
**ADDIS ABABA UNIVERSITY**
**SCHOOL OF INFORMATION SCIENCE**

# PROBABILISTIC TIGRIGNA-AMHARIC CROSS LANGUAGE INFORMATION RETRIEVAL (CLIR)

**TSEGAYE SEMERE**

**June 2013**
**Addis Ababa, Ethiopia**

**ADDIS ABABA UNIVERSITY**
**SCHOOL OF INFORMATION SCIENCE**

# PROBABILISTIC TIGRIGNA-AMHARIC CROSS LANGUAGE INFORMATION RETRIEVAL (CLIR)

**BY:**
**TSEGAYE SEMERE**

**A thesis submitted to Addis Ababa University in partial fulfillment of the requirement for the Degree of Master of Science in Information Science**

**June 2013**
**Addis Ababa, Ethiopia**

**ADDIS ABABA UNIVERSITY**
**SCHOOL OF INFORMATION SCIENCE**

# PROBABILISTIC TIGRIGNA-AMHARIC CROSS LANGUAGE INFORMATION RETRIEVAL (CLIR)

**BY:**
**TSEGAYE SEMERE**

**Name and signature of members of the examining board**

| Name | Title | Signature | Date |
|---|---|---|---|
| _____ | Chairperson | _____ | _____ |
| Dereje Teferi (PhD) | Adviser | _____ | _____ |
| Million Meshesha (PhD) | Examiner | _____ | _____ |

# DECLARATION

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

_____

TSEGAYE SEMERE

The thesis has been submitted for examination with my approval as university advisors

_____

**DEREJE TEFERI (PhD)**

**June 2013**

## ACKNOWLEDGMENTS

First and foremost extraordinary thanks go for my Almighty God, for giving me the ability to be where I am now.

Secondly, I would like to express my gratitude and heartfelt thanks to my advisor, Dr. Dereje Teferi for his keen insight, guidance, and unreserved advising. I am really grateful for his constructive comments and critical readings of the study.

Thirdly, I would like to express my sincerest gratitude and heartfelt thanks to my instructor Dr. Million Meshesha for his support. I would like to articulate my appreciation for his academic commitment.

Fourthly, My special thanks goes to my friends and classmates Tesfay Hadish and Berihu Yohannes, your support and advice has been worth considering starting from the beginning up to the completion of the program.

Finally, I would like to thank my wife Semhal Gidey and my son Michael Tsegaye for the constant encouragement and caring during the time of my study.

# Table of Content

# List of Tables

# List of Figures

# List of Abbreviations

| | |
|---|---|
| ACL | Association for Computational Linguistics |
| BLIR | Bilingual Information Retrieval |
| BIM | Binary Independent Model |
| BNM | Bayesian Network Model |
| CLEF | Cross Language Evaluation Forum |
| CLIA | Cross Lingual Information Access |
| CLIR | Cross Language Information Retrieval |
| EBM | Extended Boolean Model |
| EBMT | Example-based Machine Translation |
| FDRE | Federal Democratic Republic of Ethiopia |
| IDF | Inverse Document Frequency |
| INM | Inference Network Model |
| IR | Information Retrieval |
| LM | Language Model |
| MAP | Mean Average Precision |
| MLIR | Multilingual Information Retrieval |
| MRBD | Machine Readable Bilingual Dictionary |
| MT | Machine Translation |
| OOV | Out of Vocabulary |
| PRP | Probabilistic Ranking Principle |
| SMT | Statistical Machine Translation |
| TF | Term Frequency |
| TREC | Text Retrieval Conference |

# Abstract

Amharic is the official working language of the Federal Democratic Republic of Ethiopia. On the other hand, Tigrigna is the working language of Tigray, which is one of the regional states of Ethiopia. Furthermore Tigrigna is official working language of the neighboring country Eritrea.

As considerable amount of information is being produced in Amharic language rapidly and continuously; experimenting on the applicability of a cross language information retrieval system for Tigrigna–Amharic is important. Because academicians, offices, researchers and other individuals can be benefited. This study is, therefore, an attempt to develop Tigrigna-Amharic Cross lingual Information Retrieval (CLIR) system which enables Tigrigna native speakers to access and retrieve the online information sources that are available in Tigrigna and Amharic by writing queries using their own (native) language.

Dictionary-based approach is employed to conduct the query term translation from Tigrigna to Amharic. To this end, Tigrigna-Amharic machine-readable dictionary is used.

Because of the capability of handling the uncertain nature of Information Retrieval, the probabilistic information retrieval model is employed. Both indexing and searching module are constructed. In these modules, different text operations such as: tokenization, normalization, stemming and stop word removal for both Tigrigna and Amharic languages are included.

The performance of the system after User Relevance Feedback is measured using recall, precision, and F-measure. The system registered an average recall of 84% and 93%, an average precision of 75% and 64%, and average F-measure of 79% and 73% for Tigrigna and Amharic languages respectively. Though the performance of the system is greatly affected by the stemmer, the result obtained is encouraging,

Finally the recommendation can be drawn that the performance of the CLIR system can be improved by designing good stemmer for both languages.

# CHAPTER ONE

# INTRODUCTION

## 1.1. Background

There is an increasing amount of full text material in various languages available through the Internet and other information services. To this end cross-language information retrieval (CLIR) has become an important new research area. It refers to an information retrieval task where the language of queries is other than that of the retrieved documents [1].

In the past, this was not really a problem [2]. However, with the rapid growth of the Internet online information resources have become available in almost all major languages [4]. This condition is increasing users desire to explore documents that were written in either their native language or some other languages that they can understand [3]. Thus, became the main motivation for Cross-language Information Retrieval (CLIR) system.

Cross-Language Information Retrieval (CLIR) is where the user request and the document collection against which the request is to be matched are in two different human languages. The aim of CLIR is to match the request against the collection as if the request had been issued in the document collection language to begin with [2]. CLIR system, thus, facilitates retrieval of relevant documents written in one natural language with automated systems that can accept queries expressed in other language.

CLIR generates relevant documents to the user queries though the language of query and document is distinct. The language that the query used is referred to as the source language (e.g. Tigrigna) and the language of the documents is the target language (e.g. Amharic). Even though target (document) translation into the source (query) language is comfortable for the user, it is expensive and hard to implement as it obeys complex rules of the natural language than query translation [10]. Thus,

the query translation approach is more common in CLIR and applied in present research as well.

Cross-Language Information Retrieval has been studied widely in different languages, such as English, Chinese, Spanish, and Arabic [4]. Much research works have been reported and evaluation results have, in general, been satisfactory. In the past, research in Cross Lingual Information Access (CLIA) has been strongly practiced through, the Cross-Language Evaluation Forum (CLEF), Question-answering Workshop and cross-language named entity extraction challenges by the Association for Computational Linguistics (ACL) and the Geographic Information retrieval (GeoCLEF) track of CLEF [33].

These days, in Ethiopia, considerable amount of information is being produced, especially in Amharic language within organizations and outside organizations rapidly and continuously. This information explosion is challenging for archival and searching from the existing huge amount of information [5].

CLIR can use different approaches of query translation. The main approaches are dictionary based translation, machine translation, and corpus based methods.

Dictionary based approach, a machine–readable bilingual dictionary, is used to replace the source language query words with their target language counterparts [2]. Machine translation (MT) aims to provide human-readable translations of natural language texts. This makes it arguably a much harder task than query translation, since queries can be translated word-by-word and the translations need not be seen by the user. In corpus based approaches the translation knowledge is derived from parallel or comparable corpora.

Predicting relevant documents is one of the core issues in IR system, and IR models are responsible for determining the prediction of what is relevant and what is not [24]. A number of retrieval models have been proposed since the mid 1960s. They have evolved from specific models intended for use with small structured document

to recent models that have strong theoretical basis and which are intended to accommodate variety of full text document types such as: Boolean model, vector space model, probabilistic model, etc.

Current models handle documents with complex internal structure and most of them incorporate a learning or relevance feedback component that can improve performance when presented with sample relevant documents [29].

Probabilistic model is a statistical analysis model that estimates the probability of a document relevance given available evidences. It works based on the probability ranking principle, which states that "An information retrieval system is supposed to rank the documents based on their probability of relevance to the query, given all the evidence available" [25].

The probabilistic model incorporates relevance feedback and query terms re weighting mechanisms.

## 1.2. Statement of the Problem and Justification

There are more than 80 languages in Ethiopia and Amharic is the mother tongue language for more than 25 million people. According to federal democratic republic of Ethiopia (FDRE) population census commission it is the first language for more than 20 million and second language for over 5 million people [5]. Moreover Amharic is the official working language of FDRE and some regional states of FDRE. Accordingly there is huge collection of documents available in the form of books, magazines, newspapers, novels, officials and legal documents, etc.
On the other hand, Tigrigna is the native language for the Tigray and Eritrean people. It is one of the widely spoken languages in Ethiopia and Eritrea. It is also spoken in some parts of the world especially in the Middle East, Europe and North America by Ethiopian and Eritrean immigrants living in these areas [8].

Languages are crucial tools for the easy accessibility of the huge collection of

documents on the Internet. Usually people have one language they prefer to use. In general, while people may have working knowledge of more than one language, it is not common for people to have equal competence in all of them.

Although many Tigrigna speaker Internet users can read and understand Amharic documents, they feel uncomfortable formulating queries in Amharic. This is either because of their limited vocabulary in Amharic, or because of the possible miss usage of Amharic words. Therefore these wealth of the Amharic documents available on the WWW are inaccessible for most of the Tigrigna speakers.

Different attempts have been made to develop CLIR systems for Amharic-English [6][7] and Afan Oromo-English[4][37] languages. But no CLIR system is found for Tigrigna language. Therefore to make use of the considerable amount of online Amharic documents, the barrier found between these two languages needs to be resolved.

An automatic query translation tool would be very helpful to such users and query translation tool would also allow them to retrieve relevant documents in all the languages of interest with only one query. To achieve all these, CLIR would be a useful tool. Hence, translation of local queries into Amharic is critical for making these useful online documents accessible for the local use. Thus, the importance of CLIR is crucial [4].

Query translation can be done by using different approaches, either by using machine readable bilingual dictionary or by using bilingual dictionary constructed from parallel corpora. According to Donnla [2], the least reliant and on expensive query translation system is using the machine readable bilingual dictionary. In addition bilingual printed dictionaries already exist for a large number of language pairs. Porting such a dictionary to a machine-readable format which can be utilized for CLIR requires considerably less human effort than any of CLIR approaches.

IR system can be developed by using different models. According to Amanuel [5],

unless other modules are integrated to the models, IR systems developed for Amharic language so far using vector space model have not registered good performance because of the incapability to define users need. In comparison probabilistic model has registered better performance because it enables to defining users need through relevance feedback and query reformulation techniques.

Though IR system for Tigrigna language is not yet developed, As Probabilistic IR for Amharic language is already developed by Amanuel [5], it is extended further to design this CLIR system.

Tigrigna- Amharic CLIR is not yet developed. So the aim of this study is there fore to develop Dictionary based Tigrigna-Amharic CLIR using probabilistic model so as to ease information sharing between the users of the two languages.

This research tries to answer the following research questions

- To what extent is it possible to develop a CLIR system for Tigrigna and Amharic?
- What is the performance of dictionary based approach for Tigrigna-Amharic CLIR system?
- What is the effectiveness of Tigrigna queries translation in to Amharic by using machine readable bilingual dictionary

## 1.3. Objective of the study

### 1.3.1. General Objective

The overall objective of this research is to design Tigrigna-Amharic CLIR system using probabilistic model based on dictionary based approach to translate Tigrigna queries into Amharic in order to retrieve both Tigrigna and Amharic documents.

### 1.3.2. Specific Objectives

To achieve the general objective stated above, the following specific objectives are accomplished throughout the study: These specific objectives are:

- to review related works on Tigrigna-Amharic CLIR,
- to prepare and organize test documents and queries,
- to develop machine readable Amharic-Tigrigna bilingual dictionary,
- to develop a CLIR prototype that uses Tigrigna queries and retrieve both Amharic and Tigrigna documents from the test collection,
- to examine the effectiveness of the prototype using the queries and documents prepared for the experimentation and
- to discuss and report experimental results found and recommend further research works in the area.

## 1.4. Significance of the Study

CLIR systems provide users to retrieve documents written in one language by using a query written in another language [2]. Obviously, translation is needed in the CLIR process; either translating the query into the document languages (query translation), or translating the documents into the query language (document translation). CLIR systems can help people who are able to read in a non-native language but are not proficient enough to write a query in that language. It can also aid people who are not able to read in a non-native language but have access to translations resources. This situation increases the significance of CLIR systems which can make relevant document(s) from enormous collection accessible to the users.

The major contribution of the study is to design bilingual word based CLIR system to benefit those who are Tigrigna speakers and capable of using IR systems to obtain their information need from the web using their native queries. Users enter their query in their native language and the system retrieves relevant documents in their native language and other language, Amharic. The retrieved documents that are relevant for the given query can benefit those who can understand contents of

Amharic documents that are returned for the given query. The work of this research can also be a starting point for phrase-based query translation and also for Multilingual Information Retrieval (MLIR) system (that involves more than one CLIR) using Tigrigna query to retrieve documents written in two or more languages. Therefore, the beneficiaries of this research include: individuals, schools, offices, and other researchers.

## 1.5. Scope and Limitation of the Study

The scope of this study is restricted to the translation of Tigrigna queries into Amharic using dictionary-based approach in order to retrieve both Tigrigna and Amharic documents. The retrieval process is performed using Binary Independent Model (BIM). In addition test documents are taken only from available Tigrigna and Amharic websites. Therefore they are limited only to five domains. These are Education, Health, Religion, Politics, and Social. The translation of Tigrigna queries into Amharic is performed using bilingual dictionary. After query translation, the major information retrieval processes (indexing and searching) are carried out.

Though the system can retrieve both Tigrigna and Amharic relevant documents, it is limited to use only Tigrigna queries. In addition limited size of corpus and limited number of test queries are used because of non existence of the required corpus.

## 1.6. Methodology of the study

### 1.6.1. Literature Review

To accomplish the objectives of this research literatures form books, journals, articles, conference and Internet are reviewed. Materials concerning Tigrigna language and Amharic-English CLIR are also reviewed. Related works with CLIR have also been reviewed to understand the state of the art in the field.

### 1.6.2. Research design

In this study experimental research design was selected. Experimental research, which is also called empirical research or cause and effect method is a data-based research coming up with conclusions which are capable of being verified with observation or experiments. Experimental research is appropriate when proof is sought that certain variables affect other variables in some way [41],

### 1.6.3. Data Collection

Amharic-Tigrigna Machine readable bilingual dictionary that contains direct translation of a given word in the other language was constructed by digitizing Saba English-Amharic-Tigrigna printed dictionary. Because of non-existence of parallel Tigrigna and Amharic documents only 200 Tigrigna and Amharic documents that are publicly available in both languages such as, Bible chapters, health related documents from the official web site of Ministry of Health, Education related documents from the official web site of Ministry of Education, religion related documents from the official web site of Awake magazine, social and politics related documents from the official web sites of Walta information center and Hadas Eritrea are collected and prepared for implementing and evaluating the Tigrigna-Amharic CLIR system.

### 1.6.4. Implementation

Python programming language is used for developing the system. Python is selected because it is clear and readable. Experts and beginners can easily understand the code and everyone can become productive in python very quickly.

Python is an open source, interpreted, object-oriented, high level programming language. It enabled the researcher to implement the functionality of the sought system without any hassle and allowed writing programs that are clear and readable. It has extensive built-in help functions, which make it possible to learn new things and minimize programming errors.

### 1.6.5. Testing Process

The experimentation for evaluating the effectiveness of the system was done by using selected test documents and queries. Tigrigna queries were presented to the CLIR system to retrieve both Tigrigna and Amharic documents judged to be relevant by the system for the given query. To measure the effectiveness of the prototype system Recall, Precision and F-Measure are used,

## 1.7. Organization of the Thesis

This thesis is organized in five chapters. The first chapter presents the general overview of the thesis that comprises the Background, Statement of the problem, General and Specific Objectives, Scope and Limitation, Significance, and Methodology,

Chapter two reviews different articles regarding to Tigrigna language, CLIR together with its different approaches and IR Models. Various related researches done locally and internationally on CLIR systems are also described briefly in this chapter,

The third chapter deals with detail description of the dictionary-based Tigrigna-Amharic CLIR. In this chapter data collection, data preprocessing and the proposed architecture of the system are presented. The major components involved in the architecture of the CLIR system are also explained in detail.

In chapter four the experimentation and evaluation of the proposed system are discussed. Findings of the results obtained from the experiment and challenges that affected the system's performance are also presented in this chapter.

Finally, chapter five winds up what has been done in the research. It reviews the results obtained and forwards recommendations for future work.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1. Overview of Tigrigna and Amharic Languages

Tigrigna and Amharic are members of the Ethio-Semitic languages, which belong to Afro-Asiatic super family [8] [5].

Tigrigna is spoken primarily in Eritrea and Ethiopia. There are more than 6 million Tigrinya speakers worldwide. According to the 2007 population and housing census of Ethiopia, there are over 4.3 million Tigrigna speakers in Tigray [34] and there are 2.4 million Tigrigna speakers in Eritrea as noted by Yonas [8] (citing Lewis, 2009).

On the other hand, Amharic was the national language of Ethiopia until 1995. Following the declaration of constitution of Ethiopian federal democratic government, it becomes the working/official language [40].

