictionary. The transliteration process keeps approximate phonetic equivalents between the two languages. Since the dialects of the two languages use different scripts, inter-dialectal translation without lexical changes is quite useful. In our work, a heuristic based approach which uses phoneme mappings is implemented to solve the problem of OOV terms.

Amharic-English transliteration

The Ethiopic writing system is a rich syllabary of at least 34 consonant classes each having generally seven or eight forms, and fewer having twelve or thirteen [60]. Which means each

letter or symbol usually represents the whole syllable like ‘ዳ’, ‘ዱ’, ‘ላ’, or ‘ሉ’ for ‘da’, ‘du’, ‘la’, or ‘lu’ respectively. For this work, the conventions used for finding the closest match between the Latin and Ethiopic phonetic system called System for Ethiopic Representation in ASCII (SERA), is used as our best reference to develop the transliteration module. This convention is mostly developed for use of typing Ethiopic scripts using Latin keyboard layouts. However, in our work, the transliteration module is used for transliterating words written in Ethiopic script into Latin script based on their Amharic pronunciations. Therefore, minor convention modifications have been done in some order of the Amharic characters to fit for our system as shown in Appendix IV. For example, each fifth order (*hamis*) of each Amharic character is represented by its consonant representation plus “E” in SERA, such as: ሄ=hE, ሌ=lE, ሜ=mE, etc, but for pronunciation this is not quite useful. Therefore, we have used ሄ=hie/he, ሌ= lie/le, ሜ= mie/me, etc to pronounce these characters in English language. Table 5.3 shows some exemplary characters used in the representation of Amharic characters in English.

Table 5.3: **Character transliteration between the two languages**

Amharic Char	**1st order**	**2nd order**	**3rd order**	**4th order**	**5th order**	**6th order**	**7th order**	**8th order**
	ግዕዝ	**ካዕብ**	**ሳልስ**	**ራብዕ**	**ሃምስ**	**ሳድስ**	**ሳብዕ**	**ዳጉማ**
ሀ	he	hu	hi	ha	hie/he	h/hi	ho	
ለ	le	lu	li	la	lie/le	l/li	lo	lua
መ	me	mu	mi	ma	mie/me	m/mi	mo	mua
ሰ	se	su	si	sa	sie/se	s/si	so	sua
ረ	re	ru	ri	ra	rie/re	r/ri	ro	rua
ሸ	she	shu	shi	sha	shie/she	sh/shi	sho	shua
ቸ	che	chu	chi	cha	chie/che	ch/chi	cho	chua

The other modification is on *sadises* depending on their position in the word. For example, the Amharic proper name “ሸመልስ” can be transliterated as “***shimelis”*** in English. In this case, *sadises* can be transliterated as their Latin consonantal representation plus the vowel ***“i”*** or the Latin consonantal character only, based on their position in the word as in the example ***ለ=****li,* ***ሸ=****shi and* ***ሰ=****s* although all are *sadises*. This means when a *sadis* present at the beginning of the word, or the character before it is also *sadis,* or before the last character except for some exceptions, they are represented by consonant plus the vowel ***“i”*** (as ***ሸ=****shi,* ***ለ=****li*) but the

consonant character only (as *ሰ=s*) when they appear at the end of the word or at the remaining places and conditions [19, 59]. Figure 5.2: shows the algorithm for transliterating a word from Amharic to English.

Input: Amharic bags of words
Output: English bags of words

For each Amharic stemmed query word not found in the dictionary

Split the word into its Amharic character

For all the characters in the word

If the character is at the beginning of the word and the character is sadis

Replace with its Latin conventional 'Xi'

Put the characters at the beginning of the word

Else if the character is sadis and the character before it is also sadis

Replace with its Latin conventional 'Xi'

Append the character in its appropriate place

Else if the character is sadis and it is before the last character

Replace with its Latin conventional 'Xi' except for some exceptions

Append the character in its appropriate place

Else

Transliterate the character based on the transliteration convention

Concatenate the characters

End if

End for

End for

Figure 5.2: **Algorithm for Amharic-English transliteration**

English-Amharic transliteration

The automatic back transliteration of Amharic-English is not an easy task to implement and it needs a lot of efforts to be done, and it also needs the involvement of linguistic experts who are fluent in both languages. As a result, for our work, we have used list of manually transliterated proper names in a separate table to transliterate proper names written in English to Amharic language. Most of the collection of proper names called gazetteers have been collected from the work of Seid Muhie[54], he used it as a named entity recognizer for gazetteer based answer selection in his work. Figure 5.3 shows the algorithm used for transliteration of English proper names into Amharic.

```
Input: English bags of words
Output: Amharic bags of words
For each English stemmed query word not found in the dictionary
        Check it in the proper name table
                If the word is present
                        While next record is not end of database
                                Retrieve the corresponding transliteration
                        End while loop
                Else
                        Leave the word as it is.
                End if
End for
```

Figure 5.3: **Algorithm for English-Amharic proper name transliteration**

5.6 Web Document Crawling, Indexing and Searching

As we have discussed in the previous Section of this Chapter, we have used an open source web search engine called Nutch for our English web document crawling and indexing; and an Amharic search engine which is developed by Hassen Redwan [12] by customizing Nutch for our Amharic Web document crawling and indexing. The Nutch search engine consists of roughly four main components: ***Crawler***: which discovers and retrieves web pages, ***WebDB:*** a custom database that stores known URLs and fetched page contents, ***Indexer:*** which dissects pages and builds keyword-based indexes from them, and ***Search Web Application***: which is a JSP application that can be configured and deployed in a servlet container.

Nutch requires up to a gigabyte of free disk space and a high-speed connection for searching and indexing web documents. As a result, several configurations and customizations have been made to bring the search engine for our context. The configurations are made for both crawling and searching Web documents. In this subtopic, we have described some of the configurations made on the search engine.

5.6.1 Web Document Crawling and Indexing

In order to use the Nutch search engine crawler, first we have to configure and change the Nutch default property values. Second, we have to run the Nutch crawler using the crawl command in UNIX like environment.

Configuring the Crawler

The first step for configuring the crawler is creating a directory with a flat file of root URLs. A list of seed URLs are created under the urls directory which is the sub-directory of nutch-1.0 with a file name urls. In this file, the name of the sites intended to be crawled are listed. The sites are chosen because of the following three main reasons.

- They host and archive current issues about Ethiopia
- They frequently accessed by Web users
- Some of them host both Amharic and English language

From the sites used for English Web document crawling, some of them host only English language documents and some of them are bilingual (Amharic and English) Web sites. The

bilingual web sites are used for both Amharic and English web documents crawling. The sites used for only English Web document crawling are the following.

- ***http://www.addisfortune.com/:*** a website for Fortune, Ethiopian business newspaper.
- ***http://www.capitalethiopia.com/:*** a website for Capital, Newspaper striving to promote free enterprise in Ethiopia, and inform the public at large about economic events.
- ***http://www.ethioobserver.net/:*** a website for Ethiopian Observer, which provides news and commentaries regarding current issues on Ethiopia.

From the sites used for Amharic Web document crawling, some of them host only Amharic language documents and some of them are bilingual (Amharic and English) Web sites. The sites used for only Amharic Web document crawling are the following.

- ***http://am.wikipedia.org/:*** a Wikipedia website in Amharic language.
- ***http://www.addisadmass.com/:*** a website for Addis Admass newspaper.
- ***http://archives.ethiozena.com/:*** a website for EthioZena which consists of selected articles from well known newspapers and magazines published in Amharic.
- ***http://www.dw-world.de/amharic/:*** is an Amharic website from Germany's international broadcaster. It broadcasts news and information on Internet in Amharic language.

The bilingual websites used for both Amharic and English Web document crawling are the following.

- ***http://www.ena.gov.et/:*** a website for the Ethiopian News Agency, the state news service.
- ***http://www.ethiopianreporter.com/:*** a website for Ethiopian Reporter, a daily newspaper based in Addis Ababa.
- ***http://www.waltainfo.com/:*** Walta is pro-government news site based in Addis Ababa. It includes news coverage from EPRDF point of view.
- ***http://www.ethpress.gov.et/:*** a website of Ethiopian press agency, particularly for an official Ethiopian newspaper called Herald.

The second step is editing the file ***conf/crawl-urlfilter.txt*** and replacing ***MY.DOMAIN.NAME*** with the name of the domain we wish to crawl. In our work, the domain names of all the seed

URLs are written with the line, for example: ***+^http://([a-z0-9]*\.)*addisfortune.com/,*** to include any url in the 'addisfortune' domain.

The third step is editing the file ***conf/nutch-site.xml***. In this step, a lot of property value changes have been performed to fit the search engine for our system. The property value changes are done as shown in Appendix V. Some of the properties for which value changes are made during the configuration of the crawler include:

- *http agent name:* http User-agents request header
- *http robots agents:* the agent strings we'll look for in robots.txt files
- *http timeout:* http default network timeout in milliseconds
- *http proxy host:* http proxy host name
- *http proxy port:* the proxy port number, etc

Running the crawler

After things are configured running the crawler is performed using the *crawl* command which results the crawling processes shown Appendix VI. In this stage, we have specified some variables such as the directory which contain list of sites as *urls*, the directory to put the crawled content in as *crawl*, the depth of the crawling which indicates the link depth from the root page that should be crawled as *10*, the number of threads that will be used to fetch in parallel as *10*, and the crawling log files as *crawl.log*. Once crawling is successfully completed, the log files and the crawl directory which contains the WebDB, indexes, and segments will be created. With this set up the Amharic crawler collected 21185 Web documents and the English crawler has collected 16316 Web documents. Figure 5.4 shows the crawl command with the variable specifications in a Cygwin windows environment to run the Nutch crawler.

