ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A PROBABILISTIC INFORMATION RETRIEVAL SYSTEM FOR TIGRINYA

BY

ATALAY LUEL

June 2014

ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A PROBABILISTIC INFORMATION RETRIEVAL SYSTEM FOR TIGRINYA

A THESIS SUBMITTED TO THE SCHOOL OF INFORMATION SCIENCE OF ADDIS-ABABA UNIVERSITY

BY

ATALAY LUEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE

June 2014

ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A PROBABILISTIC INFORMATION RETRIEVAL SYSTEM FOR TIGRINYA

BY

ATALAY LUEL

Name and signature of members of the examining board

Name	Title	Signature	Date
Ato _____	Chairperson:	_____	_____
Dr. Solomon Teferra	Advisor:	_____	_____
Ato Ermias Abebe	Examiner:	_____	_____
Dr. Dereje Teferi	Examiner:	_____	_____

Dedication

To my family

DECLARATION

This thesis is my original work. It has not been presented for a degree in any other university

_____

ATALAY LUEL

This is to certify that I have examined this copy of Master's thesis by

ATALAY LUEL and have found it is complete and acceptable in all respects, and has been submitted for examination with my approval as university advisor.

Name of Advisor	Signature of Advisor	Date
SOLOMON TEFERRA ABATE (PhD)	_____	_____

June 2014

ADDIS ABABA, ETHIOPIA

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms and Abbreviations

BIM	Binary Independent Model
BLE	Bahadur-Lazarsfeld expansion
BNM	Bayesian Network Model
CLIR	Cross Lingual Information Retrieval
EBM	Extended Boolean Model
FDRE	Federal Democratic Republic of Ethiopia
GVSM	Generalized Vector Space Model
IDF	Inverse Document Frequency
IR	Information Retrieval
LSI	Latent Semantic Indexing
MAP	Mean Average Precision
NIST	National Institutes of Standards and Technology
PRP	Probabilistic Ranking Principle
SMART	System for the Mechanical Analysis and Retrieval of Text
Sim	similarity
SVD	Singular Value Decomposition
TF	Term Frequency
TREC	Text Retrieval Conference
VOA	Voice Of America
VSM	Vector Space Model

Acknowledgments

I would like to take this opportunity to acknowledge all those who helped me during this thesis work. Before everything, I would like to thank the omnipotent God for giving me the strength to achieve whatever I have achieved so far.

Then, I am deeply thankful to my advisor Dr. Solomon Teferra for his valuable suggestions and guidance during the course of this thesis work, and his patience in reviewing my thesis. He listened to all my problems I faced during this thesis and showed me the way to overcome them. He has been providing me constructive comments for the betterment of this study.

I owe my deepest heartfelt appreciation to Dr. Million Meshesha for his support, constructive suggestion, academic commitment and encouragement at the very beginning of my work. His appreciation and input to my work helped me to make my research more alive.

My special thanks also go to my friend Ato Tsegay Semere for his constant guidance and support and my parents for their moral support and encouragement during my study. I thank especially, my Fiancé Netsanet Chekole for her constant encouragement and caring during the time of my study.

Finally, I extend my heartfelt thanks and respect to all my classmates and those people who were not mentioned here but their contributions have been inspiring for the completion of this work. Thank you all.

Abstract

Many applications that handle information on the internet or other archive would be completely inadequate without the support of information retrieval technology. Nowadays, a considerable amount of information has been produced in Tigrinya. This accumulation of information is challenging for searching from the existing huge amount of information particularly written in Tigrinya. Thus, developing an IR system for Tigrinya allows searching and retrieving relevant documents that satisfy information need of Tigrinya users. Accordingly a research is conducted for Tigrinya IR system using the probabilistic model which, unlike vector space model, has the mechanism to reweighting query terms using relevance feedback and query reformulation techniques. Additionally, the model does define uncertainty that exists in IR systems. This thesis is a pioneer research on IR for Tigrinya text documents. This research is initiated to experiment the effectiveness of an IR system for Tigrinya using a rule-based Tigrinya stemmer developed by Yonas Fisseha in 2011. Yonas had recommended that researches should be conducted using Tigrinya stemmer on Tigrinya Information Retrieval system to see its impact over recall and precision.

In this thesis, the potential of probabilistic model in Tigrinya text retrieval is investigated. 300 Tigrinya documents and 10 queries were used to test the approach. The researcher presents the design and prototype implementation of the probabilistic model of the IR system for Tigrinya documents.

Both indexing and searching modules are constructed. Then, the retrieval system is tested and the experimental results show that probabilistic based IR system in Tigrinya documents returned encouraging result. The system registered, after stemming and pseudo relevance feedback, an average precision 69.1%, recalls 90%, and F- measure 74.4%. This result is achieved without controlling the problem of synonyms and polysemous of terms that exist in Tigrinya text.

The researcher has recommended that further works on the area need to see the retrieval effectiveness of Tigrinya IR system using 1) hybrid Tigrinya stemmer to mean rule based and dictionary based Tigrinya stemming algorithm or 2) ontology based stemming algorithm that conflates based on meaning understanding (recommended more). There should also a need to build hybrid system that uses vector space model to guess relevant documents for user query using non-binary weighting technique and then use probabilistic relevance feedback to improve the performance of the system and to solve the problem of the initial guess of probabilistic model based on Boolean expression.

CHAPTER ONE

Introduction

1.1 Background

The practice of archiving written information can be traced back to around 3000 BC, when the Sumerians designated special areas to store clay tablets with cuneiform[1] inscriptions. Even then the Sumerians realized that proper organization and access to the archives was critical for efficient use of information [3].

The need to store and retrieve written information became increasingly important over centuries, especially with inventions like paper and the printing press. Soon after computers were invented, people realized that they could be used for storing and mechanically retrieving large amounts of information. In 1945 Vannevar Bush published a ground breaking article titled "As We May Think" that gave birth to the idea of automatic access to large amounts of stored knowledge (information) [3]. In the 1950s, this idea materialized into more concrete descriptions of how archives of text could be searched automatically. Several works emerged in the mid 1950s that elaborated upon the basic idea of searching text with a computer. One of the most influential methods was described by H.P. Luhn in 1957, in which he proposed using words as indexing units for documents and measuring word overlap as a criterion for retrieval [4].

Several key developments in the field happened in the 1960s. SMART system were Among the most important development by Gerard Salton and his students, first at Harvard University and later at Cornell University [5] and the Cranfield evaluations done by Cyril Cleverd on and his group at the College of Aeronautics in Cranfield. The Cranfield tests developed an evaluation methodology for retrieval systems that is still in use by IR systems today. The SMART system, on the other hand, allowed researchers to experiment with ideas to improve search quality. A system for experimentation coupled with good evaluation methodology allowed rapid progress in the field, and paved way for many critical developments [6].

The 1970s and 1980s saw many developments built on the advances of the 1960s. Various models for doing document retrieval were developed and advances were made along all dimensions of the retrieval process. These new models/techniques were experimentally

[1] Based on Concise Oxford Dictionary: denoting or relating to the wedge-shaped characters used in the ancient writing systems of Mesopotamia, Persia, and Ugarit.

proven to be effective on small text collections (several thousand articles) available to researchers at the time. However, due to lack of availability of large text collections, the question whether these models and techniques would scale to larger corpora remained unanswered. This changed in 1992 with the inception of Text REtrievalConference, or TREC. TREC is a series of evaluation conferences sponsored by various US Government agencies under the auspices of NIST, which aims at encouraging research in IR from large text collections [7].

With large text collections available under TREC, many old techniques were modified, and many new techniques were developed (and are still being developed) to do effective retrieval over large collections. TREC has also branched IR into related but important fields like retrieval of spoken information, non-English language retrieval, information filtering, user interactions with a retrieval system, and so on. The algorithms developed in IR were the first ones to be employed for searching the World Wide Web from 1996 to 1998. Web search, however, matured into systems that take advantage of the cross linkage available on the web [7].

Now days, Information Retrieval is one of the major branches of Information Science discipline [1]. Information retrieval deals with the representation, storage, organization of, and access to information items. The representation and organization of the information items should provide the user with easy access to the information in which she/he is interested. Unfortunately, characterization of the user information need is not a simple problem .Because information retrieval is defined as finding documents of an unstructured nature that satisfies information need of users from within large collection [1, 16, 28].

In addition, an information retrieval system is a system that stores and manages information on documents and also enables users finding the information they need. It returns documents that contain answer to user's question (query) rather than explicit answer to their information need. Most of the time retrieved documents satisfy users'information needs. Documents which satisfy users'information needs called relevant documents, whereas documents which are not satisfying users'information need are irrelevant documents. In fact there is no perfect information retrieval system which retrieves all relevant documents and no irrelevant document [1, 7, 16].

Information Retrieval has two main subsystems [16]; Indexing and Searching. Indexing is an offline process of representing and organizing large document collection using indexing

structure such as Inverted file, sequential files and signature file to save storage memory space and speed up searching time. Searching is the process of relating index terms to query terms and return relevant hits to users query. Both indexing and searching are interrelated and dependent on each other for enhancing effectiveness and efficiency.

The focus of quality of Information Retrieval design is in evaluating both retrieval effectiveness and efficiency [6, 16]. Efficiency is about optimizing computing resource such as the needed storage space and time complexity, while effectiveness concerned with relevancy of document retrieved that satisfies users‘ information need.

There are different IR models availbele now days [1, 16]. IR models are responsible for determining the prediction of what is relevant and what is not. 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.

Among the Current models one is the Probabilistic model. 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 “If a reference retrieval system’s response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data.” [53].

The probabilistic model is the oldest but also the hottest research area in IR [1,28]. It incorporates relevance feedback and term reweighting mechanisms using the source of evidence of the statistical distribution of the terms in both the relevant and non-relevant documents. The principle takes into account that there is uncertainty in the representation of the information need and the documents [30, 56]. The detail of the most common IR models are discussed in section 2.4.

IR system is uncertain in nature [1]. The most successful approach for handling the uncertainty nature of IR is probabilistic model [30].

Experimental evidences show that other models like Vector space model (VSM) and its variant models such as, Extended Boolean Model (EBM) and Generalized Vector space model (GVSM) are not attempted to define uncertainty in IR system. They do not have relevance feedback and term reweighting mechanism by them-selves to do with the external realities of users and information needs unless other modules are integrated to those models so as to register better performance [16, 18]. See the comparison made by Amanuel [11] in table 1.1

	Probabilistic Model	**Vector Space Model**
Motivation	Address uncertainty in query representations	Simplify query formulation
Goal	Rank the output based on probability of relevance	Rank the output based on similarity of document and query
Methods	Example: BIM, BNM	Example : Cosine measure

Table 1.1: Characteristics of vector space model and probabilistic model

There are different information retrieval methods which have a probabilistic basis [30, 31, 32]. Several researches in the area of probabilistic retrieval model has been conducted globally. Robertson et.al [19] developed a probabilistic IR system for English to solve the problem of uncertainty found in IR system. From the experiment, encouraging result is found and the detail is given in section 2.7.1. However, the consideration of giving the same weights for terms in different document was one of the main challenges. Rijsbergen [20] also developed probabilistic IR system to improve the binary independent assumption and the uncertainty nature of IR by considering the semantic relationship of query terms and document.

1.2 Statement of the Problem and Justification

There are more than 80 languages in Ethiopia. Tigrinya is one of the local languages. It is a Semitic language of the Afro-Asiatic language family [40]. It is spoken in Tigray-Ethiopia as well as the contiguous borders of southern Eritrea. It is also spoken by large immigrant communities around the world, like; Sudan, Saudi Arabia, Italy, Sweden, the United Kingdom, Canada and the United States. This language has more than six million speakers worldwide [22].

Tigrinya is the official language of Tigray regional state of Ethiopia and also academic language for primary school of the region. Tigrinya literature and myths are delivered as a field of study in many universities in Ethiopia [33]. Nowadays journal, magazines, newspapers, news, online education, books, entertainment Medias, videos, pictures in Tigrinya language are available in electronic format both on the Internet and on offline sources. There is huge amount of information being released with this language, since it is the language of education and research, language of administration (in Tigray and others areas specified above) and political welfares, language of religious activities and social interaction [8,41].

As a result, the Tigrinya documents are increasing in size from time to time. This shows that there are large collections of Tigrinya document available in the web [9]. Thus, it is necessary to implement an IR system for this language.

There are different challenges in implementing an IR system [1]. Information retrieval is language dependent process which needs integrating knowledge of information retrieval techniques and natural language in relation the particular Tigrinya language. Most of IR techniques are developed for English and it is always difficult task applying it for other languages [16].

Worldwide there are a number of IR systems designed using probabilistic model in different languages [1].

There is no IR work still done for Tigrinya language except a CLIR system developed by Tsegay [12]. The attempt made by this study is to enable retrieval of document of Amharic in Tigrinya query. But this doesn‘t solve need of users all in all. Users who look for Tigrinya document with Tigrinya query may not find suitable environment to find information of their need with those prior works. From perspective of IR principles it is important to have system that works for Tigrinya and then it is better extending it to CLIR. Implementation of this work helps users of Tigrinya to find Tigrinya documents (information) simply without much difficulty using Tigrinya query [1].

IR system can be developed using different models. As the comparison made according to Amanuel [11] in table 1.1

IR systems based on vector space model developed for Amharic language so far has not registered a satisfactory performance. For the reason that vector space model and its variants do not have the mechanism to define users need using relevance feedback and query

reformulation techniques unless other modules are integrated to the models. In comparison probabilistic model itself enables defining users need using relevant feedback and query reformulation techniques. This comparison about the models also works for Tigrinya language too.

A rule-based stemming algorithm for Tigrinya language has been developed by Yonas [26] in 2011.The stemmer was the first rule-based stemmer for Tigrinya language and Yonas recommended that researches should be conducted using the stemmer on Tigrinya Information Retrieval system to see its impact over recall and precision.

Therefore, the researcher has taken this as an opportunity to develop an information retrieval system to see the effectiveness of the stemmer on Tigrinya text. The stemmer details are given in section 2.7.2.

Hence, this research is initiated to experiment the effectiveness of IR system for Tigrinya text retrieval system using Tigrinya stemmer developed by Yonas [26] that organize document corpus using indexing and search relevant ones as per users query based on probabilistic model. In addition it tries to answer the following research questions:

- ✓ What are the basics of Tigrinya language to perform text operation?
- ✓ What are the suitable components to design probabilistic based Retrieval system?
- ✓ What is the effect of the stemming on the performance of a Tigrinya probabilistic IR?

1.3 Objective of the Study

1.3.1 General Objective

The main objective of this study is to investigate a probabilistic IR processes in order to design, build and test a prototype Information Retrieval system for Tigrinya.

1.3.2 Specific Objectives

In order to meet the general objective, the following specific objectives are performed.

- ❖ Review literatures on previous works related with information retrieval system.
- ❖ Understand and explore basics of Tigrinya language and perform text operation such as, tokenization, stop words removal, stemming, and normalization.
- ❖ Design architecture for implementing probabilistic Information Retrieval System for Tigrinya language.
- ❖ Develop a prototype of probabilistic Information Retrieval System for Tigrinya language

that searches relevant documents from unstructured Tigrinya corpus.

- ❖ Integrate the Tigrinya stemmer with the prototype and measure its effectiveness
- ❖ Evaluate the effectiveness of probabilistic Information Retrieval System for Tigrinya language using recall, precision and F-measure from the prototype.

1.4 Scope and Limitation of the Study

The study focuses on designing and developing prototype information retrieval system that effectively searches within Tigrinya text corpus. This study mainly implements an indexer and searcher from corpus of Tigrinya textual documents. Other data types, such as multimedia (though spoken, image, and video data are all important), are out of focus of the research. For the indexing purpose and to identify content bearing index terms and query terms a series of text operations such as tokenization, normalization, stop word removal and stemming are applied. Index terms are organized using inverted file and searching for documents satisfying query terms are guided by probabilistic model.

The researcher used a rule based Tigrinya stemmer developed by Yonas [26] which handles only prefixes and suffixes. Infixes, reduplication, compounding and irregular words are not handled.

The system is tested using limited amount of Tigrinya text documents prepared by the researcher. This is due to lack of standard Tigrinya corpus prepared for IR purpose. In addition, lack of thesaurus to integrate for query expansion mechanism to control Tigrinya synonyms and polysemy words are the limitations of this research. Due to long processing time to construct relevance matrix for relevance judgment, considering the time constraint, the experiment was conducted using only 300 Tigrinya documents.

1.5 Methodology

This research is conducted in order to figure out challenges of implementing probabilistic Tigrinya information retrieval system. Accordingly, the following step by step procedures are followed to achieve the main objective of the study.

1.5.1 Literature Review

To have conceptual understanding and to identify the gap that is not covered by previous studies different materials, including journal articles, conference papers, books and thesis works are reviewed. In this study the review mainly concerned works that have direct relation with the topic and the objective of the study. Additional literature review and document

analyses are made to investigate and identify the feature of Tigrinya text, which is important to the research in the course of Tigrinya text operations. These include previous works done on the area of information retrieval system giving more attention to local and international works that attempt to develop information retrieval system and search engine.

1.5.2 Data Collection and preparation

Probabilistic Tigrinya IR system requires Tigrinya text corpus documents and these are collected from different sources. Mainly to conduct the research Tigrinya documents are prepared from hidiyat magazine, Tigray online, VOA program from internet and newspapers, because they are easier to access, available in electronic form and cover all domains. These items cover issues such as, politics, sport, social, religion and philosophy, health, art, and education.

1.5.3 Development Tools

The program is developed using Python 3.2.2 programming language to implement the prototype. This is because Python has rich string manipulation techniques and the researcher has some knowhow of writing programs using Python.

It is simple, strong, involves natural expression of procedural code, modular, dynamic data types, and embeddable with in applications as a scripting interface [10].

1.5.4 Testing Process and Evaluation Techniques

The experimentation for evaluating the effectiveness of the system is done by using 300 selected test documents from Tigrinya corpus and ten (10) queries in Tigrinya.

After corpus is prepared and queries are constructed then relevance judgment is made for evaluating effectiveness of the work. Recall , precision and F-measure techniques are used for measuring retrieval effectiveness of the IR system ,see section 4.4 and its sub-sections; as they are frequently used and most basic measures of IR effectiveness[16]. Paricularly, in this work the interpolated precision value at **11 standard recall level** is used to draw precision-recall curve in order to evaluate retrieval effectiveness of the system.

1.6 Significance of the Study

It is important localizing works already done in international languages like English. Generally the study has the following significance: 1) provides an opportunity for other researchers to focus on the area and to continue on the area, to come up with an applicable IR system. 2) enables Tigrinya speakers retrieve text documents in Tigrinya language effectively

3) since the study is master thesis it has also learning out comes for the researcher. It helps the researcher to investigate problems and solving it in scientific manner. Additionally, it is an academic exercise to fulfill the requirement of masters program the researcher is enrolled in.

1.7 Organization of the Thesis Paper

For ease of comprehension, this thesis has a simple structure in which five chapters are distinguished. The rest, chapter 2 to chapter 5 are organized in the following way. Chapter two is literature review. It involves 6 main topics: Historical Overview of Tigrinya Languages and its writing system, General review of the Information Retrieval Process, IR models, Query Operation, IR Systems Evaluation and Review of previous Related IR Works; including local and international works.

Chapter three is Probabilistic Tigrinya IR System Design and Architecture . In this section technique used for indexing, searching and IR system evaluation are discussed

Chapter four is experimentation and result analysis. In this part corpus preparation, query preparations, experimentations, retrieval performance evaluation, result analysis, Findings and challenges are discussed in detail.

Finally in chapter five conclusion and recommendation are given.

CHAPTER TWO
LITERATURE REVIEW

2.1 Historical Overview of Tigrinya Languages and its writing system

Tigrinya, which is spoken in Eritrea and Ethiopia, is the second largest, after Amharic, member of the Ethiopian branch of the Semitic family of languages, constituting together with Tigre and the extinct Ge‘ez (or Classical Ethiopic). The Semitic languages are the Afro-Asiatic language family. The most widely spoken Semitic languages today are Arabic, Amharic, Tigrinya, Hebrew, and Aramaic. There are different Semitic languages families in Ethiopia. These are Amharic (አማርኛ), Tigrinya (ትግርኛ), Gurage (ጉራግኛ), Tigre (ትግረ) and Ge’ez (ግዕዝ) [34].

Tigrinya is the working languages of Tigray regional state of Ethiopia [8]. Tigrinya is spoken by the Tigray people located in northern Ethiopia and Eritrea people. It is also spoken by large immigrant communities around the world, in countries including Sudan, Saudi Arabia, Israel, Germany, Italy, Sweden, the United Kingdom, Canada and the United States. Estimates of the number of speakers in Tigrinya language (in Tigray, Eritrea and including immigrant) vary from 6 to 7 million worldwide [22]. According to Federal Democratic Republic of Ethiopian (FDRE) population census commission report Tigray population was 4.3 million in 2007 [21].

Tigrinya is written in slightly expanded version of Geez script also called Ethiopic. Ethiopic Script is syllabic; each symbol represents 'consonant + vowel' characteristics. Each Tigrinya base character also known as (Fidel, ፊደል) has seven vowel combinations. Tigrinya is written from left to right. Tigrinya as a written language has a history only from the latter half of the 19th century, due in great part to the prestige of Ge‘ez as the written language of Christian Ethiopia in the past [22]. According [40,41], Tigrinya language has its own characters (alphabets), punctuation marks, and numbers systems . Converting them into Latin representation is required for IR purpose. For this purpose Tigrinya documents are saved using UTF-8 format, which is supported by most programming languages. It translates the Tigrinya text into suitable representation for computer understandability.