Both Tigrigna and Amharic are written in the Ge'ez script which these days is called Ethiopic and originally developed for Geez language. In Tigrigna and Amharic each symbol represents a consonant and vowel combination and the symbols are organized in groups of similar symbols on the basis of both the consonant and the vowel. For each consonant in each symbol, there is an unmarked symbol representing that consonant followed by a canonical or inherent vowel [8] [5].

Both Tigrigna and Amharic, like other Semitic languages such as Arabic, exhibit a root-pattern morphological phenomenon. In addition, they use different affixes to create inflectional and derivational word forms.

### 2.1.1. Origins of Tigrigna and Amharic languages

The Ethiopic writing system is used to represent the different Semitic languages such as, Ge'ez, Amharic, Tigrigna, Harerigna, Argpba, siltigna and so on [8][5][40]. These languages are confined to Ethiopia and Eritrea. Ge'ez is no more mother

tongue of any person, but it still has a very significant role in the traditional language of literature and religion. Amharic and Tigrigna are closely related to each other [40].

Ethiopic writing system is written from left to right. It does not make any distinction between upper and lower case letters and has no conventional cursive form. There are no systematic variations in the form of the symbol according to its position in the word. The Ethiopic system for Tigrigna language consists of alphabets, and punctuation marks [8].

### 2.1.2. Alphabets

Alphabets are sets of letters arranged in fixed orders of the language they used to write. They are also called phonemes which contain consonants and vowels. There are different alphabets representations in the world. The most alphabets representation is Latin or Roman alphabets which have been adapted by numerous languages. The Ethiopic writing systems have also their own writing systems. Tigrigna and Amharic have their own alphabets (ፊደል) and it is used for writing different documents of Tigrigna and Amharic languages. They have thirty-five base symbols with seven orders which represent seven vowels for each base symbol [8].

### 2.1.3. Punctuation marks

Identifying punctuation marks is vital to know word demarcation for natural language processing. Most of the Tigrigna language punctuation marks are listed in table 2.1 below. The word separator mark (:) is used in the old literature to separate one word from other words. In the current literature, it is rarely used. As, a result a single space is used to separate words instead of this punctuation marks. The end of sentence mark (::) is used to shows when an idea is finished. The sentence connector mark (፡) is used to connect two sentences in to one sentence. The list separator mark (፥) is used to list things, separate parts of a sentence, and indicate a pause in a sentence or question. Like the other punctuation marks, the beginning of the list mark (፦) is used at the beginning of the lists. In addition to these

punctuation marks, the Tigrigna language is also borrowed some punctuation marks from English language such as ?, !, and ”.

| Punctuation marks | Meaning |
|---|---|
| ፡ | word separator |
| ። | End of sentence |
| ፤ | sentence connector |
| ፥ | list separator marks |
| ፦ | beginning of the list mark |
| ? | End of question |
| ! | End of an emphatic declaration, or command. |
| “ | quote some words or sentences taken from other |

*Table 2:1 list of Tigrigna punctuation marks*

## 2.2. Overview of Cross Language Information Retrieval (CLIR)

The importance of Cross language Information Retrieval (CLIR) nowadays is patent in multiple contexts. In fact, communication is more global, and the access to multilingual information is more and more widespread within this globalized society. However, unless some lingua franca is established in specific geographic areas and discourse communities, it is still necessary to facilitate access in the native speaker’s language.[10]

CLIR can be seen as IR with a language barrier placed between the query and the collection. It is the study of retrieving information in one language (e.g Amharic) by queries expressed in another language (e.g. Tigrigna). Only the language barrier requires specific techniques in CLIR, which are mainly focused on the translation process. The different approaches differ essentially with respect to which available information is translated (queries or documents), and in the approaches used to carry out the translation [10].

The basic idea behind the CLIR system is to retrieve on-line text in a language different from a query language. The availability of resources on the Web other than English has raised the issue of CLIR, how users can retrieve documents in different languages [10].

There are three main ways in which cross-language information retrieval approaches attempt to "cross the language barrier" – through query translation, or document translation, or both. The best results are obtained by translating the collections into the language of the queries. However, this approach is computationally expensive and most of the works have focused on query translation methods [11]. This research also uses the query translation approach of CLIR.

The CLIR tasks are either bilingual or multilingual. Bilingual IR (BLIR) is concerned with one target language only while Multilingual IR (MLIR) is concerned with several target languages [12]. BLIR is useful for a user who might not be able to express his or her information need, even if he or she is capable of reading documents. MLIR benefits if a user is performing his or her retrieval in a multiple in language collection (e.g. the Web), and would like to retrieve documents in multiple languages by expressing a query in a single language. The multilingual task based on query translation is more complicated than bilingual one, because there are several target languages. CLIR is different from the monolingual or classical IR in that, it is expected to generate relevant documents to the user queries though documents and user queries are in different language. However, in case of CLIR system the query and the document may not be in the same language to retrieve relevant document. CLIR benefits users who only know one language [10], by providing mechanism of query translation to the language of the document. It differs from the monolingual IR in its consideration of crossing the language barrier that exists between the queries and documents [10]. So CLIR helps users retrieve relevant documents by expressing query using their native language.

According to Daniel [4], the following three steps are the basic processes performed by most CLIR systems.

- Query translation: The natural language query input must be translated to the target language of the documents to be searched. A translation or approximate translation of the target language of the document can be performed from the language of the query input.

- Monolingual document retrieval: For each target language, the query generated from query translation is used to retrieve relevant document written in the language of the target document. Cheng (2004) pointed out that simple string matching cannot satisfy the goal of an IR system. Usually, the documents and the user's query are converted into some internal representations (that are used during the matching process) during the indexing process. Accordingly, indexing a document shall be done at the word and phrase levels. Indexing at the word level includes all words that appeared in the document, including their morphological features (such as verb tenses, number, etc). Phrase level indexing is important to perform phrasal indexing for they may convey more content than single words.

- Result merging: To produce the unique result, it is needed to merge the results produced for the each monolingual document retrieval.

## 2.3. IR and CLIR

In recent years, there has been a rapid increase in the amount of stored and accessible electronic information. The goal of IR is, therefore, to find relevant documents from a large collection of documents or from the WWW for the users' query. Users typically formulate a query, frequently in free text, to describe their information need. The IR system then compares the query with each document in order to evaluate its similarity to the query. CLIR is, however, considerably more complex than traditional IR because some method for query or document translation must be developed before one can use traditional IR [10].

The search engines currently available on the Web are IR systems that have been created to answer users' information need. However, most of the existing search engines provide only monolingual IR; i.e.; they retrieve documents only in the same language as the query. Search engines usually do not consider the language of the keywords when the keywords of a query are matched against those of the documents [13].

Identical keywords can be matched, even if their languages are different. For example, the Tigrigna word "ሓበሬታ" can be matched with the Amharic word "መረጃ". This is possible by using cross lingual information retrieval.

## 2.4. Possible Approaches to CLIR

CLIR does not differ too much from IR and it only focuses on the techniques of translation process, to map a written word from one language-script pair to another language-script pair, to solve the language barrier [10]. For example, Tigrigna word "ሕዮ" corresponds to Amharic word "ጥያቄ". Therefore, in CLIR to retrieve relevant documents the language of the queries and the documents should not be necessarily the same.

Approaches to CLIR can be categorized according to how they solve the problem of matching the query and documents across different languages. In CLIR, either the query or the document needs to be mapped into the common representation to retrieve the relevant documents. Translating all documents into the query language performs significantly better than query translation as results of some experiments identify [10] but translating all documents into the query language requires huge storage space and it is computationally expensive [2]. Due to this, query is usually translated into the language of the target collection of documents [2]. Thus, this research is focused on the accuracy of dictionary based query translation.

As noted by Gearailt [2], the basic approaches in translation of queries into a language of target documents can involve the following approaches: Machine

Translation (MT), Machine-Readable Dictionary (MRD) and Corpus Based approaches.

### 2.4.1.Machine Translation Approach

Machine translation (MT) is the translation of text from one human language into another human language by the use of a computer. In order to accomplish this it needs texts in one specific language as input and generates texts with a corresponding meaning in another language as output. Thus, machine translation is a decision problem where we have to decide on the best of target language text matching a source language text [14].

In CLIR MT can be implemented in two different ways [1]. The first way is to use MT system to translate foreign language documents in the corpora into the language of the user's query. This approach is not applicable for large document collections, or for collections in which the documents are in multiple languages.

Document translation approach can be accomplished on an 'as-and-when-needed' basis at query time, referred as on-the-fly translation, or in advance of any query processing [10]. The second method of using MT in CLIR is where the users query language is translated into the language of the documents in the stored collection. With both methods of MT, an ambiguity problem can exist since the translated query does not necessarily represent the sense of the original query. For instance, translating the Tigrigna query "ወርሒ" to Amharic language could produce an inappropriate translation since it is not clear whether to mean "ፈለቃ" or "ወር". Due to this, MT is more efficient in document translation when the context is unambiguous.

### 2.4.2. Dictionary-based query translation Approach

Dictionary based query translation is a CLIR approach in which the query words are translated to the target language using MRD [14]. Machine Readable Dictionary (MRD) is an electronic version of printed dictionary, and may be general dictionaries or specific domain dictionaries or a combination of them. It has been adopted in

CLIR because bilingual dictionaries are widely available.

Dictionary-based approaches are straightforward to implement, and the efficiency of word translation with a dictionary is high [14]. However, due to the vocabulary limitation of dictionaries, translations of some words in a query may not be found in a dictionary. Due to this, the problem referred to as Out of Vocabulary (OOV) is created. The OOV terms are proper names or recently created words which are not included in the dictionary. As input queries are usually short, query expansion does not provide enough information to help recover the missing words. In many cases OOV terms are the crucial words in the query. As a result if a user enters a newly created term to find information about that term, he or she will be unable to find any relevant documents for the word since it is a newly created term and not included in the dictionary.

### 2.4.3. Corpus-Based Approach

In corpus-based CLIR approaches, the translation knowledge is derived from parallel or comparable corpora. Parallel corpora are preferred because they provide more accurate translation knowledge. However, because of the scarcity of parallel corpora, comparable corpora are often used in CLIR.

The aligned texts of a comparable corpus are not translations of each other, but related topically. For example, consider collections of articles appeared on Tigrigna and Amharic newspapers from the same time period. A lot of the topics and events covered in the Tigrigna newspaper would also be covered in the Amharic newspaper. A comparable corpus could be created by finding, for each article in the Finnish collection, a document in the Swedish collection that discussed the same event or topic. In this case, the Finnish collection is the source collection of the comparable corpus, and the Finnish articles form the set of source documents. By contrast, the Swedish collection is the target collection that consists of target documents. [15].

Parallel corpora contain collection of pairs of documents in more than one language which are direct translations of each others [17]. An aligned parallel corpus is

interpreted to show exactly which sentence of the source language corresponds with exactly which sentence of the target text. Parallel corpora are not always readily available and those that are available tend to be relatively small or to cover only a small number of subjects [17]. On the other hand, performance of CLIR systems using corpus based approach is highly influenced by quality, i.e., reliability and correctness, and size of the corpus [17].

Parallel text alignment procedures attempt to identify translation equivalences within collections of translated documents. It plays a critical role in multi-lingual natural language processing research. In particular, SMT systems require collections of sentence pairs as the basic elements for building statistical word and phrase alignment models [1]. Alignment is the process of establishing the correspondence between matching element in parallel corpora [1].

Corpus-based approach has four shortcomings according to [2]. First, it is difficult to get the parallel or comparable corpora. Second, the current available corpora tend to be relatively small. Third, the domain-dependent problem is involved between the query and the statistics of the corpora. Finally, the performance is dependent on how well the corpora are aligned.

## 2.5. IR Models

A model of information retrieval predicts and explains what a user will find relevant given the user query. The correctness of the model's predictions can be tested in a controlled experiment [19].

IR model is the mechanism of predicting and explain the need of the user given the query to retrieve relevance documents from the collection. IR models serves as blueprint so as to develop applicable IR system. In addition to that, IR models guide the matching process to retrieve a ranked list of relevant document given a query [19].

IR models are broadly categorized in to two approaches, which are semantic approach and statistical approach. Semantic approaches models such as, latent semantic indexing and neural network try to work on syntactic and semantic analysis. They attempt to implement some degree of understanding the natural language text that users provide. On the other hand, statistical approaches such as, vector space model and probabilistic model attempts to retrieve documents that are highly ranked in terms of statistical measure [19].

The three most widely used information retrieval model that bases on statistical approaches are : Boolean, vector space, and probabilistic. [20]

In the boolean model, a set of index terms are used to represent document and query. Therefore, the mathematical basis can be called set theoretic. In the vector space model, a vector in a t-dimensional space is used for representing document and query, therefore, the model is algebraic. In the probabilistic model, probability theory is used for representing document and query. Therefore, the model is probabilistic. Different scholars have been proposed several alternative approaches based on their mathematical basis. For set theoretic models, the alternatives are; the fuzzy and the extended boolean models. For algebraic models, the alternatives are; the generalized vector, the latent semantic indexing, and the neural network. For probabilistic model, the alternatives are; the inference network and the belief network models [5].

### 2.5.1. Boolean Model

The BIR is based on Boolean Logic and classical set theory in that both the documents to be searched and the user's query are conceived as sets of terms.[21]

Retrieval is based on whether or not the documents contain the query terms
The Boolean model is the first and the simplest model of information retrieval. It works based on set theory and Boolean algebra which allowed users to specify their information need using a combination of classical Boolean operators, ANDs, ORs

and NOT [21][22][23]. For instance, query $t_1$ and query $t_2$ ($t_1$ AND $t_2$) is only satisfied by a given document $D_1$ if and only if $D_1$ contains both terms $t_1$ and $t_2$. Likewise, the query "$t_1$ OR $t_2$" is satisfied by $D_1$ if and only if it contains $t_1$ or $t_2$ or both. On the other hand, the query "$t_1$ AND NOT $t_2$" satisfies $D_1$ if and only if it contains $t_1$ and does not contain $t_2$. The model views each document as just a set of words [28].The weight for index terms in Boolean retrieval model are all binary (i.e. $W_{ij}$ E{0, 1}. The similarity of a document dj to the query q is expressed as:

$$(dj,q) = \begin{cases} 1 = if documents atisfies the boolean query \\ 0 = Otherwise \end{cases}$$

Where, if similarity of document $d_j$ to the query q is equal to 1 (present), the document $d_j$ is relevant. Whereas, if similarity of document $d_j$ to the query q is equal to 0 (absent), the document is not relevant [5].

Boolean model have an advantage of giving users a sense of control over the system. It is immediately clear why a document has been retrieved given a query. If the resulting document set is either too small or too big, it directly clear which operators will produce respectively a bigger or smaller set. The other advantage of this model is the clean formalism behind the model and its simplicity [21][26].

However, the model has a number of clear disadvantages. First, most user find it difficult to translate their information need into a Boolean expression. Second, the model has not follow document ranking principle by nature, which means all retrieved documents are considered equally important. Third, the model either retrieves too few or too many documents, or sometimes not retrieves any, which might lead to null result. In addition it's sometimes creates its own problems, such as, misunderstanding and misrepresentation of the users information needs [24][27].

Several attempts has been done so far to make the model effective by developing different alternatives of the model such as fuzzy set model and the extended Boolean

model. However, IR communities consider that Boolean system is less effective than ranked retrieval system [25]. Accordingly, a number of models have been proposed for this process; the widely used IR models are the Vector Space Model and the Probabilistic Model [26].

### 2.5.2. Vector Space Model

The vector space model for document retrieval is based, upon building an n dimensional vector for the query and each document in the collection. The full non-optimized model causes n to be equal to the number of words in the language, whereas in reality n is usually equal to the number of different words in the document collection (words in the query are not important, because if they don't appear in the document then they are of no use for retrieval). 28]

Vector space model is one of the commonly used statistical approaches to represent each textual document as a set of terms [5]. In this model, documents and terms are represented as vectors in a k-dimension space where each dimension corresponds to a possible document feature. The word "terms" is not inherent in the model, but terms are typically words and phrases. The values assigned to elements in the vector space describe the degree of importance of term in representing the semantics of the document. Sometimes a given term may receive a different weight in each document in which it occurs. If the term is not appearing in a given document the weight of that term will be zero in that document [5]. For a given document ($d_j$) the weight of the terms in it can be expressed as the coordinates of $d_j$ in the document space. A document collection containing a total of documents ('d') described by terms ('t') is represented as term by document matrix (T x D). Each row of this matrix is a term vector and each column of this matrix is a document vector. The element at row i, column j, is the weight of term j in document I [28].

| Term list | Doc1 | Doc2 | Doc3 | Doc4 | …… | Doc n |
|---|---|---|---|---|---|---|
| Term 1 | w11 | w12 | w13 | w14 | …… | w1n |
| Term 2 | w21 | w22 | w23 | w24 | …… | w2n |
| Term 3 | w31 | w32 | w33 | w34 | …… | w3n |
| Term 4 | w41 | w42 | w43 | w44 | …… | w4n |
| Term 5 | w51 | w52 | w53 | w54 | …… | w5n |
| …… | …… | …… | …… | …… | …… | ……. |
| Term m | wm1 | wm2 | wm3 | wm4 | | Wnm |

*Table 2:2 Example of Term Document Matrix*

In this model, query is interpreted as another document in document space. If the term that is not in the collection but appear in the query, this will add additional dimension in the document space [24].

The weight of terms in the documents or in the queries assigned by using term frequency (TF) and inverse document frequency (TF*IDF) scheme which are the most successful and widely used automatic generation of weights [24]. The term frequency is the frequency of occurrence of the given term within the given document. This scheme attempts to measure the degree of importance of the term within the given document. Inverse document frequency (idf) is a frequency of a given term within an entire collection of documents. It attempts to measure how widely the term is distributed over the given collection.