Figure 5.4: **Running the crawler with the crawl command**

5.6.2 Web Document Searching

After we have verified that the crawling is successfully completed, we have proceeded to setting up the web interface to search Web documents. For searching using the web interface, we have put the Nutch war file into the Apache Tomcat web application server servlet container. Then in our web application folder, we have edited the ***nutch-site.xml*** file to set up our searcher directory as shown in Figure 5.5.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
        <property>
                <name>searcher.dir</name>
                <value>D:/nutch-1.0/crawl/</value>
        </property>
</configuration>
```

Figure 5.5: **Nutch searcher directory set up**

After the searcher directory set up has been performed as shown in the above Figure, we have displayed the user interface of our Amharic-English bilingual search engine which is designed using Java Server Page (JSP). The user interface is displayed using Microsoft Internet explorer browser window. The user interface provides facilities to select the query language and to enter a query in a text box. In the query text box, we can write a query for searching from the crawled and indexed web documents.

CHAPTER SIX

EXPERIMENTAL RESULTS

The purpose of the experiment is to evaluate the effectiveness of our dictionary based bilingual search engine. The effectiveness of cross language information retrieval systems is often measured by using precision and recall. The evaluation is usually performed by comparing the results of cross language runs with the corresponding monolingual runs of the same system by varying the queries. To do this, one needs to have the appropriate test collection such as document collection, test queries, and their relevance judgments.

The test collections that have been used for CLIR experiments are usually provided by the three major evaluation workshops; the Text REtrieval Conference (TREC) for Cross Language Track, the Cross-Language Evaluation Forum (CLEF) covering many European languages, and the NII Test Collection for IR systems (NTCIR) for Asian languages evaluation covering Chinese, Japanese and Korean[34]. In these workshops, the task was to match the queries in one language against documents collection in another language and to return a ranked list of results. The experiments of most of the previous CLIR studies were conducted using standard document collections provided by these organizations. These collections usually consist of documents prejudged and carefully selected by human experts for evaluation purposes.

However, for web based IR systems, there is no established relevance judgment available for precision and recall. As a result, precision is usually considered and reported for web based IR systems at low recall levels. This is because, there is no proper method of calculating absolute recall of search engines as it is impossible to know the total number of relevant documents in huge number of web pages. Since our study used Web pages instead of standard collections, the traditional CLIR evaluation techniques have not been applicable. As a result, in order to evaluate the performance of our system, an experiment was designed and conducted to test the precision instead of the recall since it does not require the knowledge of the test collections. In this Chapter, we have described the approaches followed to evaluate our system, the query preparation for testing the system, the discussions on the experimental results and the possible reasons behind the results found.

6.1 Experimental Approaches

6.1.1 Cross-Lingual Performance Evaluation

There are two commonly used approaches to evaluate CLIR systems [61]. The first approach is manually translating test queries that are written in one language into other language pair and using the original test queries and the translated queries to retrieve the same document collections which are written in the translated query language. Then the performance of the cross-language information retrieval system can be evaluated by comparing the results of the monolingual runs directly with the cross-language equivalents as shown in Figure 6.1. The disadvantage of this evaluation technique is the manual translation requires the application of human judgment, and evaluation collections constructed in this way exhibit some variability based on the terminology chosen by a particular translator.

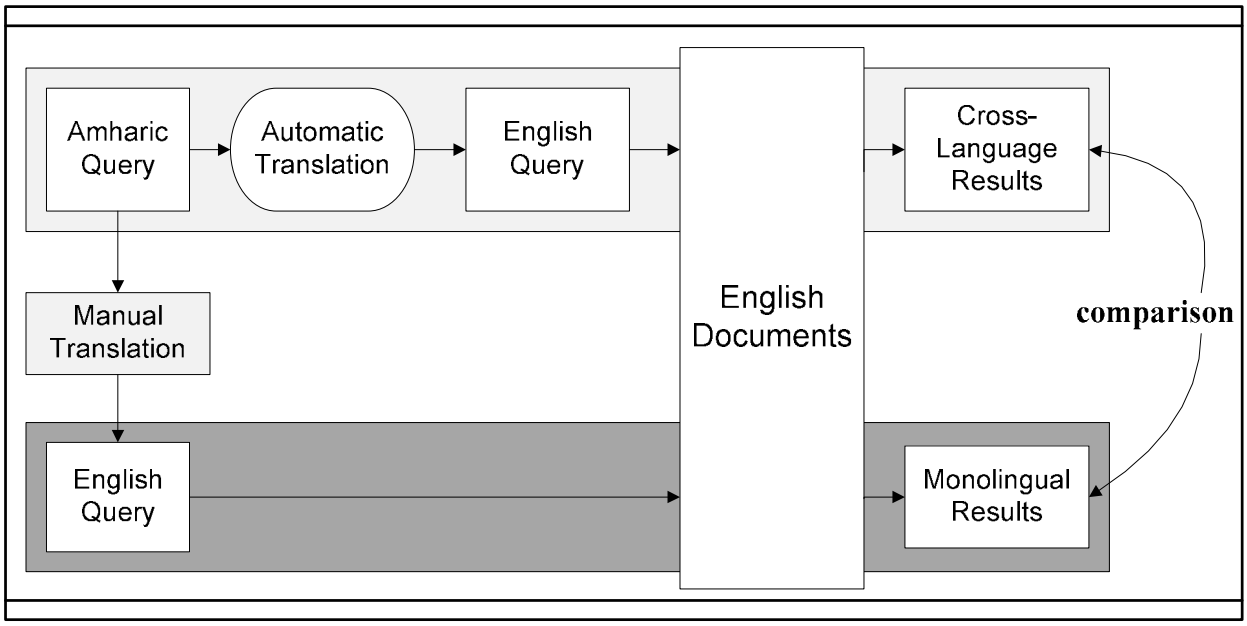


Figure 6.1: **Bilingual CLIR system evaluation using manually translated query.**

The second approach is using queries formulated in one language and retrieving documents which are written in the two language pairs and stored in a parallel corpus. Then the precision of the results of the queries for each recall for documents which are written in both languages will be compared. The disadvantage of this technique of evaluation is it requires a collection of parallel corpus which is difficult to get particularly for under resourced languages.

The first approach, i.e. using manually translated test queries is a widely used evaluation strategy [61] for under resourced languages such as Amharic because it permits existing test collections in any of the languages to be inexpensively extended to any language pair for which translation resources are available. As a result, we have used the first approach to test the performance of our bilingual search engine. We conducted our cross language retrieval experiments using both Amharic and English queries and on both Amharic and English language Web document collections since our system is a bidirectional (two-way) bilingual CLIR system.

6.1.2 Bilingual Results Evaluation

Our system is also designed and implemented to retrieve documents in two languages (Amharic and English) for a single query in Amharic or English language. As a result, to visualize the significance of our bilingual system, the relevant bilingual retrieval results of our system (Amharic and English documents) are compared with the relevant results of monolingual search engines as shown in Figure 6.2.

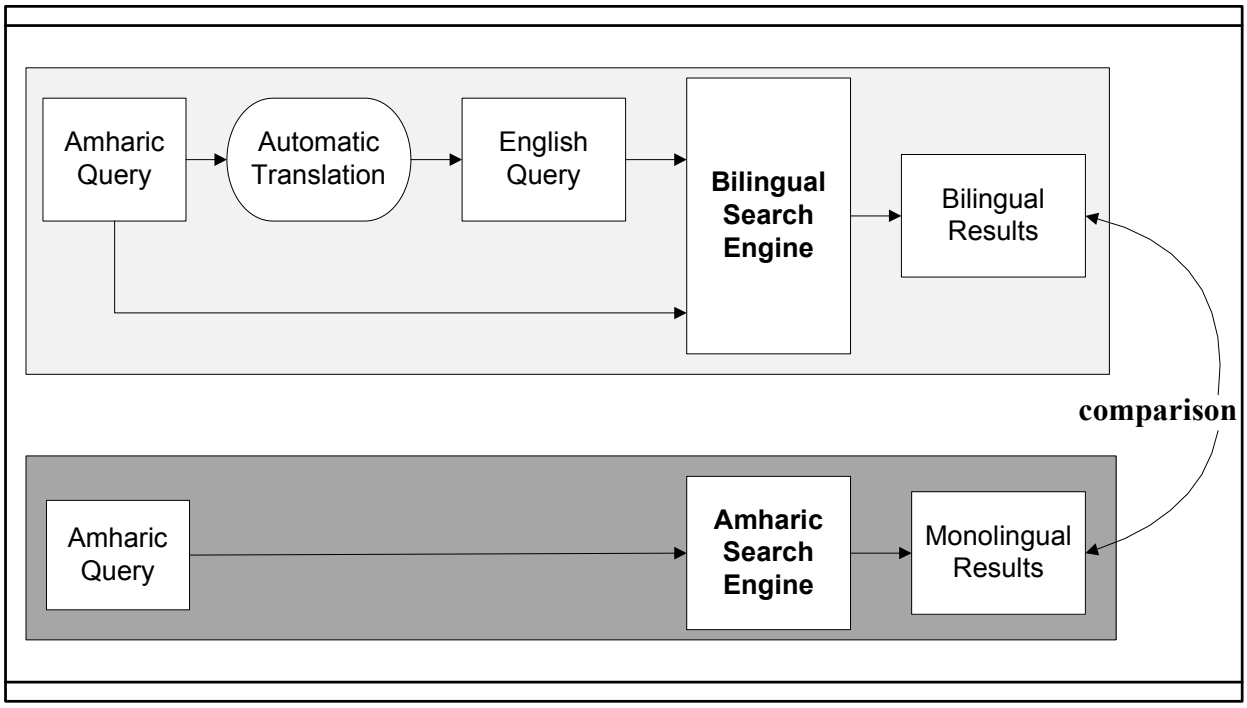


Figure 6.2: **Monolingual and Bilingual results comparison**

6.2 Test Query Preparation

The bilingual websites such as; *www.ethiopianreporter.com*, *www.ena.gov.et* (Ethiopia News Agency), *www.ethpress.gov.et* (Ethiopian Press Agency), and *www.waltainfo.com* (Walta Information Center) which are selected and used for web document crawling are the bases for our experimental query preparation. The websites present and archive news and current information mostly about Ethiopian events in both Amharic and English languages. However, it doesn't mean that the websites are exactly contains parallel aligned web documents but they do have contents about the same information in the two languages. The queries are prepared considering the current hot issues in the context of Ethiopia. After assessing these sites, we have understood that the queries prepared in one language will have a translation in another language Web documents. Our assumption is the Amharic queries selected for monolingual retrieval will have English translations in the English web documents and will be used for monolingual Amharic as well as cross-lingual English web document retrieval simultaneously, and it is true for English queries.

We have prepared 15 (fifteen) Amharic queries to evaluate our bidirectional Amharic-English CLIR system. The average length of these queries is 2.33, which is close to the average Web query length of 2.21[34]. These Amharic queries are given to a linguist (foreign language department post graduate student) who is flaunt in both Amharic and English languages and manually translated into other 15(fifteen) English queries. The English queries were used for the evaluation of English-Amharic cross-lingual retrieval and the precision comparison of the English monolingual retrieval with Amharic-English cross-lingual retrieval. With a total of 30(thirty) queries, we performed thirty different searches having two different results for each search using the system. A total of 60(sixty) different search results were retrieved for our system evaluation.

6.3 Query Translation Results of the Translation Component

The effectiveness of any cross-lingual search engine is highly dependent on the performance of the query translation component. The query formulated for cross-lingual information retrieval need to be properly translated into the target document language before it has been submitted to the search engine. To visualize the effectiveness of the translation system, the Amharic query

translation results of the 15 Amharic queries together with their manual translation and the English query translation results of 15 English queries are shown in Table 6.1 and Table 6.2 respectively. The discussions on the query translation results are also described in this Section.

Table 6.1: **Manual and Automatic Amharic query translation results**

No	Amharic Queries	Manual Translation	Amharic Query Translation Component Result
1	ምርጫ 2002	Election 2002	Election 2002
2	መለስ ዜናዊ	Meles zenawi	Meles zenawi
3	የኢትዮጵያ ምርጫ ቦርድ	Electoral Board of Ethiopian	etyopiya election board
4	የአለም ዋንጫ 2010	World cup 2010	World cup 2010
5	ብርቱካን ሚደቅሳ	Birtukan Mideksa	Orange mideksa
6	የኢትዮጵያ ፖለቲካ ፓርቲዎች	Ethiopian political parties	etyopiya politics party
7	ቢዝነስ እና ኢኮኖሚ	Business and economy	Business economy
8	ቀነኒሳ በቀለ	Kenenisa Bekele	Kenenisa Bekele
9	የት/ት ጥራት	Quality education	Quality education