2.1.1 Tigrinya 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 [8]. There are different alphabets representations in the world. The most alphabets representation is Latin or Roman alphabets which have been adapted by numerous languages [37]. The Ethiopic writing systems have also their own writing systems and alphabets representations. Tigrinya is one of the Semitic languages that use Ethiopic writing system. Tigrinya has its own alphabets (ፊደል) and they are used for writing. It has thirty-five (35) base symbols with seven orders which represent seven vowels for each base symbol, total 245 characters [8]. In addition to the basic forms, there are also additional nearly fifty three (51) characters which contains special feature representing labialization [40,41]. The script has over 296 (basic and non basic) characters, each representing a different sound. Basic and non basic Tigrinya scripts are given in appendix two of this paper. And it can be characterized by the following points [41]:

- ✓ Every letter has seven different sounds
- ✓ Pronunciation is very simple and straight forward. In Tigrinya "You write as you read it and you read as you write it".
- ✓ It is written and read from left to right as the English language.
- ✓ There are many familiar words that Tigrinya has borrowed from other languages, especially Italian and English, which reflect the history of the areas where Tigrinya is spoken.

2.1.2 Tigrinya Punctuation marks

Identifying punctuation marks is vital to know word separation for natural language processing. Most of the Tigrinya language punctuation marks are listed in table 2.1 below. There are around 17 punctuation marks [40]. However, only few of them are practically used, especially in computer-written system. These are the word separator mark or ‘ክልተ ነጥቢ ‘(:) 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 or ‘ኣርባዕተ ነጥቢ’ (::) is used to shows when an idea is finished. The sentence connector mark semicolon or ‘ድርብ ሰረዝ’ (፤,), alternatively ‘ነፀላ ሰረዝ’ (፣) 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 Tigrinya language is also borrowed some punctuation marks from English language such as?, !, and ” [37].

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: Tigrinya punctuation marks

2.1.3 Tigrinya Number system

Tigrinya sometimes and rarely uses Geez numerals for official writing purposes. It has twenty characters. They represent numbers from one to ten (፩-፲), twenty to ninety (፳-፺), hundred (፻) and thousand (፼) [41]. However, these are not suitable for arithmetic computation purposes because there is no representation for zero (0), decimal points. As a result, Tigrinya numbers are used for calendar purposes. Since today Arabic numbers are more frequently used than the original Ethiopian numbers, for arithmetic computation, Arabic numbers are used in the Tigrinya literature. The complete Tigrinya number is depicted in appendix three.

2.1.4 Problem of Tigrinya Writing System in Designing Tigrinya Text Retrieval

Leslau [8] notes that there are a number of challenges in Tigrinya language for text processing.

2.1.4.1 Redundancy of some characters

Sometimes more than one letter is used to represent similar sound in Tigrinya language. For instance, letters with their six variant forms of each character ሀ and ኀ, ጸ and ፀ, ሰ and ሠ have similar sounds in reading. As a result, words which have the same meaning may have different spelling structure. For example, the word “ፀለምᵐ” which means “black” can be written differently as; “ጸለምᵐ” [41].

In IR system since it only match the character in query words to check whether the word found in a document has the same structure as the word in the query, it consider the words as dissimilar.

2.1.4.2 Spelling variation of the same borrowed words

Even though there are feasible problems of spelling variation in current literature of Tigrinya [38]. A word may be translated by different persons using different spelling variation. For instance, when "television" word is translated into Tigrinya, it may be written as "ተለቭሽን", "ተለቭሽን" or "ተለበጅን". All these words are used to mean the word "television" in Tigrinya. Also all the following Tigrinya words are used to mean the word "millennium". It is translated into Tigrinya and may be written as "ሚሌኒየም","ሚሊኒየም", "ሚለኒየም" or "ሚለንየም". The translation of English words into Tigrinya words creates a problem in IR because they are treated differently.

2.1.4.3 Abbreviations

The abbreviations of Tigrinya words follow different formats [40]. Some time full stop '.' is used to abbreviate, while other time '/' symbol is used to abbreviate. The abbreviated words can be written without separators. For example, "ገብረሂወት" (Gebrehiwet) can be written as "ገ/ሂወት" (G/hiwot) or "ገ.ሂወት" (G.hiwot). The inconsistency in the abbreviation creates problem in IR processes [38].

2.1.4.4 Compound words

Tigrinya compound words are written in different format. Mostly space and hyphens are used to separate them. When the hyphen is used the two words are treated as one word. However, when they are separated by space their meaning differ. For example, "ቤት ትምህርቲ", "ስነ ስርዓት", "መራሕቲ ስድራ", and "ኣብያተ ፅሕፈት" are compound words separated by space. However, words "ቤት", "ስነ", "መራሕቲ", and "ኣብያተ" do not have meaning when they are used separately. So this can create problems in IR. This situation makes an IR system difficult to differentiate those words.

2.2 Information Retrieval

The meaning of the term information retrieval is very wide ranging, but in relation to computer science a general definition is provided by different scholars. For instance Manning et al. [28] defines as: "Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)".

According to P. E. R. Ingwersen [1] and R. Baeza-Yates et al [16] “Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, organization of, and retrieval to information”. The representation and organization of the information items should provide the user with easy access to the information in which he/she is interested. Unfortunately, characterization of the user information need is not a simple problem [1].

Information retrieval is the science and technology concerned with the effective and efficient retrieval of information for the subsequent use by interested parties [30]. The central problem in IR is the quest to find the set of relevant documents, amongst a large collection, containing the information sought thereby satisfying an information need usually expressed by a user with a query [16, 32]. The documents may be objects (items) in any medium, text, image, audio, or, indeed a mixture of all three but in this thesis the objects are text documents. An important area of research concentrates on the modeling of objects and processes involved in the retrieval of information. Well known models of IR are the Boolean, vector space and probabilistic models; these have been studied in detail and implemented for experimentation [1]

2.3 General review of the Information Retrieval Process

An information retrieval is a system that stores and manages information on documents, often textual documents and possibly multimedia. The system assists users in finding the information they need. It does not explicitly return information or answer questions. Instead, it informs on the existence and location of documents that might contain the desired information [1].

Some suggested documents will, hopefully, satisfy the user's information need [31]. These documents are called *relevant* documents. A perfect retrieval system would retrieve only the relevant documents and no irrelevant documents. However, perfect retrieval systems do not exist and will not exist, because search statements are necessarily incomplete and relevance depends on the subjective opinion of the user. In practice, two users may pose the same query to an information retrieval system and judge the relevance of the retrieved documents differently: Some users will like the results; others will not [32].

There are three basic *processes* an information retrieval system has to support: the representation of the content of the documents, the representation of the *user's information need*, and the *comparison* of the two representations. The processes are visualized in Figure

2.1. In the figure, squared boxes represent data and rounded boxes represent processes [1, 16, 17, 31].

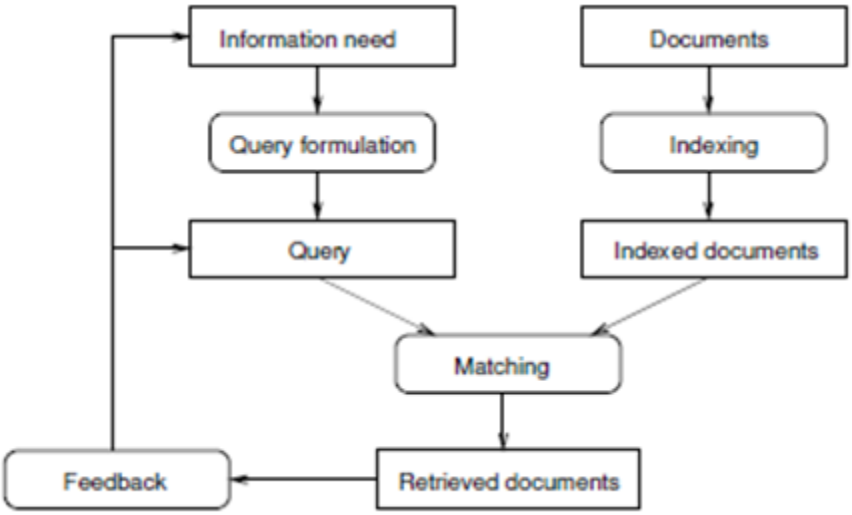

Figure 2.1: Information Retrieval processes [31]

2.3.1 The Indexing Process

Representing the documents is usually called the indexing process. The process takes place off-line, that is, the end user of the information retrieval system is not directly involved. The indexing process results in a representation of the document [16].

According to, D. Hiemstra [31], indexing process is an arrangement of index terms to permit fast searching and reading memory space requirement used to speed up access to desired information from document collection as per users query such that it enhances efficiency in terms of time for retrieval. Relevant documents are searched and retrieved quick. Index file usually has index terms in a sorted order. An index file consists of records, called index entries. Index files are much smaller than the original file (because it is recommended to use *keyword terms than full text*). The usual unit for indexing is the word called Index term. Index terms are used to look up records in a file [16].

Document representation is either (a) *full text* representation or (b) only via content bearing terms using *keyword terms*. (a) Full text set of words representation is an early way of document representation. Using full text indexing reduces retrieval efficiency and it asks higher computational costs [4,16]. Often, full text retrieval systems use a rather trivial algorithm to derive the index representations. The indexing process may include the actual storage of the document in the system, but often documents are only stored partly, for

instance only the title and the abstract, plus information about the actual location of the document.

As shown in figure 2.2 , therefore, (b) representing documents using keyword terms by applying preprocessing techniques such as, tokenization, normalization , stop word removal and stemming is advisable and is also adopted in this thesis.

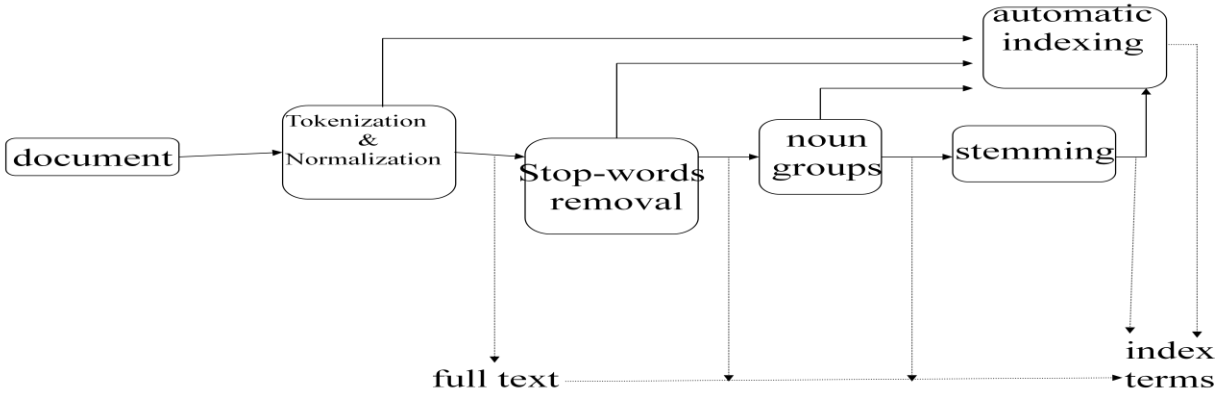

Figure 2.2: Logical view of a document: from full text to a set of index terms [16][2]

There are several index structures used for generating index terms [4]; such as, sequential file, inverted file, suffix tree, suffix array and signature file. We discuss only with the most popular indexing structure inverted file and sequential file structure.

A sequential file is the most primitive of all file structures. Sequential file is an indexing structure, which access elements of record in a predetermined ordered sequence. It has no directory and no linking pointers. The records are generally arranged serially one after another in lexicographic order on the value of some key. Even if it is easy to implement and provides fast access to the next record using lexicographic order, in this file structure it is difficult to update a record of large proportion of the file and random access is extremely slow [35].

The most popular indexing structure is inverted file, which is also adopted in this research. Inverted file stores a map from content to its locations in a database file. Inverted file is a mechanism for indexing a text collection so as to make the searching task fast. There are two elements involving in building the inverted file [16]: the vocabulary and the occurrence. The vocabulary file is the set of index terms in the text collection and it is organized by terms. The vocabulary file stores all of the keywords that appear in any of the documents in

[2] The internal structure normally present in a document and query indexing which is adopted from R. Baeza-Yates et al

lexicographical order and for each word a pointer to posting file. The occurrence contains one record per term, listing all the text locations where the words occur and frequency of each term in a document [4].

2.3.2 Query Processing

Users do search just for a need of information. The process of representing their information need is often referred to as the *query formulation process*. The resulting representation is the query. In a broad sense, query formulation might denote the complete interactive dialogue between system and user, leading not only to a suitable query but possibly also to the user better understanding his/her information need: This is denoted by the feedback process in Figure 2.1 [16, 31].

Once the document is indexed and ready for retrieval process, the next step is to translate the information need of user into query language provided by the system. This process involves a serious of steps [27]. First, the user specifies his/her information need using the natural language (e.g. English, Amharic, Tigrinya etc.) supported by the IR system. Second, the system transforms the query into logical format by applying text operation, which is also used when the document was indexed. To refine representation of users information need and improve effectiveness of the system, query operation will be employed which is discussed later in section 2.5. Finally, the query is processed to retrieve relevant documents.

2.3.3 The Matching Process

The comparison of the query against the document representations is called the *matching process*. The matching process usually results in a ranked list of documents. Users will walk down this document list in search of the information they need. Ranked retrieval will hopefully put the relevant documents towards the top of the ranked list, minimizing the time the user has to invest in reading the documents [31].

The result of the matching is a ranked list of documents according to their likelihood of relevance [27]. Baeza-Yates et al [16] stated that, one central problem of any information retrieval system is predicting which documents are relevant and which are not. The ranking algorithms are used for such decision. According to Hiemstra [31], the theory behind ranking algorithms is a crucial part of information retrieval system. They attempt to display documents in decreasing order based on their similarity score with the query. Most of the time documents that are considered as relevant to users gets the biggest score and displayed

at the top of the retrieved list. Thus, IR models guide the process of matching and ranking relevant documents.

The three classic models of Information retrieval: the Boolean model, the Vector space model and the probabilistic model are often used to accomplish these tasks. These models are briefly discussed below in the next section 2.4.

2.4 IR models

According [16] since the advent of information retrieval, a number of models have been developed to retrieve information in an effective and efficient way. Their mathematical basis spans a large spectrum, including set theory, algebra theory and probability theory. Depending on how they define and measure relevance, all the existing retrieval models are roughly classified into three broad categories. These are the Boolean model, the Vector space model and the probabilistic model.

The taxonomies of information retrieval models are depicted in figure 2.3 and we will discuss the three classic models.

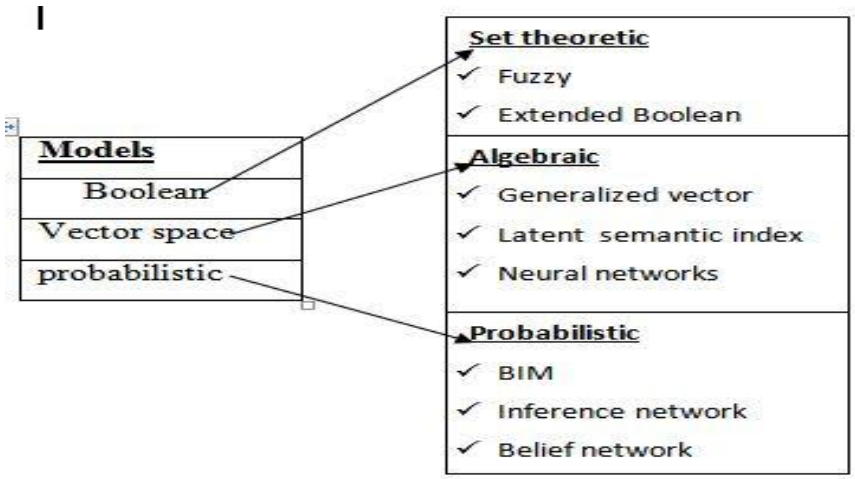

Figure 2.3: Categorization of information retrieval models with their alternative models mentioned to the right [16]

A Formal Characterization of IR Models is given as follow: Every information-retrieval model consists of four components [D, Q, F, R (q_i, d_j)] [16, 20, 28].

- D is a set composed of logical views (or representations) for the documents in the collection.
- Q is a set composed of logical views (or representations) for the user information needs. Such representations are called queries.

- F is a framework for modeling document representations, queries, and their relationships.
- R (q_i, d_j) is a ranking junction which associates a real number with a query q_i Є Q and a document representation d_j Є D. Such ranking defines an ordering among the documents with regard to the query q_i.

From the ranking function an ordering of the documents in D can be derived with respect to a particular query q_i from Q. The ranking function, and consequently the ordering of the documents, is different for each of the different models for information retrieval. In this section an overview of these models is presented.

2.4.1 The Boolean Model

The Boolean model is a simple retrieval model based on set theory and Boolean algebra [1]. The Boolean model of Information retrieval is the oldest of the three classic retrieval models and it relies on the use of Boolean operators [16]. In the Boolean model there are three basic logical operators AND, OR and NOT. AND is logical product, OR is logical sum and NOT is logical difference. AND is used to group set of terms in to single query/statement. The terms in a query are linked together with AND, OR and NOT. For example 'Information AND Technology' is two term query combined by 'AND'. In such case only document indexed with both terms will be *retrieved*. If terms in the user query are linked by operator OR, documented with either of terms or all terms will be retrieved. For example, if query is 'information OR technology' document containing information, or technology, or information Technology will be retrieved.

The Boolean model is based on Boolean algebra, which implies that the weight of an index term i associated with a document j can only have a value equal to 1 or to 0; w_{ij} ε {0,1}. In the Boolean model, the queries are specified as Boolean expressions. A query is thus composed of index terms which are linked by three Boolean operators: AND, OR, and NOT. According to the Boolean model a document is either relevant or non-relevant with respect to a particular query; there is no notion of grading. This implies that the similarity of a document d_j to a query q is binary, i.e., similarity (d_j ; q) ε {0,1}. The similarity of a document d_j to a query q is defined as

$$\text{sim}(dj, q) = \begin{cases} 1, & \text{if document satisfaies the boolean query} \\ 0, & \text{otherwise} \end{cases}_{i} \quad \text{.....equation 2.1}$$

What makes Boolean model good model is that it creates a sense of control to expert/user over the system. It is the user who is in charge for deciding what should or shouldn‘t be retrieved. Query reformulation is also simple because user is in charge of deciding what should be retrieved and should not.

However, the Boolean model has got its own drawbacks. Most users find it difficult to translate their information need into a Boolean expression [43]. It requires the users to have some knowledge of the search topic for the search to be effective [16]. A wrong word in a query could rank a relevant document as non-relevant. In addition to that, all retrieved documents are considered to be equally important.

In addition to, Boolean model may not retrieve anything if there is no matching document or, retrieves all documents if terms in query are matching with it. So there is no relevance judgment and ranking mechanism [6].

2.4.2 The Vector space model

Standard vector space model is one of the classic models of information retrieval. The drawback of binary weight assignments in Boolean model is remediated in the vector space model which projects a framework in which partial matching is possible [13]. Non-binary weights for index terms in queries and documents are used in the calculation of degree of similarity. Decreasing order of this degree of similarity for the retrieved documents gives the ranked documents with partial match.

The vector model [46] assigns non-binary weights to index terms in documents and in queries in order to achieve a better retrieval performance. A weight associated with index term k_i and document d_j is denoted by $w_{i;j}$, while a weight associated with index term k_i and query q is denoted by $w_{i;q}$. According to Salton and Buckley [44], the weights of the index terms appearing in the documents are computed as follows

$$W\,i,j = Fi,j * IDF,i \quad \text{.....equation 2.2}$$

where $F_{i,j}$ is the raw frequency of index term ki within document dj and IDF_i is the inverse document frequency for index term k_i. May we want to normalize *term frequency* Fi,j across the entire corpus:

$$\text{normalized}(Fi,j) = \frac{Fi,j}{\text{maxl }(Fi,j)} \quad \text{.....equation 2.3}$$

Where maxl(Fi, j) is the maximum term i frequency scored in document j. The inverse document frequency of an index term k*i* is computed as

$$\text{IDF}, i = \log N/ni \quad \text{.....equation 2.4}$$

Where N is the total number of documents in the collection and n*i* is the number of documents in which the index term k*i* appears we call it document frequency. For the calculation of the weights of the index terms of the query Salton and Buckley (1988) suggest the formula

$$\text{W}i; q = \left(0.5 + \frac{0.5 * \text{Freq } i;q}{\max l \text{Freq } l;q}\right) * \log N/ni \quad \text{.....equation 2.5}$$

Where Freq$i; q$ is the frequency of the term ki in the query q. Using these weights, a document can be defined as $\overrightarrow{dj} = (w1; j,\ w2; j,\ \ldots, wt; j)$ and a query can be defined as $\vec{q} = (w1; q,\ w2; q,\ \ldots, wt; q)$. Although Salton and Buckley [44] also suggest other ways of calculating both w*i;j* and w*i;q* , the above formulas provide a rather good weighting scheme. From these weight vectors the similarity between a document and a query can be computed as follows.

$$\text{sim}(dj, q) = \frac{\overrightarrow{dj}.\vec{q}}{|\overrightarrow{dj}||\vec{q}|} \quad \text{.....equation 2.6}$$

$$sim\ (dj, q) = \frac{\sum_{i=1}^{t}(\text{w}i,j)(\text{w}i,q)}{\sqrt{\sum_{i=1}^{t}(\text{w}i,j)^2}\sqrt{\sum_{j=1}^{t}(\text{w}i,q)^2}} \quad \text{.....equation 2.7}$$

Where $|\overrightarrow{dj}|$ and $|\vec{q}|$ are the norms of the document and query vectors.