The similarity of document and query in vector space model is determined by correlation between the vectors dj and vector q. The correlation is quantified by the associative coefficients based on the inner product(dot product) of the document and query vector, where documents whose vectors are close to the query vector are more relevant to the query than documents whose vectors are far from the query vector [24].

As discussed in section 2.1.3 there are different similarity measurements. However, the most widely used by vector space model and popular similarity measure is the

cosine coefficient, which measures the angle between the document vector and the query vector [24].

According to Baeza-Yates and Ribeiro-Neto[24], Vector space model (VSM) have several advantages. First, it improves retrieval performance using its term-weighting scheme. Second, the partial matching strategy of VSM allows retrieval of document approximate the query condition. Third, it sort and rank the documents according to their degree of similarity to the query using cosine similarity measurement. Finally, it is simple to implement and fast.

However, the model has also several disadvantages. It considers terms as unrelated objects in the semantic space. This means, if no common words are shared between the query and documents in text collection, the similarity value will be zero and no document will be retrieved. This is usually happened when users and authors prefer to use different words which have the same meaning [24]. Vector space model is not attempt to define uncertainty in IR system and its ranked answers sets is difficult to improve upon without the integration of query expansion and reformulation modules to the model [29].

Literature suggested that one of the alternative modeling paradigm, which is capable of solving the above mentioned problem of vector space model is probabilistic IR model [29][30].

### 2.5.3. Probabilistic Model

A major difference between information retrieval (IR) systems and other kinds of information systems is the intrinsic uncertainty of IR. Whereas for database systems, an information need can always (at least for standard applications) be mapped precisely onto a query formulation, and there is a precise definition of which elements of the database constitute the answer, the situation is much more difficult in IR; here neither a query formulation can be assumed to represent uniquely an information need, nor is there a clear procedure that decides whether a DB object is an answer or not. (Boolean IR systems are not an exception from this statement;

they only shift all problems associated with uncertainty to the user.) As the most successful approach for coping with uncertainty in IR, probabilistic models have been developed.[31]

In information retrieval process, user first start with information needs which is then translated into query representation. On the other side, documents are also translated into document representation. Finally, the system attempts to determine the relevance of the document to the information need of the users. In IR models such as, Boolean and Vector space model, given a query and document representation matching is done without considering the semantic relationship between query and documents. IR systems build upon those models has an uncertain guess of whether a document has content relevant to the information need. However probabilistic theory provides a principled foundation for such reasoning under uncertainty [28][29].

Probabilistic information retrieval is the estimation of the probability of relevance that a document $d_i$ will be judged relevant by the user with respect to query q. which is expressed as, $P(R|q,d_i)$, where, R is the set of relevant document. Typically, in probabilistic model, based on the query of user the documents are divided in to two parts. The first contain relevant documents and the second contain non-relevant (irrelevant) documents. However, the probability of any document is relevant or irrelevant with respect to user query is initially unknown. Therefore, the probabilistic model needs to guess at the beginning of searching process. The user then observe the first retrieved documents and gives feedback for the system by selecting relevant documents as relevant and irrelevant documents as irrelevant. By collecting relevance feedback data from a few documents, the model then can be applied in order to estimate the probability of relevance for the remaining documents in the collection. This process iteratively applied to improve the performance of the system so as to retrieve relevant documents which satisfies users need [24].

In probabilistic model, the order in which documents are presented to the user is to rank documents by their estimated probability of relevance with respect to the user

information need. The principle behind this assumption is called, probability ranking principle (PRP) [29][24].

Several retrieval models have been developed, which has a probabilistic basis. The mostly used and recent developed methods of probabilistic model are, Binary Independent Model, Inference Network Model and Belief Network Model [5].

#### 2.5.3.1. Binary Independent model

The classical Binary independence model (BIM) was introduced in 1976 by Roberston and Sparck Jones[24]. According to Greengrass [28], the model has been called 'binary independence' because it has been assumed that to arrive at the model one must make the simplifying assumption that the document properties that serve as clues to relevance are independent of each other in both the set of relevant documents and the set of non-relevant documents.

In BIM, binary is equivalent to Boolean; queries and documents are represented as binary incidence vectors of terms. $D=\{d_1,d_2,....,d_n\}$ where, $d_i=1$ if term i is present in document d and $d_i=0$ if term i is not present in document d. Moreover, query q represented by the incidence vector $^{q}$ . As expressed above, in BIM model, the probability of P(R|d,q) that a document is relevant through the probability in terms of term incidence vectors P(R| $^{x,q}$ ) in both document and query.

#### 2.5.3.2. Bayesian Networks Model

Bayesian networks model was introduced by Turtle and Croft [28] as information retrieval model on the basis of probabilistic theory. Bayesian networks use a form of probabilistic graphical model. They use directed acyclic graphs (DAGs) to show probabilistic dependencies between variables, in which the nodes represent random variables, the arcs depict causal relationships between these variables. They have child and parent causal relationship, which is represented by linking each parent node to the child node. Only the root nodes are without parents [24].

There are two models for information retrieval based on Bayesian networks. The first model is called inference network and the second model is called belief network.

#### 2.5.3.3. Bayesian Inference Network Model

According to Baeza-Yates amd Ribeiro-Neto [24], there are two traditional schools of thought in probability which are based on the frequentist view and the epistemologist view. The frequentist views probability as a statistical notion related to the laws of chance. The epistemologist views probability as a degree of belief whose specification might be devoid of statistical experimentation.

Bayesian Inference Network model is built up on epistemologist view of the information retrieval problem. It associates random variables with the index terms, the documents and the user queries. The model computes P(I|D), the probability that a user's information need (I) is satisfied given a particular document (D). This probability can be computed separately for each document in a collection. The documents can then be ranked by probability, from highest probability of satisfying the user's need to lowest [24].

#### 2.5.3.4. Bayesian Belief Network Model

Bayesian belief network is the use of Bayesian calculus to determine the probabilities of each node from the predetermined conditional and prior probabilities [5]. As Baeza-Yates and Ribeiro-Neto [24], stated in Bayesian belief network the users query q is modeled as a network node to its associated random variable. Whenever q completely covers the concept space C the variable is set to 1. Therefore belief network computes the probability of q, (i.e. P(q)) by the degree of coverage of the space C by q.

Document $d_j$ is modeled as network node to its associated binary random variable. If $d_j$ completely covers the concept space C the variable set to 1. To compute the probability of $d_j$ ($P(d_j)$), compute the degree of coverage of the space C by $d_j$. In belief network documents and the user query modeled as subsets of index terms. Each subset is interpreted as concept in the concept space C [24].

The ranking principle in belief network expressed as, the degree of coverage provided to the concept $d_j$ by the concept q. $P\{d_j|q)$ is adopted as the rank of the document $d_j$ with respect to the query q[24].
In general, probabilistic models attempt to capture the information retrieval problem within a probabilistic framework. Unfortunately, the probabilistic model has got its own drawbacks. First, the probability theory analysis takes much more time and efforts, and it offer unnecessary theoretical burden on the researcher. Second, probabilistic model need to guess the initial separation of documents into relevant and non-relevant sets. Third, the model does not take into account the frequency with which an index term occurs inside a document. In other word, all weights are binary [24][26].

However, probabilistic model have several potential advantages [24][26]. The first, advantage is the expectation of retrieval effectiveness that is near to optimal relative to the evidence used is high. Second, it has less reliance on traditional trial and error retrieval experiments. Third, each document's probability of relevance estimate can be reported to the user in ranked output. It would presumably be easier for most users to understand and base their stopping behavior (i.e., when they stop looking at lower ranking documents).

## 2.6. Relevance feedback

Relevance feedback is a process, introduced over 20 years ago, and designed to produce improved query formulations following an initial retrieval operation so as to improve the final result set [35][36]. There are two relevance feedback mechanisms user relevance feedback and pseudo relevance feedback.

The idea of user relevance feedback is to involve the user in the retrieval process. In particular, the user gives feedback on the relevance of documents in an initial set of results. The procedures followed in user relevance feedback are the following. First, the user provides a query based on which the IR system returns initial relevant documents. Second, the user marks some returned documents as relevant or non-

relevant. Third, the system computes a better representation of the information need based on the user feedback. Finally, the system displays a revised set of retrieval results.

On the other hand Pseudo relevance feedback, also known as blind relevance feedback, provides a method for automatic local analysis. It automates the manual part of relevance feedback, so that the user gets improved retrieval performance without an extended interaction. The method is perform normal retrieval to find an initial set of most relevant documents, then it assumes that the top k ranked documents are relevant, and finally, the system displays a revised set of retrieval results [35][36].

## 2.7. Related Works

Effective systems for mono-language information retrieval have been available for several years. Research in the area of multi-lingual information retrieval has focused on incorporating new languages into existing systems to allow them to run in several mono-language retrieval modes. In recent times, greater interest in retrieval across languages has motivated more work to study the factors involved in building a CLIR system.[4]

Very limited works have been done in the areas of IR and CLIR in the past in relation to African indigenous languages including major Ethiopian languages [4]. Very limited research works conducted so far on Amharic–English and Afaan Oromo-English CLIR. But nothing is found for Tigrigna. The following sub sections summarize the review of some related works (for different pairs of languages) in which different approaches of CLIR used to cross the language barriers.

### 2.7.1. Oromo-English Cross Language Information Retrieval

The bilingual information retrieval experiments as part of ad-hoc was conducted by Kula [36] for three different languages: Oromo-English, Hindi-English and Telugu-

English. The experimentation result of Oromo-English cross language information retrieval experiments at CLEF'06 is summarized as bellow. In the study, the dictionary based approach of CLIR was used to translate Oromo topics into a bag of words of English queries. The basic objective of the study was to design and develop an Oromo-English CLIR system with a view to enable Afaan Oromo speakers to access and retrieve the vast online information resources that are available in English by using their native language queries. In the experimentation three different official runs were submitted in Oromo-English bilingual task. Afaan Oromo queries were translated into their equivalent English queries based on Machine Readable Dictionary (MRD) and submitted to the text retrieval engine. The English test collection was provided by CLEF (Cross Language Evaluation Forum) from two newspapers, the Glasgow Herald and the Los Angeles Times. The Oromo-English dictionary the researchers used was adopted and developed from hard copies of human readable bilingual dictionaries by using Optical Character Recognition (OCR) technology. The developed dictionary was used to translate Oromo topics into a bag of words of English queries.

In order to define Afaan Oromo stop words, the researchers first created a list of the top most frequent words found in Afaan Oromo text corpus collected. Then pronouns, conjunctions, prepositions and other similar functional words in Afaan Oromo were also added in the stop word list. Once these less informative words were removed from Afaan Oromo text corpus, a light stemming algorithm was applied in order to conflate word variants into the same stem or root.

The experimentation results of the three official runs, i.e. title run (OMT), title and description run (OMTD), and title, description and narration run (OMTDN) for Oromo- English bilingual task in the ad-hoc track of CLEF'06 summarized in table 2.2. In the table the summary of the total number of relevant documents (Relevant-tot.), the retrieved relevant documents (Rel.Ret.), and the non-interpolated average precision (R- Precision) are summarized. The Mean Average Precision (MAP) and Geometric Average Precision (GAP) scores of the three runs are also presented.

| Run-Label | Relevant-tot. | Rel. Ret. | MAP | R-Prec. | GAP |
|---|---|---|---|---|---|
| OMT | 1,258 | 870 | 22.00% | 24.33% | 7.50% |
| OMTD | 1,258 | 848 | 25.04% | 26.24% | 9.85% |
| OMTDN | 1,258 | 892 | 24.50% | 25.72% | 9.82% |

*Table 2:3 Summary of average results for the three runs*

The evaluation of this experiment was also conducted by Kula [36] and the purpose of the evaluation experiment was to assess the over all performance of the dictionary approach by using these different fields of Afaan Oromo topics.

Based on the evaluation result it was concluded that limited language resources can be used in designing and implementation of a CLIR system for resource scarce languages like Afaan Oromo. The significant average results for all official runs were achieved even though limited CLIR resources used in the experiments.

### 2.7.2. English-Oromo Machine Translation

An Experiment Using a Statistical Approach was conducted by Sisay [37] focused on the translation of English documents to Oromo using statistical methods. In general, the work has two main goals: one is to test how far we can go with the scarce recourse of the parallel corpus for the English-Oromo language pair and the applicability of existing SMT systems on these language pairs. The second one is to analyze the output of the system with the objective of identifying the challenges that need to be addressed. The architecture of the English-Oromo SMT system studied includes four components: Language Modeling, Translation Modeling, Decoding and Evaluation.

The Language Modeling component takes the monolingual corpus and produces the language model for the target language (Afaan Oromo). The Translation Modeling component takes the part of the bilingual corpus as input and produces the translation model for the given language pairs. The Decoding component takes the

language model, translation model and the source text to search and produce the best translation of the given text. The Evaluation component of the system takes the system output and the reference translation and compares them according to some metric of textual similarity. The parallel documents used for the experimentation part were: Some Afaan Oromo versions of Bible chapters that are available both in English and Afaan Oromo languages and some spiritual manuscripts for which the English equivalents were accessible on the web, the United Nation's Declaration of Human Rights, the Kenyan Refugee Act, the Ethiopian Constitution, some medical documents, the proclamations, and the regional constitution of Oromia Regional State. By using these resources, an average BLEU (Bilingual Evaluation Understudy) score of 17.74% was achieved based on the experimentation.

### 2.7.3. Dictionary-based Amharic-English Information Retrieval

Amharic-English cross lingual information retrieval that is based on dictionary-based approach was designed by Atelach [7]. The experiment was conducted in two different approaches. Both experiments used a dictionary based approach to translate the Amharic queries into English bags-of-words, but while one approach removes non- content bearing words from the Amharic queries based on their IDF value, the other uses a list of English stop words to perform the same task. After the conversion of Amharic topics into English queries system retrieval was done by using translated English queries as input. The translation was done through a dictionary look up that takes Amharic words in the topic and finding the corresponding English match from MRD. The main difference between the two approaches is the way non-content bearing words were identified and removed.

At a general level, the two approaches consist of a step that transforms the Amharic topics into English queries, followed by a second step that takes the English queries as input to a retrieval system. In both approaches the translation was done through a simple dictionary lookup that takes each stemmed Amharic word in the topic set and tries to get a match and the corresponding translation from a MRD. In the first approach the conversion was not done for all Amharic words. The cutoff point depending on the inverse document frequency value of each word was set to look for

the translation words in a dictionary. In this approach those Amharic words that have IDF value below the given threshold were reduced and then looks up in the dictionary was performed for the remaining words. In the second approach the translation was done for all Amharic words, and after they were translated into their English equivalent words those found in the list of English stop words were removed.

The MRD used for the experimentation was one that consisted of any entry of Amharic words and their derivational variants. The infixed words were represented separately in the dictionary. Therefore, the stemmed words of the Amharic query were looked up for their possible translations in the dictionary. If there was a match and only one sense of the word, the corresponding English word or phrase in the dictionary was taken as possible translation. If there was more than one sense of the term, then all possible translations were picked out and a manual disambiguation was performed. For most of the proper names there was no entry in the dictionary, hence they were translated manually.

The resulting translated English queries by using both of the approaches were given to the retrieval engine to determine the average precision value obtained. The results from the two approaches differ, with the second approach (based on a list of English stop words) performing slightly better than the first approach (based on IDF values for the Amharic terms), but they both perform reasonably well. The average precision value obtained for the first approach was 0.3615 while it was 0.4009 for the second approach.

### 2.7.4. Amharic-English Cross-lingual Information Retrieval: A Corpus Based Approach

Research on Amharic-English Cross-lingual Information Retrieval that employed corpus- based approach was conducted by Aynalem [36]. The objective of the research was to experiment on Amharic- English corpus-based CLIR by using statistical method to translate Amharic queries into their respective English. Amharic query was translated into English query since the research was aimed at

making English documents available to the Amharic speakers by using queries of their native language. To achieve query translation the researcher built bilingual dictionary by using parallel corpus of Amharic and English documents collected. This dictionary was constructed by the help of GIZA++ word alignment toolkit after the data collected were preprocessed. The data preprocessing task done by the researcher includes case normalization, tokenization and transliteration. The parallel corpus employed for the research includes news and legal items. Since the two languages (Amharic and English) use different alphabets, the text must be transliterated into the other alphabet. To handle this, the text character of collected Amharic documents and Amharic queries were translated into English. In the research Amharic queries were used for the retrieval of both Amharic and English documents. The Amharic queries were converted into their equivalent English queries for the retrieval of documents written in English. With the help of the bilingual dictionary constructed. The researcher wrote python script code to search and convert the equivalent English translation of the Amharic queries from the dictionary.

Evaluation of the system was conducted for both Amharic and English documents retrieved by using Amharic queries. The experimentation of the research involves monolingual and bilingual retrieval evaluation. For monolingual evaluation Amharic queries are given to the system to retrieve Amharic documents whereas for bilingual evaluation translated Amharic queries are given to the system to retrieve English documents.

The experimentation was conducted in two phases. In the first experimentation words with high or low frequency were not used for the content representation while for the second phase of experimentation all words with the exception of stop words were used as index terms. The results found after conducting the second phase of the experimentation was a maximum precision value of 0.24 and 0.33 for Amharic and English respectively.

### 2.7.5. Cross-Lingual Information Retrieval System for Indian Languages

Microsoft Research India worked on Hindi to English cross-lingual system [11] in which a word alignment table that was learnt by a SMT system trained on aligned parallel sentences was used. The research was experimented on English corpus of LA Times 2002. The researchers worked as part of the ad-hoc monolingual and bilingual track of CLEF 2007. The task was to retrieve relevant documents from an English corpus in response to a query expressed in different Indian languages including Hindi, Tamil, Telugu, Bengali and Marathi.