10	የአማራ ልማት ማህበር	Amhara Development Association	Amara Development Association
11	እግር ኳስ	football	Foot ball
12	የጣና በለስ ፕርጀክት	Tana Beles project	Tana les project
13	ዲሞክራሲያዊ ፓርቲ	Democratic party	Democracy party
14	የአማርኛ ቋንቋ	Amharic Language	Amharic Language
15	የእርሻ ልማት	Agricultural Development	Agriculture Development

The fifteen Amharic queries selected for the cross-lingual evaluation, which have a total number of 36 words, are as shown in Table 6.1. Out of these, 11 words are proper names of which 3 words are inflected with prefixes; since they are OOV terms, they need the transliteration component for translation, 2 words are numbers which don't need any translation, 1 word is a stopword which has to be removed during query preprocessing, and from the remaining 22 words which need the bilingual dictionary for translation, 5 are inflected with suffixes and prefixes of which 1 word is a short word.

Table 6.2: **Automatic English query translation results**

No.	English Query	English Query Translation Component Result
1	Election 2002	ምርጫ 2002
2	Meles zenawi	መለስ ዜናዊ
3	Electoral Board of Ethiopian	ምርጫ ቦርድ ኢትዮጵያ
4	World cup 2010	አለም ዋንጫ 2010
5	Birtukan Mideksa	ብርቱካን ሚደቅሳ
6	Ethiopian political parties	ኢትዮጵያ ፖለቲካ ፓርቲ
7	Business and economy	ቢዝነስ ኢኮኖሚ
8	Kenenisa Bekele	ቀነኒሳ በቀለ
9	Quality education	ትምህርት ጥራት
10	Amhara Development Association	አማራ ልማት ማህበር
11	Football	እግር ኳስ
12	Tana Beles project	ጣና ንቨጀክት
13	Democratic party	ዲሞክራሲ ፓርቲ
14	Amharic Language	አማርኛ ቋንቋ
15	Agricultural Development	እርሻ ልማት

6.3.1 Discussions on the Results of Amharic-English Query Translation Component

Out of the 15 queries, 10 queries are properly translated. When we see it word wise, out of the 36 words, 31(88.57% of the words) words are properly translated. The translations of the remaining 5 words which are the cause for the improper translation of the 5 queries are not exactly the same as the manual translation. The problem arises because of different reasons. For example, the word *'ኢትዮጵያ'* and *'አማራ'* are transliterated as *'etyopiya'* and *'Amara'* instead of *'Ethiopia'* and *'Amhara'* respectively, because the transliteration component doesn't handle some 'irregular' transliterations, such as the appearance of the character *'h'* in both words. On the other hand, the proper name *'ብርቱካን'* is directly translated to *'Orange'*, the word in the bilingual dictionary, because of the absence of automatic named entity recognizer. In addition, in the query *'የጣና በለስ ንቨጀክት'*, the word *'በለስ'* is transliterated as *'les'* because the character *'በ'* is considered as Amharic prefix and stripped by the Amharic stemmer.

6.3.2 Discussions on the Results of English-Amharic Query Translation Component

To test the effectiveness of the English query translation component, the manual translations of the 15(fifteen) Amharic queries are used. As we can see in Table 6.2 above, out of the 15 queries, 14(93.33% of the queries) queries are properly translated. This shows that the English query translation component has better query translation performance than the Amharic query translation component. This is because the English translation component uses a database to store and translate proper names instead of using the character map transliterations as the Amharic query translation component. However, since it is impossible to list out all the proper names, some proper names are totally missed from the translated queries. For example, from the query *'Tana Beles project'*, the term *'Beles'* is missed from the translated query. This is because; this word is not found either from the bilingual dictionary or from the proper name database. But it could be handled if there was a transliteration module as the Amharic-English query translation component.

6.4 Evaluation of the Transliteration Component

The transliteration component is also independently evaluated for its performance. To do this, we have collected 1100 (one thousand one hundred) proper names of which most of them are person names and some are the names of countries, places, hotels, well noun cities, etc. Most of the proper names are collected from the gazetteers used for English-Amharic manual transliteration and few of the names are collected from the Web and the department of Computer Science as we are trying to include the names of Addis Ababa University Computer Science graduate students and the staff members of Computer Science department.

The manual transliteration of the gazetteer is made with the help of linguists. For words which are ambiguous for transliteration, a free English-Amharic online transliteration service of the Google API (*http://www.google.com/ta3reeb/*) is also used just for cross check although this transliteration service has also few problems.

6.4.1 Discussions on the Transliteration Results

Out of the 1100 proper names, 887(80.64% of the names) names are properly transliterated. The remaining 213(19.36% of the names) names have detected to occur having minor and major spelling problems because of different reasons. The causes of the problems that degrade the performance of our transliteration component are viewed from different angles as follows.

1. ***Transliteration Variations:*** some Amharic characters use different English characters for the same pronunciation during transliteration as shown in Table 6.3.
   For example:
   a. the character "ከ" and its derivatives use the English characters 'C', 'K', and 'Q' in different words
   b. the English characters 'ph' are pronounced as 'ፈ' when they appear as written
   c. the Amharic character 'ጀ' and its derivatives use sometimes 'g' instead of 'j'

Table 6.3: **Transliteration variations**

Amharic Word	Manual Transliteration	System Result	Cause of the problem
ሞዛምቢክ	Mozambique	mozambik	1a
ሞሮኮ	Morocco	moroko	1a
አንታርክቲካ	Antarictica	antarkitika	1a
ኳታር	Qatar	kuatar	1a
መኳንንት	Mequanint	mekuannit	1a
ሚካኤል	Michael	mikaael	1a
ፊሊፒንስ	Philippines	filipinis	1b
ኤፍሬም	Ephrem	aefrem	1b
አርጀንቲና	Argentina	Arjenitina	1c
ጆርጂያ	Georgia	jorjiya	1c
ናጄሪያ	Nigeria	nayjeriya	1c

2. ***Irregular Transliterations:*** some English words include characters which have not been pronounced or they can be transliterated in a different manner. For example: Djibouti, Jerusalem, Ghana, Amhara, etc although their Amharic transliteration is ጂቡቲ, ኢየሩሳሌም, ጋና, አማራ, etc respectively.

3. ***Double Consonants:*** some Amharic words include double characters to indicate longer consonants in pronunciation as the length of the consonant may often affect the meaning of a word. For Example: Gessess, Mohammed, Tullu, Jimma, Gojjam, Teppi, etc for '**ገሠሠ**', '**ሞሏሐዱ**', '**ቱሉ**', '**ጅማ**', '**ጎጃም**', '**ቴፒ**', etc respectively.
4. ***Vowel Usage Variations:*** the Amharic characters '**ያ**', '**ዩ**', '**ዮ**', etc are represented as 'ia' or 'ya', 'u' or 'yu', 'yo' or 'io', etc respectively. For example: Algeria for '**ኣልጄሪያ**' and yaregal for '**ያሬጋል**' for the character '**ያ**', Uganda for '**ዩጋንዳ**' and Kalayu for '**ካሳዩ**' for the character '**ዩ**', Yonas for '**ዮናስ**' and Tsion for '**ጽዮን**' for the character '**ዮ**', etc
5. ***Problems Related to Sadises and Hamises:*** when Amharic 'sadis' and 'hamis' characters are transliterated into Latin characters, they sometimes vary from the normal agreement although they appear in the same positions. For example: Adinew for '**ኣድነው**' and Admasu for '**ኣድማሱ**' for the character '**ድ**', Abiy for '**ኣብይ**' and Abdo for '**ኣብዱ**' for the character '**ብ**', Kelkilie for '**ከልኪሌ**' and Kifle for '**ክፍሌ**' for the character '**ሌ**' etc.

6.5 Bilingual Retrieval Performance Evaluation

Even though some Web-based CLIR systems such as Keizai(which accepts English queries and returns Japanese and Korean documents), TwentyOne(which supports six European languages), Arabvista(which accepts English or Arabic query to retrieve Web pages in multiple languages, including Chinese, French, and German), ECIRS(English–Chinese Web-based system), and MULINEX (which is more mature multilingual Web search and navigation tool for English, French and German) are available, for most of them, there is no systematic evaluations are available which leave their effectiveness uncertain [62]. The traditional CLIR systems are often tested on standard, readily available collections (mostly news articles). However, this is not applicable for Web-based CLIR which requires an extensive crawling (spidering) process to build multilingual collections.

Since it was impossible to read all Web page documents collected during crawling to judge the relevancy of the documents, we emphasized on precision only for the top 10 retrieved Web pages for each query as our primary performance measure. The measurement is referred to as target retrieval (34, 62). In addition, cross-language information retrieval always yields precision loss compared to traditional monolingual information retrieval. Therefore, it is not uncommon to

evaluate the performance of CLIR systems by comparing its results with the corresponding monolingual run. In a monolingual run, we manually translated the original Amharic query discussed in Section 6.2 into the target query (i.e. English query), and we have performed the retrieval on this translated query. After the precisions for both CLIR and monolingual retrieval are obtained, the precisions of the CLIR and of the monolingual retrieval have been compared.

Since our system is bidirectional, within a single query and its corresponding manual translation four different search results will be retrieved. These search results are:

- Amharic monolingual search results
- English monolingual search results
