

**ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING
DEPARTMENT OF COMPUTING**



**A PROBABILISTIC AFAAN OROMO INFORMATION
RETRIEVAL SYSTEM**

SOLOMON REGASSA GUDA

July, 2016
Adama, Ethiopia

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING
DEPARTMENT OF COMPUTING

**A PROBABILISTIC AFAAN OROMO INFORMATION
RETRIEVAL SYSTEM**

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTERS IN INFORMATION SYSTEMS ENGINEERING

SOLOMON REGASSA GUDA

July, 2016
Adama, Ethiopia

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING
DEPARTMENT OF COMPUTING

**A PROBABILISTIC AFAAN OROMO INFORMATION
RETRIEVAL SYSTEM**

SOLOMON REGASSA GUDA

Signature of Board of the Examiners for Approval

Name	Title	Signature	Date
_____	Chair Person	_____	_____
<u>Million Meshesha (PhD)</u>	Advisors	_____	_____
_____	External Examiner	_____	_____
_____	Internal Examiner	_____	_____

DEDICATED TO MY FATHER **MR. REGASSA GUDA AREDO**

Acknowledgments

I would like to express my gratitude to my advisor Dr. Million Meshesha, whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate his vast knowledge and skill in various areas.

I would also like to thank my friends Amanuel H. and Gezaheghe G. who were standing beside me for professional and friendly assistance and guidance.

I must acknowledge my wife Lily Balda, without her support, encouragement and assistance; it would be hard to complete my work on time.

In conclusion, I recognize that this research would not have been possible without the financial assistance of Adama Science and Technology University (ASTU) Graduate Studies the Department of Computing.

Abstract

Developing information retrieval (IR) system for Afaan Oromo language fills the gap between the information stored and the people who use this language as their mother tongue or as a second language to search relevant documents as per their information need. As the language is the most widely spoken language in Ethiopia, it is important to find out ways to access the information available in this language. Therefore, in this study a probabilistic Afaan Oromo IR system is developed.

There are two main operations in the designed IR system: indexing and searching. Documents are indexed using inverted index file and searching operation is done based on probabilistic concept in which *Binary Independence Model (BIM)* is used.

The retrieval performance of the designed IR system is calculated using the three most common IR effectiveness measures; *precision*, *recall*, and *F-measure*. In probabilistic based IR process there are at least two cyclic document retrieval processes: initial retrieval and after query terms are reweighted. The two iterative retrieval processes are done by three different users; each of the users used on average three queries out of the ten prepared for testing. At initial retrieval, the system scored 86% and 41% precision recall and F-measure 53% respectively. After relevance feedback, the performance became 73% precision, 64% recall and 68% F-measure respectively.

As the system retrieval performance evaluation shows, the recall at initial retrieval has improved by 23% and 15% F-measure is well after relevance feedback; which indicates that the system become more effective in terms of retrieving more relevant documents than the first attempt considering the user's relevance feedback.

There is a possibility of improvement for the designed system by extending it to another query reformulation technique such as query expansion, in which such polysemy and synonymy problems can be solved.

Keywords: *Information Retrieval, Afaan Oromo Language, Probabilistic model, Query Reformulation, Binary Independence Model*

TABLE OF CONTENTS

CHAPTER ONE	1
INTRODUCTION.....	1
1.1. BACKGROUND	1
1.2. STATEMENT OF THE PROBLEM AND ITS JUSTIFICATION	3
1.3. OBJECTIVE OF THE STUDY	4
1.3.1. GENERAL OBJECTIVE.....	4
1.3.2. SPECIFIC OBJECTIVES	4
1.4. SCOPE AND LIMITATION OF THE STUDY	5
1.5. METHODOLOGY OF THE STUDY	5
1.5.1. STUDY DESIGN	5
1.5.2. LITERATURE REVIEW.....	6
1.5.3. CORPUS PREPARATION	6
1.5.4. DEVELOPMENT TOOLS.....	7
1.5.5. EVALUATION TECHNIQUES	8
1.6. SIGNIFICANCE OF THE STUDY	8
1.7. ORGANIZATION OF THE THESIS	9
CHAPTER TWO	10
LITERATURE REVIEW.....	10
2.1. OVERVIEW OF IR SYSTEM	10
2.1.1. TEXT PREPROCESSING	11
2.1.2. INDEXING	13
2.1.3. QUERY PROCESSING	15
2.1.4. TERM FREQUENCY AND TERM WEIGHTING.....	15
2.1.5. TERM MATCHING	16
2.2. IR MODELS	17
2.2.1. BOOLEAN MODEL	17
2.2.2. THE VECTOR SPACE MODEL	18
2.2.3. THE PROBABILISTIC MODEL	19
2.3. QUERY OPERATION	21
2.3.1. RELEVANCE FEEDBACK	21
2.3.2. QUERY REFORMULATION	22
2.4. RETRIEVAL EFFECTIVENESS MEASURE	23
2.5. AN OVERVIEW OF AFAAN OROMO	24
2.5.1. ALPHABET OF AFAAN OROMO (QUBEE).....	25
2.5.2. SENTENCE STRUCTURE	26
2.5.3. GRAMMAR.....	26

2.6. CHALLENGES IN AFAAN OROMO TEXT RETRIEVAL SYSTEMS	32
2.7. RELATED WORKS	32
2.7.1. IR SYSTEM FOR AMHARIC LANGUAGE	34
2.7.2. IR SYSTEM FOR AFAAN OROMO LANGUAGE	34
CHAPTER THREE	36
DESIGN AND DEVELOPMENT.....	36
3.1. SYSTEM ARCHITECTURE	36
3.1.1. TEXT PREPROCESSING ALGORITHMS	37
3.2. INDEXING AFAAN OROMO DOCUMENTS.....	38
3.2.1. INVERTED INDEX FILE	38
3.3. PROBABILISTIC MODEL FOR SEARCHING.....	39
3.4. EVALUATING IR SYSTEM PERFORMANCE	41
CHAPTER FOUR.....	45
EXPERIMENTATION AND ANALYSIS	45
4.1. DATA FOR EXPERIMENTATION	45
4.1.1. DOCUMENT PREPARATION.....	45
4.1.2. QUERY PREPARATION.....	45
4.2. OVERVIEW IR SYSTEM PROTOTYPE	46
4.3. TESTING IR SYSTEM PROTOTYPE	50
4.4. DISCUSSION	53
CHAPTER FIVE	54
CONCLUSION AND RECOMMENDATION	54
5.1. CONCLUSION	54
5.2. RECOMMENDATION	56
REFERENCE	57
APPENDIX I: LIST OF AFAAN OROMO STOP-WORDS.....	59
APPENDIX II: USER JUDGED DOCUMENT-QUERY RELEVANCE MATRIX	60
APPENDIX III PYTHON CODE FOR INDEXING.....	63

LIST OF TABLES

TABLE 2.3 DIPHTHONGS[7]	25
TABLE 2.4 AFAAN OROMO VOWELS[7]	26
TABLE 2.5 SAMPLE OF NOUNS IN AFAAN OROMO WITH ENGLISH EQUIVALENT	27
TABLE 2.6 OROMO PERSONAL PRONOUNS	27
TABLE 2.7 VERBS IN AFAAN OROMO AND ENGLISH	28
TABLE 2.8 CONJUNCTIONS IN AFAAN OROMO	29
TABLE 2.9 ADJECTIVES IN AFAAN OROMO PAIRED WITH ENGLISH	30
TABLE 2.10 ADVERBS OF AFAAN OROMO WITH ITS ENGLISH EQUIVALENT	31
TABLE 2.11 SAMPLE PREPOSITIONS IN AFAAN OROMO WITH ENGLISH EQUIVALENT	32
TABLE 3.1 CONTINGENCY TABLE OF RELEVANCE JUDGMENT	41
TABLE 4.1 NUMBER OF DOCUMENTS IN THEIR CATEGORIES	45
TABLE 4.2 LIST OF QUERIES IN THEIR CATEGORIES.....	46
TABLE 4.3 SAMPLE FOR TERM-DOCUMENT MATRIX AT INITIAL RETRIEVAL.....	49
TABLE 4.4 SAMPLE FOR RETRIEVED DOCUMENT IN DESCENDING ORDER	50
TABLE 4.5 TERM REWEIGHTING RESULT AFTER RELEVANCE FEEDBACK	50
TABLE 4.6 PERFORMANCE EVALUATION RESULT AT INITIAL RETRIEVAL	51
TABLE 4.7 PERFORMANCE EVALUATION RESULT AFTER RELEVANCE FEEDBACK.....	52