The researchers used a word alignment table that was learnt by a SMT system trained on aligned parallel sentences, to map a query in source language (Hindi) into an equivalent query in the language of the target (English) document collection. The query in Hindi language was translated into English using word by word translation. For a given Hindi word, all English words which have translation probability above certain threshold were selected as candidate translations. Only words with the probability of above the threshold value set were selected as final translations to reduce ambiguity in the translation. This method may reduce the size of aligned words to be included in the bilingual dictionary constructed, if the probability of query word is below the threshold value set. Once the query was translated into the language of the document collection the relevant documents retrieved using a language modeling based retrieval algorithm.

The experiment was done for two different cases for both monolingual and cross lingual retrieval. In the first case when the high threshold value (selected value was 0.3) was used for the translation probability. This was done to avoid ambiguity, when there are many possible English translations for a given Hindi word. The result was presented in terms of precision at different levels of interpolated recall. In the second set of experiments, various levels of threshold values were used. The result shows that, cross- lingual performance was improved for the latter case when compared with first case. This indicate that word translations with no threshold on the translation probability gave the best results.

### 2.7.6. Corpus-Based Cross-Language Information Retrieval in Retrieval of Highly Relevant Documents

The study conducted by Talvensaari [36] was aimed to find out how corpus- based CLIR retrieve highly relevant documents which has become progressively important in the age of enormously large collections. The study was based on query translation technique. A comparable corpus of Finnish-Swedish was used as a source of knowledge for the translation of query. The test queries were selected from Finnish corpus and translated into Swedish to retrieve documents in Swedish language. Both Swedish and Finnish corpora were collected from the news articles of different time span.

The performance evaluation of the system was measured by using recall and precision measurement, which weight the retrieved documents according to their relevance level. The researchers experiment their work with the term vector matching strategy of the comparable corpus translation system (COCOT) which was evaluated with graded relevance assessments to find out how the system managed in retrieving highly relevant documents. The performance of the system was compared to that of a dictionary based query translation system. The results of the study indicate that corpus-based translation works better in the retrieval of highly relevant documents than dictionary-based translation.

# CHAPTER THREE
# DESIGNING TIGRIGNA-AMHARIC CLIR SYSTEM

In designing probabilistic Tigrigna-Amharic cross language information retrieval system, the two major components which are indexing and searching are used [4]. Indexing is an off-line process used to facilitate access to preferred information from document corpus accurately and efficiently as per users query. Searching is an on-line process used to scan document collection so as to find relevant documents that satisfy users need .

Figure 3.1 depicts the probabilistic based CLIR system architecture designed and implemented in this research. It indicates that, the CLIR system has on-line (Searching) and off-line (Indexing) processes. During the off-line process, Tigrigna and Amharic documents are organized using inverted indexing structure. To ease the indexing task, text operations are applied on the text of the documents in order to transform them in to their logical representation. The documents are indexed and the index is used to execute the search. The searching task is an on-line process that accepts users query. The query is processed to identify terms using which searching is done to identify and retrieve relevant documents. After ranked documents are retrieved, the users provide feedback that can be used to refine the query and restart the search for improved results [4].

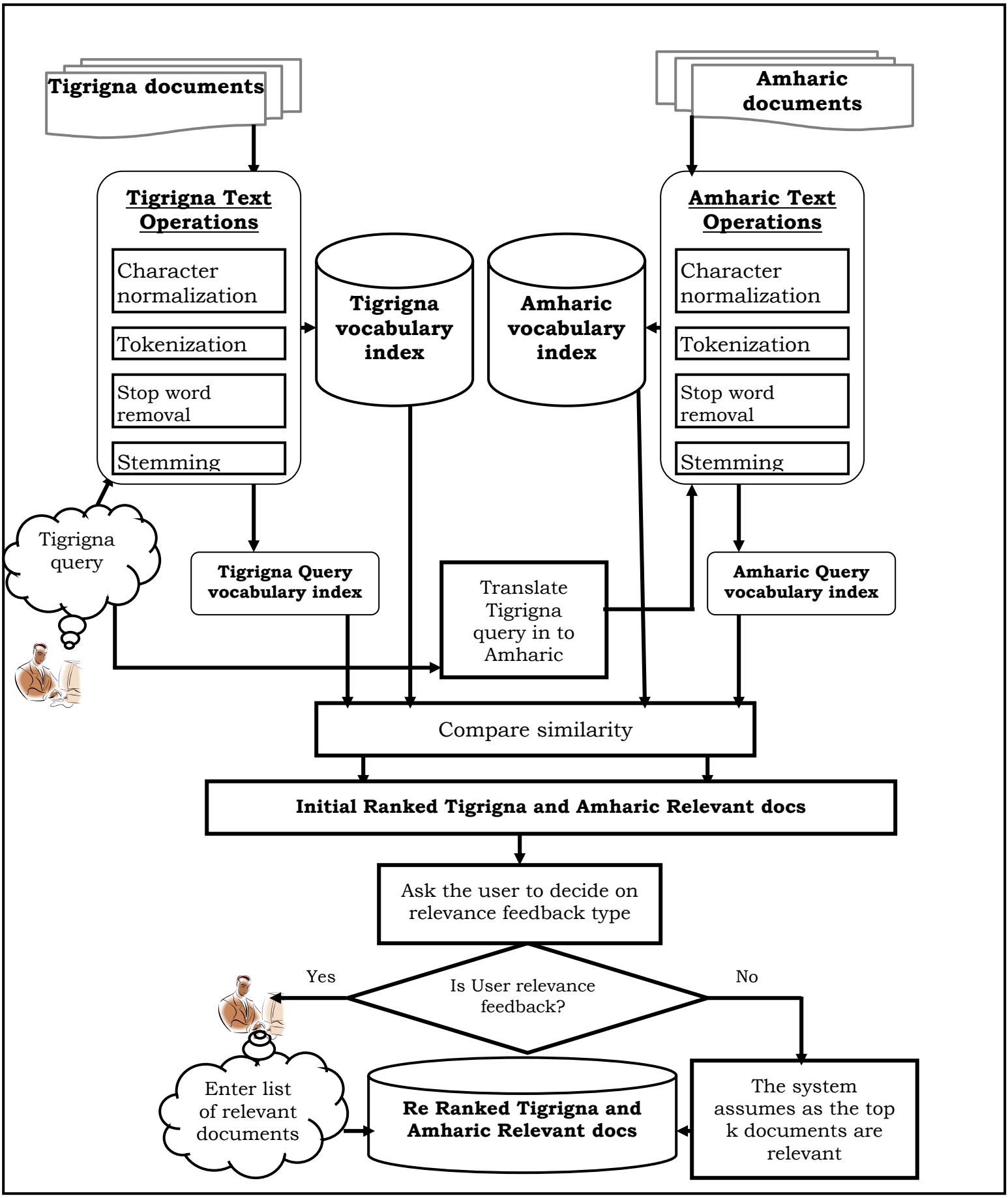



*Figure 3.1: Architecture of Tigrigna-Amharic CLIR system using probabilistic model*

## 3.1. Tigrigna and Amharic Document Indexing

In information retrieval, searching is possible or efficient when the database is small. However, in large databases searching will take much more time and space unless indexing structure is used to organize documents. Therefore, constructing and maintaining indexing on large database is necessary.

### 3.1.1. Word Stemming

In Tigrigna and Amharic writing systems, words are morphologically variant. Morphological variant words have similar semantic interpretations. In IR system, those words are considered as equivalent words [9]. Therefore, words have to be reduced to their root using stemming technique. On the other hand, stemming is also used to reduce the dictionary size (i.e. the number of distinct terms used in representing a set of documents). The smaller the dictionary size the smaller storage space and processing time required. Moreover, Stemming techniques are language dependent. Therefore, every language needs to have language specific stemming technique. In Tigrigna and Amharic text, there are many word variants/affixes [8][9]. To conflate them into stem word, stemming technique/ algorithm developed by Yonas [8] for Tigrigna and Nega [32] for Amharic languages were used. Both developed the stemmer that involves the removal of both prefixes and suffixes and also consider letter inconsistency and reiterative verb forms.

### 3.1.2. Stop Word Removal

Not all terms found in the document are equally important to represent documents they exist in. Some terms are common in most documents. Therefore, removing those terms, which are not used to identify some portions of the document collection, is important. Such terms are removed based on two methods. The first method is to remove high frequent terms by counting the number of occurrences (frequency). The second method is using stop word list for the language [22][23].

In this research the second method is used to apply stop word removal. The stop word list is adopted form yonas [8] and Nega [32]. Removing stop words were applied

before stemming is implemented. This is because, some stop words may have different look if they are stemmed and can be considered as content bearing terms.

### 3.1.3. Normalizing Characters and Words

As discussed in section 2.4.3, Both Tigrigna and Amharic writing system contains characters or symbols, which have the same sound during reading. For example, in the case of Amharic the characters "ሀ, ሐ,ኀ, አ and ዐ" produces the same sound during reading with characters "ሃ, ሓ, ኃ, ኣ and ዓ" respectively. On the other hand, there are also alphabets that share the same sound, such as, "ሀ, ሐ, and ኀ", "ሰ and ሠ", "አ and ዒ", and "ጸ and ፀ". The same is true for Tigrigna except that "ሀ and ሐ " and "አ and ዐ" have different sounds. As a result, words which have the same meaning may have different spelling structure. For example, the word "አሥራት" can be written in different way such as, "አስራት", "ዓስራት" "ኣስራት" and "ኣሥራት". In this research such kind of problems are solved by changing the characters and the word to their common standard form.

### 3.1.4.Punctuation Marks Removal

Punctuation marks are usually attached to the words which precede them this is also the case in Tigrigna and Amharic. Removal of punctuation marks is essential to prepare data for the indexing process. This is because the same word that is attached to a punctuation mark and that is not attached to a punctuation mark is considered as different words. For example, the word *"መፂኡ"* and *"መፂኡ?"* are different words when they are indexed unless the question mark (?) that is attached to the latter is removed.

## 3.2. Bilingual Dictionary

The translation of Tigrigna queries into Amharic was done by using Tigrigna-Amharic MRBD which is taken and digitized from printed dictionary called "Saba English-Amharic- Tigrigna dictionary".

The bilingual dictionary stores both source words (Tigrigna) and their corresponding translation of the target words (Amharic). The dictionary provides different possible translation for a single word. In this research the first single word is taken. Python script was developed to select the one that is found at the beginning of the statement.

### 3.2.1. Translation

This component is responsible for taking query in one language and translating it into another language, i.e. it is the query translation phase. Query translation is required to achieve CLIR with the help of a MRBD. The translation of the given query into another language is needed to retrieve documents in the translated (target) language. For this research, a given Tigrigna query was translated into its equivalent Amharic query. The translated query was passed under different text operation and sent to the document collection in the target language (Amharic) to retrieve the relevant documents. Translation of the query was done on a word-by-word basis for this study. Python script was written for the translation of a Tigrigna query to its equivalent Amharic query by searching through the MRBD.

## 3.3. Searching Using the Probabilistic Model

Because of its capability of handling the uncertain nature of information retrieval, the Binary Independent Model is used to design probabilistic Tigrigna-Amharic CLIR system. This is because, according to Greengrass [28], the first step in most of probabilistic methods is to make some simplifying assumption. Thus, BIM is the model that has been used with the probabilistic ranking principle by introducing some simple assumptions which makes estimating the probability function $P(R|d, q)$ practical.

The feedback process is also directly related to the derivation of new weights for query terms and the term re-weighting is optimal under the assumptions of term independence [24]. In addition, it is the first model that has been used in several

researches because of its clear and simple mathematical and theoretical assumptions

In binary independent model there are three steps to compute term probability. The first step compute terms when there is no retrieved document at initial stage. The second step compute terms after documents are retrieved and feedback is provided by the user. The third step compute terms when partial feedback is given [24]. At first, since the properties used to retrieve relevant information are unknown and only index terms are known properties, attempt has to make the initial guessing. The assumptions made in this step are [31];

- p(ki|R) is constant for all index terms k (usually, its equal to 0.5)

The distribution of index terms among the non-relevant documents can be approximated by the distribution of index terms among all the documents in the collection.

These two assumptions will give as;

$$P(K_i \mid R) = 0.5 \text{ and } P(K_i \mid R) = \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

Where, N is the total number of documents in the collection and $n_i$ is the number of documents which contain the index term ki.

Using this initial guess, documents are retrieved which contain query terms and provide an initial probabilistic ranking. After documents are retrieved, the user looks at the retrieved documents and marks them as relevant and non-relevant. The system then uses this feedback to refine the description of the answer set. At this stage, initial ranking is shown and more discriminating information about terms is available (i.e. relevance feedback), this will allow more accurate estimation [24][31]. Therefore, relevant documents retrieved should be improved using probabilistic relevance weighting technique. This technique uses the concept in term incidence contingency [31]. See table 3.1 shown below.

| | **Relevant** | **Non-relevant** | **Total** |
|---|---|---|---|
| Containing the term | r | n - r | n |
| Not containing the term | R - r | N – n – R + r | N - n |
| Total | R | N - R | N |

*Table 3.1: Term incidence contingency table*

Where,

r is the number of relevant documents that contain the term,

n – r is the number of non-relevant documents that contain the term,

n is the number of documents that contain the term,

R- r is the number of relevant documents that do not contain the term,

N – n – R + r is the number of non-relevant documents that do not contain the term,

N – n is the number of documents that do not contain the term, R is the number of relevant documents,

N – R is the number of non-relevant documents and N is the total number of documents in the collection.

After the knowledge of relevant documents and non-relevant documents for a given query is completed, the next step is estimating the probability of finding term (ti) in relevance document using equation 3.2 and the probability of finding term (ti) in non-relevant document using equation 3.3 [31];

$$P(t_i \mid R) = \left( \frac{r}{R} \right) \quad \text{..... 3.2}$$

$$P(t_i \mid R) = \left( \frac{n-r}{N-R} \right) \quad \text{..... 3.3}$$

According to Sparck [31], the above equations can be rewritten to compute term presence weighting function as;

$$W = \log\left( \frac{N-n \mid -R+r}{(R-r)(n-r)} \right) \text{..... (3.4)}$$

However, Robertson and Jones [31], noted different assumptions lead to a different formula for computing term weighting. They argue “in practice users may find themselves in the situation where, even if they know some relevant documents are retrieved, they wish to continue searching”. They assume that “users may not found all the relevant documents that would satisfy their need”. Therefore, the record in the center of the contingency table (i.e. N – n – R + r) may not be taken as absolute. The estimation of document relevance when considering new items has to allow for uncertainty. This estimation adds 0.5 to all the central record and it derives a specific term relevance weighting formula;

Relevance Weighting

$$RW = \log\left(\frac{(r+0.5)(N-n-R+r+0.5}{(R-r+0.5)(n-r+0.5}\right) \quad \ldots\ldots \ (3.5)$$

## 3.4. IR System Evaluation

Retrieving relevant document from the collection that satisfies users need is the heart of IR system evaluation in determining its effectiveness. Two strategies are identified in measuring the effectiveness performance of IR systems. The first is the user-centered strategy, which uses relevant judgment so as to evaluate the performance of the system and the second is system centered strategies which work based on reference judgment available prior to testing process [24]. Based on the concept of relevance (i.e. to a given query or information need), there are several techniques of measures of IR performance available, such as, precision and recall, F-measure, E-measure, MAP (Mean average precision), R-measure, etc [26]. In this study, the three widely used techniques precision, recall, and F-measure are used to measure the effectiveness of the IR system designed.

Precision and recall are the two most frequent and basic statistical measures. Recall is the percentage of relevant documents retrieved from the database in response to users query, whereas precision is percentage of retrieved documents that are relevant to the query [18]. To show these metrics, assume the document collection

be D. Let Rt be all retrieved documents from the collection D and Rl the number of relevant documents in D. The union of Rt and Rl is a set of documents retrieved and relevant, RA. Therefore, the recall and precision can be calculated using equation 3.5 and 3.6 respectively.

Recall= RA |Rt||Rl|..... (3.5)

Precision= RA |Rt||Rt|..... (3.6)

The formula shown above for precision and recall assume that, all documents retrieved (Rt) is examined by user. Thus, the retrieved document cannot be presented for the user at once. Rather, the retrieved documents are presented according to their degree of relevance as per the user query. Then, the user examines the ranked documents starting from the top. This kind of examination of documents by the user leads the recall and precision measures to vary. Therefore, for appropriate evaluation of recall and precision, plotting a precision versus recall curve is necessary [24]. To draw precision-recall curve, for example let documents retrieved by the system is Dr Where Dr = {d3, d33, d9, d1, d10, d11, d14, d7, d44, d49, d50, d53, d55, d77, d133} in ranked order. Assume Rq contain a set of relevant documents for the query. Where, Rq = {d1, d3, d7, d10, d14, d33, d44, d49, d55, d133}. Based on the given information recall- precession curve is constructed. However, when constricting the curve based on the original recall and precession may result in saw tooth curve, there is a need to smooth the curve using interpolation technique.