- Amharic-English cross-lingual search results
- English-Amharic cross-lingual search results

Then our system evaluation is performed by calculating the precision of each of the retrieved relevant documents. To evaluate the performance of the Amharic-English CLIR system, we compared the average precision of the Amharic-English cross-lingual search results with the average precision of the English monolingual search results. In addition, to evaluate the performance of the English-Amharic CLIR system, we compared the average precision of the English-Amharic cross-lingual search results with the average precision of the Amharic monolingual search results. Finally, we compared the performance of the Amharic-English and the English-Amharic retrieval engine.

6.5.1 Amharic-English Bilingual Retrieval Performance Evaluation

As we described in Section 6.2 and 6.3, we have used 15 Amharic queries to evaluate the effectiveness of our Amharic-English bilingual retrieval performance. Each query is directly submitted to the Amharic monolingual retrieval engine and at the same time to the Amharic-English cross-lingual retrieval engine. It means that within a single search click we do have both Amharic and English web documents in the same browser window separated by frames as shown in Figure 6.3 for the Amharic query "*የስማርኛ ቋንቋ*". On the other hand, these Amharic queries are manually translated into their corresponding English query and used for evaluation of our system by comparing the precision of the monolingual and cross-lingual results. As it was described in Chapter Two, precision is the number of relevant documents retrieved divided by

the total number of retrieved documents. However, relevance is subjective and different users may have different relevance measures. In our work, the relevancy of the retrieved document to the query is evaluated by two Information Science post graduate students who are regular users of the Web.

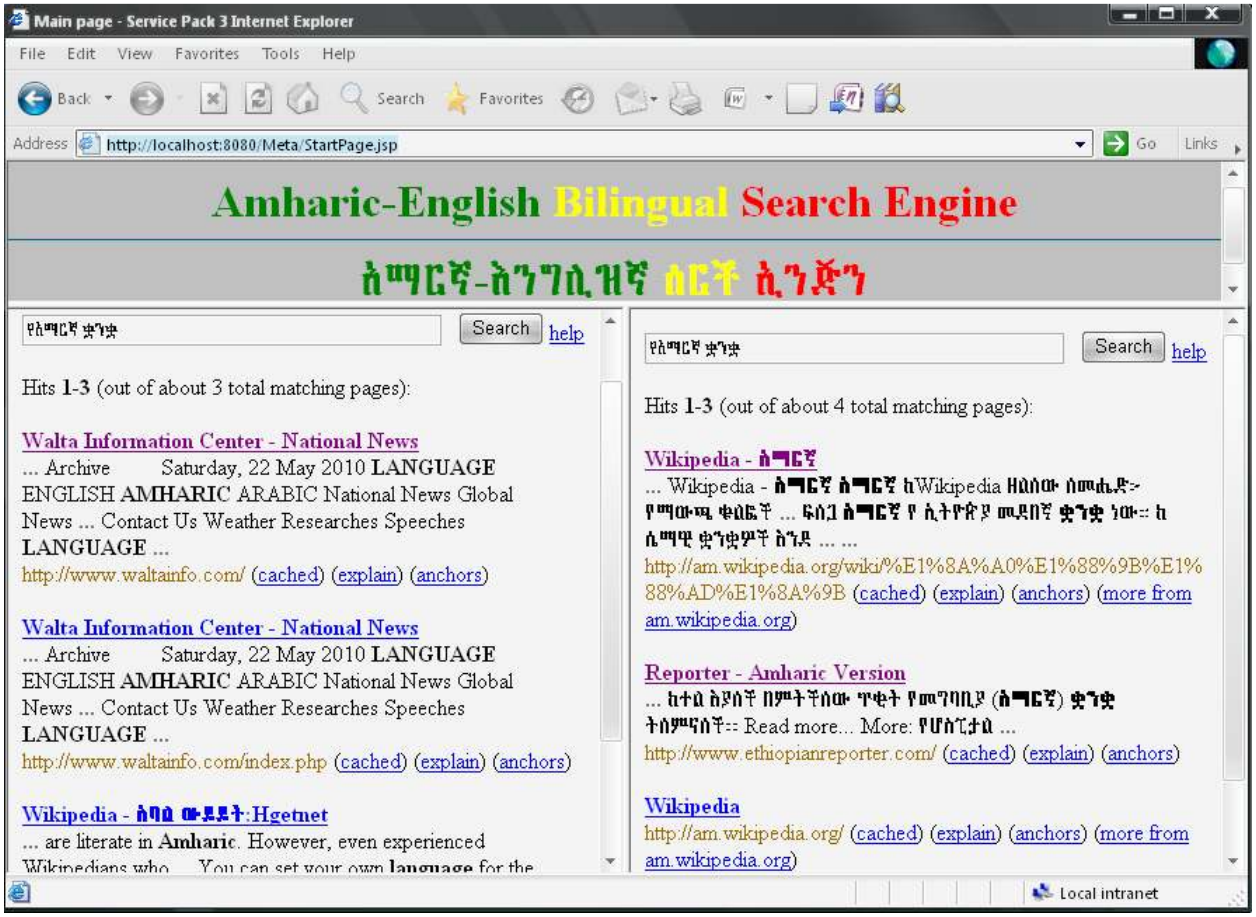


Figure 6.3: **Amharic-English search result.**

The presicions of the English monolingual and Amharic-English cross-lingual retreival for each Amharic query and the precision comparision of the two are shown in Table 6.4. The precisions for both monolingual and cross-lingual retrieval are calculated for only the top 10 retrieved English Web pages for each query.

Table 6.4: **Amharic-English precision evaluation result**

Query	Monolingual precision	Cross-lingual precision	% of monolingual
Query1	0.8	0.8	100%
Query2	0.7	0.5	71.42%
Query3	0.9	0.7	77.78%
Query4	0.8	0.3	37.5%
Query5	0.83	0.2	24.1%
Query6	0.9	0.8	88.89%
Query7	0.8	0.7	87.5%
Query8	0.83	0.8	96.39%
Query9	0.8	0.6	75%
Query10	0.8	0.4	50%
Query11	1	0.8	80%
Query12	0.88	0.38	43.18%
Query13	0.83	0.8	96.39%
Query14	1	1	100%
Query15	0.83	0.67	80.72%
Average	**0.85**	**0.63**	**74.12%**

As it can be seen from this Table, the monolingual precision of the English retrieval engine is 85% and that of the Amharic-English retrieval engine is 63%. From this, we can observe that the Amharic-English cross-lingual retrieval system out performed 74.12% of its corresponding monolingual retrieval engine.

6.5.2 English-Amharic Bilingual Retrieval Performance Evaluation

Like the Amharic queries, the English queries, which are direct manual translations of the Amharic queries, are also submitted to the English monolingual retrieval engine and at the same time to the English-Amharic cross-lingual retrieval engine. The English-Amharic search results are as shown in Figure 6.4 for the English query "*Election 2002*".

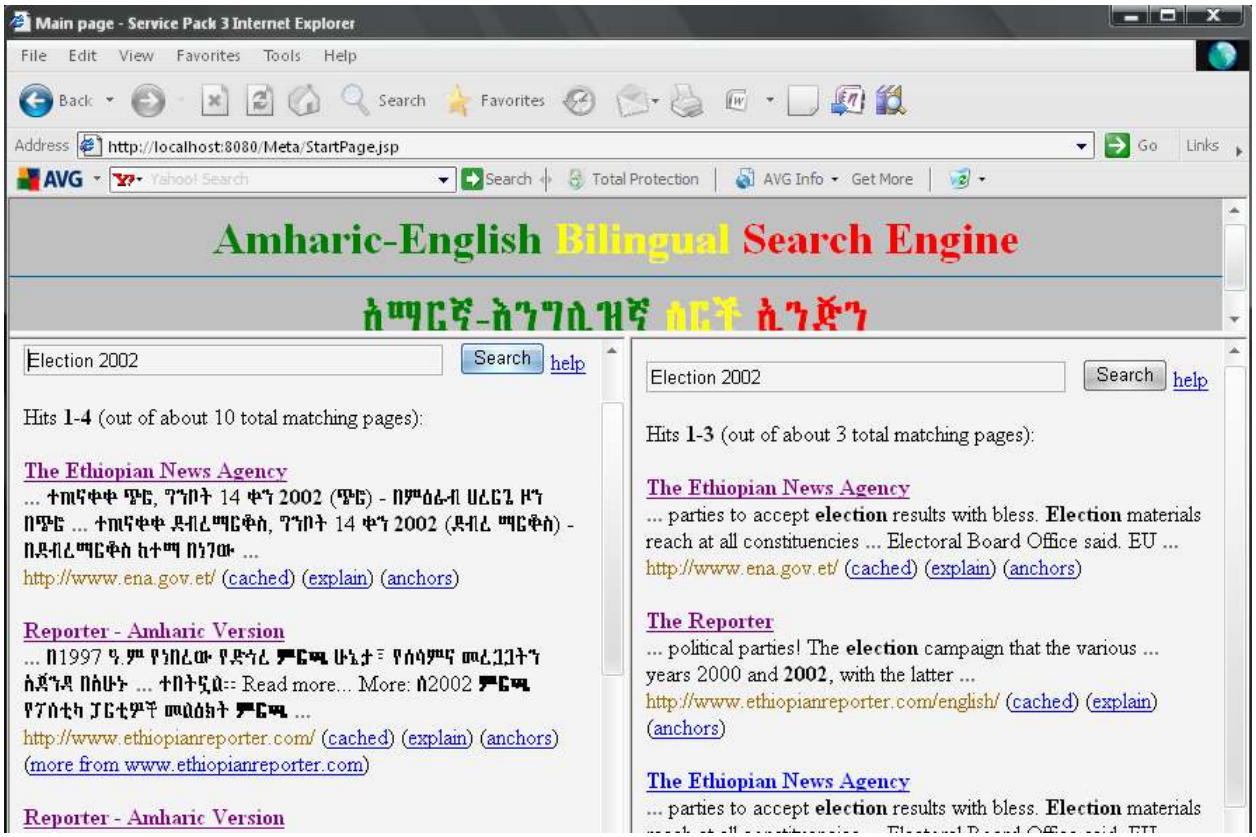


Figure 6.4: **English-Amharic search result.**

The presicions of the Amharic monolingual and English-Amharic cross-lingual retreival for each English query and the precision comparision of the two are shown in Table 6.5. The precisions for both monolingual and cross-lingual retrieval are calculated for only the top 10 retrieved Amharic Web pages for each query.

Table 6.5: **English-Amharic precision evaluation result**

English Query	Monolingual precision	Cross-lingual precision	% of monolingual
Query1	1	0.9	90%
Query2	0.75	0.6	80%
Query3	0.9	0.8	88.89%
Query4	0.8	0.5	62.5%
Query5	0.75	0.4	53.33%
Query6	1	0.7	70%
Query7	0.83	0.67	80.72%

Query8	0.75	0.6	80%
Query9	0.86	0.75	87.21%
Query10	0.89	0.71	79.78%
Query11	0.8	0.57	71.25%
Query12	0.71	0.5	70.42%
Query13	0.88	0.75	85.23%
Query14	1	1	100%
Query15	0.8	0.67	83.75%
Average	**0.85**	**0.67**	**78.82%**

As it can be surmised from Table 6.5, the Amharic monolingual retrieval engine has an average precision of 85% and its corresponding English-Amharic cross-lingual retrieval engine has an average precision of 67%. This experimental result also showed us the English-Amharic cross-lingual retrieval engine out performed 78.82% of its corresponding monolingual retrieval engine.

6.5.3 Summary on the Overall Experimental Results

The experimental results in Table 6.4 and 6.5 showed that the cross-lingual retrieval engines have lower precisions than their corresponding monolingual retrieval engines which are as expected and observed in every cross-lingual evaluation results. Furthermore, the experimental results showed that the English-Amharic cross-lingual retrieval engine out performed better than Amharic-English cross-lingual retrieval engine. This is because the English query translation component has better query translation performance than Amharic query translation component because of the reasons we discussed in Section 6.3.2 and this shows the dependency of the cross-lingual retrieval performance on the translation component.

6.5.4 The Significance of Our System Compared to General Purpose Search Engines

Our system is designed and implemented considering the needs of Web users to use Web documents in both languages and hence it handles the specific characteristics of Amharic language. As a result, to evaluate the significance of our system when compared to general purpose search engines, some Amharic queries which have morphological variations, character variations, and short words are given to our system and to 'Google' search engine to see whether our system handles these characteristics of the language and to see the result variations in Google

search results. In addition to this, the significance of the system because of being its bilingualism is also observed by looking into the relevant English Web documents retrieved by the system in addition to the Amharic documents for these Amharic queries. The relevancies of the documents are only evaluated for only the top 10 retrieved Web documents.

Table 6.6: Google versus our system results for queries that show different characteristics of Amharic language.

Language Characteristics	Query	Google Results		Our System Results			
				Amharic Documents		No. of Rel. English Documents	Total No. of Relevant Documents
		No. of Rel. Documents	Documents Similarity	No. of Rel. Documents	Documents Similarity		
Morphological Variations	የኢትዮጵያ ፓርቲዎች	10	0	7	7	5	12
	ኢትዮጵያ ፓርቲ	7		7		5	12
	የኢትዮጵያ ምርጫዎች	9	0	8	8	7	15
	ኢትዮጵያ ምርጫ	6		8		7	15
Character Variations	ስማርኛ ዜና	9	0	10	10	7	17
	ዐማርኛ ዜና	2		10		7	17
	መስስ ዜናዊ	10	0	6	6	4	10
	መስሥ ዜናዊ	3		6		4	10
Short words	የትምህርት ጥሩት	10	1	8	8	5	13
	የት/ት ጥሩት	6		8		5	13

As shown in Table 6.6, Google does not consider the characteristics of the Amharic language and retrieved different results for the same query which have different variations. However, our system results are the same for different variations of the words in the query. In addition to this, since our system is being bilingual, it also retrieved some relevant English Web documents for these queries. In general, the results in Table 6.6 showed us the significance of our system over general purpose search engines in both considering the characteristics of Amharic language and retrieving more number of relevant documents by simultaneously searching Amharic and English Web documents for the given Amharic query.

CHAPTER SEVEN

CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion

With the expansion of the World Wide Web, the amount of information in different languages and encoding schemes online on the Web is enormously growing. As a result, the number of online non-English speakers who use the Web as their major source of information and a means communication channel by realizing the importance of finding information in different languages on the Web is highly increasing. These Web users need to query search engines using their own native language query to search information that are relevant to their needs. However, the today's general purpose search engines do not handle the special characteristics of non-English languages. This creates new challenges in information search and retrieval since information search and retrieval needs language specific treatment.

Until recent years, the non-English speaking Web users use general purpose search engines to find information using English queries although they are not proficient enough to formulate the queries in the language. On the other hand, they may use their own language query although the search engines do not take into account the special characteristics of the specific language they use. In response to this, a number of language specific search engines which consider the characteristics of the users query language have been developed to allow the Web users to search information using their own native language. However, having a search engine that support only a specific language may lead to miss relevant Web documents that are not written in the same language and script of the query language particularly when the query language is not mostly used on the Web.