LIST OF FIGURES

FIGURE 2.1 ARCHITECTURE OF TEXTUAL IR PROCESS [26]	10
FIGURE 2.2 SUFFIX TREE STRUCTURE[2]	14
FIGURE 3.2 VOCABULARY FILE AND POSTING LIST	38
FIGURE 3.3 A PROBABILISTIC BASED DOCUMENT RETRIEVAL PROCESS	39
FIGURE 4.1 SAMPLE FOR INVERTED INDEX FILE (VOCABULARY AND POSTING LIST)	49
FIGURE 4.2 PRECISION-RECALL GRAPH AT INITIAL RETRIEVAL	52
FIGURE 4.3 PRECISION-RECALL GRAPH AFTER RELEVANCE FEEDBACK	53

LIST OF ALGORITHM

ALGORITHM 2.1 AFAAN OROMO TEXT STEMMER [27].....	12
ALGORITHM 3.1 TOKENIZING AFAAN OROMO TEXT	37
ALGORITHM 3.2 AFAAN OROMO STOPWORDS REMOVER	38

LIST OF CODES

CODE 4.1 FRAGMENTED PYTHON CODE FOR TOKENIZATION AND PUNCTUATION REMOVE	47
CODE 4.2 FRAGMENTED PYTHON CODE FOR CONVERTING TEXT TO LOWER CASE	47
CODE 4.3 FRAGMENTED PYTHON CODE FOR STOP-WORD REMOVAL	48
CODE 4.4 FRAGMENTED PYTHON CODE FOR STEMMING AFAAN OROMO TERMS	48

LIST OF ABBREVIATION

BIM: Binary Independence Model

GWA: Gumii Waaqeffattota Adduyaa

IDF: Inverse Document Frequency

IR: Information Retrieval

PRP: Probabilistic Ranking Principle

SOV: Subject Object Verb

TF: Term Frequency

CHAPTER ONE

INTRODUCTION

1.1. Background

During ancient Greeks and Romans period there was no efficient ways of storing information while they used to use the papyrus scroll. As the scholars began to write large works that were compilations of data of various sorts, they found it useful to devise various means of organizing the material to make locating certain passages easier for the reader that includes table of contents, Alphabetization, hierarchies of information, and indexing [1,2].

A massive work called *The Natural History in 37 Books* wrote by Pliny the Elder. It was a kind of encyclopedia that comprised information on a wide range of subjects. In order to make it a bit more user friendly, the entire first book of the work is nothing more than a gigantic table of contents in which he lists, book by book, the various subjects discussed. He even appended to each list of items for each book his list of Greek and Roman authors used in compiling the information for that book [2].

There are a few other ancient works which employed arranging material under headings in order to make the writing more users friendly and easier to consult. Valerius Maximus wrote a collection of memorable deeds. The work is divided into nine books, and each book is subdivided into chapters, and each chapter has its own heading, and all entries within that chapter contain anecdotes taken from ancient literature and history which illustrate that theme [1, 2].

Marcus Julius Frontinus, a Roman senator of the late first century A.D. and early second century A.D wrote a book of military stratagems in four books. Each book concerns itself with a specific area of warfare. Each book is then subdivided into chapters that each addresses one specific aspect of the book's major theme. Each chapter has a heading to clue the reader, and the chapter itself consists of brief extracts taken from historical works that illustrate the practical application of the topic [2].

Finally, Aulus Gellius wrote a work entitled *The Attic Nights* in 20 books. The work is a crazy quilt assortment of items on Greek and Roman history, philosophy, grammar, rhetoric, and antiquarian material. Since the work was composed with no real order but as the various topics occurred to the author, each chapter of every book concerns an isolated subject, and this subject is clearly spelled out in a title heading that stands at the beginning of the chapter.

A reader could therefore skim through a book and locate the subject by glancing over the titles of the chapters [2].

One method of information organization which we take for granted nowadays, namely alphabetization, it was probably first devised by Greek scholars of the third century B.C. at the library of Alexandria in Egypt in order to help them organize the growing numbers of Greek literary works[1,2].

The history of Indexing book goes back to ancient Rome. There, when used in relation to literary works, the term *index* was used for the little slip attached to papyrus scrolls on which the title of the work and sometimes the name of the author were written so that each scroll on the shelves could be easily identified without having to pull them out for inspection [2].

In this modern age, there is an increasing amount of full text materials available in various languages through the internet and other information suppliers which makes searching for relevant document challenging. To resolve the problem, researchers introduce a method for information retrieval (IR). IR deals with the representation, storage, organization of, and access to information items[1]. The representation and organization of the information items should provide users with easy access to the information in which they are interested in. IR usually deals with natural language text which is not always well structured and could be semantically ambiguous [2].

The purpose of an IR system is to find the most relevant materials for the user while it reduces the least relevant ones from the available collection. The ability of such system to accomplish this achievement can be evaluated using precision and recall [3].

Basically IR system is divided in two sub systems; Indexing and Searching[4]. Indexing is an offline process of organizing documents using keywords extracted from the collection so as to speed up access to desired information from document collection as per users query. Searching is an online process that scans document in corpus to find relevant documents that matches users query. The focus of quality of IR design is in evaluating both retrieval effectiveness and efficiency. Efficiency is about optimizing computing resources such as the needed storage space and time complexity, however effectiveness concerned with relevancy of document retrieved that satisfies user's information need [2].

Due to the rapidly expanding use of the Internet for communication and dissemination of information, electronic information sources are now available in an ever-increasing number of

languages [5]. World Internet Usage and Population Statistics report indicates that in June 2007 the percentage of English language users on internet was only 31.2% as compared to 45% in June 2001[6]. This figure implies that there is an increase in non-English language contents on the Web, which shows the need for more attention to non-English documents on the Web[6].

According to the World Internet Usage and Population Statistics report of 2007[6], the number of Internet users in Ethiopia with a total population of about 79 million has grown by 1,540% from December 2000 to August 2007. Based on the above data one can suggest that there is a great deal of native Afaan Oromo internet users [6]

As the number of Afaan Oromo language speakers increase the demand to use the language for communication technologies also growing. Today considering this demand and developing an IR system for Afaan Oromo is very substantial.

1.2. Statement of the Problem and its Justification

Ethiopia is a multilingual country with over 80 distinct languages [6]. Afaan Oromo is one of the major languages that is widely spoken and used in Ethiopia. Afaan Oromo is part of the Lowland East Cushitic group within the Cushitic family of the Afro-Asiatic phylum, in this Cushitic branch of the Afro-Asiatic language family, Afaan Oromo is considered as one of the most extensive languages among the rest of Cushitic languages[6]. Currently Afaan Oromo is used as an official language of State of Oromia. Besides that, this language is the instructional medium for primary and junior secondary schools throughout the state and its administrative zones[6]. As the Ethiopia's statistical report of 2007 shows there are more than 25 million (34.5%) speakers of Afaan Oromo in Ethiopia and this fact shows that, the language has the largest speaker followed by Amharic language which is the official language of Federal Democratic Republic of Ethiopia (FDRE) [7].

There are a number of medias that uses Afaan Oromo as primary language; for instance, Oromia Television and Radio (Web news), Kallacha Oromiyaa, Bariisaa, Yeroo, Voice of America (VOA) (web news), and different academic and recreational medias[6].

Development of language is highly related with development of technology[4]. The fact that initiated this study is the motivation of enabling development of Afaan Oromo to grow with current information technology support. IR is not being optional technology, it is very

important to everybody. In this information age, information is highly needed than anything else; however, finding this important information needs system support [6].

Some related works have been done by different researchers. One of the recent researches is done by Gezahegn G. [24] which is ‘Afaan Oromo text retrieval’ using Vector Space Model (VSM). The VSM has its own features such as term frequency weighting and similarity measurement. However, it misses some of the features that probabilistic model provides like term reweighting [6]. The main reason for adopting probabilistic model is the use of relevance feedback. The relevance feedback breaks down the whole searching task to a sequence of steps that makes it easy to grasp. It also provides controlled process, which is designed to emphasize that some of the documents are relevant and deemphasizing that others are non-relevant[2].

It is therefore, the aim of this study is to adopt probabilistic model for developing Afaan Oromo information retrieval system.

To this end, the following research questions are explore, and answered in this study.

- What are the linguistic features of Afaan Oromo language?
- How to adopt probabilistic model for Afaan Oromo text retrieval?
- To what extent the probabilistic model enables to design an effective Afaan Oromo text retrieval system?

1.3. Objective of the study

1.3.1. General Objective

The general objective of this study is to design, develop, and experiment a probabilistic Afaan Oromo IR system.