The vector model has three main advantages [16]. First, by weighing the index terms the information retrieval performance is improved. Second, because partial matching is allowed, also documents that approximate the query can be retrieved. Third, by using the degree of similarity, documents can be ranked according to their degree of similarity to the query. The disadvantage of the vector model is that the index terms are assumed to be mutually independent.

2.4.3 The Probabilistic Model

In this subsection we focus on probabilistic models. The distinguishing characteristic of probabilistic models is that their framework for modeling documents and queries is based on probability theory.

In IR, probabilistic modeling refers to the use of a model that ranks documents in decreasing order of their evaluated probability of relevance to a user's information need. But 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 [16, 17].

Maron and Kuhns first suggested the probabilistic retrieval model [47]. The basic idea is to rank the documents in a collection based on their probability of being relevant to the current information need.

Then after, Robertson and Sparck Jones proposed a probabilistic retrieval model based on the distribution of query terms in relevant and non-relevant documents [48].

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) [16, 17]. Probability ranking principle asserts that [49]

"If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, then overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data."

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 (BIM), Inference Network Model and Belief Network Model [16].

2.4.3.1 Binary Independent model (BIM)

Probabilistic retrieval provides formal models for incorporating end-user relevance evaluations into the search process. The simplest of these models is based on the presence or absence of independently distributed terms in relevant and non-relevant documents, and is referred to as the binary independent (BI) model [28]. The BIM was introduced in 1976 by

Roberston and Sparck Jones [19]. BIM we present in this section is the model that has traditionally been used with the Probability ranking principle (PRP).

The BI assumptions define a relevance weight, which is a function of the relative likelihood of a term appearing in relevant and non-relevant documents.

In BIM, binary is equivalent to Boolean; queries and documents are represented as binary incidence vectors of terms. D={d1,d2,….,dn} where, di=1 if term i is present in document d and di=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.

2.4.3.2 Bayesian Networks Model

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.

2.4.3.2.1 Bayesian Inference Network Model

According to Baeza-Yates et al [16], 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 [16].

2.4.3.2.2 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 [28]. As Baeza-Yates et al [16], 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 [16].

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[16].

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 [16,32].

However, probabilistic model have several potential advantages [16]. 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).

2.5 Query Operation

Without detailed knowledge of the collection make-up and of the retrieval environment, most users find it difficult to formulate queries which are well designed for retrieval purposes. In fact, the users might need to spend large amounts of time reformulating their queries to accomplish effective retrieval. This difficulty suggests that the first query formulation should be treated as an initial (naive) attempt to retrieve relevant information. Following that, the documents initially retrieved could be examined for relevance and new improved query formulations could then be constructed in the hope of retrieving additional

useful documents. Such query reformulation involves two basic steps: expanding the original query with new terms and reweighting the terms in the expanded query.

For the query operations to happen relevance feedback and query reformulation is necessary. Relevance feedback enables to identify relevant document retrieved for the users query and query reformulation enables to expand the original query with the new terms and reweight the terms in the expanded query so as to retrieve relevant documents which satisfy user's information need. There are various relevance feedback and query reformulation mechanisms as discussed below [16, 17, 4, 50, 52].

2.5.1 Relevance Feedback

Relevance feedback is a mechanism of engaging users or system in retrieval process so as to improve the final result of the IR system. The users tend to ask short queries, even when the information need is complex. Irrelevant documents are retrieved as answers because on the ambiguity of the natural language (words have multiple senses). If we know that some of retrieved documents were relevant to the query, new terms from those documents can be added to the query or the terms are reweighted in order to be able to retrieve more relevant documents. This is called relevance feedback. Often, it is not possible to ask the user to judge the relevance of the retrieved documents.

There are two relevance feedback mechanisms [16, 17, 19, 20], user relevance feedback and pseudo relevance feedback. User relevance feedback is used to improve the final result of the IR system by involving the users in relevance feedback during the retrieval process. The procedures followed in user relevance feedback are the following. First, the user provides a query based on 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 [16, 28].

2.5.2 Query Reformulation

Query reformulation is a mechanism used to enhance the performance of the retrieval system by using two different methods called query expansion and term reweighting [1, 16].

Query expansion technique is a process of adding a new term from relevant documents. There are two types of query expansion strategies [16]: global analysis and local analysis. Global analysis strategy examines all documents in the collection so as to expand query. Local analysis examine only documents retrieved automatically for a given query q to determine query expansion.

Term reweighting technique is a process of adjusting the weight of the term based on the users or system relevance judgment. There are different techniques of term reweighting [1]. Rocchio algorithm, probabilistic term reweighting etc [16].

Rocchio algorithm is one of the most widely used algorithms designed for vector space model. It finds a query vector which increases similarity with relevant document while decreases similarity with non-relevant documents [50].

Probabilistic relevance weights can be estimated from the relevant and non-relevant documents retrieved in an initial search and can be used in the next iteration of the search to improve retrieval performance. Such investigations test the benefits of modifying query statements and relevance weights in a feedback process [35]. It attempts to predict the probability that a given document will be relevant to a given query. The similarity of document dj to a query q can be expressed as [16]:

$$\text{Sim}\,(dj, q) \propto \sum_{i=1(wi,q)(wi,j)}^{t} \left(\log \frac{p(ki\ /R)}{1-p(\frac{ki}{R})} + \log \frac{p(ki/\overline{R})}{1-p(ki/\overline{R})}\right) \text{.....equation 2.8}$$

Where, P(ki|R) express the probability of getting term ki in the relevant documents of R and P(ki|R) represent the probability of getting term ki in the non-relevant documents of $\overline{R}$. However, initially equation 2.8 is not used because of the probabilities of P (ki/R) and P(ki/$\overline{R}$) are unknown. For the initial search (when there are no retrieved documents yet), two assumptions often made include: (a) P(ki/R) is constant for all terms ki (typically 0.5) and (b) the term probability distribution P(ki/$\overline{R}$) can be approximated by the distribution in the whole collection. The two assumptions are expressed as:

$$P\ (ki\ /R) = 0.5\ and\ (ki/\overline{R}) = \frac{ni}{N} \text{.....equation2.9}$$

Where, ni stands for the number of documents in the collection which contain the term ki. Substituting into equation 2.8, we obtain [16]:

$$Sim_{initial}(\ ,q)=\sum_{i}^{t}(wi,q)(wi,j)\ \ \log\frac{N-ni}{ni}$$ ……..equation 2.10

For the feedback searches, the accumulated statistics related to the relevance or non-relevance of previously retrieved documents are used to evaluate the probabilities $p(ki/R)$ and $p(ki/\overline{R})$. According to the user judgment let Dr be the set of relevant retrieved documents and Dr,i be the subset of Dr composed of the documents which contain the term ki Then, the probabilities $p(ki/R)$ and $p(ki/\overline{R})$ can be approximated by

$$p(ki/R)=\frac{|Dr,i|}{|Dr|}\quad ;\quad p(ki/\overline{R})=\frac{ni-|Dr,i|}{N-|Dr|}$$ ……equation 2.11

Using these approximations, equation 2.8 can rewritten as

$$Sim(dj,q)=\sum_{i}^{t}(wi,q)(wi,j)[\log\frac{|Dr,i|}{|Dr|-|Dr,i|}\ \div\frac{(ni-|Dr,i|)}{N-|Dr|-(ni-|Dr,i|)}]$$ ....equation 2.12

Notice that here; contrary to the procedure in the vector space model, no query expansion occurs. The same query terms are being reweighted using feedback information provided by the user or system.

Equation 2.11 poses problems for certain small values of |Dr| and |Dr,i| that frequently arise in practice (Dr| = 1,|Dr,i| =0). For this reason, a 0.5 adjustment factor is often added to the estimation of P(ki/R) and $p(ki/\overline{R})$ yielding

$$P\ (ki/\ R)=\frac{|Dr,i|+0.5}{|Dr|+1}\quad ;\quad p(ki/\overline{R}=\frac{ni-|Dr,i|+0.5}{N-|Dr|+1}$$ ---------- equation 2.13

The adjustment fact made at equation 2.13 may provide inadequate estimation in some cases. In this case, alternative adjustment factors have been formulated. For instance, replacing adjustment factor 0.5 by ni/N

$$P\ (ki/\ R)=\frac{|Dr,i|+ni/N}{|Dr|+1}\quad ;\quad p(ki/\overline{R}=\frac{ni-|Dr,i|+ni/N}{N-|Dr|+1}$$ ---------- equation 2.14

2.6 IR Systems Evaluation

Evaluation of information retrieval system is done before the system is implemented [16]. Several reasons are stated why evaluation is needed in IR. For instance, evaluation provides the ability to [28]. 1) Validate and verify the system to check whether the system is right or not. 2) Measure which one is the better system than the other one. 3) Measure how good the IR system works. 4) Identify techniques/ algorithms that work well and do not work. 5)

Identify specific components of techniques or algorithms that work better. 6) Provide future direction for further studies.

IR evaluation is highly related to the concept of relevance [48]. Relevance is the degree of correspondence between retrieved documents and information need of users. Judgments are produced by human judges and included in the standard test collections [47]. In order to evaluate the performance of an IR system we need to measure the ranked list of results of the relevant documents [32]. The typical evaluation process starts with finding a collection of documents. A set of queries needs to be formulated. Then one or more human experts are needed to exhaustively label the relevant documents for each query. This assumes binary relevance judgments: a document is relevant or not to a query. This is simplification, because the relevancy is continuous: a document can be relevant to a certain degree. Even if relevancy is binary, it can be a difficult judgment to make. Relevancy, from a human standpoint, is subjective, situational, it might be dynamic, and it can change over time.

In IR system evaluation, the two common measure of system performance are efficiency and effectiveness. Efficiency is the time and space used by the system in retrieval process. To be called efficient system, the retrieval and indexing time of the system should be shorter and the space used in indexing file should be smaller. On the other hand, effectiveness refers how much the system meets its designed objective. To be called effective system the system should be capable of retrieving relevant documents from the collection and the system should retrieved documents that satisfy users need [16] and the discussion proceeds with effectiveness system performance measures, because this research is aimed at evaluation of system retrieval electiveness.

Once the test collection is assembled, we can compute numerical evaluation measures, for each query, and find an average over all of the queries in the test set [16, 19, 51, 52]: see below all

Precision and Recall

Precision (P) measures the ability to retrieve top-ranked documents that are mostly relevant.

Recall (R) measures the ability of the search to find all of the relevant items in the corpus.

$$P = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}}$$

$$R = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents}}$$

F-measure and E-measure

The F-measure combines precision and recall, taking their harmonic mean. The F-measure is high when both precision and recall are high.

$$F = \frac{2PR}{P + R} = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

A generalization of the F-measure is the E-measure, which allows emphasis on precision over recall or vice-versa. The value of the parameter β controls this trade-off: if β = 1 precision and recall are weighted equally (E=F), if β < 1 precision weights more, and if β > 1 recall weights more.

$$E = \frac{[1 + \beta^2]PR}{\beta^2 P + R} = \frac{[1 + \beta^2]}{\frac{\beta^2}{R} + \frac{1}{P}}$$

Average Precision

Usually precision is more important than recall in IR systems, if the user is looking for an answer to a query, not for all the possible answers. Recall can be important when a user needs to know all the relevant information on a topic. A system can increase precision by decreasing recall and vice-versa; there is a precision-recall tradeoff (for example, recall can be increased by simply retrieving more documents, but the precision will go down, since many retrieved documents will not be relevant). Precision-recall curves can be used to compare two IR systems for all values of precision and recall [16].

Some good measures are: precision at 5 retrieved documents, precision at 10 retrieved documents or some other cut-off point; the R-Precision; the interpolated average precision; and the mean average precision.

The *R-precision* is the precision at the R-th position in the ranking of the results for a query that has R known relevant documents.

The *interpolated average precision* computes precision at fixed recall intervals (11 points), to allow fair average over all the queries in the test set at the same recall levels [16]. This measure is in use lately in evaluating IR systems.

The most widely-used measure is the *mean average precision* (MAP score) [1,5,13,16,19, 24]. It computes precision at each point in the ranking where a relevant document was found, then averages over these values (and then over all queries).

2.7 Review of previous Related IR Works

From reviewed literatures there is no researches made yet on IR system for Tigrinya language.

In this section the discussion proceeds with related IR Systems for international and local researches made so far.

2.7.1 Related IR Systems for International Languages

Some probabilistic information retrieval researches that have been studied so far, which results several body of literature on the topic and the related works are given below.

Probabilistic retrieval provides formal models for incorporating user or system relevance feedback. This is possible based on the presence or absence of independently distributed terms in relevant and non-relevant documents, and is referred to as the binary independent model (BIM)[19, 35]. The BIM assumptions define "a term relevance weight, which is a function of the relative likelihood of a term appearing in relevant and non-relevant documents "[56]. Term relevance weights can be estimated from the relevant and non-relevant documents retrieved in an initial search and can be used in the next iteration of the search to improve retrieval performance. Such investigations test the benefits of modifying query statements and term weights in a feedback process. In test collections for which exhaustive relevance evaluations provide complete inventories of relevant and non-relevant documents for a set of queries, term relevance weights can be computed definitively. Outcomes of such investigations establish the optimal performance of the retrieval model [19, 56].

When we back to the probabilistic model background, Maron and Kuhns [47] first suggested the probabilistic retrieval model. The basic idea is to rank the documents in a collection based on their probability of being relevant to the current information need. This is expressed as P(r/N), or the probability that the information need is met given document N. A user's information need is something internal to the user and cannot be expressed exactly to the system, so this probability must be estimated using the terms supplied by the user in a query. The estimation is simplified using Bayes' theorem to rewrite the probability as;

$$p(r|N) = \frac{p(r|N)p(r)}{p(N)} \quad \text{.....equation2.15}$$

Where r= relevant, N=total documents in the corpus

Document N can be represented as a binary vector x= (x1,x2,x3,…,xn), where xi=1 if term i appears in document N, xi=0 if otherwise, and the terms are limited to those that appear in the query. Now the estimation task amounts to estimating the probability of the terms appearing in a relevant document, p(N|r), and the a priori probability of a document, p(N). P(r) will be constant for a given query and so may be ignored.

Then after, Robertson and Sparck Jones [19], developed the well-known classical probabilistic model called the Binary Independence Retrieval model so as to estimate the probability of relevance for a given query q. (i.e. P(r |N) or p(nr| N)). Where nr=non relevant, r=relevant, N=total documents

Robertson and Sparck Jones work was based on two concepts: Independence assumptions and ordering principle.

Independence assumptions: assumed that, "terms are distributed independently and randomly". Specifically

Independence assumptions one (I1): stated that "distribution of terms in relevant documents is independent and their distribution in ***all*** documents is independent"

Independence assumptions two (I2): stated that "distribution of terms in relevant documents is independent and their distribution in ***irrelevant*** documents is independent"

Ordering principle: The ordering principle states, "documents should be ordered by their probable relevance to the query". Specifically,

Ordering principle one (O1): it stated that, "probable relevance is based only on the presence of search terms in documents" and

Ordering principle two (O2): it stated that, "probable relevance is based on both the presence of search terms in documents and their absence from documents".

Taking independence assumption and ordering principle together, the theory yields four specific weighting functions as shown in table 2.2.

	Independence Assumption I1	Independence Assumption I2
Ordering principle O1	F1	F2
Ordering principle O2	F3	F4

Table 2.2: weighting functions

The weighting functions mentioned above have been characterized in a fairly superficial way.

In fact, all four functions (F1-F4) derive from a formal probabilistic theory of relevance weighting [19]. The object of this theory is to drive an optimal ranking of the documents in a collection, on the basis of presence or absence of in each document of the request terms, when some information average performance of these is available.

For each of the weighting functions (F1 To F4), r_i = number of relevant documents containing t_i , n_i = number of documents containing t_i, R = number of relevant documents, N= number of documents in collection.

F1 formula: $w1 = \log \frac{r/R}{n/N}$, **F2 formula:** $w2 = \log \frac{r/R}{(n-r)/(N-R)}$,

F3 formula: $w3 = \log \frac{r/(R-r)}{n/(N-n)}$, **F4 formula** : $w4 = \log \frac{r/(R-r)}{(n-r)/(N-n-R+r)}$

Experimentally a comparison was also made for F1 to F4, for a large collection. The experiment was conducted using manually indexed Cranfield 1400 document collection written in English language. The result of this experiment is summarized in Table 2.3.

Weighting function	Precision	Recall
F1	50%	90%
F2	60%	90%
F3	66%	80%
F4	70%	80%

Table 2.3: Experimental result of Robertson and Sparck Jones work

Weighting function one F1is based on the simplest and most obvious choices of assumption and principle, while F4 derives from more complex and less obvious ones. However, they argue ordering principle O2 is correct and I1 is incorrect and independence assumption two I2 is likely to describe reality more closely than I1. The performance also shows that F3 and F4 performed consistently better than F1 and F2. On the other hand, the experiment depict that, the performance of the system improves when information about the occurrences of terms in relevant documents is added to information about their simple document incidence. Specifically, relevance weights give a better performance than simple term matching [19]. Thus, the theory predicts that F4 is the best of the four functions.

Several attempts have been also made to improve the binary independent representation. For instance, W. M. Shaw Jr [14] tries with the concept of term-relevance computations and perfect retrieval performance on the CF database by modified the convention computing equations for binary independent (BI) term relevance weights.

First the author tries to see and interpret the binary independent relevance weights in terms of the conventional 'contingency' table 2.4.

	No. of relevant documents	No. of non-relevant documents	Total
No. of documents including term k	r	n-r	n
No. of documents excluding term k	R-r	*N - n - R + r*	*N - n*
Total	R	*N - R*	*N*

Table 2.4: Symbolic representation for the number of relevant and non-relevant documents including and excluding term k

According to W. M. Shaw Jr [14] the relevance weight of term k, denoted wk, can be derived from a 2x2 contingency table in which the number of relevant and non-relevant documents including or excluding term k is represented symbolically. The number of documents in the four cells and marginal totals of table 2.4 is expressed in terms of four variables: the total number of documents (N), the total number of relevant documents (R), the number of documents in which term k appears (n), and the number of relevant documents in which term k appears (r).

"The probability term k appears in a relevant document, denoted by Pk, and the probability term k appears in a non-relevant document, denoted by u_k*, are defined respectively by* $\frac{r}{R}$ *and* $\frac{(n-r)}{(N-R)}$ *.The probability term k appears in a relevant document, defined by* $\frac{pk}{(1-pk)}$ *and the probability term k appears in a non-relevant document, defined by* $\frac{uk}{1-uk}$ *can be written in terms of the empirical frequencies and are given respectively by* $\frac{r}{R-r}$ *and* $\frac{n-r}{N-R-n+r}$*. If the probability is high that term k appears in relevant documents and low that term k appears in non-relevant documents, the presence of term k can discriminate the few relevant documents from the many non-relevant documents in a large document collection, which is the distinguishing characteristic of the desired term relevance function. Varying from zero to*

infinity, with one signifying equal probability, the ratio of the odds term k appears in a relevant document to the odds term k appears in a non-relevant document is the basis for the relevance function. The logarithm of the odds ratio produces a symmetric scale and constitutes the relevance weight (wk):

$$wk = log_e \left[\frac{\frac{pk}{1-pk}}{\frac{uk}{1-uk}} \right] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textit{equation 2.16}$$

Values of the term relevance function appear in the range $-\infty \leq wk \leq +\infty$*. When the odds term k appears in a relevant document are equal to the odds term k appears in a non-relevant document, wk = 0. A positive value of the term relevance function (wk > 0) indicates the odds favor term k appearing in a relevant document, and a negative value (wk < 0) indicates the odds favor term k appearing in a non-relevant document".*

The relevance function, subject to BIM assumptions, can also be derived from a formal model based on Bayesian probability theory. In the formal model, the logarithm of the odds ratio causes the value of documents to be an additive function of relevance weights (wk).

Although the meaning of relevance weights is conceptually ; computations of P*k*, u*k*, and w*k* can present difficulties, even with prior knowledge of all relevant documents; P*k* is undefined if R=0, uk is undefined if *N-R=0,* and w*k* is undefined if either P*k* or u*k* equals one or zero. Statistical theory has been invoked to resolve the problem of undefined values; leading to computing formulas of the form given in equations 2.9 and 2.10, where c is an adjustable parameter.

$$pk = \frac{r+c}{R+1} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{equation 2.17}$$

$$uk = \frac{n-r+c}{N-R+1} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{equation 2.18}$$

W. M. Shaw Jr [14] states that "*The theory demonstrates that the logarithm of* $\frac{\boldsymbol{r+c}}{\boldsymbol{(R-r+c)}}$ *is an unbiased estimate of the logarithm of* $\frac{\boldsymbol{pk}}{\boldsymbol{(1-pk)}}$ *and that the logarithm of* $\frac{\boldsymbol{(n-r+c)}}{\boldsymbol{(N-R-n+r+c)}}$ *is an unbiased estimate of the logarithm of* $\frac{\boldsymbol{uk}}{\boldsymbol{(1-uk)}}$ *when c = 0.5. Statistical theory allows one half to be added to each of the four cells of table 2.5 to guard against the effect of small cell frequencies on certain statistical calculations and causes one to be added to the marginal totals of the table. Consequently, the conventional computing formulas for pk and uk are defined by equation 2.17 and equation 2.18, with c=0.5. The conventional computing formulas do not allow wk to be undefined when r is equal to R or zero, or when n-r is equal to*

N-R or zero. Statistical theory does not, however, insure that equation 2.17 and equation 2.18, with c=0.5, provide unbiased estimates of pk and uk, or that subsequent computations of wk are meaningful in the present context".