Although Web users speak different languages and there are huge number of non-English Web documents, most of the resources are written and published in English language on the Web to be accessed by search engines. Amharic, the second most spoken Semitic language in the world after Arabic, is one of the languages which have rapidly growing content on the Web. To search these Amharic Web documents, an Amharic Search engine has already been developed. However, the majority of Web documents are published in English when compared to Amharic

language and accessing only Web documents which are written in Amharic may not satisfy user requirements since there may be relevant documents for user queries in English as well. As a result, Web users need a multilingual search engine which can give them results in English language and in their own native language.

Nowadays, multilingual search engines which allow users to formulate their query in their own native language and retrieve documents in any other language have been developed. The search engine is said to be bilingual search engine for the case of two language pairs. In this type of Web information retrieval engines, since the query and the document are written in different languages, either the query or the document should be translated. So translation system is one of the basic components which should be well addressed in any bilingual search engine.

This research concerned itself with the bilingual search process in an attempt to enable users finding information they need in Amharic and English. It also concerned with translation of keywords from one language to the other which is an important issue for bilingual search. Our work aims to be different to and is probably the first Amharic-English bilingual search engine which displays the search results in two different languages on the same page. The first one is for users query language and the other one is for its translation. It uses language specific Amharic search engine and a general purpose search engine as its underlying search engine.

In this research, we attempted to design and develop a bilingual Web search engine for Amharic and English languages. The search engine is developed considering the features of Amharic and English languages and the translations between them in its design and implementation. The search engine has different components which address the basic language specific issues for the two languages in query translation and information retrieval. These major components are: Amharic query preprocessing, English query preprocessing, Amharic query translation, English query translation, Amharic search engine, and English search engine.

The Amharic and English query preprocessing modules perform tokenization, normalization, removing stop-words, and stemming of the Amharic and English queries respectively. The Amharic and English query translation modules are also responsible for lexical transfer or dictionary lookup in the bilingual Amharic-English dictionary and transliterating the out of dictionary words in the Amharic and English queries respectively. The Amharic-English

transliteration subcomponent is developed considering the heuristic Amharic-English characters mapping. The Amharic and English search engines are the underlying search engines for the bilingual Amharic-English search engine for crawling, indexing, and ranking of the Amharic and English Web documents respectively.

Nutch, an open source search engine, is chosen and customized to crawl, index, and rank both the Amharic and English Web documents in the underlying search engines. The Web crawling was conducted by providing selected seed URLs which are considered to host Amharic Web documents for Amharic search engine and English Web documents for English search engine.

Some of the components of our system are independently evaluated for their performance. The Amharic and English query translation components are evaluated by the way they translate the queries. The Amharic query translation module translates 88.57% of the words properly in the given Amharic queries where as the English query translation module properly translates 93.33% of the words in the given English queries. The transliteration component transliterates 80.64% proper names correctly which can be considered as promising.

To evaluate the effectiveness of our Amharic-English bilingual search engine, precision measures were conducted on the top 10 retrieved Web documents. The Web documents are collected by crawling the well know and frequently used websites which hosts mostly about Ethiopian events. The queries are carefully prepared from the bilingual websites and manually translated by a linguistic professional human expert. The relevancy of the Web documents to the queries is evaluated by experienced and regular Web users who are Information Scientists by profession.

The precision was calculated for each query and the average precision measure showed promising results for both the Amharic-English and English-Amharic cross-lingual evaluations. The average cross-lingual precision evaluation results are compared with the corresponding average monolingual precision evaluation results for each retrieval engine. The experimental results showed that the Amharic-English cross-lingual retrieval engine performed 74.12% of its corresponding English monolingual retrieval engine and the English-Amharic cross-lingual retrieval engine performed 78.82% of its corresponding Amharic monolingual retrieval engine. The results of both retrieval engines are found to be promising.

7.2 Contributions of the Work

With this work several novel ideas and designs are proposed. Among which we:

- Identified the major components of Amharic-English bilingual search engine which should be considered in developing Amharic-English bilingual search engine.
- Identified the critical issues that need to be dealt in cross-lingual search engines and proposed appropriate techniques.
- Proposed a general model for Amharic-English bilingual search engine.
- Paved the way for further large scale research projects on Web based bilingual search engines by identifying language dependent components.
- Developed algorithms for query translation and transliteration. These algorithms can also be adopted for other cross-lingual information retrieval.
- Designed and created a bilingual Amharic-English dictionary for query translation purpose and a repository reference table for transliteration of words in queries.
- Designed and developed an Amharic-English bilingual search engine based on the proposed model.
- Evaluated the prototype developed for its effectiveness.

7.3 Recommendations

In this thesis work, we have presented the efforts exerted to design and implement a dictionary based bilingual Amharic-English Web search engine. However, developing a full-fledged Amharic-English bilingual Web search engine requires the involvement of several professionals from different disciplines such as computer science, information science, linguistics and other related fields. As a result, additional features, improvements, and modifications should be incorporated to come up with an effective and efficient bilingual search engine. Hence, we proposed the following recommendations for future research directions:

- ✓ Incorporating query expansion techniques and automatic relevance feedback to expand the original source query to emphasize the context of the source query which improves search results further.

- ✓ Incorporating phrasal translation and co-occurrence analysis techniques for translation disambiguation in combination with the dictionary-based approach.
- ✓ Improving the performance of the transliteration component by handling exceptions, transliteration variations and irregularities, sadises and hamises.
- ✓ Another possible extension to our work is expanding the bilingual search engine to include other more languages such as Tigrigna, Afaan Oromo, etc.
- ✓ The current Amharic stemmer that we have used should be enhanced since it has a negative impact on the query preprocessing for query translation which in turn has an impact on the search results.
- ✓ Automatic Named Entity Recognizer(NER) for Amharic should be developed and integrated to identify whether the query term is need to be checked from the bilingual dictionary or should be transliterated.
- ✓ As the main problems of dictionary based approach is limitation of word coverage, including large size commercial bilingual dictionary and online bilingual dictionary for query translation will minimize the problem.
- ✓ Amharic thesaurus and WordNet which used to expand the query using synonyms must be investigated and integrated.
- ✓ A comprehensive set of Amharic stop-words, short words, aliases should be identified and included.
- ✓ Building domain specific machine translation systems and implementing the bilingual search engine for specific applications. The bilingual search engine can be easily customized and implemented to satisfy the needs of some organizations for specific Web based application domains such as: medicine, business, and other domain specific Web portals.
- ✓ After efficient and critical masses of resources such as corpus, lexicon, morphological analyzers, stemmers, and named entity recognizers have been developed and made publicly available for Amharic language, the Amharic-English bilingual Web search engine will reach a level higher than that of academic prototype systems.