1.3.2. Specific Objectives

- To review related works done on IR for Afaan Oromo language done by many researchers.
- To understand Afaan Oromo language that helps to perform the required text operation such as, tokenization, normalization, stop-word removal, and stemming.
- To design an architecture for probabilistic Afaan Oromo IR system.
- To develop a prototype of an IR system that comprises indexing and searching modules.

- To evaluate the prototype in terms of its retrieval performance/effectiveness using precision, recall, and F-measure.
- To analyze the result found, explain strength and weakness of the study, and finally pointing out the future work.

1.4. Scope and Limitation of the Study

This study covers on designing, developing, experimenting and evaluating mono-lingual IR system for Afaan Oromo language using one of probabilistic model known as Binary Independence Model (BIM). This IR system searches and retrieves relevant Afaan Oromo text from the prepared corpus to respond for the user's query. About 116 documents and 10 queries are prepared with diversified topics. To identify content bearing index terms, a series of text operations, such as tokenization, normalization, stop-words removal, and stemming are done. Index terms are organized using inverted index file. Preparing term-document matrix, weighing terms, document-query similarity scoring, ranking, term reweighting, document-query similarity scoring, and re-ranking for documents are done using BIM and Probabilistic Ranking Principles (PRP). It is out of the scope of this study to search in multimedia documents such as image, audio, video, and other graphics information.

This study is limited in applying only term reweighting technique after user provides relevance feedback to the system. However, query expansion method is not implemented as it needs further investigation how to integrate with probabilistic information retrieval.

1.5. Methodology of the Study

Research methodology is a systematic way to solve a problem. It is a science of studying how research is to be carried out. Essentially, the procedures by which researchers go about their work of describing, explaining and predicting phenomena are called research methodology. It is also defined as the study of methods by which knowledge is gained. Its aim is to give the work plan of research[8].

1.5.1. Study Design

Broadly, there are two major categories of IR technology and research: semantic and statistical. Semantic approaches attempt to implement some degree of syntactic and semantic analysis; in other words, they try to reproduce to some (perhaps modest) degree the understanding of the natural language text that a human user would provide. In statistical approaches, the documents

that are retrieved or that are highly ranked are those that match the query most closely in terms of some statistical measure [16].

Statistical approaches fall into a number of categories: Boolean, extended Boolean, vector space, and probabilistic. Statistical approaches break documents and queries into *terms*. These terms are the population that is counted and measured statistically. Most commonly, the terms are words that occur in a given query or collection of documents [16].

This study follows experimental research based on statistical approach. The research involves in designing, building, experimentation and analysis to understand and reason out the fundamental IR processes in probabilistic model particularly in Binary Independence Model (BIM) and Probabilistic Ranking Principle (PRP) for Afaan Oromo.

Three different users are participated; each user use 3 queries on average for issuing them to the system for two iterative attempts at initial and after relevance feedback document retrieval process. This iterative testing process helps to identify the optimal performance of the system with respect to different users information need.

The three most known IR system performance evaluation techniques: precision, recall, and F-measure are used to measure IR system's performance. These effectiveness measurement techniques are used in each of cyclic retrieval processes.

Based on the results found, analysis and discussion are done. Finally, the strength and weakness of the system is analyzed which also used to indicate the future work.

1.5.2. Literature Review

In order to identify the gap in previous related works and obtain detail understanding of IR system; available related materials such as books, journal articles, conference papers, and related web sources have been reviewed. These reviews are more of focused on local mono-lingual and Cross Lingual Information Retrieval (CLIR) works.

1.5.3. Corpus Preparation

For the purpose of building Afaan Oromo text corpus, various sources are used such as an official web site of state of Oromia ¹, Voice of America Afaan Oromo language ², Gumii

Waaqeffattota Adduyaa (GWA) ³, International bible society official web site ⁴, Oromia Radio and Television Organization (ORTO) ⁵.

The main data set includes news articles with no domain specific documents that include sport, culture, socio-economic, politics, religious scripts, educational materials, and other governmental policies and agendas. Building diversified data set helps to evaluate the system's retrieval performance.

1.5.4. Development Tools

The development platform and the programming language used are Windows 8 operating system and Python 2.7 version. Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands [9].

Python is selected for the this study, since it is a high-level and an interactive object-oriented-scripting language with many features like easy-to-learn, easy-to-read, easy-to-maintain, a broad standard library, interactive mode, portable, extendable, databases, GUI programming, and its scalability[9].

¹<http://www.ethiopia.gov.et/stateoromia>

²<http://www.voanews.com/oromoo>

³<http://www.waaqeffannaa.org>

⁴<http://www.biblica.org>

⁵<http://www.orto.gov.et>

1.5.5. Evaluation Techniques

After completing the development of the IR system, evaluating the performance of the prototype is one of the required tasks in the study. The evaluation process used 10 queries which work with the corpus already prepared. The main objective of evaluating this IR prototype is, to measure the retrieval performance of the system.

In this study, the most common retrieval performance evaluation techniques: *Precision*, *Recall*, and *F-measure* are used. *Precision* is a fraction of retrieved documents that are relevant; *recall* is a fraction of relevant document that retrieved. *F-measure* is a single measure that trades off precision versus recall. The 11-standard interpolated precision value is used to draw precision-recall curve that uses to show the retrieval effectiveness of the system graphically. To measure performance of the system for multiple queries mean average precision is calculated[10] .

1.6. Significance of the Study

The major contribution of this study is to benefit those peoples who use Afaan Oromo and capable of using IR systems, in which they can satisfy their information need by using their own language to formulate the queries. Users specify their query in their language (Afaan Oromo) and the system retrieves relevant documents written in the same language. The retrieved documents that are relevant for the given query can benefit those who can read and understand contents of Afaan Oromo documents and also the users are not forced to use other language to retrieve documents prepared in their own language. This research work used probabilistic model to search, rank and retrieve relevant documents with respect to the user query.

Therefore, the beneficiaries of this research include: individuals, schools, offices, and other researchers.

The following points indicate the significance of this study.

- Avoid using non-native languages to access and use the resources available in Afaan Oromo language.
- Help to develop the language itself and make it one of communication technology language.
- Give advantages for the writers who use Afaan Oromo language.

- Opens the opportunities for the societies to introduce and develop their social and cultural values using their own language and the technology.
- Help to create motivation and show the directions for interested researchers in the given area.

1.7. Organization of the Thesis

This thesis is organized in five chapters. The first chapter started with introducing the historical background of IR, the problem area followed by the statement of the problem, objective of the study, scope and limitation of the work and methodology of the study.

The second chapter mainly focused on the two sub-topics: Overviewing IR and Afaan Oromo. These topics are intensively reviewed including related works which have been done that regards Afaan Oromo text retrieval systems in which the unsolved problems and gaps to be filled are identified.

The third chapter covers more of designing the probabilistic based IR system, in which the methodologies and techniques selected for this thesis work is discussed in detail. Basically, indexing and searching techniques took more coverage. Furthermore, term weighting and term re-weightings techniques are discussed from the perspectives of probabilistic retrieval approach.

Experimentation, analysis, and interpretation of the result are discussed in the fourth chapter in which data set preparation, query selection, implementation processes are covered in detail including sample retrieval process at initial and after relevance feedback.

Finally conclusion and recommendations are presented in chapter five in which findings of the study, including both strength and challenges are discussed. In the recommendation future research directions are pointed out.

CHAPTER TWO

LITERATURE REVIEW

For the purpose of this thesis work, different related topic materials such as Books, journals articles, local and international research works, conference papers, and other web resources are reviewed and discussed. IR concept is discussed in detail mainly focusing on indexing and searching, IR system evaluation, and an overview of Afaan Oromo language.

2.1. Overview of IR System

Information retrieval (IR) is the process of finding relevant documents of an unstructured text that satisfies an information need from within large collections stored on computers [1]. As depicted in figure 2.1, the IR process includes text processing (*Tokenization, Normalization, Stop-word removal, and Stemming*) for the document and query side, the process of indexing documents, term weighting, term matching, and finally provide the relevant documents in decreasing order to the user[26]. The architecture of textual IR process is presented in the following figure 2.1.