In general, precision and recall have been used widely to evaluate information retrieval system performance. However, they measure two different aspects of the system and thus they are inversely related. If recall of a system is improved then the precision is reduced. The reason behind this is that, when an attempt is made to include many of the relevant documents, more and more irrelevant documents exist in the answer set. On the other hand, if precision of a system is improved then the recall is reduced. This is because; there are retrieved relevant documents among the

whole relevant documents found in the corpus. Achieving both precision and recall 100% is ideal but impossible in reality [24]. Several problems have been distinguished. First, to make appropriate estimation of maximum recall for a query, it needs deep knowledge of all the documents in the collection. Second, even if many situations consider the use of a single measure, which combines both, recall and precision capture different aspects of the set of retrieved documents. One of the methods developed to alleviate the above recall and precision problems is the F-measure [18]. In this research probabilistic model is used to enhance the performance of Tigrigna-Amharic CLIR system by increasing the precision without affecting the recall ability of the system. Thus, the F-measure is used to measure the performance of the system since it balances the precision and recall values. F-measure is a single value measure that trades off precision versus recall. It is the weighted harmonic mean of precision and recall expressed as follows [18];

$$f_j = 2PR/P+R \quad \ldots\ldots\ldots\ldots\ldots (3.7)$$

Where R=(r1,r2,….r11,…rn) and P=(p1,p2,….p11,…pn) are recall and precision at the jth document respectively. F-measure will be 0 when no relevant document is retrieved and 1 when all the first ranked documents are relevant. Moreover, the harmonic mean F assumes a high value only when both recall and precision are high.

In summary, there are different methods in designing probabilistic based IR system. However, the binary independent method is used to develop probabilistic based IR system for Amharic language so as to enhance the performance of the IR and to alleviate the problem of uncertainty that exists in IR. To evaluate the performance of the method, the model is implemented and tested using Tigrigna and Amharic documents.

# CHAPTER FOUR

# EXPERIMENTATION AND ANALYSIS

## 4.1. Document Selection

The researcher has utilized different sources of text for the development of the Tigrigna - Amharic CLIR system and the experiments. Since there is no document collection available for Tigrigna and Amharic languages like the TREC, and the CLEF collections for the English language, the researcher used his own collections. The researcher took documents from different sources.

A total of 200 Tigrigna and Amharic documents were used as a document corpus to test the prototype system. The documents are obtained from different web sites. Minister of Health, Minister of Education, Awake news articles, online bible chapters, Walta Information Center, and Hadas Eritrea were sources of the researcher's corpus collection.

As shown in table 4.1, the document corpus contains five clusters, which are health, education, Religion, social and politics.

| **No** | **Types of Documents** | **Number of Documents** |
|---|---|---|
| 1 | Health | 30 |
| 2 | Education, | 30 |
| 3 | Religion, | 60 |
| 4 | Social | 40 |
| 5 | Politics | 40 |
| | Total | 200 |

*Table 4.1: Types of articles used*

Each document is saved under common folder using UTF8 format, which is supported by most programming languages. Additionally, 6 test queries were selected by the researcher to test the performance of the system. Relevance judgment is also done for identifying which document is relevant for a given test query. Both user relevance feedback, where a user is asked to select the list of relevant documents out of the retrieved ones, and pseudo relevance feedback, where the system takes some of the top ranked documents as relevant, are used,

## 4.2. Query Selection

Queries were prepared for the selected sample test documents of Tigrigna. The preparation was done in such a way that it is relevant for the given selected test documents and it was prepared for Tigrigna selected test documents only. As pointed out, the MRBD is used for the query translation techniques in order to bridge the gap between the query language (Tigrigna) and the document language (Amharic). So Tigrigna test queries were translated into Amharic to retrieve both Tigrigna and Amharic documents that are relevant for the given queries. To accomplish this task Tigrigna queries were passed through the translation module to be translated into their Amharic equivalent. Then, the translated queries were used to retrieve Amharic documents.

## 4.3. Description of the Prototype System

Preprocessing on the queries and test document collections is performed. Then queries are projected to the term-document matrix. However, unlike vector space model, which obtains frequency of each query terms in the documents, Binary Independent probabilistic model obtains the absence and presence of the query terms in documents. In addition to that, queries are weighted two times (i.e. before and after relevance feedback is given).

In developing the prototype system of Dictionary based Tigrigna–Amharic CLIR several components are integrated together as depicted in Figure 3.1. Those

components describe the major activities done in the CLIR system development. The system is developed using python version IDLE 3.2. Figure 4.1, presents a screen shot which shows the first list of retrieved document using a given query.


*Figure 4.1: screen shot of the main page of the prototype*

The main objective of the prototype system is to project query terms and documents of both Tigrigna and Amharic documents in the matrix to make comparison between documents and a query. Then having calculated the weight of each query terms based on the notion implemented by probabilistic model, it calculates the score of each document and rank the retrieved documents and list them in decreasing order starting from the most relevant documents to the least .

The indexing process is implemented to construct inverted index. According to Amanuel [5], when implementing inverted index there are several tasks that need to be done. Such as, stemming, normalization, tokenization and stop word removal. The first step in constructing inverted index is generating index terms from document collections. In this step, the first task is tokenizing terms. Figure 4.2, shows python code implemented to tokenize terms found in Tigrigna document collection.

```
for j in range(1,d_no+1):

    dline='tdct\\'+docu_List[j-1]
    ddn=docu_List[j-1]
    ddn=ddn.replace('Tgfile','')
    ddn=ddn.replace('.txt','')
    dname=''
    dn=int(ddn)
    dline=dname+dline.rstrip()

    encoding_doc=open(dline,encoding='utf-8')
    while True:
        v_String=encoding_doc.readline()
        fstring=remove_puc(v_String)
        fstring=Tnormaliz(fstring)
        v_List=fstring.split()
```

*Figure 4.2: Python code for tokenization*

The code first reads the first line of the first file. Then, it removes punctuations and performs normalization finally it splits that line and stores the list. It performs likewise iteratively till all lines of all documents are processed.

As discussed in section 2.5.3, in both Tigrigna and Amharic writing system there are characters, which have the same sound during reading. In addition, there are alphabets that share the same word sound. To convert characters and words to their common form, normalization module is implemented.

One of the problems faced when implementing normalization was finding standard list of characters for both languages. In this research, list of characters is prepared. Figure 4.3, shows how Tigrigna characters are converted to their common form.

```
def Tnormaliz(text):
    h1=["ሀ","ሁ","ሂ","ሃ","ሄ","ህ","ሆ"]
    h3=["ኀ","ኁ","ኂ","ኃ","ኄ","ኅ","ኆ"]

    s1=["ሰ","ሱ","ሲ","ሳ","ሴ","ስ","ሶ"]
    s2=["ሠ","ሡ","ሢ","ሣ","ሤ","ሥ","ሦ"]

    t1=["ጸ","ጹ","ጺ","ጻ","ጼ","ጽ","ጾ"]
    t2=["ፀ","ፁ","ፂ","ፃ","ፄ","ፅ","ፆ"]
    for i in range(len(h1)):
        text=text.replace(h3[i],h1[i])
        text=text.replace(t2[i],t1[i])
        text=text.replace(s2[i],s1[i])

    return text
```

*Figure 4.3: Python code for normalization*

The code first takes a line of text from a document then checks for the existence of each character in h3, s2, and t2 then it replaces with the corresponding h1, s1, and t1 characters accordingly.

In Tigrigna and Amharic writing system words are morphologically rich. Since morphological variant words have similar semantic interpretations, there is a need to stem words to their root. In this research, the stemming implementation has two modules, the prefix removal module and the suffix removal module. Figure 4.4, depicts python code implementation of prefix and suffix removal of Tigrigna words.

```
def sufpre(v_List):
    prfx=open("Prefixlist.txt",encoding='utf-8')
    prefix=prfx.read()
    prefix=prefix.split()
    sfx=open("SuffixList.txt",encoding='utf-8')
    suffix=sfx.read()
    suffix=suffix.split()
    for n in range(0,len(v_List)):
        stemed_query=''
        stemed_query=stemed_query+v_List[n]
        for preffix_l in range(0,len(prefix)-1):
            if(len(stemed_query)>2):
                if(stemed_query.startswith(prefix[preffix_l])):

stemed_query=stemed_query.replace(prefix[preffix_l],'')
                    preffix_l=len(prefix)
        for suffix_l in range(0,len(suffix)-1):
            if(len(stemed_query)>2):
                if(stemed_query.endswith(suffix[suffix_l])):

stemed_query=stemed_query.replace(suffix[suffix_l],'')
                    suffix_l=len(suffix)
        v_List[n]=stemed_query
    sfx.close()
    prfx.close()
    return v_List
```

*Figure 4.4: Python code for prefix and suffix removal*

The code takes list of index terms and check if the length is greater than two or not. If the length of the word is less than two, the word is returned without further processing. If the length of the word is greater than two, the code iterates on word and check characters if they matched with one of the prefix found in prefix list. If the character is matched, the stemmed word is returned. It follows similar procedure to remove suffix.

One of the challenges faced when implementing stemming on Tigrigna and Amharic word is the algorithm adopted from Nega [39]. Sometimes, the algorithm over stem words that are inflected using prefix. This results the stem to become too short word. For instance, the word "**ማግኘት**" meaning 'to find' and "**ማግባት**" meaning 'to get married' are over stemmed to "**ማግ**"; and the algorithm adopted from Yonas[8] made

similar errors for Tigrigna words like **ኸሰብክ** meaning to “ when preaching ” **ብሰብኣይ** meaning to “ with a man ” over stemmed to “**ሰብ**”, as a result of which both words are indexed as the same word located in different document. This situation in turn misrepresents two different words as similar and happened irrelevant documents to be retrieved.

As one of the goals of this prototype is to provide relevant Amharic documents for a given Tigrigna query in addition to the Tigrigna results, the Tigrigna queries have to be translated in to Amharic. Figure 4.5, depicts python code implementation of query terms translation.

The code accepts Tigrigna query then it opens the dictionary file to search for Amharic meanings for the Tigrigna query words. If it finds equivalent meaning it upends the equivalent Amharic word to the Amharic query word list. If it doesn’t, it upends the original Tigrigna query word in to the Amharic query word list. Finally the query is passed under different text operations.

```
    query=input("ኣንታ·ይ ይደልዩ/ይደልያ፡    ")

    while len(query)==0:
      query= input(" ዝደልይዎ ሓበሬታ· ይጻሓፉ: ")
    fstring1=remove_puc(query)
    fstring1=re.sub('[፡።.!:¤()"፣?/!\ufeff]',",query)
    fstring1=Tnormaliz(fstring1)
    qv_List=[]
    qt_List=[]
    qt_List=fstring1.split(" ")
############ meaning #################
    dic=open("dictionary.txt",encoding='utf-8')
    am=[]
    tig=[]
    ii=0
    while(ii < 5530):
       lin=dic.readline()
       lin=lin.replace('\n',")
       t=lin.split('\t')
       if len(t)>1:
             am.append(t)
       ii+=1
    dic.close()

    for i in range(len(am)):
          am[i][0]=Anormaliz(am[i][0])
          am[i][1]=Tnormaliz(am[i][1])

    aall=[]
    c=0
    for i in range(len(qt_List)):
              c=0
              for j in range(len(am)):
                    if qt_List[i]==am[j][1]:
                          aall.append(am[j][0])
                          c+=1
                          break

              if c==0:
                    aall.append(qt_List[i])
```

*Figure 4.5: python code implementation for query terms translation*

The main challenges that the researcher faced word translation is discussed

- Word variant

As discussed so far the morphological structure of Tigrigna and Amharic words is complex. For example "መፀ" and "መፂኡ" have the same meaning in Amharic

““መጣ“” meaning ‘he comes’ . But as the dictionary is suppose to contain word to word translation such kind of words were considered as unknown words.

- Polysemous

Depending on the context of the statement that contains the word, the same word can get several meanings in the target language. For example, “**ፅንዓትን ምርምርን**” and “**ፅንዓትን ተወፋይነትን**” are two different phrases that uses “**ፅንዓትን**“ for different purpose. In the first phrase it has a meaning “**ጥናት**” in Amharic and “research” in English while in the second phrase it means “**ጥንካሬ**” in Amharic and “strength” in English.

- Synonyms

Different words may represent similar ideas. For example direct translation of “**ከይዱ**” is “ **ሂደ**” in Amharic and “gone” in English similarly “**ተመሊሱ**” is “**ተመለሰ**” in Amharic and “got back” in English but some times “**ተመሊሱ**” and “**ሂደ**” may represent the same meaning as in the following “**ናብ ዓዱ ተመሊሱ**” and “**ወደ ሃገሩ ሂደ**” both statements say “he went back to his country”.

## 4.4. System Evaluation Method

As discussed in section 3.5, once the proposed information retrieval system is designed and developed it is essential to carry out its evaluation. Evaluation of IR system can be examined from the view point of its effectiveness and efficiency. The type of the evaluation considered depends on the objectives of the retrieval system, even though complete evaluation process requires evaluation of both system effectiveness and efficiency. The objective of this research is to examine the ability of dictionary-based approach (one type of query translation approach of CLIR) for Tigrigna - Amharic CLIR. In addition, retrieval effectiveness of a system is evaluated on a given set of documents, queries and relevance judgments. Therefore, for this research only effectiveness of IR system is taken into consideration to determine the performance of the system for the baseline and translated queries.

There are different techniques for measuring effectiveness of IR system. In many IR

systems recall, precision and F-measure are considered as basic measures for retrieval effectiveness and majority of the studies used these measurements [18]. For this research also these techniques are used to evaluate the performance effectiveness of the system.

Recall measures the ability of a retrieval system to find out relevant documents. It considers how many percentages of relevant documents are correctly retrieved by the system. Using recall alone is not enough when measuring the effectiveness of a retrieval system since retrieved documents contains both relevant and irrelevant documents. Therefore, precision which measures the ability of a retrieval system to find out only relevant documents is also essential. In addition F-measure is used to measure the performance of the system since it balances the precision and recall values.

Retrieval effectiveness of relevant Amharic documents depends on the translation performance of Tigrigna queries in to Amharic. To this end three different ways of translations is evaluated. These are translation before Tigrigna query terms are stemmed, translation after Tigrigna query terms are stemmed, and translating stemmed Tigrigna query terms using stemmed dictionary. Table 4.2 depicts summery of the translation performances.

| QNo | Translation before stemming | Translation after stemming | Translation after stemming using stemmed dictionary |
|---|---|---|---|
| q1 | 100% | 0% | 100% |
| q2 | 80% | 0% | 80% |
| q3 | 100% | 71% | 100% |
| q4 | 80% | 80% | 80% |
| q5 | 83% | 17% | 83% |
| q6 | 100% | 25% | 100% |
| Average | 91% | 32% | 91% |

Table 4.2: Translation performances for different ways of translation

As shown in table 4.2, translation before stemming and translation after stemming using stemmed dictionary registered exactly the same translation performance but Translation after stemming registered low translation performance.

### 4.4.1. Evaluation for Translation before stemming

#### 4.4.1.1 Before Relevance Feedback

Using the information given in table 4.3, evaluation is done by measuring the recall, precision and F-measure of each test queries and the average of each queries result are obtained as the initial performance of the system. Table 4.3 presents relevant documents retrieved from the Tigrigna and Amharic corpus for each test query.

| Query No | List of queries | relevant documents for each query | Relevant documents retrieved |
|---|---|---|---|
| 1 | ኢትዮጵያ ናይ መልቐቒ ሰርትፍኬት ፈተና | 20, 22, 85, 82, 41, 21, 36, 86, 90, 92, 95 | 20, 22, 21, 90 |
| | ኢትዮጵያ መልቀቂያ ሰርትፍኬት ፈተና | 1, 10, 11, 76, 63, 80, 84, 94, 7, 25, 31, 56 | 9, 10, 1, 11, 76 |
| 2 | ዝለዓለ ዕርየት ዘለዎ ናይ መድሓኒት ኣቕርቦት | 14, 99, 12, 84, 6, 16, 13, 17, 82, 19, 4, 29 | 14, 12, 13, 6, 84, 99 |
| | ከፍተኛ ንጽህና ያለው መድሃኒት አቅርቦት | 4, 7, 2, 5, 67, 65, 97, 9, 10, 11, 34, 72, 63, 76, 80, 94 | 2, 65, 72, 97, 4 |
| 3 | መልኣኽ መለኽት ነፍሔ ኰነ፡ብ ድማ ናብ መሬት ወረደ | 42, 52, 43, 54, 55, 48, 44, 67, 95 | 42, 43, 44, 52, 54, 55 |
| | መልአክ መለከት ነፋ ኮከብ መሬት ወረደ | 31, 30, 24, 22, 77, 43, 8, 28, 19, 60, 61, 93 | 8, 22, 24, 28, 30, 31 |
| 4 | መጻጉዕ ሰብኣይ ብእምነቱ ሓጢኣቱ ተሐዲጉሉ | 60, 68, 71, 59, 62, 98 | 60, 62, 68, 71, 59 |
| | ሽባ ሰው በእምነቱ ሃጢኣቱ ተተወለት | 35, 45, 34, 36, 46, 56, 61 | 35, 36, 45, 46, 34 |
| 5 | ቅዱሳት ዕላማታት ህዝብና ኣብ መዕርፎኣም ከም ዝበጽሑ ንተኣማመን | 82, 86, 84, 91, 89, 96, 97, 46, 51, 74, 90 | 82, 84, 89, 91, 96, 86 |
| | የተቀደሱ ኣላማ ህዝብ መዕርፍ የሚደርሱ እንተማመን | 91, 92, 63, 64, 16, 66, 79, 83 | 91, 92 |
| 6 | ብዝበፅሐ ግምዐ መሬት ነዝተፈናቐለ ሰባት ተወሳኪ መሬት ተዋሂቡ | 42, 48, 28, 16, 69, 44, 82, 51, 55, 27, 10 | 42, 48 |
| | በደረሰው ናዳ መሬት ለተፈናቀሉት ሰዎች ተጨማሪ መሬት ተሰጠ' | 62, 63, 69, 70, 74, 75, 73, 86, 72, 79, 42, 65, 83, 94, 44, 19, 28, 56, 22, 49, 21 | 62, 63, 70, 74, 69 |