References

[1]Atelach Alemu Argaw. "Amharic-English Information Retrieval with Pseudo Relevance Feedback". In: Peters, C., et al. (eds.) Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers, pp. 119-126. Springer, Berlin / Heidelberg. 2008

[2]Tessema Mindaye Mengistu. “Design and Implementation of Amharic Search Engine”. A Thesis Submitted to the School of Graduate Studies of the Addis Ababa University in Partial Fulfilment for the Degree of Master of Science in Computer Science, 2007

[3]Atelach Alemu Argaw and Lars Asker. "Amharic-English Information Retrieval". Working Notes of CLEF 2006, Alicante, Spain. September 2006.

[4]Atelach Alemu Argaw, Lars Asker, Rickard Cöster and Jussi Karlgren. "Dictionary-based Amharic - English Information Retrieval". In Proceedings of Cross Language Evaluation Forum (CLEF 2004), Bath, UK. September 2004.

[5]J. Deepa Devi, Ranjani Parthasarathi and T.V. Geetha. “Tamil Search Engine”. Sixth Tamil Internet 2003 Conference, Chennai, Tamilnadu, August 2003.

[6]Prasad Pingali, Jagadeesh J and Vasudeva Varma. “WebKhoj: Indian language IR from multiple character encodings”. In Proceedings of the 15th International Conference on World Wide Web Edinburgh, Scotland, May 2006.

[7]Marcello Federico and Nicola Bertoldi. “Statistical Cross-Language Information Retrieval using N-Best Query Translations”. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, Tampere, Finland. ACM New York, NY, USA, 2002.

[8]Yan Qu and Gregory Grefenstette. “Finding Ideographic Representations of Japanese Names Written in Latin Script via Language Identification and Corpus Validation”. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Barcelona, Spain. Association for Computational Linguistics, Morristown, NJ, USA, 2004.

[9]Kristen Parton, Kathleen R. McKeown, James Allan, and Enrique Henestroza. "Simultaneous Multilingual Search for Translingual Information Retrieval". In Proceeding of the 17th ACM conference on Information and knowledge management, Napa Valley, California, USA. ACM New York, NY, USA, 2008.

[10]Yiming Yang, Jaime G. Carbonell, Ralf D. Brown, and Robert E. Frederking. "Translingual information retrieval: learning from bilingual corpora". Elsevier Science Publishers Ltd. Essex, UK, 1998.

[11]Karunesh Arora, Ankur Garg, Gour Mohan, Somiram Singla, and Chander Mohan. "Cross Lingual Information Retrieval Efficiency Improvement through Transliteration". In Proceedings of ASCNT – 2009, CDAC, Noida, India, pp. 65 – 71, 2009.

[12]Paola Virga and Sanjeev Khudanpur. "Transliteration of Proper Names in Cross-Lingual Information Retrieval". In Proceedings of the ACL 2003 workshop on Multilingual and mixed-language named entity recognition - Volume 15, Sapporo, Japan. Association for Computational Linguistics, Morristown, NJ, USA, 2003.

[13]Judit Bar-Ilan and Tatyana Gutman: "How do search engines handle non-English queries? - A case study". WWW (Alternate Paper Tracks), Budapest, Hungary, 2003.

[14]Kulwadee Somboonviwat, Takayuki Tamura, and Masaru Kitsuregawa, "Simulation Study of Language Specific Web Crawling". In Proceedings of the 21st International Conference on Data Engineering Workshops. IEEE Computer Society Washington, DC, USA, 2005.

[15]Amharic-WIKIPIDIA, the free encyclopaedia.

Available at http://en.wikipedia.org/wiki/Amharic, Accessed on 21 September 2009.

[16]Amharic Ethiopia Language. Available at http://www.free-press-release.com/news/200907/1248234344.html, Accessed on July 21, 2009.

[17]Hassen Redwan Hussen. "Enhanced Design of Amharic Search Engine (An Amharic Search Engine with Alias and Multi-character Set Support)". A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment for the Degree of Master of Science in Computer Science, 2008.

[18]Wen-hui Zhang, Hua-lin Qian, Wei Mao and Guo-nian Sun. “A Multilingual (Chinese, English) Indexing, Retrieval, Searching Search Engine”. Available at

http://www.isoc.org/inet99/proceedings/posters/210/index.htm, Accessed on August 10, 2009.

[19]David Appleyard, “Colloquial Amharic: The Complete Course for Beginners”, simultaneously published in USA and Canada by Routledge, 2003.

[20]Joanne Capstick, Abdel Kader Diagne, Gregor Erbach and Hans Uszkoreit. “MULINEX: Multilingual Web Search and Navigation”, Accessed on August 25, 2009, Available at http://eprints.kfupm.edu.sa/52030/1/52030.pdf, Published on 08.02.99.

[21]Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. “An Introduction to Information Retrieval”. Cambridge University Press, England Online edition (c), 2009.

[22]Yaser Al-Onaizan and Kevin Knight. “Translating Named Entities Using Monolingual and Bilingual Resources.” In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002.

[23]Nasreen AbdulJaleel and Leah S. Larkey. “English to Arabic Transliteration for Information Retrieval: A Statistical Approach.” In Proceedings of the twelfth international conference on information and knowledge management , New Orleans, LA, 139-146, 2003.

[24]Amit Kirschenbaum and ShulyWintner. “Lightly Supervised Transliteration for Machine Translation.” Association for Computational Linguistics Morristown, NJ, USA, 2009.

[25]Baeza-Yates, Ricardo and Ribeiro-Neto, Bertheir. “Modern Information Retrieval”. New York: ACM Press, 1999

[26]Brian Pinkerton. “WebCrawler: Finding What People Want”: A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy University of Washington, Department of Computer Science and Engineering, 2000.

[27]Monica Peshave and Kamyar Dezhgosha, "How Search Engines Work and a Web Crawler Application". Department of Computer Science, University of Illinois, Springfield USA, 2005.

[28]Sergey Brin and Lawrence Page. “The anatomy of a large-scale hypertextual Web search engine”. Computer Networks and ISDN Systems, April 1998.

[29]Amit Singhal. “Modern Information Retrieval: A Brief Overview”. In IEEE Data Engineering Bulletin 24(4), pages 35-43, 2001.

[30] Kiduk Yang. “Information Retrieval on the Web”. Annual Review of Information Science and Technology, Volume 39 Issue 1, Pages 33 – 80, American Society for Information Science and Technology, 2008.

[31]Andrew Graves, Mounia Lalmas. “Video retrieval using an MPEG-7 based inference network”. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. ACM New York, NY, USA, 2002.

[32]Chunyu Kit, Haihua Pan and Jonathan J. Webster. “Example-Based Machine Translation: A New Paradigm”. Chinese University of HK Press, 2007.

[33]ADAM LOPEZ. “Statistical Machine Translation”. ACM Computing Surveys, Vol. 40, No. 3, Article 8, August 2008.

[34]Jialun Qin, Yilu Zhou, Michael Chau and Hsinchun Chen. Multilingual Web retrieval: An experiment in English–Chinese business intelligence. John Wiley & Sons, Inc. New York, NY, USA, 2006

[35]Mohammed Aljlayl, Ophir Frieder, and David Grossman. On Arabic-English Cross-Language Information Retrieval: A Machine Translation Approach. IEEE Computer Society Washington, DC, USA, 2002

[36]P. L. Nikesh, Sumam Mary Idicula, S. David Peter. English-Malayalam Cross-Lingual Information Retrieval – an Experience, In Proceedings of IEEE International Conference on Electro/Information Technology, Ames, Iowa State University, May 2008

[37]H. Moukdad and H. Cui (2005). How Do Search Engines Handle Chinese Queries? Webology, 2 (3), Article 17.

Available at: http://www.Webology.ir/2005/v2n3/a17.html

[38]H. Moukdad. Lost In Cyberspace: How Do Search Engines Handle Arabic Queries? The 12th International World Wide Web Conference, Budapest, Hungary, May 2003.

[39] Interlingual Machine Translation.

Available at: http://en.wikipedia.org/wiki/Interlingual_machine_translation. Accessed on December 20, 2009.

[40]Machine Translation.

Available at: http://en.wikipedia.org/wiki/Machine_translation. Accessed on: December 20, 2009.

[41]Y. Al-Onaizan, R. Florian, M. Franz, H. Hassan, Y. S. Lee, S. McCarley, K. Papineni, S. Roukos, J. Sorensen, C. Tillmann, T. Ward, and F. Xia. TIPS: A Translingual Information Processing System. In Proceedings of HLT-NAACL 2003, Demonstrations , pp. 1-2, Edmonton, May-June 2003.

[42]OSMaTran: Open-Source Machine Translation.

Available at: http://www.torsimany.ua.es/OSMaTran/ . Accessed on January 15, 2010.

[43]Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, and Mikel L. Forcada. "Opentrad Apertium open-source machine translation system: an opportunity for business and research." In Proceedings of Translating and the Computer 28 Conference, London, November 16--17, 2006.

[44]Mikel L. Forcada. Open-source machine translation: an opportunity for minor languages. In Proceedings of Strategies for developing machine translation for minority languages (5th SALTMIL workshop on Minority Languages). 2006

[45]Dorr, Bonnie, Eduard Hovy and Lori Levin, "Machine Translation: Interlingual Methods", Encyclopedia of Language and Linguistics 2nd edition ms. 939, Brown, Keith (ed.), 2004.

[46]Alexander Franz, Keiko Horiguchi, Lei Duan, Doris Ecker, Eugene Koontz, and Kazami Uchida. An integrated architecture for example-based machine translation. Association for Computational Linguistics Morristown, NJ, USA. 2000.

[47]Jimmy O'Regan. Apertium: Open source machine translation.

Available at: http://linuxgazette.net/152/oregan.html. Accessed on January 15, 2010.

[48]Atelach Alemu, Lars Asker, and Mesfin Getachew. “Natural language processing for Amharic: Overview and suggestions for a way forward”. In Proceedings of the 10th Conference on Traitement Automatique des Langues Naturelles. Batzsur-Mer, France, June 2003.