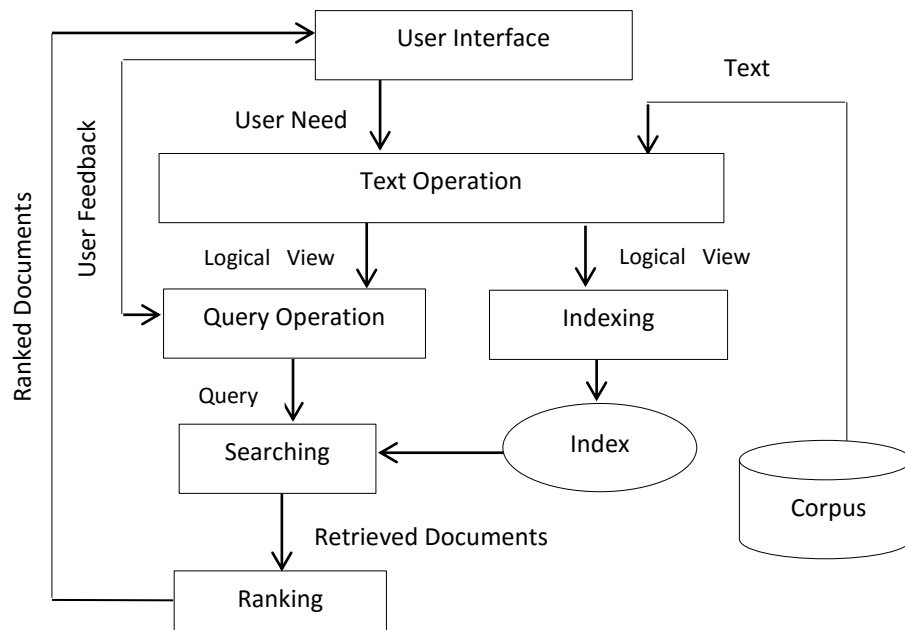


Figure 2.1. Architecture of textual IR Process [26]

2.1.1. Text Preprocessing

The text operation is sequential process of *tokenizing*, *normalizing*, *stop-word removal*, and *stemming* of texts in the documents and the query[1]. Each of this processes are discussed as follows.

Tokenization

Tokenization is the process of chopping on white spaces and throwing away punctuation and special characters other than letters and numbers[11]. For example, if the original document is ‘*Oromiyaan, magaalaa fi baadiyyaa qabdi*’ the tokens will be *Oromiyaan, magaalaa, fi baadiyyaa, qabdi*. So we can define *token* as an instance of a sequence of characters. Each such token is now a candidate for an index entry after further processing. One of challenges related to tokenization is differentiating single word from compound word; for instance, *Hewlett-Packard* Versus *Hewlett and Packard*, *NewYork* Versus *New York*. The other challenge is identifying numerical values like dates, phone numbers, and Internet Protocol (IP) addresses. Additionally, Chinese and Japanese has no space between words, which makes difficult space and punctuation mark based tokenization. That is why tokenization is called a natural language dependent task [1].

Normalization

The common practice in the processes of text operations is normalization which involves in standardization of terms usually in to lower case. This process avoids the problem caused by variations of characters, such as all upper case or mixed upper and lower case of the terms should be converted in to lower case; e.g. ‘ABBA’ to ‘abba’ or in English ‘FATHER’ converted in to ‘father’ [1].

Stop-word Removal

Words which are too frequent among the documents in the collection are not good discriminators. In fact, a word which occurs in 80% of the documents in the collection is useless for purposes of retrieval [1]. Such words are frequently referred to as *stop-words* and are normally filtered out as potential index terms. Articles, prepositions, and conjunctions are natural candidates for a list of stop-words [1]. Afaan Oromo language uses these stop-words such as, ‘fi’ in English ‘and’, ‘yookaan’ in English ‘or’ are some conjunctions considered as stop-words in Afaan Oromo language. List of Afaan Oromo stop-word are presented in Appendix II.

Stemming

Stemming is a process of removing portion of terms or its affixes (i.e., prefix and suffix) in order to reduce the terms in to their root form. A typical example of a stem is the word *compute* which is the stem for the variants *computer*, *computing*, *computed*, and *computation*. Stems are thought to be useful for improving retrieval performance because they reduce variants of the same root word to a common concept [1]. Afaan Oromo stemmer algorithm developed by Debela, 2010 is adopted for this study which is presented as follows in Algorithm 2.1.

```
1. READ the next word to be stemmed
2. OPEN stop word file
   Read a word from the file until match occurs or End of File reached
   IF word exists in the stop word list
       Go to 3
   Else
       Go to 5
3. If word matches with one of the rule
   Remove the suffix and do the necessary adjustments
   Go back to 3
E LSE
   Go o 6
4. Return the word and RECORD it in stem directory
5. IF end of file not reached
   Go to 1
   ELSE
       Stop processing
6. IF there is no application and action exist
   Remove vowel and return the result
   Go to 4
```

Algorithm 2.1 Afaan Oromo Stemmer [27]

The advantage of stemming is that it handles the problems related to inflectional and derivational morphology. Inflections occurred in Afaan Oromo when making plurals (-ootaa, -oolii, -wwam, -lee, -an, -een, -eetii, eeyyi, -ii, etc), when making gender (-ttii, -tuu, -eessa, -a, itti, etc.), in possession (-ke, -ko, -sa,), postposition (-bira, -dura, -irra, -jala, akka, gara, gad, -n, -dha, -jira), conjunction (-fi, -lee,) and in articles (-fi, -lee,). Derivation is the form of the root terms without being inflected. The following table 2.1 presented some examples in Afaan Oromo plural morph [24].

Base form	Inflected form	Gloss
waggaa	wagg-ootaa	‘years’
gowwaa	goww-ootaa	‘fools’
ganda	gand-ootaa	‘Kebeles’
hayyuu	hayy-ootaa	‘scholars’
Hoolaa	hool-ootaa	Sheeps

Table 2.1 Sample for Afaan Oromo plural morph-(o)ootaa on nouns

2.1.2. Indexing

An index is a critical data structure because it allows fast searching over large volumes of data. In the process of searching not all words are equally significant for representing the semantics of a document. In written language, some words carry more meaning than others. Usually, noun words or groups of noun words are the ones which are most representative of document content[2].

Different index structures might be used, such as *suffix array*, *signature file*, and *an inverted file* [2]. Different types of indexing methods are discussed as follows.

Suffix array

A suffix tree also known as PAT tree, it is a compressed tree that contains all the suffixes of the given text as their keys and positions in the text as their values. A suffix tree is made for a set of words instead of only for a single word. It represents all suffixes from its set of words. Each word must be terminated by a different termination symbol or word. Suffix tree allows particularly fast implementation of many important string operations. [29].

Suffix arrays are used for indexing text with an efficient usage of memory space. This type of index can answer more complex queries efficiently. The high cost for constructing process is its main drawback, in which the text must be readily available at query time, and that the

results are not delivered in text position order. This structure can be used to index only words not stopwords as the inverted index as well as to index any text character. This makes it suitable for a wider spectrum of applications, such as genetic databases. However, for word-based applications, inverted files perform better unless complex queries are an important issue[2]. The following example considers string $S = \{abcabx\}$. Figure 2.2 shows the suffix tree corresponding to S .

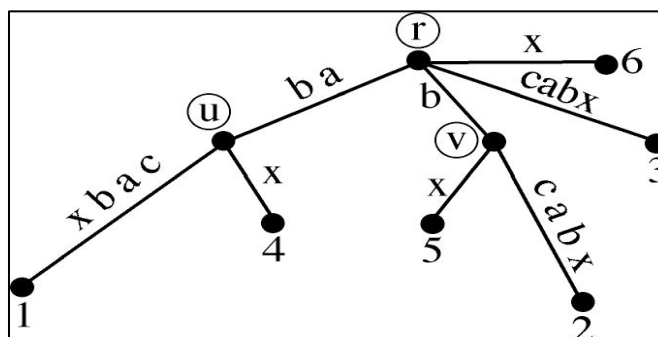


Figure 2.2 Suffix tree structure [2]

Signature file

Signature files are word-oriented index structures based on hashing, that maps words to bit masks of B bits. It divides the text in blocks of b words each. To each text block of size b , a bit mask of size B will be assigned. This mask is obtained by bitwise, the signatures of all the words in the text block. Hence, the signature file is no more than the sequence of bit masks of all blocks and a pointer to each block. The main idea is that if a word is present in a text block, then all the bits set in its signature are also set in the bit mask of the text block. Hence, whenever a bit is set in the mask of the query word and not in the mask of the text block, then the word is not present in the text block [2].

Signature files pose forcing a sequential search over the index. However, although their search complexity is linear, its constant is rather low, which makes the technique suitable for not very large texts. Nevertheless, inverted files outperform signature files for most applications[2].