*Table 4.3: Tigrigna Test queries, Amharic test queries translated by the system using the digital Tigrigna - Amharic dictionary relevant documents list and ranked list of retrieved documents.*

| QNo | List of queries | Relevant | Retrieved | Relevant Retrieved | Precision | Recall | F-Measure |
|---|---|---|---|---|---|---|---|
| 1 | ኢትዮጵያ ናይ መልቐቒ ሰርትፍኬት ፈተና | 6 | 11 | 4 | 0.36 | 0.67 | 0.47 |
| 2 | ዝለዓለ ፅርየት ዘለዎ ናይ መድሓኒት ኣቕርቦት | 7 | 12 | 6 | 0.5 | 0.86 | 0.63 |
| 3 | መልኣኽ መለኽት ነፍሔ ኰኸብ ድማ ናብ መሬት ወረደ | 7 | 9 | 6 | 0.67 | 0.86 | 0.75 |
| 4 | መጻጉዕ ሰብኣይ ብእምነቱ ሓጢኣቱ ተሐዲጉሉ | 6 | 6 | 5 | 0.83 | 0.83 | 0.83 |
| 5 | ቅዱሳት ዕላማታት ህዝብና ኣብ መዕርፎኦም ክም ዝበጽሑ ንተኣማመን | 6 | 11 | 6 | 0.55 | 1 | 0.71 |
| 6 | ብዝበፅሐ ግምዐ መሬት ነዝተፈናቐሉ ሰባት ተወሳኪ መሬት ተዋሂቡ | 2 | 11 | 2 | 0.18 | 1 | 0.31 |
| | | | | **Average** | **0.52** | **0.87** | **0.62** |

*Table 4.4 shows the effectiveness of the probabilistic Tigrigna IR system based on 6 queries selected for the experiment.*

| QNo | List of queries | Relevant | Retrieved | Relevant Retrieved | P | R | F |
|---|---|---|---|---|---|---|---|
| 1 | ኢትዮጵያ የ መልቀቂያ ሰርትፍኬት ፈተና | 5 | 14 | 5 | 0.35 | 1 | 0.52 |
| 2 | ከፍተኛ ንጽህና ያለው የ መድሃኒት አቅርቦት | 5 | 16 | 5 | 0.31 | 1 | 0.47 |
| 3 | መልአክ መለከት ነፋ ነከብ ድማ ወደ መሬት ወረደ | 7 | 12 | 6 | 0.5 | 0.86 | 0.63 |
| 4 | ሸባ ሰው በእምነቱ ሃጢኣቱ ተተወለት | 6 | 7 | 5 | 0.71 | 0.83 | 0.77 |
| 5 | የተቀደሱ ኣላማዎች ህዝባችን በ ማረፊያቸው እንደ የሚደርሱ እንተማመን | 2 | 8 | 2 | 0.25 | 1 | 0.4 |
| 6 | በደረሰው ናዳ መሬት ለተፈናቀሉት ሰዎች ተጨማሪ መሬት ተሰጠ | 5 | 21 | 5 | 0.24 | 1 | 0.39 |
| | | | | **Average** | **0.39** | **0.95** | **0.53** |

*Table 4.5 shows the effectiveness of the probabilistic Amharic IR system based on 6 translated queries selected for the experiment.*

As shown in table 4.4, the average result of precision and recall for Tigrigna using the initial guess made by the model about the relevance of documents is 52% and 87% respectively. On the other hand, as shown in table 4.5, the average result of precision and recall for Amharic using the initial guess is 39% and 95% respectively. The F-measure score registered 62% for Tigrigna and 53% for Amharic, which indicates the performance of the system is not good. This is because; documents containing one of the query terms but that are not-relevant are retrieved. These documents are irrelevant because, the query term found in those documents does not express the meaning of the query with respect to other terms found in the query. For example, for query "**ኢትዮጵያ ናይ መልቐቒ ሰርትፍኬት ፈተና**" which express the high school leaving certificate examination , the system retrieved irrelevant

documents such as “Tgfile82”, “Tgfile86”, “Tgfile36” and “Tgfile95” because they contains query term “ፈተና”. However, in these documents the term “ፈተና“ is used to expresses financial crises in Ethiopia which are not related with examination. The same problem is also revealed for other queries. On the other hand, in probabilistic model the initial guess of relevant document is based on Boolean expression. Thus, all terms that match one of user queries are retrieved which increases the number of denominator used for calculating precision, thereby decreasing the percentage of precision.. Therefore, in order to increase the performance of the system, the probabilistic model uses either user relevance feedback or pseudo relevance feedback so as to apply query terms re-weighting in order to increase the weight of terms found in relevant documents and decrease the weight of terms found in non-relevant documents.

#### 4.4.1.2 After Relevance Feedback

As can be seen on the architecture of the prototype in figure 3.1 both user relevance feedback and pseudo relevance feedback are tested. Table 4.6 and Table 4.7 depicts summery of recall and precision of the final result for Tigrigna queries after relevance feedback is applied,

| QNo | Recall | | | Precision | | |
|---|---|---|---|---|---|---|
| | Recall before Relevance Feedback | After user relevance Feedback | After Pseudo Relevance Feedback | Precision before Relevance Feedback | After user relevance Feedback | After Pseudo Relevance Feedback |
| q1 | 0.67 | 0.5 | 0.5 | 0.36 | 0.5 | 0.5 |
| q2 | 0.86 | 0.86 | 0.86 | 0.5 | 0.67 | 0.75 |
| q3 | 0.86 | 0.86 | 0.86 | 0.67 | 0.67 | 0.75 |
| q4 | 0.83 | 0.83 | 0.83 | 0.83 | 1 | 1 |
| q5 | 1 | 1 | 1 | 0.55 | 0.67 | 0.67 |
| q6 | 1 | 1 | 1 | 0.18 | 1 | 1 |
| average | **87%** | **84%** | **84%** | **52%** | **75%** | **78%** |

*Table 4.6 summery of recall and precision for Tigrigna queries after relevance feedback is applied,*

| QNo | Recall | | | Precision | | |
|---|---|---|---|---|---|---|
| | Recall before relevance feedback | After user relevance Feedback | After Pseudo Relevance Feedback | Precision before relevance feedback | After user relevance Feedback | After Pseudo Relevance Feedback |
| q1 | 1 | 1 | 1 | 0.35 | 0.62 | 1 |
| q2 | 1 | 0.86 | 0.6 | 0.31 | 0.8 | 0.5 |
| q3 | 0.86 | 0.86 | 0.86 | 0.5 | 0.6 | 0.75 |
| q4 | 0.83 | 0.83 | 0.33 | 0.71 | 0.83 | 1 |
| q5 | 1 | 1 | 0 | 0.25 | 0.67 | 0 |
| q6 | 1 | 1 | 1 | 0.24 | 0.31 | 0.31 |
| **Average** | **95%** | **93%** | **63%** | **39%** | **64%** | **59%** |

*Table 4.7 summery of recall and precision for Amharic queries after relevance feedback is applied,*

| | F-measure before Relevance Feedback | F-measure after User Relevance Feedback | F-measure after Pseudo Relevance Feedback |
|---|---|---|---|
| Tigrigna | 62% | 79% | 80% |
| Amharic | 53% | 73% | 55% |

*Table 4.8: summery of F-measure before and after Relevance Feedback*

As indicated in Table 4.6 and Table 4.7, both types of relevance feedbacks registered exactly the same recall value for Tigrigna documents. Though the difference is not significant, the pseudo relevance feedback registered better precision value than that of user relevance feedback. On the other hand, on Amharic documents the user relevance feedback has significantly dominated the pseudo relevance feedback.

As it is shown in Table 4.6 and Table 4.7, the system registered better performance when the user provides relevance feedback. However Recall is decreased from 87% to 84% for Tigrigna queries and decreased from 95% to 93% for Amharic queries, Precision is increased from 52% to 75% for Tigrigna queries and increased from 39% to 64% for Amharic queries.

### 4.4.2. Evaluation for Translation After stemming

| | Translation performance | Language similarity | Recall | Precision | F-measure |
|---|---|---|---|---|---|
| q1 | 0% | 75% | 100% | 100% | 100% |
| q2 | 0% | 20% | 0% | 0% | 0% |
| q3 | 71% | 43% | 29% | 100% | 44% |
| q4 | 80% | 40% | 33% | 100% | 50% |
| q5 | 17% | 17% | 0% | 0% | 0% |
| q6 | 25% | 25% | 100% | 50% | 67% |
| average | 32% | 37% | 44% | 58% | 44% |

Table 4.9: Experiment detail for translation after stemming

## 4.5. Findings and Challenges

The MRBD is used as a tool for translating query terms in this research. The translation for the given sample queries registered promising performance. However some challenges are seen that makes working with dictionary base approach difficult. The following are some of them:

**Translations are made before stemming**

All stemmed words may not give meanings. The presence of infix is one of the potential reasons to produces meaningless words. Tigrigna and Amharic languages contain considerable amount of infix. So when translation is tried after stemming a query terms could not get their corresponding meaning. To this end, the researcher has found difficult translating words after stemming.

**Written language is highly influenced by writer's accent**

Especially in Tigrigna written language is highly influenced by writer's accent. We can consider the case of Eritrean and Ethiopian Tigrigna. For example Table 4.7 depicts some list of Tigrigna words along with their Amharic meaning.

| Tigray | Eritrea | Amharic |
|---|---|---|
| ሓዝ | ሕዝ | ያዝ |
| ይክእል | እክእል | ይቸላል |
| በለዐ | በለጌ | በላ |

*Table 4.10: List of similar words with different accent*

**Word variant**

The morphological structure of Tigrigna and Amharic words is complex[8][39]. For example "**መፀ**" and "**መፂኡ**" have the same meaning in Amharic ""**መጣ**"" meaning 'he comes' . But, unless and otherwise a huge MRBD that contains all words and word variants along with their translation is developed, as the dictionary is supposed to contain only root word to root word translation, such kind of words are considered as unknown words.

**Polysemous**

Depending on the context of the statement, the same word can be attributed to several meanings in the target language. For example "**ፅንዓትን ምርምርን**" and "**ፅንዓትን ተወፋይነትን**" are two different phrases that uses "**ፅንዓትን**" for different purpose in the first phrase is has a meaning "**ጥናት**" in Amharic and "research" in English while in the second phrase it means "**ጥንካሬ**" in Amharic and "strength" in English. Such kind of words biased the translation process.

**Synonyms**

Different words may represent similar ideas, For example direct translation of "**ከይዱ**" is " **ሂደ**" in Amharic and "gone" in English similarly "**ተመሊሱ**" is "**ተመለሰ**" in Amharic and "got back" in English but some times "**ተመሊሱ**" and "**ሂደ**" may represent the same meaning like in the following case "**ናብ ዓዱ ተመሊሱ**" and "**ወደ ሃገሩ ሂደ**" both statements say "he went back to his country".

Another reason for the low performance of the bilingual run (Amharic documents retrieval using Tigrigna queries) was the incorrect translation of the Tigrigna queries

into Amharic. A given test document can be represented by either single word or multiple words of query. For the multiple words of query the translation could be completely correct (i.e. all words of query are correctly translated), partly correct (i.e. not all words of query are correctly translated) or all words in a query are wrongly translated. For example, "**ዝለዓለ ፅርየት ዘለዎ ናይ መድሓኒት ኣቕርቦት**" is translated to "**ከፍተኛ ንጽህና ያለው የመድሃኒት አቅርቦት**". The Tigrigna query discusses about quality of medicine while the Amharic query discuss about clean medicine.

# CHAPTER FIVE
# CONCLUSION AND RECOMMENDATIONS

## 5.1. Conclusion

Cross Lingual Information Retrieval (CLIR) system helps a user to pose a query in one language and retrieve documents in another language. In this study, Tigrigna - Amharic CLIR that is based on Dictionary Approach was developed for Tigrigna users to specify their information need in their native language and to retrieve documents in both Tigrigna and Amharic. Performance of CLIR systems using dictionary based approach is highly dependant either on the performance of the stemmer, if translation is to be carried out after stemming, or on the size of the MRBD, if translation is to be carried out before stemming.

For the reason that the stemmer used for this research was incapable of producing meaningful stemmed words, the researcher used translation before stemming. The other challenge that the researcher faced is, the dictionary used for this research was limited in size. Those stated problems affected the level of performance to be achieved.

Even though the dictionary-based approach is highly dependant on the size of the dictionary, the result obtained is encouraging to develop Tigrigna-Amharic CLIR by using this approach. An average F-measure of 0.79 and 0.73 for Tigrigna and Amharic is obtained respectively.

The low performance achieved for the bilingual run (Amharic document retrieval by using Tigrigna) was because of incorrect query term translation which affects the accuracy of the query translation. This incorrect translation was caused due to the word variants. Thus, it can be concluded that the performance of the system obtained was encouraging given the limited size of the MRD used.

## 5.2. Recommendations

It is believed that there is much room for improvement of the performance of Tigrigna-Amharic CLIR system developed in this research. Therefore, the following recommendations should be looked at in the future so that effective Tigrigna-Amharic dictionary-based CLIR system can be developed to help users in their information need:

- Since the capability of the stemmer developed so far was not found to be efficient, translation before stemming was used. But getting a dictionary that encompasses the meaning of all words and word variants is challenging. Therefore an efficient stemmer should be developed.

- Stemmed words may not always be meaningful as a result stemmed words may not have matching term in the dictionary while translation is performed. Hence research should be carried out on lemmatization.

- The proposed system implemented with word-based query translation, which reduces the effectiveness of query translation. Therefore Use of bilingual phrases instead of single words significantly improves translation quality. The study conducted by Bian [32] indicates that effectiveness of phrasal translation enhances performance by ~ 14 - 31 % than the word level translation. Therefore, it is recommended to test the achievement of phrasal translation techniques.

- Query expansion technique improves performance of retrieval systems by including synonymous terms in search. Therefore, it is necessary to see the effect of query expansion on the retrieval performance of the system.

## References:


[1] A. Pirkola, T. Hedlund, H. Keskustalo, and K. Järvelin, "Dictionary-Based Cross-Language Information Retrieval: Problems, Methods, and Research Findings", paper, Department of Information Studies, University of Tampere, Finland, 4(3/4):209-230, 2001.

[2] D. N. Gearailt, "Dictionary characteristics in cross-language information retrieval", Technical Report, University of Cambridge, United Kingdom, UCAM-CL-TR-616, ISSN 1476-2986, Number 616, 2005.

[3] M. Aljlayl, O. Frieder, and D. Grossman, "On Bidirectional English-Arabic Search", Journal of the American Society for Information Science and Technology, 53(13):1139-1151, 2002.

[4] D. Bekele, "Afaan Oromo-English Cross-Lingual Information Retrieval (CLIR): A Corpus based approach", MSc Thesis, School of Information Science, Addis Ababa University, 2011.

[5] A. Hirpa, "Probabilistic Information Retrieval for Amharic Language", MSc Thesis, School of Information Science, Addis Ababa University, 2012.

[6] A. Tesfaye. "Amharic-English cross lingual information retrieval (CLIR): A corpus based approach", MSc Thesis, School of Information Science, Addis Ababa University, 2009.

[7] A. Alemu, L. Aske, "Dictionary Based Amharic English Information Retrieval", In CLEF 2004 Bilingual Task, Stockholm University/KTH, Sweden, 2004.

[8] Y. Fisseha, "Development of Stemming Algorithm for Tigrigna Text", MSc Thesis, School of Information Science, Addis Ababa University, 2011.

[9] G. Assefa, "A Two Step Approach for Tigrigna Text Categorization", MSc Thesis, School of Information Science, Addis Ababa University, 2011.

[10] X. Saralegi and M. L. D. Lacalle. "Comparing different approaches to treat Translation Ambiguity in CLIR: Structured Queries vs. Target Co-occurrence Based Selection", Journal, DEXA WORKSHOPS, 2009.

[11] M Braschler, C. Peters, and P. Schäuble," Cross-Language Information Retrieval (CLIR) Track Overview", Article, Eurospider Information Tech. AG, Schaffhauserstr, 18, CH-8006 Zürich, Switzerland.

[12] D. W. Oard, "Multilingual Information Access," Article, in Encyclopedia of Library and Information Sciences, 3rd Ed., edited by Marcia J. Bates, Editor, and Mary Niles Maack, Associate Editor, Taylor & Francis, 2009.

[13] C. Cheng and R. Shue, H. Lee, “Evaluations for Multilingual and Cross-Lingual Information Retrieval”, Article, paper, Department of Information Management Huafan University, Taiwan, 2007.

[14] T. Hedlund, “Dictionary-Based Cross-Language Information Retrieval”, MSc Thesis, Department of Information Studies, University of Tampere, Finland, 2003.

[15] T. Talvensaari, “Comparable Corpora in Cross-Language”, MSc Thesis, department of computer sciences, university of Tampere Finland, 2008.

[16] L. Ballesteros and W. B. Croft, “Phrasal Translation and Query Expansion Techniques for Cross-Language Information Retrieval”, Article, Center for Intelligent Information Retrieval, Computer Science Department, University of Massachusetts, Amherst, MA 01003-4610 USA, 1997.

[17] M. Abusalah, J. Tait, and M. Oakes, “Literature Review of Cross Language Information Retrieval”, Article, World Academy of Science, Engineering and Technology, 2005.

[18] Y.Y. Yao, “Measuring Retrieval Effectiveness Based on User Preference of Documents”, MSc Thesis, Department of Mathematical Sciences, Lakehead University, Thunder Bay, Ontario Canada P7B 5E1, 1999.