The conventional computing formulas [equation 2.17 and equation 2.18, with c=0.5] can over estimate P*k* and w*k*, particularly when R is small [20]. Consider, for example, a database with any number of documents and a query with one relevant document (R= 1) and suppose term *k* does not appear in the relevant document (r=0). Term *k* can appear in almost one-quarter of the non-relevant documents and still yield a positive term relevance weight, suggesting that the term is more likely to appear in a relevant document than a non-relevant document. In this case, term *k* enhances the value of many non-relevant documents. The precision of a search for a few relevant documents can be diminished by the conventional computing equations.

The conventional computing formulas can also over estimate Pk and wk, when R is large. If term k appears in one non-relevant document and no relevant document (n= 1 and r=0), the relevance weight of term *k, wk,* is positive, when almost one-quarter of the documents are relevant. The conventional computing equations can produce illogical outcomes when c=0.5 dominates the computation of pk [14].

Robertson [51] has demonstrated that term relevance weights (wk) are logically equal to zero when no relevance information is available, if $\boldsymbol{c = \frac{n}{N}}$. For large databases and most retrieval test collections, $\frac{\boldsymbol{n}}{\boldsymbol{N}}$ is likely to be small compared to 0.5, and the modified computing formulas [equation 2.17 and equation2.18, with $\boldsymbol{c = \frac{n}{N}}$**]** can be expected to resolve the problems of undefined and over estimated values of wk, in most cases. However, if the universe of documents *(N)* is composed of those retrieved and evaluated by an end-user, for the purpose of initiating a feedback process, it would not be surprising to find a term appearing in all documents. If term k appears in all documents, n= *N*, r = R, Pk = uk = 1, and wk is undefined. Applying the modified computing formulas to a small set of subject related documents for the purpose of computing relevance weights could lead to mathematical singularities in the absence of further modifications of c.

It is unnecessary and inappropriate, however, to modify all calculations of defining equations for Pk and uk to resolve isolated mathematical singularities. Introduced here is a third set of computing equations for Pk and *uk,* in which singularities are treated as special cases. If pk=0, including the case when R=*0,* let $p\boldsymbol{k = \frac{1}{(N)^2}}$; similarly, if uk=0, including the case

when *N-R=0*, let $uk = \frac{1}{(N)^2}$. If Pk or uk equals one, set the probability to $\left(1 - \frac{1}{(N)^2}\right)$. The square of collection size insures that probabilities of magnitude zero are reasonably estimated in a small set of retrieved documents or a small test collection. Equation2.19 and equation2.20 alter the defining equations only as needed and resolve previously described computational difficulties.

$$pk = \begin{cases} \frac{r}{R}\left[\frac{1}{(N)^2}\right], if\ r = 0 \\ \frac{r}{R}\left[1 - \frac{1}{(N)^2}\right], if\ r = R \end{cases} \quad \text{.....equation2.19}$$

$$uk = \begin{cases} \frac{n-r}{N-R}\left[\frac{1}{(N)^2}\right], if\ n - r = 0 \\ \frac{n-r}{N-R}\left[1 - \frac{1}{(N)^2}\right], if\ n - r = N - R \end{cases} \quad \text{.....equation2.20}$$

For the experiment three sets of computing equations were evaluated: these were

1. Equation 2.17 and 2.18 when c=0.5:
2. Equation 2.17 and 2.18 when c=n/N:
3. Equation 2.19 and Equation 2.20

Experiment was held on a subset of the National Library of Medicine's MEDLINE file, referred to as the CF database [14]. The CF document collection includes 1239 English language documents published from 1974 to 1979.

The CF database includes 100 queries with exhaustive relevance evaluations from physicians and scientists involved in CF care and research [57]. The principal author of the queries, a group of faculty colleagues, and a group of post-doctoral associates examined the full text of each document and judged the document to be "highly relevant," a direct response, "marginally relevant," topically related, or "not relevant" to each query. Consequently, the relevance representation then controlled in the CF database, and two distinct representations were examined in this investigation. These were

(1) "*A document was considered relevant to a query if at least one expert considered it to be topically related or a direct response to the query; the associated search was referred to as comprehensive, because marginally relevant, as well as highly relevant, documents are sought"*.

(2) "*A document is considered relevant to a query if at least two of three experts considered it to be a direct response to the query; the associated search is referred to as specific, because only highly relevant documents are to be retrieved*".

In table 2.5 average values of recall (R), precision (P), and effectiveness (E) for three sets of term relevance weight equations as a function of query representations and retrieval expectations are given.

		Retrieval Performance Evaluation For the three Computing Equations								
Query	Retrieval expectation	Equation2.17 &2.18 when c=0.5			Equation2.17 &2.18 when c=n/N			Equation2.19 &2.20		
		R	P	E	R	P	E	R	P	E
Query terms	Comprehensive	0.445	0.534	0.557	0.450	0.532	0.554	0.450	0.533	0.554
	Specific	0.566	0.531	0.515	0.587	0.590	0.467	0.589	0.590	0.466
All terms	Comprehensive	0.560	0.651	0.430	0.996	0.989	0.008	1.000	0.996	0.002
	Specific	0.540	0.395	0.642	1.000	0.994	0.003	1.000	0.999	0.001

Table 2.5: Retrieval Performance Evaluation for the three Computing equations in CF database

As the result shows, the binary independence probabilistic model appears to be capable of producing perfect retrieval results for document and query representations composed of simple word stems.

Deficiencies of conventional computing equations, particularly equation 2.17 and equation 2.18, in the BIM and the merits of alternative formulations, particularly equation 2.19 and equation 2.20, are likely to be revealed by query expansion, which appears to be essential to the success of probabilistic retrieval and relevance feedback [43,58].

Thus, term relevance weights derived from the BIM have been investigated by W. M. Shaw Jr [14]. Analyses reveal the optimal performance of three sets of term relevance computing formulas, as a function of query representations and retrieval expectations.

Optimal performance of conventional computing equations 2.17 and 2.18 overestimate some term relevance values, and modified equations resolve the over estimation problem. This is possible when the query is constrained to include only the few, general terms of a typical query statement. Optimal retrieval effectiveness is mediocre. Feedback operations adjust only the weights of a few, fixed query terms cannot be expected to raise retrieval performance

beyond the standard of mediocrity. Conventional equations continue to produce mediocre levels of performance when the query is expanded to include all terms in the database [14].

Modified equations, particularly equation 2.19 and equation 2.20, produce essentially perfect levels of performance for both comprehensive searches and specific searches when the query representation includes all terms in the database. The theoretical limit of the BIM is perfection. A feedback process, in which discriminating terms are added to the original query terms and term relevance weights are accurately computed experimental outcomes [14].

In addition to the above, other research by B.-H. Cho, et al. [48] tries to see term dependences in probabilistic information retrieval model.

This was an attempt that has been made to improve the binary independent representation with different test collection. For instance a new method of incorporating term dependence in probabilistic retrieval model was by adapting Bahadur–Lazarsfeld expansion (BLE) to compensate the weakness of the BIM assumption by extending the 2-poison model [48]. This was through the experiments on two standard document collections, HANTEC2.0 in Korean and WT10g in English; they demonstrate that incorporation of term dependences using the BLE significantly contribute to the improvement of performance in at least two different language IR systems. From the results, a statistically significant improvement of performance was obtained on both document collections by incorporating the term dependences using the BLE [48].

They conclude that incorporating term dependences using BLE into a 2-Poisson model was a viable and appropriate technique to overcome inconsistency under the linked dependence assumption model.

The greatest disadvantage in using the BLE was, nevertheless, it has been clarified by [59, 60, 61) that incorporation of a term dependence model actually improved the performance, when a higher order model of term dependence is used, the easily reached formal representation of the model (in fact, the greatest merit of term independence) cannot be maintained and we become extremely difficult estimating the probabilities of the model.

In addition to, the retrieval cost was very high because co-occurrence information between the two terms must be obtained at the search time when the user query was given. The longer the size of the query, the greater the number of term pairs which incur a much higher retrieval cost. To reduce this cost, useless term pairs can be removed at a certain threshold. Because

this was not the essential solution, that's way they recommended for the future to develop effective algorithms or auxiliary DBs to pre-obtain the co-occurrence information.

2.7.2 IR Systems for other Ethiopian Languages

The review also has been done to find out work done for local languages in Ethiopia. Very limited research works conducted so far on Amharic IR, Afaan Oromo IR, Amharic–English and Afaan Oromo-English CLIR. But nothing is found for Tigrinya IR, except one study Tigrinya–Amharic CLIR by Tsegay in 2013.

When we came to local researches, a number of IR studies have been conducted so far for Amharic language but in probabilistic model only one had been developed by Amanual in 2012.

Amanuel [11], tried to design and develop a probabilistic based information retrieval system for Amharic language based on the probabilistic model. To test the prototype system developed, 300 Amharic News articles were used as a document corpus. All news articles are obtained from the web site of Walta Information Center. Additionally, 10 test queries were selected by the researcher to test the performance of the system. In designing the IR system, Binary Independent Model (BIM) model was selected and implemented. System evaluation was based on the F-measure and registered on the average of 73% F-measure.

Tessema and Solomon [23] designed and implemented Amharic search engine, which retrieve web documents written in Amharic language. Even if general search engine such as Google, Yahoo have the way to accept Amharic query and retrieve relevant document, they simply match patterns without considering the feature of Amharic language that affect the retrieval performance. In this research, a complete language specific process has been done, such as, crawler, indexer and query engine component. The experiment result shows, the average precision and recall of 99% and 52% respectively. In future work, the need for considering additional features of Amharic text was recommended.

In addition to these mentioned above, numerous Amharic and Oromifa IR systems had developed like Tewodros [23] had tried to see the performance of Amharic text retrieval using latent semantic indexing (LSI) with singular value decomposition and Hassen Redwan et al [2] developed search engine for Amharic web content. These are few to mention.

An attempt also was made to develop Tigrinya-Amharic Cross lingual Information Retrieval (CLIR) systems which enable Tigrinya native speakers to access and retrieve the information that are available in Tigrinya and Amharic .

The performance of the system after User Relevance Feedback was 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 Tigrinya and Amharic languages respectively. Finally recommendation was drawn that the performance of the CLIR system can be improved by designing good stemmer for both languages [12].

But still this could not give satisfactory performance for the peoples of Tigrinya speakers as retrieving Tigrinya documents by Tigrinya query.

Generally CLIR is used to enable retrieval of document in specific language with query in other language different from the language in which the document was written [1, 16, 17]. But such systems may not consider the artifacts in the language, as a result of which they may not be as effective as possible to satisfy information need of users. So developing an IR system following modern IR principles that takes into account language related issues is necessary before designing a cross language retrieval system[1]. And in this research, apart from CLIR, a complete Tigrinya language IR specific process has been done with the same model.

There are some stemming researches developed for Tigrinya languages so far. For instance the first attempt to develop Tigrinya stemmer was made by Girma in 2001.Then developed by Yonas [26] in 2011.

In this paper the researcher used the rule based stemming algorithm developed by Yonas [26] which removes prefix and suffix in successively applied steps. The stemmer was evaluated using error counting method. The system was tested and evaluated based on the counting of actual understemming and overstemming errors using a total of 5437 word variants derived from two data sets. Results show that the stemmer had an average accuracy of 86.1%.

To generalize, several IR systems have been developed in the last decade, most of the works were attempt to design an Amharic IR system using vector space model except Amanual's work that have been used probabilistic model. However, the use of vector space model may not control uncertainty nature of IR system [1, 20, 30, 31]. Thus in this study an attempt is

made to develop probabilistic based Tigrinya IR system to check the performance of the rule-based Tigrinya stemmer developed by Yonas [26].

CHAPTER THREE

3.1 Probabilistic Tigrinya IR System Design and Architecture

By definition, an Information Retrieval system is a field concerned with the structure, analysis, organization, storage, searching, organization of, and retrieval to information"[1,16]. The function of any IR system is to process a user request for information and retrieve materials that have contents that could potentially satisfy the information need of the user.

According to the architecture of this system in Figure 3.1, the IR system takes both documents and queries as input. Two major distinct processes are involved; one is the indexing of documents and the other is the processing of queries (description and formulation). Queries must be represented by a set of terms just like documents before they can be matched with documents. Query formulation refers to the preparation of the query for input to the matching system.

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

Figure 3.1 depicts the probabilistic based IR system architecture designed and implemented in this research. It indicates that, the IR system has online (Searching) and offline (Indexing) processes. During the offline process, Tigrinya 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 online 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 [16, 20]. See the system architecture in figure 3.1.

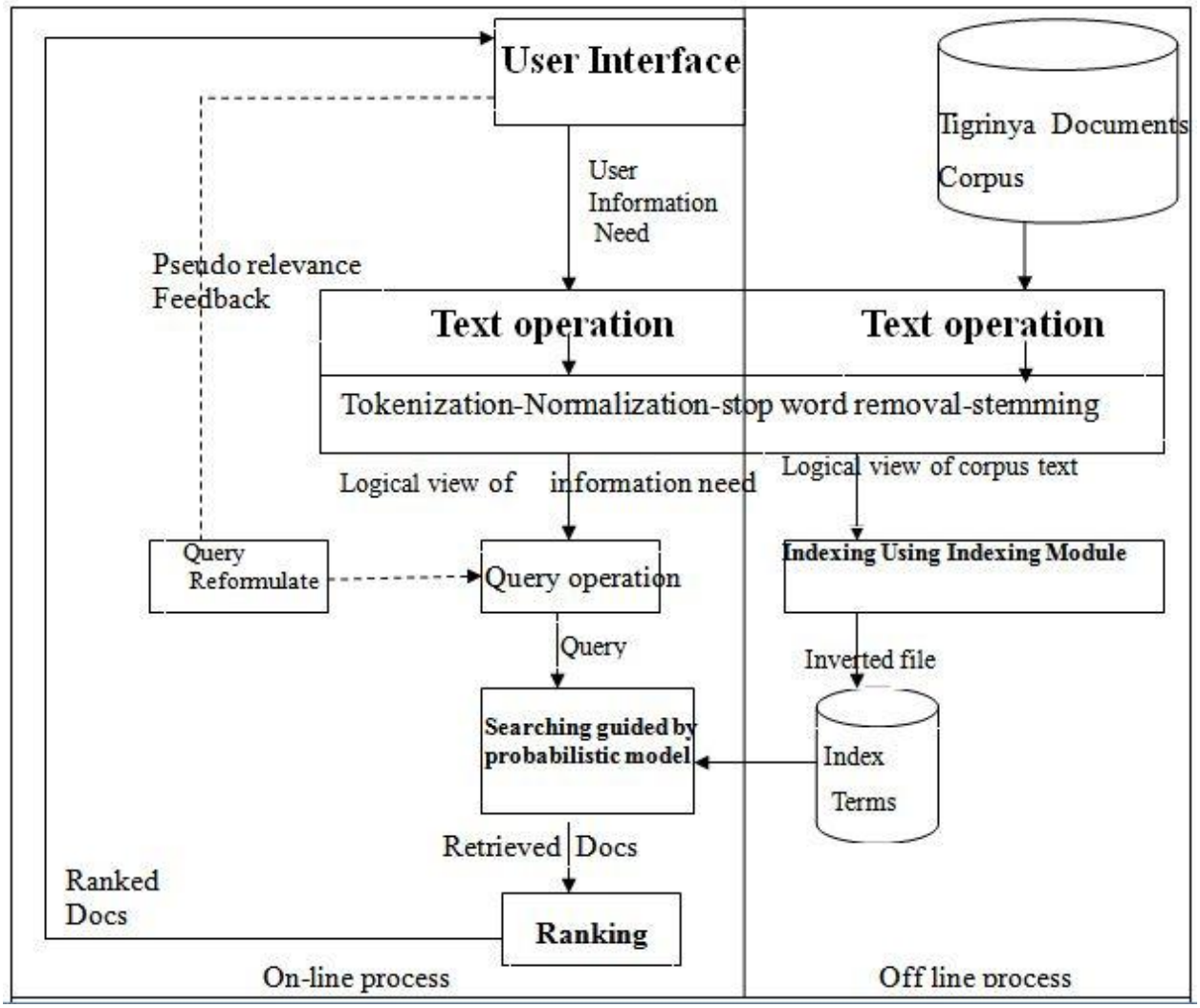

Figure 3.1[3]: Probabilistic based Tigrinya IR system architecture [16]

3.2 Data pre-processing and corpus preparation for Tigrinya Document Indexing

In Information retrieval system corpus is needed for evaluation of the system [28]. Tigrinya document collected from different news articles and other online resources passes through different procedures in order to index and use the retrieval system. Since 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. Matches are then more likely to be relevant, and since the documents are smaller it

[3] Explanation of Figure 3.1: Text operations are applied from text of the documents and on the description of the user information need in order to transform them in a simplified form needed for computation. The documents are indexed and the index is used to execute the search. After ranked documents are retrieved, the user can provide feedback which can be used to refine the query and restart the search for improved results.

will be much easier for the user to find the relevant once in the document [28]. Several works has been done so as to index the document collection/corpus used in this research. These tasks include; tokenization, normalization, stop-word removal, and stemming,

3.2.1 Tokenization

Tokenization is the process of chopping character streams in to tokens, while linguistic preprocessing then deals with building equivalent classes of tokens which are the set of terms that are indexed. Tokenization in this work also used for splitting document in to tokens and detaching certain characters such as punctuation marks. The numbers, punctuation marks and control characters in the text of each file were not considered for indexing because they content description [28]. All punctuation marks, control characters, numbers and special characters are removed from the text before the data is processed. All punctuation marks are converted to space and space is used as a word demarcation. Hence, if a sequence of characters is followed by space, that sequence is identified as a word. A consecutive sequence of valid characters was recognized as a word in the tokenization process. The algorithm given below is for tokenization

```
Open the file/corpus for processing
Create string container
Do
        Read the content of the file line by line and split to string by space
        Put to container for each strings
        For word in container
                If word contains punctuation marks, numbers, special characters
                        Replace them with a space
      End for
While end file
```

Algorithm3.1: Tokenization [15]

The above algorithm tokenizes the text documents as follows: first, the content of file is read line by line. Second, split them by space in to list of words. Third, check whether the word within the list contains punctuation marks, control characters or special characters of Tigrinya language; if any exist within the word replace it with space. This step continues until end of line reaches. Similarly, digits are also removed using the python built in function called “sub“ which takes a digits symbol “d+” as argument and removes digits from the whole list of words.

3.2.2 Normalization

The Ethiopic Script includes different letters that have the same sound in a language. The letters 'ሰ' (se) and 'ሠ' (se), letters 'ጸ' (Tse) and ፀ (Tse) are some examples. These forms, by some writers use them interchangeably. Thus, a single Tigrinya word may exist in two different variations on many documents. For example, መጽሂት(meShEt) and መፅሂት (me'ShEt) are two variants of the same word meaning 'pamphlet'. Such variant forms have negative effect on precision of retrieval. Thus, a routine is added to convert such variants in to a single form.

Input corpus/files
Read the character from the corpus/ files
if the character is "ሓ","ሑ","ሒ","ሕ","ሖ","ሗ","ኖ"
Change each to "ሀ","ሁ","ሂ","ሃ","ሄ","ህ","ሆ" orderly respectively
else if it is "ሠ","ሡ","ሢ","ሣ","ሤ","ሥ","ሦ"
Change each to "ሰ","ሱ","ሲ","ሳ","ሴ","ስ","ሶ" orderly respectively
else if it is "ጸ","ጹ","ጺ","ጻ","ጼ","ጽ","ጾ"
Change each to "ፀ","ፁ","ፂ","ፃ","ፄ","ፅ","ፆ" orderly respectively

Algorithm3.2: Normalization algorithm

3.2.3 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 [54].

In this research the second method is used to apply stop word removal. 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.

In this study, stop words are identified manually by consulting books and dictionaries of the Tigrinya languages. The consulted books and dictionaries help to identify preposition, conjunction, articles, pronoun and auxiliary verbs of the Tigrinya language. After identifying the stop words in the Tigrinya documents, algorithm 3.3 remove the stop words from the document corpus.

```
Read stop word list file
Open the documents for processing
Do
        Read the content of the file line by line and assign the content to string
        If word is in stop word list
                Remove word from the index term
         Else
                Continue
          End if
      While end file
```

Algorithm 3.3: Stop word removal [15]

As shown in algorithm 3.3 above, the system reads the list of stop word from stop word list file and will not indexed those words.