Inverted file

An inverted file is a word oriented mechanism based on sorted list of keywords which have the link to the document that contains the indexed term. An inverted file comprises two files the

vocabulary and *posting file* which helps to keep the pointer in the vocabulary to the list that posting file lets, the vocabulary kept in memory at search time while the posting list kept in disk to access the documents[2, 12]. Due to the high cost and the complex construction process in suffix arrays and signature files; inverted file structure is used for this study [26].

2.1.3. Query Processing

As the process of indexing the documents completed the next task is processing the user query. The user first specifies his/her information need using natural language; then transform it in to term-document matrix or logical view. In this process text preprocessing tasks are repeated that have been done for the documents. Processing the query is very important to represent the user need and use it as a search key with in indexed documents. Finally query reformulation is done following by user relevance feedback in order to refine the retrieved documents which previously retrieved using the initial query[13].

2.1.4. Term Frequency and Term Weighting

Term weighting scheme referring term frequency which usually denoted as $tf_{t,d}$, the subscript t,d denoting the term and document order. For the document d , the weight for the terms in the document can be determined by the number of occurrence of the terms in the document. The term frequency helps to measure how well that term describes the document contents[1, 2].

The inter-cluster dissimilarity can be computed by measuring the inverse of the frequency idf_t of a term t among the documents in the collection. This factor is usually referred to as the *inverse document frequency* or the *idf factor*. The reason to use an idf factor is that terms which appear in many documents are not very useful for distinguishing a relevant document from a non-relevant one[2].

The *document frequency* (df_t) defines the number of documents in the collection that contain a term t . This is because in trying to discriminate between documents for the purpose of scoring, it is better to use the number of documents containing a term than to use a *collection frequency* (cf_t) for the term. The reason why prefer to use df than cf is that cf and df can behave rather differently. The document frequency df of a term used to scale its weight by denoting N for the total number of documents in a collection then we can define the idf_t of a term t as follows [1].

$$idf_t = \frac{N}{df_t} \log \quad (2.1)$$

Tf-idf Weighting

In order to produce the composite weight for each term in the document we need to combine the definition of term frequency tf and inverse document frequency idf as follows[1, 2].

$$tf \times idf_{t,d} = tf_{t,d} \times idf_t \quad (2.2)$$

or

$$tf \times idf_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t}$$

Therefore, $tf-idf_{t,d}$ assigns a weight to term t in document d that score highest when t occurs many times within a small number of documents or it score lower when the term occurs fewer times in a document[1].

2.1.5. Term Matching

Term matching process is the next step being performed. In this process the preprocessed terms from the user query are used as a search key for the term matching against indexed documents in order retrieve the relevant documents which satisfies the user need. For the use of term matching process, different term similarity score measurement techniques are designed such as Euclidean distance, Cosine similarity measure, dot product, etc.[2]

Euclidian distance calculates “the root of square differences between coordinates of a pair of document and query terms”[2]. The similarity between vectors for document d_i and query q can be expressed as:

$$sim(dj, q) = |dj - q| = \sqrt{\sum_{i=1}^n (w_{ij} - w_{iq})^2} \quad (2.3)$$

Where w_{ij} is the weight of term i in document j and w_{iq} is the weight of term i in the query.

Cosine similarity which is normalized an inner product, calculates “the cosine angle between query and documents vectors by projecting them into a term space”[2]. It measures similarity between d_j and q captured by the cosine of the angle θ between them[10].

$$sim(dj, q) = \frac{\vec{d_j} \cdot \vec{q}}{|\vec{d_j}| |\vec{q}|} = \frac{\sum_{i=1}^n w_{ij} w_{iq}}{\sqrt{\sum_{i=1}^n w_{ij}^2} \sqrt{\sum_{i=1}^n w_{iq}^2}} \quad (2.4)$$

Where w_{ij} is the weight of term i in document j and w_{iq} is the weight of term i in the query q . The denominator involves the lengths of the document and query vectors [10].

Dot product/inner product calculates “the product of the magnitudes of query and document vectors”. It is normalized because it favors long documents with a large number of unique terms and it measure only how many terms matched but not how many terms are not matched. Similarity between vectors for the document d_i and query q can be computed as[10]:

$$sim(dj, q) = dj \cdot q = \sum_{i=1}^n w_{ij} \cdot w_{iq} \quad (2.5)$$

Where w_{ij} is the weight of term i in document j and w_{iq} is the weight of term i in the query q . For this study dot product/inner product similarity measurement technique is used because of its simplistic way of calculation.

2.2. IR Models

There are two good reasons for having models of IR. The first is that models guide research and provide the means for academic discussion. The second reason is that models can serve as a blueprint to implement an actual retrieval system[13].

The classic models in IR consider that each document is described by a set of representative keywords called *index terms*. Given a set of index terms for a document, it is a notice that not all terms are equally useful for describing the document contents. In fact, there are index terms which are simply vaguer than others. Deciding on the importance of a term for summarizing the contents of a document is not a trivial issue. Despite this difficulty, there are properties of an index term which are easily measured and which are useful for evaluating the potential of a term as such[10].

The three classic models in IR are the *Boolean*, *the vector space*, and *the probabilistic models*.

2.2.1. Boolean model

The Boolean model is a simple retrieval model based on set theory and Boolean algebra. Since the concept of a set is quite intuitive, the Boolean model provides a framework which is easy to grasp by a common user of an IR system. Furthermore, the queries are specified as Boolean expressions which have precise semantics[3].

Boolean model is the oldest model of IR. In the Boolean 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 or statement. 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[14].

The Boolean model predicts that each document is either *relevant* or *non-relevant*. There is no notion of a *partial match* to the query conditions [2].

The main *advantages* of the Boolean model are the clean formalism behind the model and its simplicity. The main *disadvantages* are that exact matching using logical operators and binary weighting may lead to retrieval of too few or too many documents. Today, it is well known that index term weighting can lead to a substantial improvement in retrieval performance [4].

2.2.2. The Vector Space Model

The vector space model is one of the first models to appear [15] and has been developed and refined up to the present. Documents and queries are vectors in an n-dimensional space, where each dimension corresponds to an index term. In general, the number of query terms defines this dimensionality. Term weights are calculated by *tf.idf* (discussed in section 2.2.4.) values, discrimination ratios found and documents are ranked in decreasing order of similarity to the query using the cosine correlation $\cos \theta$ to retrieve documents d_i closest to the query q in vector space where d_i is the *tf.idf* weight and q_i the weight for query term i , with $0 \leq \cos \theta \leq 1$. The model can be used at indexing time, but provides higher performance at search time, because of the dynamic nature of the collection. It may be supported by a thesaurus to include important relationships among words or to expand terms to classes, since it is those relationships and classes that are relevant to an individual query, that should be identified[16].

Advantages of using VSM are: term weighting improves answer set quality, partial matching allows retrieval of documents that approximate conditions, and cosine ranking formula sorts documents according to degree of similarity to the query [1,2].

Disadvantages of using VSM are: it assumes independence of index terms; not clear that this is bad still [1,2].

2.2.3. The Probabilistic Model

Probability theory provides a principled foundation for such reasoning under uncertainty[1].

Many IR models might be described as probabilistic. The term probability is usually associated with the methods arising from the work of Stephen Roberts, Karen Spark Jones, and their colleagues at the City University of London, at the University of Cambridge, and at Microsoft Research, Cambridge beginning in the early 1970s. Over that period they developed the probabilistic retrieval model through a series of innovation that have led to its current status as one of the most successful IR models[3].

There are many probabilistic based models developed by different statisticians. The most widely used are *Binary Independence Model*, *Inference Network Model*, and *Belief Network Model*[17, 18]. For this study BIM is used because of some features it has as it is discussed below.