[49]Saba Amsalu & Sisay Fissiha Adafre. “Machine Translation for Amharic: Where we are”. In Proceedings of LREC-2006: Fifth International Conference on Language Resources and Evaluation. 5th SALTMIL Workshop on Minority Languages: “Strategies for developing machine translation for minority languages”, Genoa, Italy. 2006.

[50]Wessel Kraaij, Jian-Yun Nie, and Michel Simard. “Embedding Web-based statistical translation models in cross-language information retrieval.” MIT Press Cambridge, MA, USA, 2003.

[51]A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke, and S. Raghavan. “Searching the Web”. ACM Transactions on Internet Technology (TOIT), 2001.

[52]Nega Alemayehu and Willet P. “Stemming of Amharic Words for Information Retrieval”. In Literary and Linguistic Computing. Oxford, Oxford University press, Vol. 17, No.1, pp 1-17, 2002.

[53]Atelach Alemu Argaw and Lars Asker. “An Amharic Stemmer: Reducing Words to their Citation Forms”. In Proceedings of the 5th Workshop on Important Unresolved Matters, pages 104–110. Association for Computational Linguistics, 2007.

[54]Seid Muhie Yimam. “TETEYEQ (ተጠየቅ): Amharic Question Answering For FactoID Questions”. A Thesis Submitted to the School of Graduate Studies of the Addis Ababa University in Partial Fulfillment for the Degree of Master of Science in Computer Science, 2009.

[55]Melese Tamiru. “Automatic Amharic Text Summarization Using Latent Semantic Analysis”. A Thesis Submitted to the School of Graduate Studies of the Addis Ababa University in Partial Fulfillment for the Degree of Master of Science in Computer Science, 2009.

[56]Lu Chengye, Xu Yue, and Geva Shlomo. “Web-Based Query Translation for English-Chinese CLIR”. Computational Linguistics and Chinese Language Processing (CLCLP), Vol. 13, No. 1, pp. 61-90, 2008.

[57]Amsalu Aklilu and G.P Mosback. “English Amharic Dictionary”. Oxford University press 1973 and reprinted by Makusa publishing PLC, 1996.

[58]Munir Abrar. “Nur Amharic-English-Arabic Dictionary”. Mega Publisher, Addis Ababa, Ethiopia, 2009.

[59]Serge Obolensky, Debebow Zelelie, and Mulugeta Andualem. “Amharic Basic Course”. Foreign Service Institute, Washingiton D.C, 1995.

[60]Mubarek Sied Mahammed. “Amharic-English Bilingual Information Retrieval Experiment: A parallel Corpus Based Approach for Query Translation.” A Thesis Submitted to the Graduate School of Telecommunications and Information Technology (GSTIT) in Partial Fulfillment of the requirements for the Degree of Master of Science in Information Technology, 2009.

[61]Anne R. Diekema and Wen-Yuan Hsiao. “Cross-Language Information Retrieval using Dutch Query Translation”. Springer Berlin / Heidelberg, January 01, 2001.

[62]Yilu Zhou, Jialun Qin, Hsinchun Chen, and Jay F. Nunamaker. “Multilingual Web Retrieval: An Experiment on a Multilingual Business Intelligence Portal”. In the Proceedings of the 38th Annual Hawaii International Conference on System Sciences. IEEE Computer Society Washington, DC, USA, 2005.

Appendix I: Lists of Amharic Stop Words

ሁሉ
ሁሉም
ኗላ
ሁኔታ
ሆነ
ሆኑ
ሆኖም
ሁል
ሁሉንም
ሳይ
ሴላ
ሴሎች
ልዩ
መሆኑ
ማለት
ማለቱ
መሳከል
የሚገኙ
የሚገኝ
ማድረግ
ማን
ማንም
ሰሞኑን
ሲሆን
ሲል
ሲሉ
ስለ
ቢቢሲ
ቢሆን
ብለዋል
ብቻ
ብዛት
ብዙ
ቦታ
በርካታ
በሰሞኑ
በታች
በኋላ
በኩል
በውስጥ
በጣም
ብቻ
በተለይ
በተመለከተ
በተመሳሳይ
የተለያየ
የተለያዩ
ተባለ
ተገለጸ
ተገልጿል
ተጨማሪ
ተከናውኗል
ችግር
ታች
ትናንት
ነበረች
ነበረ
ነበረ
ነው
ነይ
ነገር
ነገሮች
ናት
ናቸው
አሁን
አለ
አሉ
አስታወቀ
አስታውቀዋል
አስታውሰዋል
አስካሁን
አሳሰበ
አሳስበዋል
አስፈላጊ
አስገንዝቡ
አስገንዝበዋል
አብራርተዋል
አበራርተው
አስረድተዋል
አስከ
አሳክህ
አሳክሽ
አሳክሞ
እንድ
እንዳር
እስኪደርስ
እንኳ
እስከ
እዚሁ
እና
እንደ
እንደገለጹት
እንደተገለጸው
እንደተናገረት
እንደአስረዱት
እንደገና
ወቅት
እንደሁም
እንጂ
እዚህ
እዚያ
እያንዳንዱ
እያንዳንዳችው
እያንዳንጷ
ከ
ከኋላ
ከሳይ
ከመሳከል
ከሰሞኑ
ከታች
ከውስጥ
ከጋራ
ከፊት
ወዘተ
ወይም
ወደ
ዋና
ወደፊት
ውስጥ
ውጪ
ያስ
ያሱ
ደገባል
የኋላ
የሰሞኑ
የታች
የውስጥ
የጋራ
ያ
ደታወሳል
ደህ
ደግሞ
ድረስ
ጋራ
ግን
ገልጿል
ገልጸዋል
ግዜ
ጥቂት
ፊት
ደግሞ
ዛፌ
ጋር
ተናግረዋል
የገለጹት
ደገልጻል
ሲሉ
ብለዋል
ስለሆነ
ኦች
ሆኖም
መግለጹን
አመልክተዋል
ደናገራሉ

Appendix II: Lists of English Stop Words

a
about
above
across
after
again
against
all
are
almost
alone
along
already
also
although
always
among
an
and
another
any
anybody
anyone
anything
anywhere
are
area
areas
around
as
ask
asked
asking
asks
at
away
b
back
backed
backing
backs
be
became
because
become
becomes
been
before
began
behind
being
beings
best
better
between
big
both
but
by
c
came
can
cannot
case
cases
certain
certainly
clear
clearly
come
could
d
did
differ
different
differently
do
does
done
down
down
downed
downing
downs
during
e
each
early
either
end
ended
ending
ends
enough
even
evenly
ever
every
everybody
everyone
everything
everywhere
f
face
faces
fact
facts
far
felt
few
find
finds
first
for
four
from
full
fully
further
furthered
furthering
furthers
g
gave
general
generally
get
gets
give
given
gives
go
going
good
goods
got
great
greater
greatest
group
grouped
grouping
groups
h
had
has
have
having
he
her
here
herself
high
high
high
higher

highest
him
himself
his
how
however
i
if
important
in
interest
interested
interesting
interests
into
is
it
its
itself
j
just
k
keep
keeps
kind
knew
know
known
knows
l
large
largely
last
later
latest
least
less
let
lets
like
likely
long
longer
longest
m
made
make
making
man
many
may
me
member
members
men
might
more
most
mostly
mr
mrs
much
must
my
myself
n
necessary
need
needed
needing
needs
never
new
new
newer
newest
next
no
nobody
non
noone
not
nothing
now
nowhere
number
numbers
o
of
off
often
old
older
oldest
on
once
one
only
open
opened
opening
opens
or
order
ordered
ordering
orders
other
others
our
out
over
p
part
parted
parting
parts
per
perhaps
place
places
point
pointed
pointing
points
possible
present
presented
presenting
presents
problem
problems
put
puts
q
quite
r
rather
really
right
right
room
rooms
s
said
same
saw
say
says
second
seconds
see
seem
seemed
seeming
seems
sees
several
shall
she

should
show
showed
showing
shows
side
sides
since
small
smaller
smallest
so
some
somebody
someone
something
somewhere
state
states
still
still
such
sure
t
take
taken
than
that
the
their
them
then
there
therefore
these
they
thing
things
think
thinks
this
those
though
thought
thoughts
three
through
thus
to
today
together
too
took
toward
turn
turned
turning
turns
two
u
under
until
up
upon
us
use
used
uses
v
very
w
want
wanted
wanting
wants
was
way
ways
we
well
wells
went
were
what
when
where
whether
which
while
who
whole
whose
why
will
with
within
without
work
worked
working
works
would
x
y
year
years
yet
you
young
younger
youngest
your
yours
z

Appendix III: Lists of Amharic Short Words

ት/ቤት
ት/ት
ት/ክፍል
ሃ/አለቃ
ሃ/ስላሴ
ደ/ዘይት
ደ/ታቦር
መ/ር
መ/ቤት
መ/አለቃ
ክ/ከተማ
ክ/ሀገር
ወ/ር
ወ/ሮ
ወ/ሪት
ወ/ስላሴ
ፍ/ስላሴ
ፍ/ቤት
ጽ/ቤት
ሲ/ር
ፕ/ር
ጠ/ሚንስትር
ዶ/ር
ገ/ገዮርጊስ
ቤ/ክርስትያን
ም/ስራ
ም/ቤት
ተ/ሃይማኖት
ሚ/ር
ኮ/ል
ሜ/ጀነራል
ብ/ጀነራል
ሌ/ኮለኔል
ሊ/መንበር
አ/አ
ር/መምህር
ፕ/ት
ዓ.ም
ዓ.ዓ
ዶ.ር

Appendix IV: Amharic-English Character Mapping

No.	1st order	2nd order	3rd order	4th order	5th order	6th order	7th order	8th order
	ግዕዝ	ካዕብ	ሳልስ	ራብዕ	ሃምስ	ሳድስ	ሳብዕ	ዲፕቃላ
1.	ሀ = ha	ሁ = hu	ሂ = hi	ሃ = ha	ሄ = he/hie	ህ = h/hi	ሆ = ho	
2.	ለ = le	ሉ = lu	ሊ = li	ላ = la	ሌ = le/lie	ል= l/li	ሎ = lo	ሏ = lua
3.	መ = me	ሙ = mu	ሚ = mi	ማ = ma	ሜ = me/mie	ም = m/mi	ሞ = mo	ሟ = mua
4.	ረ = re	ሩ = ru	ሪ = ri	ራ = ra	ሬ = re/rie	ር = r/ri	ሮ = ro	ሯ = rua
5.	ሰ = se	ሱ = su	ሲ = si	ሳ = sa	ሴ = se/sie	ስ = s/si	ሶ = so	ሷ = sua
6.	ሸ = she	ሹ = shu	ሺ = shi	ሻ = sha	ሼ = she/shie	ሽ = sh/shi	ሾ = sho	ሿ = shua
7.	ቀ = ke	ቁ = ku	ቂ = ki	ቃ = ka	ቄ = ke/kie	ቅ = k/ki	ቆ = ko	ቋ = kua
8.	በ = be	ቡ = bu	ቢ = bi	ባ = ba	ቤ = be/bie	ብ = b/bi	ቦ = bo	ቧ = bua
9.	ቨ = ve	ቩ = vu	ቪ = vi	ቫ = va	ቬ = ve/vie	ቭ = v/vi	ቮ = vo	ቯ = vua
10.	ተ = te	ቱ = tu	ቲ = ti	ታ = ta	ቴ = te/tie	ት = t/ti	ቶ = to	ቷ = tu
11.	ቸ = che	ቹ = chu	ቺ = chi	ቻ = cha	ቼ = che/chie	ች = ch/chi	ቾ = cho	ቿ = chua
12.	ነ = ne	ኑ = nu	ኒ = ni	ና = na	ኔ = ne/nie	ን = n/ni	ኖ = no	ኗ = nua
13.	ኘ = gne	ኙ = gnu	ኚ = gni	ኛ = gna	ኜ = gne/gnie	ኝ = gn/gni	ኞ = gno	ኟ = gnua
14.	**አ = E**	**ኡ = u**	**ኢ = i**	**ኣ = a**	**ኤ = A**	**እ = e**	**ኦ = o**	**ኧ = e**
15.	ከ = ke	ኩ = ku	ኪ = ki	ካ = ka	ኬ = ke/kie	ክ = k/ki	ኮ = ko	ኳ = kua
16	ወ = we	ዉ = wu	ዊ = wi	ዋ = wa	ዌ = we/wie	ው = we	ዎ = wo	
17.	ዘ = ze	ዙ = zu	ዚ = zi	ዛ = za	ዜ = ze/zie	ዝ = z/zi	ዞ = zo	ዟ = zua
18.	ዠ = xe	ዡ = xu	ዢ =xi	ዣ = xa	ዤ = xe/zie	ዥ = x/xi	ዦ = xo	ዧ = xua
19	የ = ye	ዩ = yu	ዪ = yi	ያ = ya	ዬ = ye/yie	ይ = y/yi	ዮ = yo	
20.	ደ = de	ዱ = du	ዲ = di	ዳ = da	ዴ = de/die	ድ = d/di	ዶ = do	ዷ = dua
21.	ጀ = je	ጁ = ju	ጂ = ji	ጃ = ja	ጄ = je/jie	ጅ = j/ji	ጆ = jo	ጇ = jua
22.	ገ = ge	ጉ = gu	ጊ = gi	ጋ = ga	ጌ = ge/gie	ግ = g/gi	ጎ = go	ጓ = gua
23.	ጠ = te	ጡ = tu	ጢ = ti	ጣ = ta	ጤ = te/tie	ጥ = t/ti	ጦ = to	ጧ = tua
25.	ጨ = che	ጩ =chu	ጪ = chi	ጫ = cha	ጬ = che/chie	ጭ = ch/chi	ጮ = cho	ጯ = chua
26.	ጰ = pe	ጱ = pu	ጲ = pi	ጳ = pa	ጴ = pe/pie	ጵ = p/pi	ጶ = po	ጷ = pua
27	ጸ = tse	ጹ = tsu	ጺ = tsi	ጻ = tsa	ጼ = tse/tsie	ጽ = ts/tsi	ጾ = tso	ጿ = tsua
28	ፈ = fe	ፉ = fu	ፊ = fi	ፋ = fa	ፌ = fe/fie	ፍ = f/fi	ፎ = fo	ፏ = fua
29	ፐ = pe	ፑ = pu	ፒ = pi	ፓ = pa	ፔ = pe/pie	ፕ = p/pi	ፖ = po	ፗ = pua

Appendix V: Basic Configurations of the Amharic-English Bilingual Search Engine

Property Name	Value	Description
Http.agent.name	AEBL Spider	HTTP 'User-Agent' request header. A single word uniquely related to our work.
http.robots.agents	AEBL Spider,*	The agent strings we'll look for in robots.txt files, comma-separated, in decreasing order of precedence. We put the value of http.agent.name as the first agent name, and keep the default * at the end of the list.
http.timeout	200,000	The default network timeout, in milliseconds.
http.max.delays	100	The number of times a thread will delay when trying to fetch a page. Each time it finds that a host is busy, it will wait fetcher.server.delay. After http.max.delays attempts, it will give up on the page for now.
http.content.limit	65536	The length limit for downloaded content, in bytes. If this value is nonnegative (>=0), content longer than it will be truncated; otherwise, no truncation at all.
http.proxy.host	proxy.aau.edu.et	The proxy hostname. If empty, no proxy is used.
http.proxy.port	8080	The proxy port.

Appendix VI: Sample Crawling Process Test Result