3.2.4 Word Stemming

Stemming is a normalization step that reduces the morphological variants of words to a common form usually called a stem by the removal of affixes. Among the many normalization steps that are usually performed before indexing, stemming has significant effect in both the efficiency and the effectiveness of IR for many languages [28, 36]. The complexity of stemming process varies with the morphological complexity of a natural language. Tigrinya belongs to the Semitic language family that includes languages like Amharic, Arabic and Hebrew. Those languages are highly inflected and derived. Therefore, words have to be reduced to their root using stemming technique [40]. 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 Tigrinya text, there are many word variants/affixes [40]. To conflate them into stem word, stemming technique/ algorithm developed by Yonas [26] for Tigrinya languages were used. Yonas developed the stemmer that involves the removal of both prefix and suffix.

3.3 Searching Using the Probabilistic Model

Because of its capability of handling the uncertain nature of information retrieval, the Binary Independent Model (BIM) is used to design probabilistic Tigrinya IR system. This is because, according to C. D. Manning, et al [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 [53].

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 [16]. 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 two 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 automatically [24, 53]. 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 [18]; 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;

$$P(K_i \mid R) = 0.5 \text{ and } P(K_i \mid R) = \log\left(\frac{N - ni + 0.5}{ni + 0.5}\right) \quad \text{.....equation3.1}$$

Where, N is the total number of documents in the collection and ni 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 system looks at the top 10 retrieved documents by assuming them as 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. pseudo relevance feedback), this will allow more accurate estimation [16,19, 20]. Therefore, relevant documents retrieved should be improved using probabilistic relevance weighting technique. This technique uses the concept in term incidence contingency [18]. 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 [19]

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 [18];

$$P(t_i \mid R) = \left(\frac{r}{R}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\text{equation3.2}$$

$$P(t_i \mid R) = \left(\frac{n-r}{N-R}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\text{equation3.3}$$

According to C. J. van Rijsbergen and K. S. Jones [56], the above equations can be rewritten to compute term presence weighting function as;

$$W = \log\left(\frac{N-n\,|\,-R+r}{(R-r)(n-r)}\right) \ldots\ldots\ldots\ldots\ldots\ldots\text{equation3.4}$$

However, Robertson and Jones [19], 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

$$RW = \log\left(\frac{(r+0.5)(N-n-R+r+0.5}{(R-r+0.5)(n-r+0.5}\right) \ldots\ldots\ldots\ldots\text{equation3.5}$$

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. IR evaluation is highly related to the concept of relevance. Relevance is the degree of correspondence between retrieved documents and information need of users.

Even if relevance is subjective concept there is no possibility of ignoring it. One of the approaches to deal with subjectivity is by generating "user profiles". User profile includes knowledge about user's needs or preferences. This helps the IR system to give what "meant" not what they asked for. To enhance this user profile should be generated automatically.

The test corpus of this research consists of a collection of documents, a set of information requests and a set of relevant documents evaluated by the researcher). Given a retrieval strategy S, the evaluation measure quantifies (for each information request) the similarity between the set of documents retrieved by S and the set of relevant documents provided by the researcher. This provides an estimation of the goodness of the retrieval strategy S [16]

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-precision [16,20].

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 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 [1,35,16] and F-measure is a single measure that trades-off precision versus recall. It is the weighted harmonic mean of precision and recall. The recall, precision and F-measure can be calculated using equation 3.6, 3.7 and 3.8 respectively using information from table 3.2.

	Relevant	Not relevant
Retrieved	A	B
Not retrieved	C	D

Collection size=A+B+C+D

Relevant =A+C

Retrieved=A+B

Table 3.2: Retrieved versus relevant documents

Therefore, Recall $= \frac{|Relevant\ | \cap |Retrived\ |}{Relevant}$ ……..equation 3.6

Precision= $\frac{|Relevant\ | \cap |Retrieved\ |}{Retrieved}$…….equation3.7

F-measure= $\frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$……..equation3.8

The above formula for precision and recall, assume that, all relevant documents (A+C) are examined by user manually. Then, the retrieved documents (A+B) are presented according to their degree of relevance as per the user query by the system prototype. Then, the system examines the ranked documents starting from the top. Using this output recall and precision are measured. Therefore, for appropriate evaluation of recall and precision, plotting a precision versus recall curve is necessary [16, 35]. After the recall- precision curve is constructed, based on the original recall and precision may result in saw tooth curve. Thus, there is a need to smooth the curve using interpolation technique. which states that the "the interpolated precision at the j-th standard recall level is the maximum known precision at any recall level between the j^{th} and $(j+1)^{th}$ level: $P(r_j) = \max_{r_j \le r \le r_{j+1}} P(r)$ "

In general, precision and recall have been used widely so as to evaluate information retrieval system performance. However, they measure two different aspects of the system and thus they are inversely relative. If recall of a system is improved then the precision is reduced. The

reason behind this is that, when attempt is made to include many of the relevant documents, irrelevant documents more and more 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 [16].

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 [16].

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 Tigrinya language so as to enhance the performance of the IR and to ease the problem of uncertainty that exists in IR. To evaluate the performance of the method, the model is implemented and tested using Tigrinya documents.

CHAPTER FOUR

EXPERIMENTATION AND ANALYSIS

4.1 Introduction

This chapter reports on the experiments conducted using the architecture designed in chapter three, and the findings from the experiment. It describes the test environment. In this research an attempts has been made to design a probabilistic information retrieval system for Tigrinya language. The system has both the indexing and searching parts as in Figure 3.1. Inverted file indexing structure is used to organize documents so as to speed up searching. The probabilistic model that attempts to simulate the uncertainty nature of an IR system guides the searching and the prototype system has been developed using Python programming language.

4.2 Document and Query preparation for the Experimentation

4.2.1 Test corpus preparation

According to R. N. Oddy [62] a test collection in IR consists of a static collection of documents. In a setup for an experiment, the numbers of documents and queries are usually small (reasonable). The use of a reasonable size collection for experiment tests is justified, as Oddy explains from the point of view of the labor and time required to set up the test collection with complete relevance judgments.

So far there is no standard established test collection for Tigrinya information retrieval testing. Experiments in this study are therefore based on sets of documents and queries set up by the researcher [62][4].

For the purpose of this research a corpus with 300 short documents were used. Those documents were obtained from different web sites. Minister of Health, Minister of Education, different news, online bible chapters, Walta Information Center, and Hadas Eritrea and Hidiyat megazin (ሕድያት መፅሄት) were sources of the researcher's corpus collection and very few of these were used in a previous research by Tsegay semere [12].

To test the performance of the system 10 (ten) queries were selected by the researcher. The preparation of the query was done in such a way that it is relevant for the given selected test

[4] R. N. Oddy said that "The evaluation uses test data, that is documents and queries chosen by the experimenter and with known characteristics"

documents. Each document is saved under common folder using .txt format, which is supported by most of programming language.

As shown in table 4.1, the document corpus contains seven (7) groups, which are health, education, Religion, social, politics, sport and art related areas.

No	Types of Documents	Number of
1	Health related	40
2	Education related	30
3	Religion and philosophy	40
4	Social related	40
5	Politics related	80
6	Sport related	30
7	Art related	40
	Total	300

Table 4.1: Corpus used for the development of Tigrinya IR system

4.2.2 Query preparation

In order to make the experiment, ten (10) queries are identified. These queries are marked across each document as either relevant or irrelevant to make relevance evaluation as in appendix one for each document. The main importance of having identified queries is to evaluate the performance of the system [16]. These queries are selected subjectively after reviewing content of each document by the researcher. For this performance evaluation purpose the selected queries are given in table 4.2. In column three (3) of table 4.2 query short- cuts are given which can help to use these short- cuts instead of rewriting those queries again in further process (processes).

No	Query	Query short-cuts
1	መርቹኒ ብይርጋ ገብረመድህን	A
2	ምክልካል ተሓላላፍቲ ሕማማትን ናይ ህፃናትን ኣደታትን ጥዕና ኣገልግሎት መርሃግብሪ	B
3	ብዛዕባ ምርመራ ኤች ኣይ ቪ / ኤድስ	C
4	ዘመናዊ ትምህርቲ ምስፍሕፋሕን ናይ ትምህርቲ ስርዓት ኣወቓቕራን ኣብ ኢትዮጵያ	D
5	ወሕለል ኣገባብ ኣጸናንዓ መጽሓፍ ቅዱስ	E
6	ስርዓት ቃል ኪዳንን መርዓን ባህሊታትን ኣብ ገለገለ ክፍልታት ትግራይ	F
7	ታሪኻዊ ፣ ጅኦፖለቲካዊ ፣ ኢኮኖሚያዊን ሃይማኖታዊን ምትእስሳራት ኢትዮጰያን ግብፂን	G
8	ማእከላይ መሪሕነት ኤርትራዊ ዲሞክራስያዊ ኪዳንን ብረታዊ ቃልሲን ሕዝቢ ኤርትራ	H
9	ታሪክ ካብ ባሕርና ህዝቢ ትግራይ	I
10	ዓፈና ድምፂ ኣሜሪካን ሕገ መንግስቲ ኢትዮጵያን	J

Table 4.2: query terms with their assigned short-cuts

4.3 Indexing and searching components of the Experimentation

4.3.1 Document indexing

In the preprocessing stage, this study addresses tokenization, normalization, stop word removal and word stem (stemming) of Tigrinya documents [28]. After preprocessing stage the indexing is done.

4.3.1.1 Tokenization

All punctuation marks, control characters, numbers, borrowed characters (like $, &, ?) and special characters are removed from the text before the data is processed. All punctuation marks are converted to space and space is used as a word demarcation. Hence, if a sequence

of characters is followed by space, that sequence is identified as a word[5]. A consecutive sequence of valid characters was recognized as a word in the tokenization process [28].

The below code is a fragment of code that removes each of the above mentioned characters.

```
def tokenization(v_String):
    lp=['[',']','{','}','.',',','?','፧','።','፡','፨',':',';','!','=','፣','፤','%','#','\'','*','-','Ō','&'
        ,'፦','«','·','ú','¢','_','•']
    v_String=re.sub('[??o=?.??():",<>-``/"??"??!\'' '\ufeff]','',v_String)
    for i in lp:
            v_String=v_String.replace(i,'')
            v_String=v_String.replace('','')

    v_String= re.sub('[\d+]','', v_String)
    v_String= re.sub('[A-Za-z]','', v_String)
```

Figure 4.1: Python code for tokenization

4.3.1.2 Normalization

As discussed in section 2.2.4.1 (the problem of redundancy of some characters), in Tigrinya writing system, there are alphabets, which have the same sound during reading. To convert such characters to their common form, normalization module is implemented.

These different symbols must be considered as equivalent because they do not cause changes in meaning. As a result, in this research, all different symbols of the same sound were converted to one common form. For example, the characters ሠ and ሰ have similar sound (with the sound se). These two characters with equivalent sound are converted to ሰ (se). ፀ and ጸ are characters with equivalent sound and are changed to ፀ (tse). ሀ (he) and ኀ (he) also have the same sound then converted to ሀ (he). This conversion is for its entire seven form variant.

In the normalization module, these words written in abbreviation also normalized into their normal word structure. Its code is given below.

From the python codes of (Figure 4.1 and Figure 4.2) given tokenized and normalized documents are returned for next process and the process iterates till the whole documents in the corpus are analyzed.

[5] A word (*token)* is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing according.

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

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

        t1=["ፀ","ፁ","ፂ","ፃ","ፄ","ፅ","ፆ"]
        t2=["ጸ","ጹ","ጺ","ጻ","ጼ","ጽ","ጾ"]

        w1=["ኣሎ","እየ","እታ","እቲ","እዮም","እውን","ኣላ"]
        w2=["ዐሎ","ዒየ","ዓታ","ዒቲ","ዒዮም","ዒውን","ዒላ"]
        e1=["ኣለው","እዘኣቶም","እታ","ኣብእታ","እዚ","እምበር","ዶርከተር"]
        e2=["ዐለው","ዚኣቶም","ዓታ","ኣብዓታ","ዚ","ምበር","ዶር."]
        d1=["እቶም","እኳ","እዚኤን","ማሕበረሰብ ","እንድየ","ፅ","ዖ"]
        d2=["ዚቶም","ዚኳ","ዚኤን","ማ/ሰብ ","ንድየ","ጽ","ጾ"]

        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])
                text=text.replace(w2[i],w1[i])
                text=text.replace(t2[i],t1[i])
                text=text.replace(e2[i],e1[i])
                text=text.replace(d2[i],d1[i])
        return text
```

Figure 4.2: Python code for normalization

4.3.1.3 Stop-words removal

After tokenization and normalization the next step in the preprocessing stage is Stop word removal. The index terms selected in this study are content bearing terms which are not part of stop word list [28].

For the purposes of this research stop word list are inspected manually. This is to remove all verbs and nouns from stop word list which are more or less directly related with the main subjects of the underlying collections and to include those non-content bearing word in the stop word list.

The general stop word lists for this research are compiled in Appendix four.

The code in figure4.3; read stop word list from text file and compared it with tokenized and stemmed index term. Then, if word is similar, it removes from index terms.

This is the step where most of the inflections are removed. The affix removal algorithm depends on a list of prefixes and suffixes. In Tigrinya text, there are many word variants/affixes. To conflate them into stem word, stemming technique/ algorithm developed by Yonas [26] was used. Yanas developed the stemmer that involves the removal of both prefixes and suffixes.

```
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()
    line=line.strip()
    stp_w.append(line)
stopw.close()
```

Figure 4.3: Python code for removing stop words

4.3.1.4 Stemming: prefix and suffix Removal

Figure 4.4, depicts python code implementation module of prefix and suffix removal of Tigrinya 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 words 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. This procedure is possible for all prefix and suffix list. The prefix and suffix lists are attached in Appendix five and six respectively.

4.3.2 Searching Using the Probabilistic Model

4.3.2.1 Initial search

The searching component of the prototype uses probabilistic model. Users'information need is represented as query. The detail of query term preparation is discussed in section 4.2.2. Query text pre-processing is done in similar way to indexing part. Pre-processing part involves tokenization, normalization, stop word removal and stemming. Each query term should pass through each of these processes as it was elaborated in Figure 3.1.

The main objective of the prototype system is to map query terms and documents in the matrix to make comparison between documents and query, then calculating the weight of each query terms based on the notion implemented by probabilistic model and finally calculating the score of each document relevance and ranks in decreasing order as discussed in section 3.3.

From the actual code used in the prototype a screen shot of the initial output which shows the first list of retrieved document using a given query is given in figure 4.5.

```
በይዘአለም/አን እንታይ ይደልዩ/ይደልያ ? መሕተቲለም የአትዉ/መሕተቲለን የአትዋ:ዓፈና ድምፂ አሜሪካን ሕገ መንግስቲ ኢትዮጵያን

 Initial search result which is before relevance feedback:

በቲ መሕትት መሰረት ዝተረከቡ ፋይላት ለዘም ዝስዕቡ እዮም

shortcuts: RW= relevance weight,        Doc. List= document list


Rank       Doc. List                RW
|-----------------------------------------------|
1 :      Tgfile91 :             :  6.86168
|-----------------------------------------------|
2 :      Tgfile92 :             :  6.86168
|-----------------------------------------------|
3 :      Tgfile93 :             :  6.86168
|-----------------------------------------------|
4 :      Tgfile94 :             :  6.86168
|-----------------------------------------------|
5 :      Tgfile95 :             :  6.86168
|-----------------------------------------------|
6 :      Tgfile119 :            :  6.86168
|-----------------------------------------------|
7 :      Tgfile90 :             :  5.91382
|-----------------------------------------------|
8 :      Tgfile272 :            :  3.056
|-----------------------------------------------|
9 :      Tgfile275 :            :  3.056
|-----------------------------------------------|
10 :     Tgfile265 :            :  2.51695
|-----------------------------------------------|