Binary Independence Model

The *Binary Independence Model* (BIM) is the model that has traditionally been used with the *Probabilistic Ranking Principle* (PRP). It introduces some simple assumptions, which make estimating the probability function $P(R|d, q)$ practical. Here, “binary” is equivalent to Boolean: documents and queries are both represented as binary term incidence vectors. That is, a document $d_t = 1$ if term t is present in document d and $d_t = 0$ if term t is not present in d . Similarly, we represent q for the term in the query q . “*Independence*” means that terms are modeled as occurring in documents independently. The model recognizes no association between terms. This assumption is far from correct, but it nevertheless often gives satisfactory results in practice; it is the “naive” assumption of Naive Bayes models. Indeed, Under the BIM model the probability, $P(r|d, q)$ that a document is relevant via the probability in terms of term incidence vectors $p(r|d, q)$. Then, using Bayes rule, we have[1].

$$p(R|d, q) = \frac{P(d, q | R)P(R)}{P(d, q)} \quad (2.6)$$

$$p(\bar{R}|D, Q) = \frac{P(d, q | \bar{R})P(\bar{R})}{P(d, q)} \quad (2.7)$$

Since a document is either relevant or irrelevant to a query, we must have that:

$$\begin{aligned} p(R|d, q) + p(\bar{R}|d, q) &= 1 \\ p(R|d, q) &= 1 - p(\bar{R}|d, q) \end{aligned} \quad (2.8)$$

Driving Ranking Function

Given a query q , wishes are to order returned documents by descending $P(R=1|d, q)$. Under the BIM, this is modeled as ordering by $P(R=1|d, q)$. We can rank documents by their odds of relevance (as the odds of relevance are monotonic with the probability of relevance). This makes things easier because we can ignore the common denominator in equation (2.6 and 2.7), and then can be expressed as[3]:

$$\log \frac{p(R|d,q)}{1-p(R|d,q)} = \log \frac{p(R|d,q)}{p(\bar{R}|d,q)} \quad (2.9)$$

$$= \log \frac{p(d,q|R)p(R)}{p(d,q|\bar{R})p(\bar{R})} \quad (2.10)$$

An immediate benefits of switching to log-odds is that the term $p(d, q)$ cancels out and doesn't need to be estimated. Expanding the joint probabilities in Equation (2.10) in to conditional probabilities using equality $p(d,q|r)=p(d|q,r).p(q|r)$ the joint probabilities in this way and applying Bayes' Theorem a second time gives[3]:

$$\log \frac{p(d|q,R)}{p(d|q,\bar{R})} \quad (2.11)$$

For our first attempt to estimate a value, we consider only the presence and absence of terms in documents and queries. We re-define the random variable d , representing documents in to a vector binary random variables $d=(d_1,..d_2)$ with one dimension for each term in the vocabulary V and with $d_i=1$ indicating term presence and $d_i=0$ indicating term absence. Similarly, we refine the random variable q , representing queries, in to a vector of binary random variables $q=(q_1,q_2)$ with $q_i=1$ indicating term presence and $q_i=0$ indicating term absence[3].

Assumption 1: Given relevance, terms are statistically independent. Then equation (2.11) becomes:

$$\log \frac{p(d|q,R)}{p(d|q,\bar{R})} = \sum_{i=1}^{|v|} \log \frac{p(d_i|q_i,R)}{p(d_i|q_i,\bar{R})} \quad (2.12)$$

The second assumption links the appearance of a term in the query to the probability of its appearance in a relevant document, thus clarifying the role of the query as a bridge between the user's need and a document's relevance.

Assumption 2: The presence of a term in a document depends on relevance only when that term is present in the query[3].

The refined equation (2.8) under assumptions 1 and 2, and considering only the presence and absence of terms is known as *Binary Independence model* as it is shown below[3].

$$\sum_{t \in (q \cap d)} \log \frac{p(d_t=1|R)p(d_t=0|\bar{R})}{p(d_t=1|\bar{R})p(d_t=0|R)} \quad (2.13)$$

Where each d_t either 0 or 1, making explicit the values of random variables and d_t represent the value of random variable from $(d_1, d_2, \dots)$ corresponding to the term t [3].

2.3. Query Operation

Probabilistic Term Weighting at Initial Retrieval

At the beginning of searching for the documents, its relevance (R) is completely unknown based on this concept it is assumed that $p(t_i|R)$ is constant with the value of 0.5 and $p(t_i|\bar{R})$ can be approximated by distribution in the whole collection[10].

$$p(t|\bar{R}) = \log \frac{N-n+0.5}{n+0.5} \quad (2.14)$$

Where N is the total number of documents in the collection and n_i is number of documents contains the term t this equation can be used as the first initial ranking using IDF as the first term weighting[13].

The results at the initial retrieval may not met the user interest due to several reasons such as queries can't specify the user's need precisely; as a result some, answers are correct but others are not. In order to obtain better result, there is a need to apply query reformulation and re-submitting the query iteratively until the results become satisfactory [3].

Query reformulation involves a series of steps. First relevance feedback is obtained to have the necessary information about relevance of documents returned. This is followed by query modification. The relevance feedback can be managed in two ways, user relevance feedback and pseudo relevance feedback[1, 2].

2.3.1. Relevance Feedback

The idea of relevance feedback (RF) is to allow the user involvement in retrieval process of relevant documents so as to get an improved result. In particular, the user gives feedback on the relevance of documents at initial set of results and then the system returned the refined relevant

documents which consider the user's input labeled as relevant[1, 2]. As it is simple, uses short length and no precise query, the user relevance feedback method is used in this study.

Pseudo relevance feedback, on the other hand works by itself rather than depend on the user to choose the top relevant documents. The system simply assumes that the top-ranked documents are relevant, and uses these documents to expand the query with a relevance feedback ranking algorithm. This procedure has been found to be highly effective in some settings, most likely those in which the original query statements are long and precise[2].

2.3.2. Query Reformulation

The query reformulation involves in modifying the given query with two methods. The first one is expanding the original query with new terms and the second one is reweighting the terms. 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; Rocchio algorithm is one of the approaches which used to design vector space model as well as probabilistic term reweighting[1, 2].

The probabilistic model dynamically ranks documents similar to a query q according to the probabilistic principle. The similarity of a document d_j to a query q can be expressed as[10]:

$$sim(d_j, q) \propto \sum_{i=1}^t \left(\log \frac{P(t_i|R)}{1-P(t_i|R)} + \log \frac{1-P(t_i|\bar{R})}{P(t_i|\bar{R})} \right) \quad (2.15)$$

Where $P(t_i|R)$ stands for the probability of observing the term t_i in the set R of relevant documents and $P(t_i|\bar{R})$ stands for the probability of observing the term t_i in the set $\bar{R}$ of non-relevant documents [10].

Initially equation (2.15) cannot be used because the probabilities $P(t_i|R)$ and $P(t_i|\bar{R})$ are unknown. For the initial search assumption often made include:

- a. $P(t_i|R)$ is constant for all terms t_i (typically 0.5) and
- b. The term probability distribution $P(t_i|\bar{R})$ can be approximated by the distribution in the whole collection. These two assumptions produce the following[10].

$$P(t_i | R) = 0.5 \text{ and} \quad (2.16)$$

$$P(t_i | \bar{R}) = \frac{ni}{N} \quad (2.17)$$

Where n_i stands for the number of documents in the collection which contain the term t_i . Substituting into equation (2.15), it becomes [10]

$$sim_{initial}(d_j, q) = \sum_i^t w_{iq} \cdot w_{ij} \log \frac{N-n_i}{n_i} \quad (2.18)$$

After initially documents are retrieved the user labels those relevant documents as relevant. Then the information is used so as to evaluate the probability of $P(t_i|R)$ and $P(t_i|\bar{R})$. The probabilities $P(t_i|R)$ and $P(t_i|\bar{R})$ can be approximated by [10]

$$P(t_i|R) = \frac{|Dr,i|}{|Dr|} \text{ and } P(t_i|\bar{R}) = \frac{n_i-|Dr,i|}{N-|Dr|} \quad (2.18)$$

Where, D_r is the set of relevant retrieved documents as the user judgment and $D_{r,j}$ is the subset of D_r composed of the documents which contain the term t_i . However, equation (2.18) have a problem for some small values of $|D_r|$ and $|Dr,i|$. Thus, a 0.5 adjustment factor is added to the estimation of $P(t_i|R)$ and $P(t_i|\bar{R})$. This will give [10]

$$P(t_i|R) = \frac{|Dr,i| + 0.5}{|Dr| + 0.5} \text{ and } P(t_i|\bar{R}) = \frac{n_i - |Dr,i| + 0.5}{N - |Dr| + 0.5} \quad (2.19)$$

The adjustment factor made at equation (2.19) may provide inadequate estimation in some cases. In this case alternative adjustment factors have been formulated. For instance, replacing adjustment factor 0.5 by $\frac{n_i}{N}$ [7].

$$P(t_i|R) = \frac{|Dr,i| + \frac{n_i}{N}}{|Dr| + 0.5} \text{ and } P(t_i|\bar{R}) = \frac{n_i - |Dr,i| + \frac{n_i}{N}}{N - |Dr| + 0.5} \quad (2.20)$$

2.4. Retrieval Effectiveness Measure

Traditional IR is based on two fundamental assumptions[3]:

- Given a user's information need, represented by a search query, each document in a given text collection is either relevant or non-relevant to this information need.
- The relevance of document d depends on only on the information need and d itself. It is independent of the search engine's ranking of the other documents in the collection.