[19] D. Hiemstra, “Information Retrieval Models, “Information Retrieval: Searching in the 21st Century”, John Wiley and Sons, Ltd, University of Twente, ISBN-13: 978-0470027622, 2009.

[20] A. Singhal, “Modern Information Retrieval: A Brief Overview”, Article, Google, Inc, 2002.

[21] C.J. Rijsbergen, “Information Retrieval”, Butterworth-Heinemann publisher, USA, 2nd edition, 1979.

[22] N A. and P. Willett, “Stemming of Amharic Words for Information Retrieval”, in Literary and Linguistic Computing. Oxford University press, Oxford London, 17(1):1-18, 2002.

[23] M.J. Gerard Salton, Introduction to Modern Information Retrieval, Available at: http://www.sifaka.cs.uiuc.edu/course/410s12/mir.pdf, Accessed Date; 12/2/2013, 2013.

[24] B. R. Ribeiro Neto, "Modern Information Retrieval", 2nd Edition, Addison-Wesley-Longman Publishers, New York, USA, 1999.

[25] "Informaiton Retrieval Model", Available at; http://comminfo.rutgers.edu /~aspoerri/InfoCrystal/ Ch_2.htm, Accessed Date: 12/1/2013, 2013.

[26] Ed Greengrass, "Information Retrieval: A Survey", Available at: www.csee.umbc.edu/cadip/readings/IR.report.120600.book.pdf, Accessed Date: 12/1/2013, 2013.

[27] H.R. Turtle and W.B. Croft, "A Comparison of Text Retrieval Models", The Computer Journal, 35(2): 279-290, 1992.

[28] Mark A. Greenwood, “Implementing a Vector Space Document Retrieval System”, Dept. of Computer Science, University of Sheffield Regents Court, 211 Portobello St, Sheffield. S1 4DP. UK.

[29] N. Fuhr, "Probabilistic Models in Information Retrieval", The Computer Journal, 35(3):243-255, 1992.

[30] S.W. K. Sparck Jones, "A probabilistic Model of Information Retrieval: Development And Comparative Experiments", Information Processing and Management, 36(4): 779-808, 2000.

[31] N Fuhr, “Probabilistic Models in Information Retrieval”, Journal, Oxford university press, Oxford, UK, pp 243-245, 1992.

[32] G. Bian, and H. Chen, “New Hybrid Approach for Chinese-English Query Translation”, Proceedings of the First Asia Digital Library Workshop, pp. 156-167, 1998.

[33] IJCNLP, “The Third International Joint Conference on Natural Language Processing", Article, Asian Federation of Natural Language Processing, International Institute of Information Technology, Hyderabad, India, 2008 .

[34] “Population and Housing Census of Ethiopia”, Available at: http://www.gcao.gov.et/index.php?option=com_jdownloads&Itemid=218&view=finish&cid=37&catid=12&m=0&lang=en , Accessed Date; 12/3/2013, 2007.

[35] Y. Shipeng, and C. Deng, W. Ji-Rong and M. Wei-Ying, “Improving Pseudo-Relevance Feedback in Web page Segmentation”, Proceedings of the 12th International Conference on World Wide Web, 1(1):11 - 18, 2003.

[36] F. Arnaud and D. Renata, “Rocciho's Relevance feedback Algorithm in Basic Vector Space Comparison and LSI Models”, Available at; http://www.mpi-inf.mpg.de /projectproposals/renata_dividino_arnaud_fietzke.pdf, Accessed date: 15/4/2013, 2013.

[37] K. Kula, V. Varma, and P. Pingali, “Evaluation of Oromo-English Cross-Language Technologies Research Center”, Article, Information Retrieval IIIT, Hyderabad, India. 2008.

[38] A. Sisay, “English-Oromo Machine Translation: An Experiment Using a Statistical Approach”, M.Sc Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2009.

[39] A. Nega and P. Willett, "Stemming of Amharic Words for Information Retrieval", in Literary and Linguistic Computing. Oxford University press, Oxford London, 17(1):1-18, 2002.

[40] T. Bloor, "The Ethiopic Writing System: a Profile", Journal of the Simplified Spelling Society, 19(1):30-36, 1995.

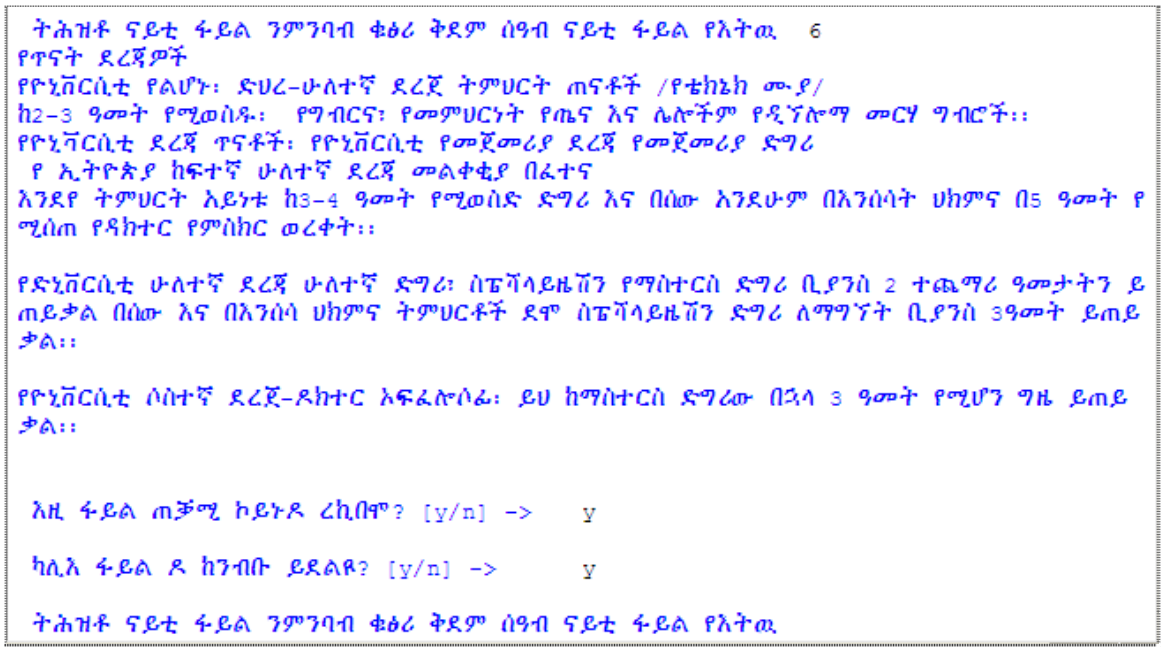
[41] “Research design”, Available at: http://en.wikipedia.org/wiki/Research_design, Accessed Date; 12/5/2013, 2013.

# APPENDIX

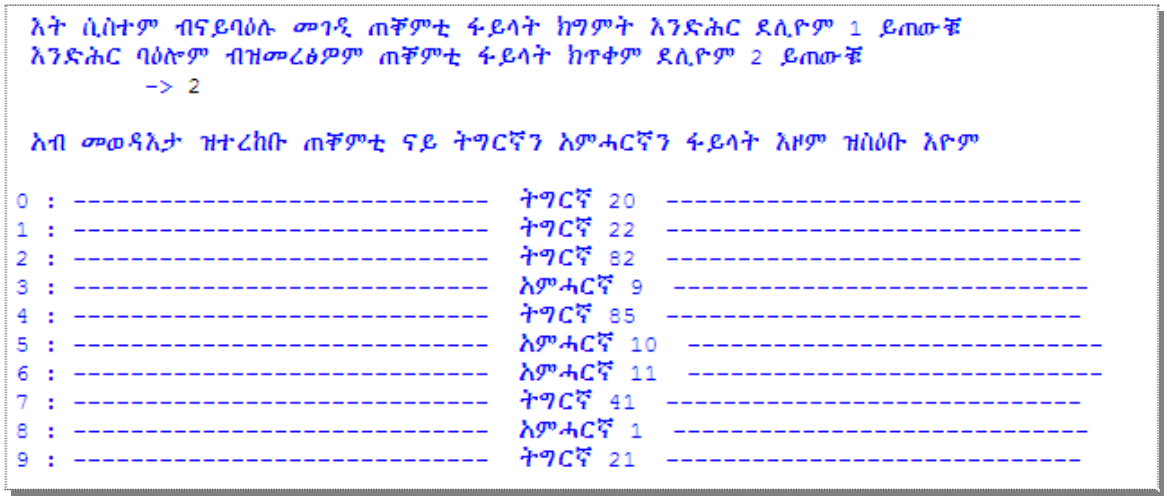
## Appendix 1: Tigrigna and Amharic character set

| | | | | | | |
|---|---|---|---|---|---|---|
| ሀ he | ሁ hu | ሂ hi | ሃ ha | ሄ hie | ህ h | ሆ ho |
| ለ le | ሉ lu | ሊ li | ላ la | ሌ lie | ል l | ሎ lo |
| ሐ He | ሑ Hu | ሒ Hi | ሓ Ha | ሔ Hie | ሕ H | ሖ Ho |
| መ me | ሙ mu | ሚ mi | ማ ma | ሜ mie | ም m | ሞ mo |
| ሠ se | ሡ su | ሢ si | ሣ sa | ሤ sie | ሥ s | ሦ so |
| ረ re | ሩ ru | ሪ ri | ራ ra | ሬ rie | ር r | ሮ ro |
| ሰ se | ሱ su | ሲ si | ሳ sa | ሴ sie | ስ s | ሶ so |
| ሸ xe | ሹ xu | ሺ xi | ሻ xa | ሼ xie | ሽ x | ሾ xo |
| ቀ qe | ቁ qu | ቂ qi | ቃ qa | ቄ qie | ቅ q | ቆ qo |
| ቐ Qe | ቑ Qu | ቒ Qi | ቓ Qa | ቔ Qie | ቕ Q | ቖ Qo |
| በ be | ቡ bu | ቢ bi | ባ ba | ቤ bie | ብ b | ቦ bo |
| ቨ ve | ቩ vu | ቪ vi | ቫ va | ቬ vie | ቭ v | ቮ vo |
| ተ te | ቱ tu | ቲ ti | ታ ta | ቴ tie | ት t | ቶ to |
| ቸ ce | ቹ cu | ቺ ci | ቻ ca | ቼ cie | ች c | ቾ co |
| ኀ he | ኁ hu | ኂ hi | ኃ ha | ኄ hie | ኅ h | ኆ ho |
| ነ ne | ኑ nu | ኒ ni | ና na | ኔ nie | ን n | ኖ no |
| ኘ Ne | ኙ Nu | ኚ Ni | ኛ Na | ኜ Nie | ኝ N | ኞ No |
| አ ae | ኡ au | ኢ ai | ኣ aa | ኤ aie | እ a | ኦ ao |
| ከ ke | ኩ ku | ኪ ki | ካ ka | ኬ kie | ክ k | ኮ ko |
| ኸ Ke | ኹ Ku | ኺ Ki | ኻ Ka | ኼ Kie | ኽ K | ኾ Ko |
| ወ we | ዉ wu | ዊ wi | ዋ wa | ዌ wie | ው w | ዎ wo |
| ዐ eee | ዑ eeu | ዒ eei | ዓ eea | ዔ eeie | ዕ ee | ዖ eeo |
| ዘ ze | ዙ zu | ዚ zi | ዛ za | ዜ zie | ዝ z | ዞ zo |
| ዠ Ze | ዡ Zu | ዢ Zi | ዣ Za | ዤ Zie | ዥ Z | ዦ Zo |
| የ ye | ዩ yu | ዪ yi | ያ ya | ዬ yie | ይ y | ዮ yo |
| ደ de | ዱ du | ዲ di | ዳ da | ዴ die | ድ d | ዶ do |
| ጀ je | ጁ ju | ጂ ji | ጃ ja | ጄ jie | ጅ j | ጆ jo |
| ገ ge | ጉ gu | ጊ gi | ጋ ga | ጌ gie | ግ g | ጎ go |
| ጠ Te | ጡ Tu | ጢ Ti | ጣ Ta | ጤ Tie | ጥ T | ጦ To |
| ፈ Ce | ፉ Cu | ፊ Ci | ፋ Ca | ፌ Cie | ፍ C | ፎ Co |
| ጰ Pe | ጱ Pu | ጲ Pi | ጳ Pa | ጴ Pie | ጵ P | ጶ Po |
| ጸ Se | ጹ Su | ጺ Si | ጻ Sa | ጼ Sie | ጽ S | ጾ So |
| ፀ Se | ፁ Su | ፂ Si | ፃ Sa | ፄ Sie | ፅ S | ፆ So |
| ፈ fe | ፉ fu | ፊ fi | ፋ fa | ፌ fie | ፍ f | ፎ fo |
| ፐ pe | ፑ pu | ፒ pi | ፓ pa | ፔ pie | ፕ p | ፖ po |

**Appendix 2: Screen shot of the prototype CLIR system when it displays content of a file and ask the user whether the file is relevant or not**




**Appendix 3: Screen shot of the prototype CLIR system when it asks the user to select type of relevance feedback and displaying the final list of relevant documents base on the user's choice**