```
crawl started in: AmharicCrawl
rootUrlDir = urls
threads = 10
depth = 10
Injector: starting
Injector: crawlDb: AmharicCrawl/crawldb
Injector: urlDir: urls
Injector: Converting injected urls to crawl db entries.
Injector: Merging injected urls into crawl db.
Injector: done
Generator: Selecting best-scoring urls due for fetch.
Generator: starting
Generator: segment: AmharicCrawl/segments/20100914160014
Generator: filtering: false
Generator: topN: 2147483647
Generator: jobtracker is 'local', generating exactly one partition.
Generator: Partitioning selected urls by host, for politeness.
Generator: done.
Fetcher: starting
Fetcher: segment: AmharicCrawl/segments/20100914160014
Fetcher: threads: 10
fetching http://am.wikipedia.org/
fetching http://www.addisadmass.com/
fetching http://www.ena.gov.et/
fetching http://archives.ethiozena.com/
fetching http://www.dw-world.de/amharic/
fetching http://www.waltainfo.com/
fetching http://www.zethiopia.com/
fetching http://www.ethpress.gov.et/
fetching http://www.ethiopianreporter.com/amharic/
fetch of http://www.addisadmass.com/ failed with: Http code=500, url=http://www.addisadmass.com/
Fetcher: done
CrawlDb update: starting
CrawlDb update: db: AmharicCrawl/crawldb
CrawlDb update: segments: [AmharicCrawl/segments/20100914160014]
CrawlDb update: additions allowed: true
CrawlDb update: URL normalizing: true
CrawlDb update: URL filtering: true
CrawlDb update: Merging segment data into db.
CrawlDb update: done
Generator: Selecting best-scoring urls due for fetch.
Generator: starting
Generator: segment: AmharicCrawl/segments/20100914160038
Generator: filtering: false
Generator: topN: 2147483647
Generator: jobtracker is 'local', generating exactly one partition.
```

Generator: Partitioning selected urls by host, for politeness.
Generator: done.
Fetcher: starting
Fetcher: segment: AmharicCrawl/segments/20100914160038
Fetcher: threads: 10
fetching http://www.ena.gov.et/EnglishNews/2010/Sep/13Sep10/121256.htm
fetching http://www.ena.gov.et/EnglishNews/2010/Sep/13Sep10/121225.htm
fetching http://www.ethpress.gov.et/../forum/forums.php
fetching http://www.ethpress.gov.et/../../ethpress/main/viewsOpinion.php
fetching http://www.ena.gov.et/EnglishNews/2010/Sep/13Sep10/121233.htm
fetching http://www.ethpress.gov.et/ethpress/main/economy.php
fetching http://www.ethpress.gov.et/../../ethpress/main/politics.php
fetching http://www.ena.gov.et/EnglishNews/2010/Sep/14Sep10/121278.htm
fetching http://www.ena.gov.et/#Menu1_SkipLink
fetching http://www.ethiopianreporter.com/amharic/#
fetching http://www.addisadmass.com/
fetching http://www.ena.gov.et/EnglishNews/2010/Sep/13Sep10/121274.htm
fetching http://www.ethiopianreporter.com/amharic/
fetch of http://www.addisadmass.com/ failed with: Http code=500, url=http://www.addisadmass.com/
fetching http://www.ethiopianreporter.com/amharic/).removeClass(
fetching http://www.ethpress.gov.et/ethpress/main/viewsOpinion.php
fetching http://www.ethpress.gov.et/ethpress/main/politics.php
fetching http://www.ethpress.gov.et/ethpress/main/contactUs.php
fetching http://www.ethpress.gov.et/ethpress/subscribe/subscribe.php
fetching http://www.dw-world.de/amharic/#
fetching http://www.dw-world.de/amharic/#mainContent
fetching http://www.ethpress.gov.et/../../ethpress/pictureGallery/pictureGallery.php
fetching http://www.ethiopianreporter.com/amharic/).addClass(
fetching http://www.ena.gov.et/Videos/Meles_InterviewII_081310.wmv
fetching http://www.ena.gov.et/EnglishNews/2010/Sep/13Sep10/121222.htm
fetching http://www.ethiopianreporter.com/amharic/application/x-shockwave-flash
fetching http://www.ethpress.gov.et/../subscribe/subscribe.php
fetching http://www.ethpress.gov.et/aboutus.php
fetching http://www.dw-world.de/amharic/#headSearch
fetching http://www.ethpress.gov.et/ethpress/main/survey.php
fetching http://www.ethpress.gov.et/../../ethpress/main/msgMinInfo.php
fetching http://www.ethpress.gov.et/../../ethpress/main/editorial.php
fetching http://www.ethpress.gov.et/ethpress/main/news.php
fetching http://www.ena.gov.et/Videos/Meles_Interview_EngII_081310.wmv
fetching http://www.ena.gov.et/Videos/Meles_Interview_Eng_081310.wmv
fetching http://www.ethpress.gov.et/ethpress/main/home.php
fetching http://www.ena.gov.et/EnglishNews/2010/Sep/13Sep10/121265.htm
fetching http://www.dw-world.de/amharic/8.0.0
fetching http://www.ethpress.gov.et/ethpress/security/login.php
fetching http://www.waltainfo.com/
fetching http://www.ena.gov.et/Videos/Meles_Interview_081310.wmv
fetching http://www.ena.gov.et/EnglishNews/2010/Sep/13Sep10/121273.htm
fetching http://www.dw-world.de/amharic/#headLanguage

fetching http://www.ethpress.gov.et/ethpress/main/aboutUs.php
fetching http://www.ethpress.gov.et/ethpress/forum/forums.php
fetching http://www.dw-world.de/amharic/#multi
fetching http://www.ena.gov.et/EnglishNews/2010/Sep/13Sep10/121275.htm
fetching http://www.ethpress.gov.et/ethpress/security/register.php
fetching http://www.ena.gov.et/Default.aspx
fetching http://www.ethpress.gov.et/../../ethpress/main/economy.php
fetching http://www.ethpress.gov.et/ethpress/advertise/reqAdvertise.php
fetching http://www.ethpress.gov.et/../../ethpress/main/news.php
fetching http://www.ena.gov.et/EnglishNews/2010/Sep/13Sep10/121277.htm
fetching http://www.ena.gov.et/EnglishNews/2010/Sep/13Sep10/121246.htm
fetching http://www.zethiopia.com/text/javascript
fetching http://www.ethpress.gov.et/contactus.php
fetching http://www.ena.gov.et/AmharicNews/2010/Sep/13Sep10/121270.htm
fetching http://www.ena.gov.et/EnglishNews/2010/Sep/13Sep10/121262.htm
fetching http://www.dw-world.de/amharic/#nav
fetching http://www.dw-world.de/amharic/#headLinks
fetching http://www.ethpress.gov.et/ethpress/main/editorial.php
Fetcher: done
CrawlDb update: starting
CrawlDb update: db: AmharicCrawl/crawldb
CrawlDb update: segments: [AmharicCrawl/segments/20100907044047]
CrawlDb update: additions allowed: true
CrawlDb update: URL normalizing: true
CrawlDb update: URL filtering: true
CrawlDb update: Merging segment data into db.
CrawlDb update: done
LinkDb: starting
LinkDb: linkdb: AmharicCrawl/linkdb
LinkDb: URL normalize: true
LinkDb: URL filter: true
LinkDb: adding segment: file:/d:/nutch-1.0/AmharicCrawl/segments/20100906232355
LinkDb: adding segment: file:/d:/nutch-1.0/AmharicCrawl/segments/20100906232451
LinkDb: adding segment: file:/d:/nutch-1.0/AmharicCrawl/segments/20100906232858
LinkDb: adding segment: file:/d:/nutch-1.0/AmharicCrawl/segments/20100906233350
LinkDb: adding segment: file:/d:/nutch-1.0/AmharicCrawl/segments/20100907003613
LinkDb: adding segment: file:/d:/nutch-1.0/AmharicCrawl/segments/20100907024804
LinkDb: adding segment: file:/d:/nutch-1.0/AmharicCrawl/segments/20100907031321
LinkDb: adding segment: file:/d:/nutch-1.0/AmharicCrawl/segments/20100907034852
LinkDb: adding segment: file:/d:/nutch-1.0/AmharicCrawl/segments/20100907044047
LinkDb: done
Indexer: starting
Indexer: done
Dedup: starting
Dedup: adding indexes in: AmharicCrawl/indexes
Dedup: done
Merging indexes to: AmharicCrawl/index
Adding AmharicCrawl/indexes/part-00000

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged.

Declared by:

Name: _____

Signature: _____

Date: _____

Confirmed by advisor:

Name: _____

Signature: _____

Date: _____

Place and Date of Submission: Addis Ababa, November, 2010