Recall and Precision

Recall and Precision, perhaps the two oldest measures used in IR evaluation, apply to an unordered set of documents retrieved in response to a query. *Recall* is the fraction of relevant documents contained in the set. Let *Res* be the set of retrieved documents, and *Rel* the set of relevant documents. recall is computed as follows[3].

$$Recall = \frac{|Res \cap Rel|}{|Rel|} \quad (2.21)$$

An effective measure, recall quantifies how exhaustively the search results satisfy the user's information need. It models a retrieval task in which the user wants to find all relevant documents [3].

A search engine could trivially achieve a recall level of 1.0 simply by returning all documents in the collection. Of course, if it did that, most of the documents in the result list would be non-relevant. Precision measures the fraction of retrieved documents relevant and calculated as follows[3].

$$Precision = \frac{|Res \cap Rel|}{|Res|} \quad (2.22)$$

F-Measure

F-measure is the harmonic mean of recall and precision which combines both recall and precision and compared to arithmetic mean, both need to be high for harmonic mean to be high[2].

$$F = \frac{2PR}{P+R} = \frac{2}{\frac{1}{R} + \frac{1}{P}} \quad (2.23)$$

2.5. An Overview of Afaan Oromo

The language Afaan Oromo is one of the major Languages that are widely spoken and used in Ethiopia [19]. Currently it is an official language of State of Oromia (which is the largest region in Ethiopia). Afaan Oromo is part of the Lowland East Cushitic group within the Cushitic family of the Afro-Asiatic phylum. In this Cushitic branch of the Afro-Asiatic language family Afaan Oromo is considered as one of the most extensive languages among the forty Cushitic languages[20].

The writing system of Afaan Oromo is called “Qubee” (Latin-based alphabet) which has been adopted and become the official script of Afaan Oromo since 1991[6].

Like a number of other African and Ethiopian languages, Afaan Oromo has a very rich morphology[7]. It has the basic features of agglutinative languages where all bound forms (morphemes) are affixes. In agglutinative languages like Afaan Oromo, Amharic and Zulu most of the grammatical information is conveyed through affixes (prefixes, infixes and suffixes) attached to the roots or stems. Both Afaan Oromo nouns and adjectives are highly inflected for number and gender. For instance, in comparison to the English plural marker s (-

es), there are more than 12 major and very common plural markers in Afaan Oromo nouns (e.g. -oota, -ooli, -wwan, -lee, -an, een, -oo, etc.). Afaan Oromo verbs are also highly inflected for gender, person, number and tenses. Moreover, possessions, cases and article markers are often indicated through affixes in Afaan Oromo. Since Afaan Oromo is morphologically very productive, derivations and word formations in the language involve a number of different linguistic features including affixation, reduplication and compounding. Obviously, these high inflectional forms and extensive derivational features of the language are presenting various challenges for text processing and IR experiments in Afaan Oromo[6].

2.5.1. Alphabet of Afaan Oromo (Qubee)

Afaan Oromo uses Qubee (Latin based alphabet) that consists of twenty-nine basic letters of which five are vowels, twenty-four are consonants, out of which 18 letters are basic consonants and the remaining six are pair letters/Diphthongs and fall together[20].

The Afaan Oromo alphabet characterized by capital and small letters as in the case of the English alphabet. In Afaan Oromo language, as in English language, vowels are sound makers and are sound by themselves. As presented in table 2.4, Vowels in Afaan Oromo are characterized as short and long vowels.[21]

The basic alphabet in Afaan Oromo does not contain ‘p’, ‘v’ and ‘z’. This is because there are no native words in Afaan Oromo that formed from these characters. However, in writing Afaan Oromo language they are used to refer to foreign words such as “*polisii*” (“police”)[21].

Letters	Pronounced as	Letters	Pronounced as
B	ba in <i>bad</i>	M	ma in <i>man</i>
C	Cha	N	na in <i>narrow</i>
D	da in <i>dad</i>	Q	
F	fa in <i>fat</i>	R	ra in <i>rabbit</i>
G	ga in <i>God</i>	S	sa in <i>sad</i>
H	ha in <i>hard</i>	T	ta in <i>target</i>
J	ju in <i>jump</i>	W	wa in <i>water</i>
K	ca in <i>cat</i>	X	
L	la in <i>large</i>	Y	ya in <i>yard</i>

Table 2.2 Basic Consonants in Afaan Oromo [7]

Letters	Ch	Dh	Dz	Ny	Ph	Sh
Pronounced as	ch in <i>charge</i>		s in <i>vision</i>	Ñ in <i>Españ</i>		sh in <i>shark</i>

Table 2.3 Diphthongs[7]

Short Vowels	Pronounced as	Long Vowels	Pronounced as
a	u in <i>but</i>	Aa	a in <i>bag</i>
e	a in <i>lake</i>	Ee	a in <i>late</i>
i	i in <i>big</i>	Ii	ee in <i>week</i>
o	o in <i>got</i>	Oo	o in <i>doll</i>
u	u in <i>shoot</i>	Uu	oo in <i>book</i>

Table 2.4 Afaan Oromo Vowels[7]

2.5.2. Sentence Structure

Afaan Oromo uses subject-object-verb (SOV) structure (like Amharic). SOV is the type of language structure in which the subject, object and verb appear in that order. For instance, in the Afaan Oromo sentence “Ganzabeen atileeti dha”. “Ganzabeen” is a subject, “atileeti” is an object and “dha” is a verb. The translation of the sentence in English is (“Ganzabe is an athlete”) which has different SVO structure[24].

Afaan Oromo adjectives follow a noun or pronoun; their normal position is close to the noun, they modify while in Amharic adjectives usually precede the noun. For instance, ilma gaarii, gaarii (adj.) follows ilma (noun) but in English “good boy” good (adj.) and boy (noun)[24].

2.5.3. Grammar

Like most other Afro-Asiatic languages, Oromo has two grammatical genders, masculine and feminine, and all nouns belong to either one or the other[7].

Nouns and Genders in Afaan Oromo

There is nothing in the form of most nouns that indicates their gender. A small number of nouns pairs for people, however, end in *-eessa* (m.) and *-eettii* (f.), as do adjectives when they are used as nouns [24]. The following table 2.5 presents nouns in Afaan Oromo with English equivalents.

Afaan Oromo	Obboleessa	Obboleettii	Dureessa	Hiyyeettii	Harmmee
English	Brother	Sister	The rich one (m.)	The poor one (f.)	Mother
Afaan Oromo	Nama	Jabbi	Indaaqo	Hiriyaa	Hojii
English	Person	Calf	Chicken	Friend	Work
Afaan Oromo	Qayyee	Mana	Kophee	Barsiisa	Abbaa
English	Village	House	Shoe	Teacher	Father

Table 2.5 Sample of nouns in Afaan Oromo with English equivalent

Grammatical gender normally agrees with biological gender for people and animals; thus nouns such as *abbaa* 'father', *ilma* 'son', and *sangaa* 'ox' are masculine, while nouns such as *haadha* 'mother' and *intala* 'girl, daughter' are feminine. However, most names for animals do not specify biological gender[7].

Names of astronomical bodies are feminine: *aduu* 'sun', *urjii* 'star'. The gender of other inanimate nouns varies somewhat among dialects[7]. Personal pronouns in Afaan Oromo presented in the following table 2.6 with English equivalents.

English	Base	Subject	Dative	Instrumental	Locative	Ablative	Possessive adjectives
I	<i>ana, na</i>	<i>ani, an</i>	<i>naa, naaf, natti</i>	<i>Naan</i>	<i>Natti</i>	<i>narraa</i>	<i>koo, kiyya (too, tiyya (f.))</i>
you (sg.)	<i>Si</i>	<i>Ati</i>	<i>sii, siif, sitti</i>	<i>Siin</i>	<i>Sitti</i>	<i>sirraa</i>	<i>Kee (tee (f.))</i>
He	<i>Isa</i>	<i>Inni</i>	<i>isaa, isaa(tii)f, isatti</i>	<i>Isaatiin</i>	<i>Isatti</i>	<i>isarraa</i>	<i>(i)saa</i>
She	<i>isii, ishii, i see, ishee</i>	<i>isiin, etc.</i>	<i>ishii, ishiif, ishiitti, etc.</i>	<i>ishiin, etc.</i>	<i>ishiitti, etc.</i>	<i>ishiirraa, etc.</i>	<i>(i)sii, (i)shii</i>
We	<i>nu</i>	<i>nuti, nu'u, nu y, nu</i>	<i>nuu, nuuf, nutti</i>	<i>Nuun</i>	<i>Nutti</i>	<i>nurraa</i>	<i>keenna, keenya (teenaa, teenya (f.))</i>
you (pl.)	<i>isin</i>	<i>Isini</i>	<i>isiinii, isiniif, isinitti</i>	<i>Isiniin</i>	<i>Isinitti</i>	<i>isinirraa</i>	<i>keessan(i) (teessan(i) (f.))</i>
They	<i>isaan</i>	<i>Isaani</i>	<i>isaanii, isaaniif, isaanitti</i>	<i>Isaaniitiin</i>	<i>Isaanitti</i>	<i>Isaanirraa</i>	<i>(i)saani</i>

Table 2.6 Oromo personal pronouns

Afaan Oromo Verbs

Verbs are forms which occur in clause final positions and belong to a distinct category from that of nouns (Nominal), for example [25].