 እንድሕር ዝተረኸቡ መረጻእታታት እኹል ከይኖም (3) ይጠውቐ/ይጠውቃ:
 እንተዘይኮይኑ ግን : እቲ ሲስተም ባዕሉ መረጻእታ ንምርካብ (1) ይጠውቐ/ይጠውቃ:
        :
```

Figure 4.5: A Screen shot of retrieved document for a given query

4.3.2.2 Relevance Feedback

The users tend to ask short queries, even when the information need is complex. Irrelevant documents are retrieved as answers because on the ambiguity of the natural language (words have multiple senses) [16]. If we know that some of retrieved documents were relevant to the query, terms from those documents can be added to the query in order to be able to retrieve more relevant documents. This is called relevance feedback. Often, it is not possible to ask the user to judge the relevance of the retrieved documents. In this case a pseudo-relevance feedback method is used. It assumes the ten (10) retrieved documents are relevant and use the most important terms from them to expand the query and document relevance is reweighted again then ranks in decreasing order.

4.4 Retrieval Performance Evaluation

Once 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 or efficiency [28]. The type of the evaluation considered depends on the objectives of the retrieval system [16], even though complete evaluation process requires evaluation of both system effectiveness and efficiency. The objective of this research is to examine the effectiveness of probabilistic IR system using Tigrinya language corpus. Therefore, for this research only effectiveness of IR system is taken into consideration to determine the performance of the system. As a result, retrieval effectiveness of a system is evaluated on the given set of documents (corpus), queries and relevance judgments. The performance of the system is evaluated before and after relevance feedback using ten (10) queries.

This IR system effectiveness is evaluated in various ways [28]. These are precision, recall and F-measure. As discussed in section 2.6, Precision, Recall and F-measure are the most frequent and basic statistical measures which are widely used measures to assess the effectiveness of IR system. These three parameters are used in this research so as to measure the effectiveness of the designed probabilistic based Tigrinya IR system.

The relevance judgments are prepared to construct document query matrix that shows all relevant documents for each test query prepared as shown in appendix one. Then, each test query is measured by precision, recall and F-measure, and then the average of each test query represents the performance registered by the system.

The performance of the prototype system is evaluated before and after stemming in four different ways. These are:

A) Evaluation before stemming; including

1) Evaluation before relevance feedback

2) Evaluation after pseudo relevance feedback

B) Evaluation after stemming under it:

3) Evaluation before relevance feedback

4) Evaluation after pseudo relevance feedback

The performance of the system is evaluated using B(4) which means the final result expected for the system is the system after stemming text of the documents and after these stemmed text documents are evaluated using a relevance feedback by the system so called pseudo relevance feedback. Numerous testing has been made to fix threshold. For retrieving a set of relevant documents that can achieve the maximum optimal performance, initially before relevance feedback, the relevance weight greater than or equal to two (>=2) has been found that the optimal threshold. The relevance weight greater than or equal to three (>=3) is the optimal threshold value fixed for retrieving a set of relevant documents after pseudo relevance feedback.

As can be seen on the architecture of the prototype in figure 3.1 the system is designed with pseudo relevance feedback. Thus in table 4.3, table 4.4, table 4.5 and table 4.6 the researcher used pseudo relevance feedback.

4.4.1 Retrieval performance evaluation of the system before stemming

			Before relevance feedback		**After pseudo relevance feedback**	
No	List of queries	Relevant docs from corpus manually evaluated	Ranked retrieved docs	Ranked relevant and retrieved	Ranked retrieved docs	Ranked relevant and retrieved
1	A	164, 165, 166, 167, 168, 169, 229, 230, 231, 232, 233 Total=11	166, 133 Total=2	166 Total=1	166, 165, 164, 229, 230, 167, 168, 169, 231, 232, 233 Total =11	166, 165, 164, 229, 230, 167, 168, 169, 231, 232, 233 Total =11
2	B	12, 13, 14, 15, 28, 29, 30, 32, 33, 34, 65, 256, 257, 260, 261	30, 257, 260, 261, 12, 14, 15, 276,278, 29,32, 34, 89 Total=13	30, 257, 260, 261, 12, 14, 15, 29,32, 34, Total=10	30, 12, 14 , 15, 276, 278, 29, 32, 34, 257, 260, 261, Total=12	30, 12, 14 , 15, 29, 32, 34, 257, 260, 261 Total=10

		Total=15				
3	C	1, 4, 5, 7, 8, 15, 28, 256, 257 Total=9	256,4, 8, 255, 1, 5 , 7 Total=7	256,4,8, 1,5,7 Total=6	256, 8, 4, 1, 5, 7, 255, 257 Total=8	256, 8, 4, 1, 5, 7, 257 Total=7
4	D	1, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 31, 35, 36, 37, 38, 39, 40, 41, 131 Total=22	26, 36, 27, 39, 20, 22, 40, 41, 21, 13, 30, 18, 25, 131, 37, 38, 17, 35, 16, 19, 24, 31, 77, 220, 266, 267, 268, 280, 290 Total=29	26, 36, 27, 39, 20, 22, 40, 41, 21, 18, 25, 131, 37, 38, 17, 35, 16, 19, 24, 31 Total=20	26, 27, 20, 22, 40, 41, 36, 39, 25, 21, 17, 35, 37, 38, 131, 18, 13, 30, 16, 19, 24, 31, 77, 220,266, 267, 268, 280, 290, 67, 130, 231, 221, 221, 223, 224, 243, 248, 255, 256, 257, 258, 261, 262, 264, 265, 274, 276, 277, 278, 279, 287, 297 Total=52	26, 27, 20, 22, 40, 41, 36, 39, 25, 21, 17, 35, 37, 38, 131, 18, 16, 19, 24, 31 Total=20
5	E	56, 281, 282, 283, 284, 285, 286, 287, 288, 289, 291,292, 293, 294, 295, 296, 297, 298, 299, 300 Total=20	281, 282, 283, 284 ,286, 287 ,289, 290, 291, 292, 293, 294, 296, 297, 299, 300, 285, 288, 295, 298, 49, 56 Total=21	281, 282, 283, 284, 286 , 287, 289, 291, 292, 293, 294, 296, 297, 299, 300, 285, 288, 295, 298, 56 Total=20	281, 282, 283, 284, 286, 287, 289, 291, 292, 293, 294, 296, 297, 299, 300, 285, 288, 295, 298, 49, 56, 247 Total=22	281, 282, 283, 284, 286, 287, 289, 291, 292, 293, 294, 296, 297, 299, 300, 285, 288, 295, 298, 56 Total=20
6	F	2, 3, 9, 10, 11 Total=5	3, 2, 79, 9, 257, 76, 127, 251, 12, 36, 185, 271, 259 Total=13	3, 2, 9, Total=3	3, 9,2, 127, 251, 79 Total=6	3, 9, 2 Total=3
7	G	98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 117, 118 Total=12	100, 117, 107, 99, 118, 70 Total=6	100, 117, 107, 99, 118, Total=5	100, 117, 99 , 107, 118 Total=5	100, 117, 99 , 107, 118 Total=5
8	H	66, 67, 68, 70, 72, 74, 75, 76, 77, 78, 79, 80 Total=12	66, 67, 80, 79, 271, 68, 76, 70, 74, 287, 297, 72, 272, 269, 240 Total=15	66, 67, 80, 79, 68, 76, 70, 74, 72, Total=9	66,67, 79, 80, 271, 68, 76, 70,74, 287, 297, 72, 269, 240, 69, 75, 205, 78, 77, 272, 270 Total=21	66, 67, 79, 80, 68, 76, 70,74, 72, 75, 78, 77 Total=12
9	I	10, 96 , 116, 187, 225, 227, 234 Total=7	139, 225, 187, 227, 133, 234, 96, 128, 240, 265, 268, 269 Total=12	225, 187, 227, 234, 96 Total=5	139, 225, 187, 133, 227 ,96, 128, 240, 265, 268, 269, 234 Total=12	225, 187, 227 , 96, 234 Total=5

10	J	91, 92, 93, 94, 95, 119, 90 Total=7	91, 92, 93, 94, 95, 119, 90, 272, 275, 163,270 Total=11	91, 92, 93, 94, 95, 119, 90, Total=7	91, 92, 93, 94 , 95, 119, 90, 272, 275, 163, 257, 260, 261, 263, 264, 265, 268, 173, 44, 3, 270 Total=21	91, 92, 93, 94 , 95, 119, 90, Total=7

Table 4.3: Before stemming: List of relevant documents from corpus manually evaluated, total ranked retrieved documents for each query and ranked list of (retrieved & relevant) documents before and after pseudo relevance feedback

Using the information from table 4.3 above precision (P), recall (R) and F-measure (F) are evaluated and given in table 4.4. This evaluation is done before stemming; including **before relevance feedback** and **after pseudo relevance feedback.** See table 4.4

		Before relevance feedback					**After relevance feedback**				
Query	**2**	**3**	**4**	**P**	**R**	**F**	**3**	**4**	**P**	**R**	**F**
A	11	2	1	0.5	0.091	0.154	11	11	1	1	1
B	15	13	10	0.77	0.667	0.714	12	10	0.833	0.667	0.741
C	10	7	6	0.86	0.6	0.706	8	7	0.875	0.7	0.778
D	22	29	20	0.69	0.909	0.785	52	20	0.385	0.909	0.541
E	20	21	20	0.95	1	0.975	22	20	0.909	1	0.952
F	5	13	3	0.23	0.6	0.334	6	3	0.5	0.6	0.545
G	12	6	5	0.83	0.417	0.556	5	5	1	0.417	0.589
H	14	15	9	0.6	0.643	0.621	21	12	0.571	0.857	0.685
I	7	12	5	0.42	0.714	0.527	12	5	0.417	0.714	0.527
J	7	11	7	0.64	1	0.778	21	7	0.333	1	0.5
Average values of P, R and F				0.65	0.664	0.615			0.682	0.786	0.686

Table 4.4: Experiment one before stemming (before and after relevance feedback): the effectiveness of the probabilistic Tigrinya IR system on 10 selected queries

Keys the researcher used in table 4.4:

2: Number of relevant documents evaluated manually from corpus for each query

3: Number of total retrieved documents for each query before stemming

4: Number of retrieved and relevant documents for each query before stemming

As it is observed from table 4.4, the average result of precision, recall and F-measure of the system using the result **before relevance feedback** by the model about the relevance of document is 65 % precision, 66.4 recall and 61.5 F-measure. Since, in probabilistic model the initial guess of relevant document is based on Boolean expression, thus, all terms that match one of user queries will be 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 pseudo relevance feedback so as to apply query terms reweighting in order to increase the weight of terms found in relevant documents and decrease the weight of terms found in non-relevant documents.

As can be seen from Table 4.4, **after pseudo relevance feedback** on initially retrieved documents as relevant and non-relevant, the average percentage of precision is increased by 3.2%, recall is increased by 12.2% and F-measure is increased by 7.1%.

4.4.1 Retrieval performance evaluation of the system after stemming

			Before relevance feedback		**After pseudo relevance feedback**	
No	List of queries	relevant docs from corpus manually evaluated	Ranked retrieved docs	Ranked relevant and retrieved	Ranked retrieved docs	Ranked relevant and retrieved
1	A	164, 165, 166, 167, 168, 169, 229, 230, 231, 232, 233 Total=11	166, 133 Total=2	166 Total=1	166, 165, 164, 229, 230, 167, 168, 169, 231, 232, 233 Total =11	166, 165, 164, 229, 230, 167, 168, 169, 231, 232, 233 Total =11
2	B	12, 13, 14, 15, 28, 29, 30, 32, 33, 34, 65, 256, 257, 260, 261 Total=15	89, 30, 260, 14, 15, 40, 33, 256, 257, 258, 261, 12, 13, 276, 278, 32, 34,29, 65, 209, 88 Total=21	30, 260, 14, 15, 33, 256, 257, 261, 12, 13, 32, 34,29, 65 Total=14	30, 89, 260, 14, 15, 40, 33, 256, 257, 258, 261, 13, 12,65, 209, 88 Total=15	30, 260, 14, 15, 33, 256, 257, 261, 13, 12, 65, Total=11
3	C	1, 4, 5, 7, 8, 15, 28,61 256, 257, Total=10	256,4, 8,1, 255, 257, 5 , 7 ,61 Total=9	256,4, 8,1, 257, 5 , 7, 61 Total=7	256, 8, 4, 1, 5, 7, 257, 255, Total=8	256, 8, 4, 1, 5, 7, 257 Total=7
4	D	1, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 31, 35, 36, 37, 38, 39, 40, 41, 131 Total=22	26, 36, 37, 39, 27,20, 22, 40, 41, 21, 18, 13, 30, 25, 79, 132, 131, 38, 17, 35, 102, 287, 297, 259 Total=24	26, 36, 37, 39, 27,20, 22, 40, 41, 21, 18, 25, 131, 38, 17, 35 Total=16	26, 37, 39, 20, 22, 40, 41, 36, 27, 21, 38, 25, 131, 17, 35, 18, 287, 297, 16,19, 24, 31, 67, 77, 220, 266, 267, 268, 280, Total=30	26, 37, 39, 20, 22, 40, 41, 36, 27, 21, 38, 25, 131, 17, 35, 18, 16,19, 24, 31, 1 Total=21

5	E	56, 281, 282, 283, 284, 285, 286, 287, 288, 289, 291,292, 293, 294, 295, 296, 297, 298, 299, 300 Total=20	281, 282, 283, 284 ,286, 287 ,289, 290, 291, 292, 293, 294, 296, 297, 299, 300, 285, 288, 295, 298, 49, 56 Total=21	281, 282, 283, 284, 286 , 287, 289, 291, 292, 293, 294, 296, 297, 299, 300, 285, 288, 295, 298, 56 Total=20	281, 282, 283, 284, 286, 287, 289, 291, 292, 293, 294, 296, 297, 299, 300, 285, 288, 295, 298, 49, 56, 247 Total=22	281, 282, 283, 284, 286, 287, 289, 291, 292, 293, 294, 296, 297, 299, 300, 285, 288, 295, 298, 56 Total=20
6	F	2, 3, 9, 10, 11 Total=5	2, 3 , 257, 185, 79, 272, 275, 9, 10, 12, 76 Total=10	2, 3 , 9, 10, Total=4	2,252, 185, 3, 79, 272, 275, 9, 10, 171,11 Total=10	2, 3, 9, 10,11 Total=5
7	G	98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 117, 118 Total=12	100, 107, 98,99, 101, 117, 118, 102, 105, 70 Total=10	100, 107, 98,99, 101, 117, 118, 102, 105 Total=9	100, 98, 99, 107, 101, 117, 118, 102, 105 Total=9	100, 98, 99, 107, 101, 117, 118, 102, 105 Total=9
8	H	66, 67, 68, 70, 72, 74, 75, 76, 77, 78, 79, 80, 99,185 Total=14	66, 67, 80, 68, 271, 79, 76, 74, 272, 70, 75, 77, 78, 270, 107, 240, 269, 185, 69, 72, 99,205,287, 297,275 Total=25	66, 67, 80, 68, 79, 76, 74, 70, 75, 185, 77, 78, 72, 99 Total=14	66, 67, 80, 68, 79, 76, 74, 70, 75, 77, 78, 270, 272, 107, 69, 72, 99, 205, 287, 297,240, 185, 65, 71, 73, 97, Total=26	66, 67, 80, 68, 79, 76, 74, 70, 75, 77, 78, 72 Total=12
9	I	10, 96 , 116, 187, 225, 227, 234 Total=7	139, 75, 225, 187, 133, 234, 96, 116, 128, 227 Total=10	225, 187, 234, 96, 116, 227 Total=6	139, 75, 10, 187, 133, 234, 96, 116, 128, 240, 265, 268, 269, 227, 225 Total=15	187, 234,10, 96, 116, 227, 225 Total=7
10	J	91, 92, 93, 94, 95, 119, 90 Total=7	91, 92, 93, 94, 95, 119, 90, 272, 275, 265, 270, 161, 183 Total=13	91, 92, 93, 94, 95, 119, 90, Total=7	91, 92, 93, 94, 95, 119, 90, 272, 275, 265, 270, 3, 42, 51, 260, 261, 267, 269, 271, 70, 107, 131, 139, 163, 254, 161, 183 Total=27	91, 92, 93, 94 , 95, 119, 90, Total=7

Table 4.5: after stemming: List of relevant documents from corpus manually evaluated, total ranked retrieved documents for each query and ranked list of (retrieved & relevant) documents before and after pseudo relevance feedback

Using the information from table 4.5 above precision (P), recall (R) and F-measure (F) are evaluated and given in table 4.6. This evaluation is done after stemming; including **before relevance feedback** and **after pseudo relevance feedback.** See table 4.6

	Before relevance feedback						**After pseudo relevance feedback**				
Query	**2**	**3**	**4**	**P**	**R**	**F**	**3**	**4**	**P**	**R**	**F**
A	11	2	2	1	0.182	0.308	11	11	1	1	1
B	15	21	14	0.667	0.933	0.778	15	11	0.733	0.733	0.733
C	10	9	7	0.778	0.7	0.737	8	7	0.875	0.7	0.778
D	22	24	16	0.667	0.727	0.696	30	21	0.7	0.955	0.808
E	20	21	15	0.714	0.75	0.732	22	20	0.909	1	0.952
F	5	10	4	0.4	0.8	0.533	10	5	0.5	1	0.667
G	12	10	9	0.9	0.75	0.818	9	9	1	0.75	0.857
H	14	25	14	0.56	1	0.718	26	12	0.462	0.857	0.6
I	7	10	6	0.6	0.857	0.706	15	7	0.467	1	0.637
J	7	13	7	0.538	1	0.7	27	7	0.259	1	0.411
Average values of P, R and F				0.682	0.77	0.673			0.691	0.9	0.744

Table 4.6: Experiment two after stemming (before and after relevance feedback): the effectiveness of the probabilistic Tigrinya IR system on 10 selected queries

Keys the researcher used in table 4.6:

2: Number of relevant documents evaluated manually from corpus for each query

3: Number of total retrieved documents for each query after stemming

4: Number of retrieved and relevant documents for each query after stemming

As it is observed from table 4.6, the average result of precision, recall and F-measure of the system using the result **before relevance feedback** by the model about the relevance of document is 68.2% precision, 77% recall and 67.3 % F-measure. Since, in probabilistic model the initial guess of relevant document is based on Boolean expression, thus, all terms that match one of user queries will be 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 pseudo relevance feedback so as to apply query terms reweighting in order to increase the

weight of terms found in relevant documents and decrease the weight of terms found in non-relevant documents.

As can be seen from Table 4.6, **after pseudo relevance feedback** on initially retrieved documents as relevant and non-relevant, the average percentage of precision is increased by 0.9 %, recall is increased by 13 % and F-measure is increased by 7.1 %.

4.5 Result analysis, Findings and challenges

4.5.1 Result analysis

The performance of the system is evaluated before and after stemming in four different ways. These are:

A) Evaluation before stemming; including

1) Evaluation before relevance feedback

2) Evaluation after pseudo relevance feedback

B) Evaluation after stemming under it:

3) Evaluation before relevance feedback

4) Evaluation after pseudo relevance feedback

Using the information given in table 4.3 and table 4.4 evaluations is done by measuring the recall, precision and F-measure of each test query and the average of each query result are obtained as the performance of the system before stemming.

Evaluation results before stemming, as indicated in Table 4.4 both before relevance feedback and after pseudo relevance feedback registered good performance where in the case of the performance of before relevance feedback, it registered 65 % precision, 66.4 % recall and 61.5 % F-measure and after pseudo relevance feedback it registered 68.2% precision, 78.6 % recall and 68.6% F-measure.

For each of the parameters precision, recall and F-measure, the pseudo relevance feedback registered better value than that of before relevance feedback. Where precision increases by 3.2%, recall increases by 12.2% and F-measure increases by 7.1%.

Also using the information given in table 4.5 and table 4.6 evaluations is done by measuring the recall, precision and F-measure of each test query and the average of each query result are

obtained as the performance of the system after stemming. This evaluation is also done in two ways before relevance feedback and after relevance feedback.

As indicated in Table 4.6 both parameters results performs significantly better performance than the parameter result of table 4.4 where the performance after stemming: before relevance feedback , it registered 68.2 % precision, 77 % recall and 67.3 % F-measure and after pseudo relevance feedback it registered 69.1 % precision, 90 % recall and 74.4% F-measure.

For each of the parameters precision, recall and F-measure, the pseudo relevance feedback registered better value than that of before relevance feedback where precision increases by 0.9%, recall increases by 13 % and F-measure increases by 7.1 %. The summary of this discussion is given in the below table 4.7. The table below shows the difference of the result achieved before and after stemming for each parameters that is P, R and F; where considering before and after relevance feedback.

		Before stemming =A	After stemming =B	The gap A and B that is ; B-A
Before relevance feedback	P	65%	68.2%	3.2%
	R	66.4%	78.6%	12.2%
	F	%61.5	68.6%	7.1%
After pseudo relevance feedback	P	68.2%	69.1%	0.9%
	R	77%	90%	13%
	F	67.3%	74.4%	7.1%

Table 4.7: summary of the result achieved before and after stemming

Recall (R) measures the ability of the search to find all of the relevant items in the corpus [16] whereas precision (P) measures the ability to retrieve top-ranked documents that are mostly relevant and F-measure (F) is the harmonic mean of the two measures that is P and R. In every result always after stemming achieves better result than before stemming. Because always B-A achieves positive result in table 4.7. In this study the maximum recall and precision registered are in after stemming - after pseudo relevance feedback which are 90 % and 69.1 % respectively. And as we observed from table 4.7 the nature of probabilistic model in using relevance feedback, after pseudo relevance feedback the results are improved.

Note that precision and recall are set-based measures. They evaluate the quality of a set of retrieved documents. To evaluate ranked lists, recall-precision curves are used [28]. For those cases to have smooth curve, Precision **at 11 standard recall levels** is used. Each recall-

precision point is computed by calculating the precision at the specified recall level value. For the rest of recall values, the precision is interpolated as in Figure 4.6:

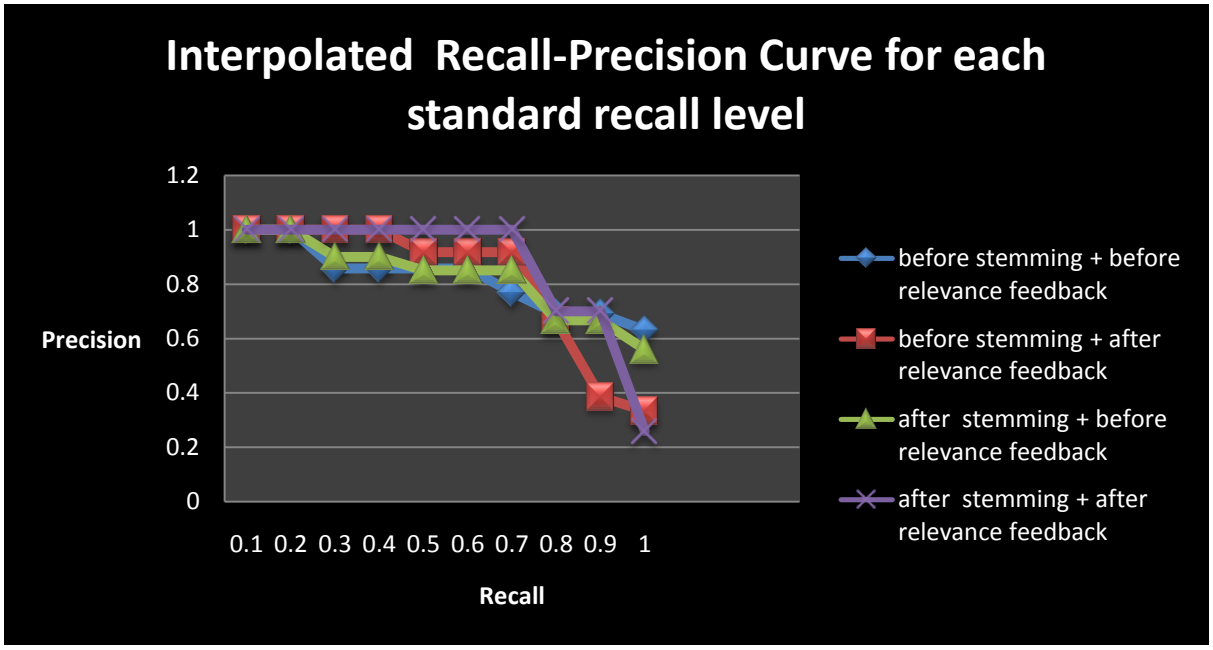

Figure 4.6: Interpolated precision at 11 standard recall levels

The interpolation of precision/recall curve is done for each available 10 queries in figure 4.6. This is important so as to show the performance of the system. As it is observed from Figure 4.6, the performance of the system registers better performance (after stemming +after relevance feedback). The curve at the upper right-hand corner of the graph indicates the best performance registered by the system, in which recall and precision reaches maximum point. At recall level, 0.9 the maximum precision registered is 0.7 this represents the average performance registered by the system.

However, there are several challenges which limit to register best promising performance. This is because of synonymy and polysemy of terms. The system is greatly affected by synonymous and polysemous nature of Tigrinya words. For instance the Query (‘**ብዛዕባ ምርመራ ኤች ኣይ ቪ**’) *meaning (*about HIV examination) is one of the queries which registered lower recall. In this case some relevant documents are not retrieved. Example document ‘**Tgfile15.txt’** is relevant for this query but not retrieved. This happens because this query is not directly available at document ’**Tgfile15.txt’** but this document contains the term (‘**ብዛዕባ ምርመራ ኤድስ**’) meaning (about AIDS examination) in which they are synonym. Thus there are such problems in other documents also. This problem can handle by considering synonymous and polysemy words.

Because morphological inflection nature of Tigrinya writing system, terms are highly inflected for number, genders, Person, adjectives.