**Appendix 4: Probabilistic Tigrigna-Amharic CLIR system python code**

```
print("\t\t\t"'እንኳዕ ናብ ማእከል ሓበሬታ ብዳሓን መፀ!!')
print("\n")
print("\t"'ብናይ ትግርኛ መሕትት ጠቀምቲ ናይ ትግርኛን አምሓርኛን ፋይላት ይርከቡ ')

import re
import math
import string
import os
def no_docs():
	docs=','
	for i in os.listdir('tdct'):
		docs=docs+','+i
	docs=docs.replace(',,',')
	doc_List=docs.split(",")
	return doc_List
########################## Amharic ##########################
def ano_docs():
	docs=','
	for i in os.listdir('adct'):
		docs=docs+','+i
	docs=docs.replace(',,',')
	doc_List=docs.split(",")
	return doc_List

########################## tig ##################################
def Tnormaliz(text):
	h1=["ሀ","ሁ","ሂ","ሃ","ሄ","ህ","ሆ"]
	h3=["ሐ","ሑ","ሒ","ሓ","ሔ","ሕ","ሖ"]
	s1=["ሰ","ሱ","ሲ","ሳ","ሴ","ስ","ሶ"]
	s2=["ሠ","ሡ","ሢ","ሣ","ሤ","ሥ","ሦ"]
	t1=["ጸ","ጹ","ጺ","ጻ","ጼ","ጽ","ጾ"]
	t2=["ፀ","ፁ","ፂ","ፃ","ፄ","ፅ","ፆ"]
	for i in range(len(h1)):
		text=text.replace(h3[i],h1[i])
		text=text.replace(t2[i],t1[i])
		text=text.replace(s2[i],s1[i])
```

```
        return text
def Anormaliz(text):
        h1=['ሀ','ሁ','ሂ','ሃ','ሄ','ህ','ሆ']
        h2=['ሐ','ሑ','ሒ','ሓ','ሔ','ሕ','ሖ']
        h3=['ኀ','ኁ','ኂ','ኃ','ኄ','ኅ','ኆ']
        a1=['አ','ኡ','ኢ','ኣ','ኤ','እ','ኦ']
        a2=['ዐ','ዑ','ዒ','ዓ','ዔ','ዕ','ዖ']
        s1=['ሰ','ሱ','ሲ','ሳ','ሴ','ስ','ሶ']
        s2=['ሠ','ሡ','ሢ','ሣ','ሤ','ሥ','ሦ']
        t1=['ጸ','ጹ','ጺ','ጻ','ጼ','ጽ','ጾ']
        t2=['ፀ','ፁ','ፂ','ፃ','ፄ','ፅ','ፆ']
        for i in range(len(h1)):
                text=text.replace(h3[i],h1[i])
                text=text.replace(h2[i],h1[i])
                text=text.replace(t2[i],t1[i])
                text=text.replace(s2[i],s1[i])
                text=text.replace(a2[i],a1[i])
        return text
def remove_puc(v_String):
    lp=['[',']','{','}','.',',','?','፧','።','፤','፤፤',':','::',': ','፣','፡',':']
    v_String=re.sub('[??o=?.??():“,<>-``/”??"??!“" '\ufeff]','',v_String)
    for i in lp:
            v_String=v_String.replace(i,'')
    v_String= re.sub('[\d+]','', v_String)
    return v_String
def sufpre(v_List):
    prfx=open("Prefixlist.txt",encoding='utf-8')
    prefix=prfx.read()
    prefix=prefix.split()
    sfx=open("SuffixList.txt",encoding='utf-8')
    suffix=sfx.read()
    suffix=suffix.split()
    for n in range(0,len(v_List)):
        stemed_query=''
        stemed_query=stemed_query+v_List[n]
        for preffix_l in range(0,len(prefix)-1):
            if(len(stemed_query)>2):
                if(stemed_query.startswith(prefix[preffix_l])):
```

```
                stemed_query=stemed_query.replace(prefix[preffix_l],'')
                preffix_l=len(prefix)
        for suffix_l in range(0,len(suffix)-1):
            if(len(stemed_query)>2):
                if(stemed_query.endswith(suffix[suffix_l])):
                    stemed_query=stemed_query.replace(suffix[suffix_l],'')
                    suffix_l=len(suffix)
        v_List[n]=stemed_query
    sfx.close()
    prfx.close()
    return v_List
######################### am suf pri ##############################
def stemtq(text):
        tp=['ሀ','አ','ን']
        for i in tp:
            if text.startswith(i):
                text=text.replace(text[0:1],'')
        return text
def asufpre(v_List):
    prfx=open("prefix.txt",encoding='utf-8')
    prefix=prfx.read()
    prefix=prefix.split()
    sfx=open("sufix.txt",encoding='utf-8')
    suffix=sfx.read()
    suffix=suffix.split()
    for n in range(0,len(v_List)):
        stemed_query=''
        stemed_query=stemed_query+v_List[n]
        for preffix_l in range(0,len(prefix)-1):
            if(len(stemed_query)>2):
                if(stemed_query.startswith(prefix[preffix_l])):
                    stemed_query=stemed_query.replace(prefix[preffix_l],'')
                    preffix_l=len(prefix)
        for suffix_l in range(0,len(suffix)-1):
            if(len(stemed_query)>2):
                if(stemed_query.endswith(suffix[suffix_l])):
                    stemed_query=stemed_query.replace(suffix[suffix_l],'')
                    suffix_l=len(suffix)
```

```
            v_List[n]=stemed_query
        sfx.close()
        prfx.close()
        return v_List
    def Tmain():
        prfx=open("Prefixlist.txt",encoding='utf-8')
        prefix=prfx.read()
        prefix=prefix.split()
        prfx.close()
        sfx=open("SuffixList.txt",encoding='utf-8')
        suffix=sfx.read()
        suffix=suffix.split()
        sfx.close()
        dfile={}
        d_no=1
        stp_w=[]
        v_Index={}
        docu_List=[]
        docu_List=no_docs()
        d_no=len(docu_List)
        nsw=0
        stopw=open("StopwordList.txt",encoding='utf-8')
        while stopw.readline()!='':
            nsw= nsw+1
        stopw.close()
        stopw=open("StopwordList.txt",encoding='utf-8')
        for i in range(1,nsw):
            line=stopw.readline()
            line=line.rstrip()
            stp_w.append(line)
        stopw.close()
        for j in range(1,d_no+1):
            dline='tdct\\'+docu_List[j-1]
            ddn=docu_List[j-1]
            ddn=ddn.replace('Tgfile','')
            ddn=ddn.replace('.txt','')
            dname=''
            dn=int(ddn)
```

```
    dline=dname+dline.rstrip()
    encoding_doc=open(dline,encoding='utf-8')
    while True:
        v_String=encoding_doc.readline()
        fstring=remove_puc(v_String)
        fstring=Tnormaliz(fstring)
        v_List=fstring.split()
        v_List=sufpre(v_List)
        s=0
        while s < len(v_List):
                if v_List[s]=='':
                    del v_List[s]
                s+=1
        for i in range(0,len(v_List)-1):
                if v_Index.__contains__(v_List[i]):
                    if v_Index[v_List[i]].__contains__(dn):
                        continue #v_Index[v_List[i]][j]+=1
                    else:
                        t={dn:1}
                        v_Index[v_List[i]].update(t)
                else:
                     if stp_w.__contains__(v_List[i]):
                        continue
                     v_Index[v_List[i]]={dn:1}
        if len(fstring)==0:
            break
tpf=open('Posting File.txt','w',encoding='utf-8')
tpf.write('\tDoc_ID\tTF \t \t \t position')
tpf.write("\n")
for i in v_Index:
    tpf.write('\t')
    tpf.write(i)#DocId and TF for each term
    tpf.write('\t')
    tpf.write('\t')
    tpf.write('\t')
    tpf.write('\t')
    tpf.write(str(v_Index[i]))# The position of each terms
    tpf.write("\n")
```

```
    tpf.close()
############################## Amharic main ##############################
def Amain():
    prfx=open("prefix.txt",encoding='utf-8')
    prefix=prfx.read()
    prefix=prefix.split()
    prfx.close()
    sfx=open("sufix.txt",encoding='utf-8')
    suffix=sfx.read()
    suffix=suffix.split()
    sfx.close()
    dfile={}
    d_no=1
    stp_w=[]
    v_Index={}
    docu_List=[]
    docu_List=ano_docs()
    d_no=len(docu_List)
    nsw=0
    stopw=open("stopw.txt",encoding='utf-8')
    while stopw.readline()!='':
        nsw= nsw+1
    stopw.close()
    stopw=open("stopw.txt",encoding='utf-8')
    for i in range(1,nsw):
        line=stopw.readline()
        line=line.rstrip()
        stp_w.append(line)
    stopw.close()
    for j in range(1,d_no+1):
        dline='adct\\'+docu_List[j-1]
        ddn=docu_List[j-1]
        ddn=ddn.replace('Amfile','')
        ddn=ddn.replace('.txt','')
        dname=''
        dn=int(ddn)
        dline=dname+dline.rstrip()
        encoding_doc=open(dline,encoding='utf-8')
```

```
        while True:
            v_String=encoding_doc.readline()
            fstring=remove_puc(v_String)
            fstring=Anormaliz(fstring)
            v_List=fstring.split()
            v_List=asufpre(v_List)
            s=0
#to remove empty words
            while s < len(v_List):
                    if v_List[s]=='':
                        del v_List[s]
                    s+=1
            for i in range(0,len(v_List)-1):
                    if v_Index.__contains__(v_List[i]):
                        if v_Index[v_List[i]].__contains__(dn):
                            continue #v_Index[v_List[i]][j]+=1
                        else:
                            t={dn:1}
                            v_Index[v_List[i]].update(t)
                    else:
                         if stp_w.__contains__(v_List[i]):
                            continue
                         v_Index[v_List[i]]={dn:1}
            if len(fstring)==0:
                break

    tpf=open('Posting.txt','w',encoding='utf-8')
    tpf.write('\tDoc_ID\tTF \t \t \t position')
    tpf.write("\n")
    for i in v_Index:
        tpf.write('\t')
        tpf.write(i)#DocId and TF for each term
        tpf.write('\t')
        tpf.write('\t')
        tpf.write('\t')
        tpf.write('\t')
        tpf.write(str(v_Index[i]))# The position of each terms
        tpf.write("\n")
```

```
    tpf.close()
#############################################################################
def search():
    docu_List=no_docs()
    N=len(docu_List)
    v=[]
    c=0
    posting=open("Posting File.txt",encoding='utf-8')
    check=posting.readline()
    while check!='':
        post=check
        post=post.replace('\t','')
        s1=post
        s2=s1.split('{')
        if len(s2)<=1:
            check=posting.readline()
            continue
        else:
            s1=s2[1]
        s2=s1.split(',')
        l=[]
        for i in range(len(s2)):
            s1=s2[i]
            ll=s1.split(':')
            lll=int(ll[0])
            l.append(lll)
        s=len(s2)
        postt=post.split(':')
        n=(len(postt))-1
        t=postt[0].split('{')
        w=t[0]
        ww=math.log10((N-n+0.5)/(n+0.5))
        v.append([w,n,ww,l])
        c+=1
        check=posting.readline()
    m=0
    while m < len(v):
        n=0
```

```
        while n < len(v[m][3]):
            #v[m][3][n]=v[m][3][n].replace(' ','')
            n+=1
        m+=1
    posting.close()
################################## Amharic searching ###################
    docu_List=ano_docs()
    aN=len(docu_List)
    av=[]
    ac=0
    posting=open("Posting.txt",encoding='utf-8')
    check=posting.readline()
    while check!='':
        post=check
        post=post.replace('\t','')
        s1=post
        s2=s1.split('{')
        if len(s2)<=1:
            check=posting.readline()
            continue
        else:
            s1=s2[1]
        s2=s1.split(',')
        l=[]
        for i in range(len(s2)):
            s1=s2[i]
            ll=s1.split(':')
            lll=int(ll[0])
            l.append(lll)
        s=len(s2)
        postt=post.split(':')
        an=(len(postt))-1
        t=postt[0].split('{')
        w=t[0]
        ww=math.log10((aN-an+0.5)/(an+0.5))
        av.append([w,an,ww,l])
        ac+=1
        check=posting.readline()
```

```
    m=0
    while m < len(av):
            n=0
            while n < len(av[m][3]):
                    #av[m][3][n]=av[m][3][n].replace(' ','')
                    n+=1
            m+=1
    posting.close()
#########################################################################
    d_no=1
    stp_w=[]
    nsw=0
    stopw=open("StopwordList.txt",encoding='utf-8')
    while stopw.readline()!='':
        nsw= nsw+1
    stopw.close()
    stopw=open("StopwordList.txt",encoding='utf-8')
    for i in range(1,nsw):
        line=stopw.readline()
        line=line.rstrip()
        stp_w.append(line)
    stopw.close()
    query=input("አንታ·ይ ይደልዩ/ይደልያ፡    ")
    while len(query)==0:
       query= input(" ዝደልይዎ ሓበሬታ· ይጸሓፉ: ")
    fstring1=remove_puc(query)
    fstring1=re.sub('[፡።.!:¤()"፧?/!\ufeff]','',query)
    fstring1=Tnormaliz(fstring1)
    qv_List=[]
    qt_List=[]
    qt_List=fstring1.split(" ")
################################### meaning ######################
    dic=open("dictionary.txt",encoding='utf-8')
    am=[]
    tig=[]
    ii=0
    while(ii < 5530):
        lin=dic.readline()
```

```
        lin=lin.replace('\n','')
        t=lin.split('\t')
        if len(t)>1:
                am.append(t)
        ii+=1
    dic.close()
    for i in range(len(am)):
            am[i][0]=Anormaliz(am[i][0])
            am[i][1]=Tnormaliz(am[i][1])
    aall=[]
    c=0
    for i in range(len(qt_List)):
                c=0
                for j in range(len(am)):
                     if qt_List[i]==am[j][1]:
                             aall.append(am[j][0])
                             c+=1
                             break
                if c==0:
                      aall.append(qt_List[i])
    for w in range(len(aall)):
            aall[w]=Anormaliz(aall[w])
    print (aall)
#########################################################################


    qu=len(qt_List)
    if qu>15:
       for q in range(15):
            qv_List.append(qt_List[q])
    else:
         qv_List=qt_List
    indexQ={}
################################ amharic searching ##################
    aall=asufpre(aall)
    aws={}
    for i in range(len(aall)):
            for j in range(len(av)):
```

```
                if aall[i]==av[j][0]:
                        for k in range(len(av[j][3])):
                                t=av[j][3][k]
                                if aws.__contains__(t):
                                        aws[t]+=av[j][2]
                                else:
                                        aws[t]=av[j][2]
    for ii, j in aws.items():
            i=str(ii)
            i="አምሓርኛ "+i
            t=[i,j]
            aresult.append(t)
#####
    qv_List=sufpre(qv_List)
    ws={}
    for i in range(len(qv_List)):
            for j in range(len(v)):
                    if qv_List[i]==v[j][0]:
                            for k in range(len(v[j][3])):
                                    t=v[j][3][k]
                                    if ws.__contains__(t):
                                            ws[t]+=v[j][2]
                                    else:
                                            ws[t]=v[j][2]

    result=[]
    for ii, j in ws.items():
            i=str(ii)
            i="ትግርኛ "+i
            t=[i,j]
            result.append(t)
    tot=[]
    tot=result
    for k in range(len(aresult)):
            tot.append(aresult[k])
    result=tot
    for i in range(len(result)):
            for j in range (len(result)-1):
```

```
            if result[j][1] < result[j+1][1]:
                t=result[j]
                result[j]=result[j+1]
                result[j+1]=t
print("\n በቲ መሕትት መሰረት ዝተረክቡ ናይ ትግርኛን አምሓርኛን ፋይላት እዞም ዝስዕቡ እዮም \n")
for i in range(len(result)):
    if result[i][1]>1:
        print(i,": ---------------------------- ",result[i][0]," ---------------------------- ")
total=[]
total=result
for k in range(len(aresult)):
    total.append(aresult)
rr=[]
rep='y'
f=0
trr=[]
arr=[]
while (rep=='y'):
    dn=0
    ree=input("\n ትሕዝቶ ናይቲ ፋይል ንምንባብ ቁፅሪ ቅደም ሰዓብ ናይቲ ፋይል የእትዉ \t")
    red=int(ree)
    if result[red][0].startswith('አምሓርኛ'):
        fn=result[red][0]
        fn=fn.replace('አምሓርኛ ','')
        dn=int(fn)
        fn='adct\\Amfile'+fn+'.txt'
        doc=open(fn,encoding='utf-8')
        docc=doc.read()
        print(docc)
        f=2
    elif result[red][0].startswith('ትግርኛ '):
        fn=result[red][0]
        fn=fn.replace('ትግርኛ ','')
        dn=int(fn)
        fn='tdct\\Tgfile'+fn+'.txt'
        doc=open(fn,encoding='utf-8')
        docc=doc.read()
        print(docc)
```

```
            f=1
        else:
            continue
        feedbk=input("\n እዚ ፋይል ጠቓሚ ኮይኑዶ ረኪበሞ? [y/n] ->\t")
        if feedbk=='y':
            if f==2:
                arr.append(dn)
            elif f==1:
                trr.append(dn)
            else:
                f=0
        rep=input("\n ካሊእ ፋይል ዶ ክንብቡ ይደልዩ? [y/n] ->\t")
    tr=[]
    ar=[]
    trf=input("\n እት ሲስተም ብናይባዕሊ መገዲ ጠቕምቲ ፋይላት ክግምት እንድሕር ደሊዮም 1 ይጠውቹ \n እንድሕር ባዕሎም ብዝመረፅዎም ጠቕምቲ ፋይላት ክጥቀም ደሊዮም 2 ይጠውቹ \n\t -> ")
    if trf=='1':
        for kk in range(5):
            if result[kk][0].startswith('አምሓርኛ'):
                fn=result[kk][0]
                fn=fn.replace('አምሓርኛ ','')
                dn=int(fn)
                ar.append(dn)
            elif result[kk][0].startswith('ትግርኛ'):
                fn=result[kk][0]
                fn=fn.replace('ትግርኛ ','')
                dn=int(fn)
                tr.append(dn)
            else:
                continue
    elif trf=='2':
        tr=trr
        ar=arr
    R=len(tr)
    for k in range(len(v)):
        c=0
        for j in range(len(v[k][3])):
            for i in range(len(tr)):
```

```
                    if v[k][3][j]==tr[i]:
                        c+=1
            v[k].append(c)
            n=v[k][1]
            nw=math.log10(((c+0.5)*(N-n-R+c+0.5))/((n-c+0.5)*(R-c+0.5)))
            v[k].append(nw)
############################
    R=len(ar)
    for k in range(len(av)):
            c=0
            for j in range(len(av[k][3])):
                for i in range(len(ar)):
                    if av[k][3][j]==ar[i]:
                        c+=1
            av[k].append(c)
            n=av[k][1]
            nw=math.log10(((c+0.5)*(aN-n-R+c+0.5))/((n-c+0.5)*(R-c+0.5)))
            av[k].append(nw)
############################3
    ws={}
    for i in range(len(qv_List)):
            for j in range(len(v)):
                if qv_List[i]==v[j][0]:
                    for k in range(len(v[j][3])):
                        t=v[j][3][k]
                        if ws.__contains__(t):
                            ws[t]+=v[j][5]
                        else:
                            ws[t]=v[j][5]
#################
    wss={}
    for i in range(len(aall)):
            for j in range(len(av)):
                if aall[i]==av[j][0]:
                    for k in range(len(av[j][3])):
                        t=av[j][3][k]
                        if wss.__contains__(t):
                            wss[t]+=av[j][5]
```

```
                        else:
                            wss[t]=av[j][5]
#################
    result=[]
    for i, j in ws.items():
        ii=str(i)
        ii='ትግርኛ '+ii
        t=[ii,j]
        result.append(t)
    for i, j in wss.items():
        ii=str(i)
        ii='አምሓርኛ '+ii
        t=[ii,j]
        result.append(t)
    for i in range(len(result)):
        for j in range (len(result)-1):
            if result[j][1] < result[j+1][1]:
                t=result[j]
                result[j]=result[j+1]
                result[j+1]=t
    print("\n አብ መወዳእታ ዝተረክቡ ጠቐምቲ ናይ ትግርኛን አምሓርኛን ፋይላት እዞም ዝስዕቡ እዮም \n")
    for i in range(len(result)):
        if result[i][1]>3:
            print(i,": ---------------------------- ",result[i][0]," ---------------------------- ")
def menu():
    print("\n                        ሚኑ          ")
    print("                    |ትግርኛ ኢንደክሰ ንምግባር ----------1 የእትው|")
    print("                    |አምሓርኛ ኢንደክሰ ንምግባር ----------2 የእትው|")
    print("                    |ሓበሬታ ንምረካብ ------------------3 የእትው|")
    print("                    |ካብቲ ሲስተም ንምውጻእ -------------4 የእትው|")
    print("                    |--------------------------------------|")
    cho=int(input("ይምረጽ: "))
    if cho==1:
        Tmain()
        print("ትግርኛ ኢንደክሰ ተገቢሩ አሎ ቻው ")
        menu()
    elif cho==2:
        Amain()
```

```
            print("አምሓርኛ ኢንደክሰ ተገቢሩ አሎ ቻው ")
            menu()
        elif cho==3:
            search()
            menu()
        else:
          while True:
            print ('----------------------------------------------------------------')
            print(" \nካብ ልቦም ክገድፍዎ? ")
            yes= input(" እንድሕር ዘይገድፍዎ/ኦ ኣይደልን ወይ 1  በሉ/ላ,እንድሕር ክገድፍዎ/አ  እወ ወይ 2 በሉ/ላ:")
            if yes in ('ኣይደልን','1'):
                menu()
            if yes in ('እወ','2'):
                print ("\t\t\t  የቐንየልና!!")
                print ('----------------------------------------------------------')
                break
menu()
```