Often, morphological variants of words have similar semantics and can therefore be considered as equivalent in information retrieval. Stemming algorithms reduce words to their stem or root form. Thus, instead of using the original document or query terms, the stems of these terms are used for information retrieval. Consequently, using this approach can significantly increase the number of relevant documents found, thus precision can increase also. Stemming algorithms do not always produce stems which are real words. However, this should not be a problem, provided that (a) words with different semantics are reduced to different stems and (b) different words with the same semantics are reduced to the same stem. However, these requirements are not always met. Requirement (a) is not met when dealing with polysemy (words with two different meanings). A polysemy is always reduced to the same stem, regardless of its specific meaning. Requirement (b) is not always met when dealing with synonyms (different words with the same meaning). It is possible that the synonyms differ to such an extent that they are reduced to different stems. From these observations it is clear that polysemy and synonyms can cause serious problems in stemming, thus in information retrieval [28]. Also these problems are also appeared in this stemmer used.

4.5.2 Findings and challenges

Generally the work done has registered promising performance. Better performance will be achieved if stemmer algorithm is improved, and there is mechanism to handle polysemy and synonymy of Tigrinya words.

Still there was no probabilistic IR research conducted for Tigrinya language except cross information retrieval for Tigrinya-Amharic language in 2013 by Tsegay [12]. The obtained result of this research indicates that probabilistic based IR system for Tigrinya language register encouraging performance with 69.1% precision, 90.0% recall and 74.4% F-measure. Basically the research is conducted to check the performance of IR system for Tigrinya language using the Tigrinya stemmer developed by Yonnas [26]. However, still here are problems with stemmer. The stemmer is a rule based stemmer. It is unable to handle the in-fixes.

With this performance registered there are several challenges which limits to register the best possible performance from the model. These are

1) **Nature of the probabilistic model**

In probabilistic model the initial guess is made based on Boolean expression. There is a high probability of having similar weight. This will result retrieving no documents or retrieving all documents containing those terms. In this case, the precision decreases highly and the recall becomes high.

In initial guess of relevant document, it does not consider the importance of the frequency of the terms in the document. For this reason, sometimes those documents having query terms with highest frequency than others could be ranked behind. In this case, users faced with the problem of having to choose the appropriate words that are also used in the relevant documents. Hence, poor result could be displayed when the system retrieve documents after feedback is given.

2) **The problem of polysemy and synonym of terms**

Some Words in Tigrinya language has polysemy and / or synonym words. In IR system unless there is a mechanism to control those kinds of words, the performance of the system highly decreases because relevant documents containing synonym word for the query term are not retrieved, while irrelevant documents that contains polysemy /synonym words are retrieved. For instance, for a query "**ምስፍሕፋሕ**" meaning they *broadening*, a document contain word "**ምዝርጋሕ**" meaning *spreading out* could not be retrieved unless it contain the query word itself "**ምስፍሕፋሕ**". The combination of the above results leads to a decrease in the performance of the system in both precision and recall.

3) **No standard test corpus for Tigrinya language**

Finding a large size and standard corpus for Tigrinya language is also considered as one of the challenge faced in this research. There is no any developed standard corpus for Tigrinya language. Thus, in this research the researcher uses small size corpus from differnt sources as discussed in section 4.2.1. This result not only weakens the performance of the system but also makes it difficult to compare the result obtained with several researches since there is different in test queries, document content and size used for testing.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Text retrieval system is very important for retrieval of textual documents. The study attempts to develop probabilistic IR system for Tigrinya. The developed prototype has two modules: indexing and searching. The indexing part of the work involves tokenization, normalization, stop word removal and stemming. The stemmer is adopted from Yonas [26].

The indexing component is developed to enable the arrangement of index terms to permit fast searching and reading memory space requirement used to speed up access to desired information from document collection as per users query.

The developed Tigrinya text retrieval system has also searching components. The main parts are the representation of the user's information need, query text operation, query operation to have an actual query, the comparison of document term and query through probabilistic IR model, ranking of documents and query reformulation through pseudo relevance feedback (as required) to improve the initial result of the model.

In this IR system, the Binary Independent Model (BIM) is chosen and implemented. At first step when the search component initiated the system generates the first ranked list of relevant documents then using terms from the initial guess made the system also searches again using pseudo relevance feedback. Finally based on the pseudo relevance feedback the system improves its performance. This leads us to conclude that pseudo relevance feedback is useful for the improvement of an IR system. However, as the researcher observed from literatures, since probabilistic model used Boolean expression for initial guess of relevant document, it does not consider the importance of the document based on the frequency of the terms in the document. For this reason, sometimes those documents having query terms with highest frequency than others could be ranked lately. In this case, users faced with the problem of having to choose the appropriate words that are also used in the relevant documents. Hence, poor result could be displayed when the system retrieve documents after pseudo relevance feedback is given.

According to the experimentation made the system registered an average precision of 69.1%, recall 90.0%, and F-measure 74.4%. This is a promising result to design an applicable IR system. However, we have observed that synonym and polysemy nature of Tigrinya words

affects the performance of the IR system. It is possible to conclude that if Tigrinya and similar language challenges are controlled with the help of thesaurus, a better performing stemmer, and other sub-modules, it is possible to develop a usable IR system for Tigrinya.

In general, the researcher concludes that it is possible to develop a complete IR system for Tigrinya using a probabilistic approach which can be further improved by increasing the corpus size in the language. We can also conclude that existing systems like stemmers developed by MSc researchers can be integrated in to bigger ones like IR systems to get performance improvement.

5.2 Recommendation

Tigrinya retrieval system has wide open place for future study. The area is just at the beginning level. It needs collaboration of researchers and funding organizations. In fact the researcher would like to recommend the following. Specially for further researchers in the area

- ✓ Probabilistic model make the initial guess based on Boolean expression, which blocks to know important words to represent a document and, accordingly may not retrieve relevant documents that contain large number of terms found in a given query. Hence, there is a need to build hybrid system that uses vector space model to guess relevant documents for user query using non-binary weighting technique and then use probabilistic relevance feedback to improve the performance of the system.
- ✓ One of the problems in enhancing the performance of Tigrinya IR system is the existence of synonym and polysemy terms in Tigrinya text. The researcher recommends integrating mechanisms of controlling synonyms/ polysyms terms in the probabilistic model to enhance precision and recall of the system.
- ✓ According Yonas [26] the stemming algorithm used in this research has understeming and overstemming problem. It frequently overstems (sometimes understems) word variants. This greatly affects the performance of the system. Therefore, potential work need to consider 1) hybrid of rule based and dictionary based Tigrinya stemming algorithm or 2) ontology based stemming algorithm that conflates using meaning understanding (recommended more) in order to see the retrieval effectiveness of the system on the area.
- ✓ Finding a standard corpus and test queries with relevance judgment for testing the designed system is one of the challenges faced in this research. Therefore, future research need to consider the development of standard Tigrinya corpus that can be

used by researchers to evaluate progress made in designing Tigrinya IR systems. Thus, the future researchers need to give an emphasis on the development of corpus for Tigrinya to reach an applicable information retrieval.

References

[1] P. E. R. Ingwersen, "Information Retrieval Interaction," a book by Taylor Graham publishing, Denmark, 1992.

[2] H. Redwan, et al., "Search engine for Amharic Web content," in AFRICON, 2009.

[3] J.Hauben, "Vannevar Bush and JRC Licklider: Libraries of the Future 1945–1965", 1st ed, 2005.

[4] I. Kocabas, et al., "Investigation of Luhn's claim on information retrieval," Turkish Journal of Electrical Engineering & Computer Sciences, pp. 993-1004, 2011.

[5] C. Buckley, et al., "Automatic query expansion using SMART: TREC 3," NIST special publication sp, pp. 69-69, 1995.

[6] C. Cleverdon, "The Cranfield tests on index language devices," in Aslib proceedings, pp. 173-194, 1967.

[7] D. K. Harman, "The first text retrieval conference (TREC-1) Rockville, MD, USA, 4–6 November, 1992," Information Processing & Management, vol. 29, pp. 411-414, 1993.

[8] W.Leslau, "Documents Tigriya (éthiopien septentrional): grammaire et textes," Librairie Klincksieck ,1941.

[9] S. Amanuel, "Sewasiw Tigrinya bisefihu," Red Sea Pr, 1998.

[10] W. J. Chun, " Core python programming", Prentice Hall PTR, 2006.

[11] H. Amanuel," Probabilistic Information Retrieval System For Amharic Language," MSc Thesis, School of Information Science, Addis Ababa University,2012.

[12] S. Tsegay, " Probabilistic Tigrinya-Amharic Cross Language Information Retrieval (CLIR)," MSc Thesis, School of Information Science, Addis Ababa University ,2013.

[13] C. Alberto' " Optimization Algorithm For Improving The Efficacy Of An Information, Retrieval Model," Proceedings Of TOGO, École Polytechnique of France, Pp. 1 - 4,2010.

[14] W. M. Shaw Jr , " Term-Relevance Computations And Perfect Retrieval Performance ", School of Information and Library Science, University of North Carolina, Information Processing & Management, pp. 491-498. 1995

[15] A. Gebrehiwot, "A Two Step Approach For Tigrinya Text Categorization," Msc Thesis,School Of Information Science, Addis Ababa University, 2011.

[16] R. Baeza-Yates et al, "Modern information retrieval," ACM press New York, USA, 1999.

[18] N. Fuhr, "Probabilistic models in information retrieval," The Computer Journal, vol. 35, pp. 243-255, 1992.

[19] S. E. Robertson and K. S. Jones, "Relevance weighting of search terms," Journal of the American Society for Information science, vol. 27, pp. 129-146, 1976.

[20] C. J. van Rijsbergen, "A theoretical basis for the use of co-occurrence data in information retrieval," Journal of documentation, vol. 33, pp. 106-119, 1977.

[21] F. D. R. o. E. P. C. Commission, "Summary and statistical report of the 2007 population and housing census–population size by age and sex," Addis Ababa, December, 2008.

[22] K. Brown and S. Ogilvie, "Concise encyclopedia of languages of the world," Elsevier Ltd, 2010.

[23] T. Mindaye and S. Atnafu, "Design and Implementation of Amharic Search Engine," in Signal-Image Technology & Internet-Based Systems (SITIS), 2009 Fifth International Conference, vol. 3, pp. 318-325, 2009.

[24] K. S. Jones, "A statistical interpretation of term specificity and its application in

retrieval," Journal of documentation, vol. 28, pp. 11-21, 1972.
[25] W. R. Greiff, "A theory of term weighting based on exploratory data analysis," in Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 11-19, 1998.
[26] F. Yonas, "Development of stemming algorithm for Tigrinya text," MSc Thesis, School of Information Science, Addis Ababa University ,2011.
[27] S. M. Katz, "Distribution of content words and phrases in text and language modeling," Natural Language Engineering, vol. 2, pp. 15-59, 1996.
[28] C. D. Manning, et al.," Introduction to information retrieval," Cambridge University Press , vol. 1: Cambridge, England, 2008.
[29] K. S. Jones, "Information retrieval and artificial intelligence," Elsevier Science B.V ,Artificial Intelligence, vol. 114, pp. 257-281, Cambridge CB2 3QG, UK, 1999.
[30] F. Crestani and M. Lalmas, "Logic and uncertainty in information retrieval," in Lectures on information retrieval, ed: Springer, pp. 179-206, London, UK , 2001.
[31] D. Hiemstra, "Information retrieval models," Information Retrieval: searching in the 21st Century, pp. 1-19, 2009.
[32] J. Teevan and D. R. Karger, "Empirical development of an exponential probabilistic model for text retrieval: using textual analysis to build a better model," in Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 18-25, 2003.
[33] G. Negash, "A History of Tigrinya Literature in Eritrea: The Oral and the Written, 1890–1991," Research In African Literatures, vol. 43, 2012.
[34] J. Hammond "A Chronicle of the Revolution in Tigray region of Ethiopia" Eritrea: Red Sea Press, 1999.
[35] C. J. van Rijsbergen " Information Retrieval. Second edition, Department of Computing Science University of Glasgow
[36] G. Salton, et al., "A vector space model for automatic indexing," Communications of the ACM, vol. 18, pp. 613-620, 1975.
[37] Y. Firdyiwek and D. Yaqob, "The system for Ethiopic representation in ASCII," 1997.
[38] T. Bloor, "The Ethiopic writing system: a profile," Journal of the Simplified Spelling Society, vol. 19, pp. 30-36, 1995
[39] L. Bender and C. Ferguson, "The Ethiopian writing system," Language in Ethiopia. Edited by ML Bender, JD Bowen, RL Cooper, and CA Ferguson. London: Oxford University press, 1976.
[40] YN. Takahashi and D. Yacob, "Adding Ethiopic Features to Emacs," in First International Conference on Information Technology in Ethiopia and Computational Ethiopics Symposium, 1997.
[41] G . Kassa , "A Tigrinya Language Dictionary", Addis Ababa, Ethiopia: EMAY Printers, 2003.
[42] G. SALTON, et al., "Automatic Routing And Retrieval Using Smart: TREC-2," Information Processing & Management, Vol. 31, No. 3, pp. 315-326, 1995.
[43] G. Salton and C. Buckley, "Improving Retrieval Performance by Relevance Feedback," Journal Of The American Society For Information Science, Department of Computer Science, Cornell University, Ithaca, NY 14853-7501, vol. 41, No. 4, pp.288-297, 1990 .
[44]G. Salton & C. Buckliy, "term weighting approaches in automatic text retrieval,"

information processing & management, vol. 24, No.5, pp.513-523,1988.
[46] H. Wu and G. Salton, "A comparison of search term weighting: term relevance vs. inverse document frequency," in ACM SIGIR Forum, pp. 30-39, 1981.
[47] M. E. MARON, and J. L. KUHNS, "On relevance, probabilistic indexing and retrieval," ACM 7, pp. 216–244, 1960.
[48] B.-H. Cho, et al., "Exploring term dependences in probabilistic information retrieval model," Information Processing & Management, vol. 39, pp. 505-519, 2003.
[49] S. E. Robertson, "Progress In Documentation Theories And Models In Information sRetrieval," Journal of Documentation, Vol. 33, No. 2, pp. 126-148, School of Library, Archive and Information Studies, University College, London, June 1977.
[49] S. E. Robertson, "Theories and models in information retrieval," Journal of documentation, vol. 33, No. 2, pp. 126-148, London,1977.
[50]J. J. Rocchio, "Relevance feedback in information retrieval," Prentice Hall Inc, pp.313-323, 1971.
[51] S. E. Robertson, "On relevance weight estimation and query expansion," Journal of documentation, vol. 42, pp. 182-188, 1986.
[52] S. E. Robertson, "On term selection for query expansion," Journal of documentation, vol. 46, pp. 359-364, 1990.
[53] S. E. Robertson, "The probability ranking principle in IR," Journal of documentation, vol. 33, pp. 294-304, 1977.
[54] N. Alemayehu and P. Willett, "The effectiveness of stemming for information retrieval in Amharic," *Program: electronic library and information systems*, vol. 37, pp. 254-259, 2003.
[55] K. S. Jones and C. J. van Rijsbergen, "Information retrieval test collections," Journal of documentation, vol. 32, pp. 59-75, 1976.
[56] C. J. van Rijsbergen and K. S. Jones, "A test for the separation of relevant and non-relevant documents in experimental retrieval collections," Journal of documentation, vol. 29, pp. 251-257, 1973.
[57] W. M. Shaw, et al., "The cystic fibrosis database: Content and research opportunities," Library & information science research, vol. 13, pp. 347-366, 1991.
[58] C. T. Yu, et al., "A generalized term dependence model in information retrieval," Information Technology: Research and Development, vol. 2, pp. 129-154, Cornell University, 1983.
[59] P. Bollmann-Sdorra and V. V. Raghavan, "On the necessity of term dependence in a query space for weighted retrieval," JASIS, vol. 49, pp. 1161-1168, Germany, 1998.
[60] W. B. Croft and D. J. Harper, "Using probabilistic models of document retrieval without relevance information," Journal of documentation, vol. 35, No. 4, pp. 285-295, 1979.
[61] R. M. Losee Jr, "Term dependence: truncating the Bahadur Lazarsfeld expansion," Information Processing & Management, vol. 30, pp. 293-303, 1994.
[62] R. N. Oddy, "9 Laboratory tests: automatic systems," 1981.

Appendix One: Document-Query matrix used for relevance judgment; where R= relevant, NR=non relevant

	A	B	C	D	E	F	G	H	I	J
Tgfile1.txt	NR	NR	R	R	NR	NR	NR	NR	NR	NR
Tgfile2.txt	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Tgfile3.txt	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Tgfile4.txt	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Tgfile5.txt	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Tgfile6.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile7.txt	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Tgfile8.txt	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Tgfile9.txt	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Tgfile10.txt	NR	NR	NR	NR	NR	R	NR	NR	R	NR
Tgfile11.txt	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Tgfile12.txt	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile13.txt	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile14.txt	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile15.txt	NR	R	R	NR	NR	NR	NR	NR	NR	NR
Tgfile16.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile17.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile18.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile19.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile20.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile21.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile22.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile23.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile24.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile25.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile26.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile27.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile28.txt	NR	R	R	NR	NR	NR	NR	NR	NR	NR
Tgfile29.txt	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile30.txt	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile31.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile32.txt	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile33.txt	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile34.txt	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile35.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile36.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile37.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile38.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile39.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile40.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR

Tgfile41.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile42.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile43.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile44.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile45.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile46.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile47.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile48.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile49.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile50.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile51.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile52.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile53.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile54.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile55.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile56.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile57.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile58.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile59.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile60.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile61.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile62.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile63.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile64.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile65.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile66.txt	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Tgfile67.txt	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Tgfile68.txt	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Tgfile69.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile70.txt	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Tgfile71.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile72.txt	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Tgfile73.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile74.txt	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Tgfile75.txt	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Tgfile76.txt	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Tgfile77.txt	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Tgfile78.txt	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Tgfile79.txt	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Tgfile80.txt	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Tgfile81.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile82.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile83.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile84.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile85.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile86.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile87.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile88.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR

Tgfile89.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile90.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Tgfile91.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Tgfile92.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Tgfile93.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Tgfile94.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Tgfile95.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Tgfile96.txt	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Tgfile97.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile98.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Tgfile99.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Tgfile100.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Tgfile101.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Tgfile102.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Tgfile103.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Tgfile104.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Tgfile105.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Tgfile106.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Tgfile107.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile108.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile109.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile110.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile111.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile112.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile113.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile114.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile115.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile116.txt	NR	NR	NR	NR	NR	NR	NF	NR	R	NF
Tgfile117.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Tgfile118.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Tgfile119.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Tgfile120.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile121.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile122.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile123.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile124.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile125.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile126.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile127.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile128.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile129.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile130.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile131.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Tgfile132.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile133.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile134.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile135.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile136.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR

Tgfile137.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile138.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile139.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile140.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile141.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile142.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile143.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile144.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile145.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile146.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile147.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile148.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile149.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile150.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile151.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile152.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile153.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile154.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile155.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile156.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile157.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile158.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile159.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile160.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile161.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile162.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile163.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile164.txt	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile165.txt	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile166.txt	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile167.txt	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile168.txt	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile169.txt	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile170.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile171.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile172.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile173.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile174.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile175.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile176.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile177.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile178.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile179.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile180.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile181.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile182.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile183.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile184.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR

Tgfile185.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile186.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile187.txt	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Tgfile188.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile189.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile190.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile191.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile192.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile193.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile194.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile195.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile196.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile197.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile198.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile199.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile200.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile201.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile202.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile203.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile204.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile205.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile206.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile207.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile208.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile209.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile210.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile211.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile212.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile213.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile214.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile215.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile216.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile217.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile218.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile219.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile220.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile221.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile222.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile223.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile224.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile225.txt	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Tgfile226.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile227.txt	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Tgfile228.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile229.txt	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile230.txt	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile231.txt	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile232.txt	R	NR	NR	NR	NR	NR	NR	NR	NR	NR

Tgfile233.txt	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile234.txt	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Tgfile235.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile236.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile237.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile238.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile239.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile240.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile241.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile242.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile243.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile244.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile45.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile2246.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile247.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile248.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile249.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile250.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile251.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile252.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile253.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile254.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile255.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile256.txt	NR	R	R	NR	NR	NR	NR	NR	NR	NR
Tgfile257.txt	NR	R	R	NR	NR	NR	NR	NR	NR	NR
Tgfile258.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile259.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile260.txt	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile261.txt	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile262.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile263.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile264.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile265.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile266.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile267.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile268.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile269.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile270.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile271.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile272.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile273.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile274.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile275.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile276.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile277.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile278.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile279.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile280.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR

Tgfile281.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile282.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile283.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile284.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile285.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile286.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile287.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile288.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile289.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile290.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Tgfile291.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile292.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile293.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile294.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile295.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile296.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile297.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile298.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile299.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Tgfile300.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR

Appendix Two: Tigrinya Scripts

ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ	ሏ				
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ	ሗ				
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ	ሟ				
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ	ሧ				
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ	ሯ				
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ	ሷ				
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ	ሿ				
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቊ	ቋ	ቌ	ቍ
ቐ	ቑ	ቒ	ቓ	ቔ	ቕ	ቖ	ቘ	ቚ	ቛ	ቜ	ቝ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ	ቧ				
ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ	ቯ				
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ	ቷ				
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ	ቿ				
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኊ	ኋ	ኌ	
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ	ኗ				
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ	ኟ				
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ	ኧ				
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	ኲ	ኳ	ኴ	
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ	ዀ	ዂ	ዃ	ዄ	
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ	ዟ				

Appendix Three: Tigrinya (Ethiopic) Vs Arabic Numbers

ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ	ዧ				
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ					
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ	ዷ				
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ	ጇ				
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጓ	ጔ	ጕ	
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ	ጧ				
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ	ጯ				
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ	ጿ				
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ	ፏ				
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ	ፗ				

፩	1
፪	2
፫	3
፬	4
፭	5
፮	6
፯	7
፰	8
፱	9
፲	10
፳	20
፴	30
፵	40
፶	50
፷	60
፸	70
፹	80
፺	90
፻	100
፼	10000

Appendix Four: Stop-word list used in this research

ሓልሓሊፉ	ብምቕፃል	ንሳተን	ኣይዐመፅኩኻን	ከምዘይብልኪ	ዘይብልኩም
ሓሰር	ብምባል	ንሳቶም	ኣይዐመፅኩኻን	ከምዘይብልካ	ዘይብልኪ
ሓሰበ	ብምንታይ	ንስኩም	ኣይከኣለን	ከምዘይብልክን	ዘይብልካ
ሃረግ	ብምኽንይዶ	ንስካ	ኣይኮነን	ከምዘይብሎም	ዘይብልክን
ሓቂዩ	ብሞት	ንስክን	ኣይፈልጥኩኹምን	ከምዙይ	ዘይብልኹም
ሀብዋ	ብሰንኪ	ንስኺ	ኣይፈልጥኩኹምን	ከምዚ	ዘይብልኺ
ሀብዎ	ብሳልሰይቲ	ንስካ	ኣዳነመሰላት	ከምዚአ	ዘይብልካ
ሐዚ	ብስወሰድዎ	ንስካትኩም	ዓድን	ከምዚኣቶም	ዘይብልክን
ሓደ	ብተን	ንስካትክን	ኣገልግሎትካን	ከምዚኣን	ዘይብሎም
ሓደሓደ	ብተወሳኪ	ንስካኽ	እሎም	ከምዚኦም	ዘይትኣትዉ

ሓጢኣይት	ብትሰትይዎን	ንስጋ	እምበር	ከምዛ	ዘይትኣትዉ
ሕሩያት	ብቶም	ንሶም	እምቢ	ከምዝኮነ	ዘይተፈርድ
ሕዚ	ብነብዩ	ንብምልኣም	እምነቶም	ከምዝኮኑ	ዘይኮነስ
ሕድሕድ	ብንያም	ንተን	እምነቶም	ከቶ	ዘይኰኖምከ
ሕጂ	ብእን	ንቶም	እሞ	ከዓ	ዘይኾነስ
ህግድፍ	ብኡ	ንዑኡ	እሞ	ከእሊ	ዘዳለወላ
ለዉጢ	ብኡብኡ	ንዓና	እስኪ	ከከም	ዘጠቓልሎ
ላዕለዎት	ብኢስልምናን	ንዓኣ	እስካብ	ከዚ	ዚሕልውዎን
መሰረታቱ	ብኢራን	ንዓዓቶም	እተበሀለ	ከይትንዕቓ	ዚህቦ
መሰረታዊ	ብኣምሆይ	ንዓኩም	እተዐጥቀ	ከፈተ	ዚመፁን
መብልዕሲ	ብኣተን	ንዓካ	እተዛረቦ	ከፈቶ	ዚስዕብዎ
መን	ብኣና	ንዓካትክን	እቱይ	ኩለካትክን	ዚተኽኸ
መንእሰያትን	ብኣንፃሩ	ንዓክን	እቲ	ኩሉ	ዚንእስ
መንነ	ብኣኩም	ንዓኹም	እቲ	ኩላተን	ዚእዘዙሉስ
መኣዝ	ብኣኪ	ንዓኽን	እቲኣን	ኩላትና	ዚኸውን
መዓዝ	ብኣካ	ንዓይንንዓክን	እቲኣ	ኩላትኩም	ዚወፅእ
መደባት	ብኣካትኩም	ንዓይንዓክን	እቲኣ	ኩላቶም	ዚፅውዑ
ሚዛን	ብኣካትክን	ንኣፍሪቃውያን	እቲኣተን	ኩላኩም	ዚፈልጦ
ማሕላኻ	ብኣክን	ንኤስሮም	እቲኣቶም	ኩልና	ዝሐባእካዮ
ማለተን	ብኣይ	ንዕአን	እቲአም	ኩልክን	ዝለመነ
ማለተይ	ብእአም	ንዕዓተን	እታ	ኩሎም	ዝመልኤ
ማለቱ	ብከምዚ	ንዕአም	እታ	ኩን	ዝመልእ
ማለታ	ብኩላኩም	ንዕዉርስ	እቶም	ኪስዕበኒ	ዝመልኩ
ማለት	ብኩልና	ንኪይንሰማማዕ	እና	ኪርእይዎ	ዝመፅእ
ማለት	ብኩልክን	ንኩለ	እናተ	ኪብሎም	ዝምልስዎን
ማለትና	ብኩሎም	ንኩለን	እናተዛተየ	ኪትሐዝ	ዝሰለብዎም
ማለትኩም	ብውሽጥን	ንኩሉ	ዕንቅርቢት	ኪነድድ	ዝሸጦምን
ማለትኪ	ብዘይ	ንኩሊኡ	እንተለና	ኪኰኑ	ዝበለ
ማለትካ	ብዘይካ	ንኩላትና	እንተለኩ	ኪኽይድ	ዝበሉ
ማለትክን	ብዘፍርስ	ንኩሎም	እንተለኩም	ኪፈትሖ	ዝበልና
ማለቶም	ብዛዕባ	ንከወፅእ	እንተለኪ	ኪፍተን	ዝበልኩም
ማባእካ	ብዛዕባኡ	ንዘለና	እንተለካትኩም	ካሊእ	ዝበልኩኹም
ማንም	ብዝብል	ንዘለወን	እንተለክን	ካልኦት	ዝበልኹም
ማዕረ	ብደገን	ንዘለዉ	እንተለዉ	ካልኦትባ	ዝበልኽን
ማዕዶ	ብፅቡቅ	ንዘለዎ	እንተላ	ካልኦትን	ዝበክን
ምሳኩም	ተለወጠ	ንዘለዎም	እንተሎ	ካበይከ	ዝበዝሑ
ምሳካትኩም	ተልእኾ	ንዘን	እንተኮነእውን	ካባይሲ	ዝበዝሕ
ምሳክን	ተመጣጣኒ	ንዘየብለን	እንተኮነግና	ካብ	ዝበዝሑ
ምሳኹም	ተመፀት	ንዘየብልና	እንተኮነግን	ካብተን	ዝብሉ
ምሳኺ	ተቐጥዓ	ንዘየብሎም	እንተኮይነ	ካብታ	ዝብላ
ምሳኻ	ተባሂሉ	ንዘይብልና	እንተኮይነን	ካብቶም	ዝብል
ምሳኻትኩም	ተነሳሕ	ንዘም	እንተኮይኑ	ካብኡ	ዝተኻየደ

ምሳኻትኹም	ተኸላኻሊ	ንጀው	እንተኮይና	ካብዚ	ዝተወሓስዎ
ምሳኸን	ተጋፈጡ	ንፅህናን	እንተኮይና	ካብዚኦም	ዝተወሓስዎ
ምሳይ	ታሕቲ	ንፖሊቲካውን	እንተኮይንኩም	ካናቲራታት	ዝተጣመረ
ምስ	ትህቡናኢሎም	ኣለካ	እንተኮይንኪ	ኬድኩም	ዝተፍለላዩ
ምስ	ትካላትጅማን	ኣለኹ	እንተኮይንካ	ክልተ	ዝከውን
ምስ	ትኹረት	ኣለዉዎ	እንተዘይተጠርኒፉ	ክምስሎ	ዝኮነ
ምስመን	ትፀልኦ	ኣሎ	እንተዘይተጠርኒፉ	ክሳዕ	ዝኮነት
ምስምስ	ነህዛብን	ኣሎና	እንተዘይኮነ	ክስታይ	ዝኮና
ምስተን	ነበረ	ኣሎኹ	እንተዘይኮነ	ክስቶ	ዝኸበደኩምን
ምስቲ	ነበሩ	ኣሎኻ	እንተዘይኮይነ	ክስይ	ዝኸውን
ምስቲ	ነተን	ኣብቲ	እንተይኮነስ	ክሪስታል	ዝኸሰሱን
ምስታ	ነቱይ	ኢለ	እንተድኣ	ክብል	ዝኾነ
ምስቶም	ነቲ	ኢለን	እንተድኣ	ክትሰምዕን	ዝኾነት
ምስአን	ነታ	ኢሉ	እንታወይቲ	ክትኣርዩ	ዝኾና
ምስኡ	ነቶም	ኢላ	እንታዋይ	ክትክደንን	ዝያዳ
ምስኣ	ነናተን	ኢላተን	እንታዎት	ክንኣቱ	ዝጉበሩ
ምስኣቶም	ነናይ	ኢላትኩም	እንታይ	ክንደይ	ዝጉበሮ
ምስኦም	ነዘን	ኢልና	እንትባሃል	ክንዲ	ዝፈጠረ
ምስዚ	ነዚ	ኢልኩም	እንትትሕፀ	ክንድቲ	ዝፈጠሮ
ምሽታዊ	ነዚአን	ኢልካ	እንትትሕፀ	ክንፈነ	ዝፍለጥእቲ
ምተደግመት	ነዚኣ	ኢልክን	እንትከውን	ክከውን	ዞና
ምንታይሲ	ነዚኣተን	ኢሎም	እንትኮኑ	ክኾና	የለናን
ምዓዝ	ነዚኣቶም	ኢና	እንትኮና	ክዋ	የለን
ምእንተዙይ	ነዚኦም	ኢና	እንከለና	ክፍለ	የለኩን
ምእንቲ	ነዞም	ኢኹም	እንኮ	ኮነ	የለካን
ምእንቲዚ	ነይሮም	ኢኺ	እንደገና	ኮይኑ	የለኹን
ምኻኑን	ነገር	ኢኻ	እንዳ	ኮይና	የለኻን
ምኺን	ነገርን	ኢዩ	እከለ	ኮይኖም	የለዋን
ምግባሩ	ነገርዚ	ኢዮ	እከሊት	ኸምዚ	የለውን
ምግባር	ነገድ	ኢዱ	እኳ	ኸባብያዊ	የላን
ሞተት	ነጋንቲ	ኢዳ	እወ	ኸኣ	የልቦን
ሰለስተ	ነጋዳይ	ኢድ	እውን	ኸኣ	የስሕቶም
ሰብ	ነጎድጓድ	ኣሕዋትካን	እዙይ	ኸይትዕንቀፍ	የረጋግፅ
ሰዓት	ናባና	ኣለኹ	እዚ	ኸይፈልጥ	የፀምልዉ
ሰይፉ	ናባኩም	ኣለዋ	እዚአን	ኹሉ	ይመፅእ
ሰይፊ	ናባካ	ኣለዎ	እዚኣ	ኹሉን	ይንገስ
ሱዑዲያን	ናባክን	ኣላ	እዚኣ	ኺመፅእ	ይኣይ
ስለ	ናባይ	ኣልያቶስ	እዚኣተን	ኺትንስኡን	ይኩን
ስለቲ	ናብ	ኣሎ	እዚኣቶም	ኺገልፀሉ	ይኩንምበር
ስለዘለና	ናብቲ	ኣሎና	እዚኦም	ኻልኣይ	ይኩንደኣ
ስለዘላ	ናብታ	ኣሎንቲ	እዚዩ	ኻብ	ይኩንደኣምበር
ስለዘሎ	ናብቶም	ኣሎኹ	እዛ	ኽትምፅውት	ይኩንደኣእምበር

ስለዘየለ	ናብቶም	ኣመልከቱ	እዞም	ኽፅሕፍ	ይኹን
ስለዘየለዉ	ናብ'ቶም	ዓመቱ	እየ	ኾነ	ይኹንእምበር
ስለዘየላ	ናብእን	ዓመታዊ	እየን	ኾኑ	ይኹንደኣ
ስለዙይ	ናብኡ	ዓመት	እዩ	ወላእኳ	ይኹንደኣምበር
ስለዚ	ናብኣ	ኣመጋግባ	እያ	ወላውን	ይኹንደኣእምበር
ስለዝኾነ	ናብኣተን	ኣማራፅ	እያተን	ወርሕን	ይኽበርዩ
ስለዝይ	ናብኣቶም	ኣምፅእዎ	እያቶም	ወትሩ	ይውሃብ
ስሙ	ናብኣም	ኣሰሩ	እይኣይ	ወዘተ	ይግባእ
ስም	ናብዚ	ኣሳናበተ	እዮም	ወይ	ይፍለጥ
ስረ	ናብዚኣን	ኣስሓይታ	ኣም	ወይስ	ዮናስ
ስሩ	ናብዚኣም	ኣርባዕተ	ከሕቅፎም	ወይኒውን	ደሌኹምሲ
ስነ	ናተን	ኣርከተይ	ከለና	ወይክ	ደቆምዶ
ረጂሒታት	ናተይ	ኣቅሓ	ከለዋ	ወይኽ	ደናግል
ረኣየ	ናቱ	ኣቕሓ	ከመይ	ወይውን	ደንገፀሎም
ረኣያ	ናታ	ኣበይ	ከማና	ወገን	ደኣ
ርሕቐ	ናታተን	ኣባና	ከማኩም	ዋላ	ደፋእቲ
ርእስኹም	ናታተን	ኣባኩም	ከማኪ	ዋላእኳ	ዲኹም
ሸውዓተ	ናታትኩም	ኣባኪ	ከማካ	ዋላውን	ድሕሪ
ሺሕ	ናታትክን	ኣባካ	ከማክን	ውሑዳት	ድሕሪሕዚ
ቁፅሉ	ናታቶም	ኣባክን	ከማይ	ውሑድ	ድሕሪት
ቅድሚ	ናትና	ኣባኻትክን	ከም	ውልቀ	ድሕሪትን
ቅድሚ	ናትኩም	ኣባይ	ከምተን	ውን	ድማ
ቅድሚት	ናትኪ	ኣብ	ከምቱ	ውዳሴን	ድኣ
ቅድም	ናትካ	ኣብቲ	ከምቲ	ዉን	ድያብሎስ
ቆፎይ	ናትክን	ኣብእን	ከምቲአን	ዘለና	ገለ
ቐዳማይ	ናቶም	ኣብኡ	ከምቲኦም	ዘለኩ	ገለገለ
ቐፅሩ	ናይ	ኣብኣተን	ከምታ	ዘለኩም	ገምገም
በ	ናይመን	ኣብኣቶም	ከምቶም	ዘለኪ	ገቢርካዮም
በሎም	ናይቲ	ኣብኦም	ከምናታ	ዘለካ	ገና
በብሓደ	ናይዘን	ኣብዙይ	ከምናቶም	ዘለክን	ገንሸልን
በተን	ናይዚ	ኣብዚ	ከምአን	ዘለኹ	ገንሸልን
በቲ	ናይዛ	ኣብ'ዚ	ከምኡ	ዘለኹም	ገገለ
በቲአን	ናይዞም	ኣብዛ	ከምኡውን	ዘለኺ	ጉዱል
በቲኣ	ናፍተን	ኣብዝሓ	ከምኣተን	ዘለኻ	ግራት
በቲኣተን	ናፍቲ	ኣተሓሕዛ	ከምኣቶም	ዘለኽን	ግራት
በቲኦም	ናፍታ	ኣታኽልቲ	ከምኦም	ዘለዋ	ግቡእ
በታ	ናፍቶም	ኣትማን	ከምዘለና	ዘለው	ግብረይ
በቶም	ን	ኣትያላ	ከምዘለኩ	ዘለውካ	ግብርታት
በእጋሮምውን	ንሕና	ኣትያላ	ከምዘለኩም	ዘለውኻ	ግና
በዚ	ንሕያዎትን	ኣቶም	ከምዘለኪ	ዘለዉ	ግን
በየነይቲ	ንላዕለዎት	ኣነ	ከምዘለካ	ዘለዎ	ግዳ
ባ	ንመላእ	ዓንተዎ	ከምዘለክን	ዘላ	ጠቕምቲ

ባእታ	ንመን	ኣዕራብን	ከምዘለወን	ዘሎ	ጠጠው
ቤ	ንመንፈስ	ኣዕፃውን	ከምዘለዋ	ዘሎን	ጥራሕ
ብ	ንመደባቱን	ኣኮነን	ከምዘለዉ	ዘቕርቡ	ጥራይ
ብሓፈሻ	ንመጓዲ	ኣወቓቕራን	ከምዘላ	ዘበሉ	ጥዑይኣምራቲን
ብሉይን	ንሙሴ	ኣወፈዩ	ከምዘይብለን	ዘይሰምዖም	ጥፋኣት
ብልሂ	ንምርግጋፅን	ኣዋጅንናይ	ከምዘይብለይ	ዘይሰምዖም	ፅቡቕ
ብልቡ	ንምንታይ	ኣዝዩ	ከምዘይብሉ	ዘይብለን	ፈርሁ
ብልባ	ንምፍላጥ	ኣይሰተውዕሉን	ከምዘይብላ	ዘይብለይ	ፈርሄ
ብልዒ	ንሰን	ኣይሳዕሳዕኩምን	ከምዘይብልና	ዘይብሉ	ፍሽለትን
ብልዓያ	ንሱ		ከምዘይብልኩም	ዘይብላ	ፍቱንን
ብመበል	ንሳ			ዘይብልና	ፕሮግራምውን
					ፖሊሲ

Appendix Five: Tigrinya prefix lists

ን	ዘይንምስ	ብዝተ	እንተይ	ክሳብዝ
ዝ	ምስተ	ተይ	እንተይተ	ክተ
ዘይ	ምስተተ	ነን	እንት	ክተተ
እት	ምት	ንመ	እንኪይ	ክተት
እተ	ስለ	ንም	እንዳ	ክት
ብኡ	ስለተ	ንብ	እንዳተ	ክነ
ብ	ስለት	ንኪይ	እንድሕር	ክን
በብዝ	ስለዘ	ንክ	እንድሕርዘይ	ክንዲ
ንኽ	ስለዘየ	ንክን	እንድሕርዘይተ	ክንድት
እንተ	ስለዘይተተ	ንዘ	እንድሕርዘይት	ዘተ
እንተዘይ	ስለዘይት	ንዝ	እንድሕርዘይት	ዘይ
ኣይ	ስለዚይ	ንዝተ	ከም	ዘይም
ከይ	ስለዝ	ኣይን	ከምተ	ዘይተ
ከይን	ስለዝተ	ኢ	ከምቲ	ዝተ
ዝተ	ስለዝይ	ኣይ	ከምት	ዝተተ
እንዳ	በቢ	ኣይምተ	ከምእን	ዝተት
	በብ	ኣይተ	ከምዘ	ዝተት
	ባተ	እተ	ከምዘይ	
	ብምት	እት	ከምዘይተ	
	ብብ	እና	ከምዘይተተ	
	ብተ	እናተ	ከምዝ	
	ብኣ	እን	ከምዝተ	
	ብዘይ	እንተ	ከየ	

ብዘይም	እንተተ	ከይ
ብዝ	እንተዝ	ከይተ

Appendix Six: Tigrinya suffix lists

ሎም	ናልኩም	ኣዊ	ክናለይን	ኽኖም
ሎምን	ናልኪ	ኣዊነት	ክናሉ	ኽዋ
ምለይ	ናልካ	ኣዋይ	ክናላ	ወታይ
ምሉ	ናልኹም	ኣይነቶም	ክናልና	ዊ
ተናታት	ናልኺ	እሎም	ክናሎም	ዊነት
ቲታት	ናልኻ	እት	ክናኒ	ዋ
ታቱ	ናሎም	እን	ክናና	ዋይ
ታታት	ናታት	እይ	ክኖም	ዋይነት
ታት	ናኒ	ኦም	ክዋ	ውቲ
ታትና	ናና	ኦምኒ	ክየን	ዎ
ታትናን	ናን	ኦምን	ኹልኪ	ዎም
ታትን	ናኩም	ኦታት	ኹልካ	የን
ታትኩም	ናኪ	ኦት	ኹም	ያ
ታትኹም	ናካ	ኦን	ኹምለን	ይነት
ታቶም	ናኹም	ኩላ	ኹምለይ	ይነትን
ታቶምን	ናኺ	ኩልኪ	ኹምለይን	ዮም
ታይ	ናየን	ኩልካ	ኹምኒ	
ትለን	ናዮም	ኩም	ኹምና	
ትለይ	ናዮምን	ኩምለን	ኹምዎም	
ትሉ	ኡምኒ	ኩምለይ	ኺየን	
ትላ	ኡኒ	ኩምለይን	ኻለን	
ትልና	ኡና	ኩምኒ	ኻለይ	
ትልኪ	ኡኩም	ኩምና	ኻለይን	
ትልካ	ኡክን	ኩምን	ኻሉ	
ትሎም	ኡኹም	ኩምወን	ኻልኩም	
ትታት	ኡውን	ኩምዎም	ኻልኹም	
	ኡዎ	ካለን	ኻሎም	
	ኡዎም	ካለይን	ኻሎምን	
	ኡዎን	ካሉ	ኻኒ	
	ኢሎም	ካልኩም	ኻና	
	ኢት	ካልኹም	ኻየን	
	ኣልኩም	ካሎም	ኻዮም	
		ካሎምን	ኽለን	
			ኽለይን	

ትን ትኩም ትኹም ቶም ነቱ ነቱ ነታዊ ነት ነትን ና ናለን ናለይ ናሉ ናላ	ኣልኹም ኣተን ኣታይ ኣት ኣትኒ ኣትና ኣትናን ኣትን ኣትኩም ኣትክን ኣትኹም ኣቶም	ካኒ ካና ካን ካዮም ክለን ክለይ ክለይን ክሉ ክልና ክሎም ክሎምን ክነን ክኑ ክኒ	ኽልና ኽሎም ኽሎም ኽሎምን ኽነን ኽና ኽናሉ ኽናልና ኽናኒ ኽናና ኽን ክና ክናለን	