Leensaan dhufte. (Lensa has come)

Tulluun dheeraadha. (Tullu is tall)

Afaan Oromo verbs are also highly inflected for gender, person, number and tenses[6]. Here are examples of each with second person singular subject agreement and the verb stem deem-‘go’ [24], in the following table 2.7.

	Afaan Oromo	English equivalent
Present	Deem-t-a	You go
Past	Deem-t-e	You went
Perfect	Deem-t-eerta	You have gone
Imperative	Deem-i	Go
Subordinate	(akka) deem-t-u	(that) you go
Contemporary	(otuu) deem-t-uu	(while) you are/were going

Table 2.7 Verbs in Afaan Oromo and English

Several optional suffixes are also possible with finite verbs. Case is normally marked on nouns in Afaan Oromo, but with certain verbs, the case suffixes-f (“dative”) and -n (“instrumental”) may appear on the verbs instead. In these cases the suffixes refer to arguments of the verb that are left unstated, corresponding to pronouns in English: *kenneef* ‘he gave (him/her/them)’, *jedheen* ‘he said’ said (to him/her/them)’. Note that these suffixes, like some others in Oromo, lengthen a preceding short vowel [25].

Afaan Oromo has several suffixes functioning as conjunctions that can be attached to verbs: -f, -s, -ti, -tti, -llee, and -yyuu. Some of these are normally combined with other conjunctions consisting of separate words appearing before the verb: *waan hin arganneef* ‘because it (he, etc.) was not found’ [25].

Punctuation Mark

In Afaan Oromo punctuation mark used as any language who use Latin alphabets except apostrophe (‘) which use differently in Afaan Oromo. Apostrophe mark (‘) in English shows possession but in Afaan Oromo it is used in writing to represent a glitch (called *hudhaa*) sound. It plays an important role in the Afaan Oromo reading and writing system. For example, it is used to write the word in which most of the time two vowels are appeared together like “*du’a*”

to mean (“*die*”) with the exception of some words like “*har’a*” to mean “*today*” which is identified from the sound created[21].

Numerals

Oromo has singular and plural number, but nouns that refer to multiple entities are not obligatorily plural. That is, if the context is clear, a formally singular noun may refer to multiple entities: *nama* "man" or "people", *nama shan* "five men" or "five people". Another way of looking at this is to treat the "singular" form as unspecified for number[21].

When it is important to make the plurality of a referent clear, the plural form of a noun is used. Noun plurals are formed through the addition of suffixes, the most common plural suffix is -*oota*; a final vowel is dropped before the suffix.

Conjunctions in Afaan Oromo

Conjunction	English equivalent
Fi	And
Yookaan (yookiin)	Or
Waan ta’eef	So, therefore
Yoo	Unless
Waan	For

Table 2.8 Conjunctions in Afaan Oromo

Word Segmentation

The word is the smallest unit of a language. There are different methods for separating words from each other. This method might vary from one language to another. In some languages, the written or textual script does not have whitespace characters between the words. However, in most Latin languages a word is separated from other words by white space character. Afaan Oromo uses white space character to separate words from each other’s called word segmentation [21].

Adjectives

Adjectives are grouped under different categories, like nouns; Oromo adjectives can be either primitive or derived. Adjectives come after the nouns they describe. For example, the adjective *guraacha* “black” in *saree guraacha* comes after the noun it modified. It is also true in the adjective case that not all words after nouns could be adjectives. For example, in *mana citaa* “thatched house”, the word *citaa* “thatched” is not adjective but noun as adjective. Moreover, nouns but not adjectives occur in subject and/or object positions. On the other hand, like nouns,

Oromo adjectives can be either primitive or derived [7]. The following table 2-9 presents adjectives in Afaan Oromo with their equivalent meaning in English.

Testes---	Qualities ---
<i>Afaan Oromo—English</i>	<i>Afaan Oromo—English</i>
<i>Hadhaawaa-- bitter</i>	<i>Hamaa-- bad</i>
<i>Asheeta-- fresh</i>	<i>Qulqulluu-- clean</i>
<i>Sogidaawaa-- salty</i>	<i>Dukkanaawaa-- dark</i>
<i>Oganaawaa-- sour</i>	<i>Ulfaataa-- difficult</i>
<i>kaurgoo baayyatu-- spicy</i>	<i>Xuraawaa-- dirty</i>
<i>mi'aawaa-- sweet</i>	<i>Qooraa-- dry</i>
<i>nyaata hadhaa'aa-- a salty food</i>	<i>Alpha—easy</i>
Boca—Shapes	<i>Duwwaa-- empty</i>
<i>Afaan Oromo—English</i>	<i>mi'aa/qaalii—expensive</i>
<i>Geengoo—circular</i>	Bifoota—Colors
<i>Sirrii-- straight</i>	<i>Afaan Oromo—English</i>
<i>Golarfee-- square</i>	<i>Gurraacha-- Black</i>
<i>Golsadee-- triangular</i>	<i>Baluu-- Blue</i>
<i>manicha geengoo-- The circular house</i>	<i>bifa bunaa-- Brown</i>
Boca—Shapes	<i>Daalacha-- Gray</i>
<i>Afaan Oromo—English</i>	<i>Magariisa-- Green</i>
<i>Geengoo—circular</i>	<i>Diimaa-- Red</i>
<i>Sirrii-- straight</i>	<i>Adii-- White</i>
<i>Golarfee-- square</i>	<i>muka magariisa-- A green tree</i>

Table 2.9 Adjectives in Afaan Oromo paired with English

Adverb

Oromo adverbs are part of speech. Generally they're words which modify any part of the language other than a noun. Adverbs can modify verbs, adjectives (including numbers), clauses, sentences and other adverbs [24].

Adverb in Afaan Oromo divided in to four categories: adverb of time, adverb of manner, and adverb of frequency. In table 2.10 adverbs are listed with equivalent meaning of English adverbs [24].

Adverbs of time		Adverbs of manner	
Afaan Oromo	English	Afaan Oromo	English
Kaleessa	Yesterday	Baayyee	Very
Harr'a	Today	Baayyee	Quite
Bor	Tomorrow	Dhugummaan	Really
Amma	Now	Dafee	Fast
Gaafas	Then	Gaarii	Well
Eger	Later	Cimaa	Hard
Edans	Tonight	Dafee	Quickly
Amma isa ammaa	Right now	Suuta	Slowly
Ganama kana	This morning	Qalbiidhan	Carefully
Turban dhufu	Next week	Matuma	Absolutely
Dhiyeenya kana	Recently	Walii wajjin	Together
Dhiyootti	Soon	Qophaa	Alone
Hatattamaan	immediately		
Adverbs of place		Adverbs of frequency	
Afaan Oromo	English	Afaan Oromo	English
As	Here	Yeroo hunda	Always
Achi	There	Gaaffii gaaf	Sometimes
Gara sana	Over there	Gaaffii gaaf	Occasionally
Iddoo hunda	Everywhere	Darbee darbee	Seldom
Eessayyu	Nowhere	Darbee darbee	Rarely
Mana	Home	Yoomiyyuu	Never
Fagoo	Away		
Ala	Out		

Table 2.10 adverbs of Afaan Oromo with its English equivalent

Prepositions

Prepositions in Afaan Oromo associated with nouns, pronouns and phrases to other words in a sentence. The word or phrase that the preposition introduces is called the object of the preposition. Here are list of some prepositions [24]. The following table 2.11 presents sample preposition in Afaan Oromo with English equivalent.

Afaan Oromo preposition	English preposition
gama	across
waa'ee	about
gubbaa / gararraa	above
booddee / booda	after
faallaa	against
jara giddu	among
naannoo	around
akka	as
itti	at
dura	before
dudduuba / dugda duuba	behind
jala / gajjallaa	below

Table 2.11 Sample prepositions in Afaan Oromo with English equivalent

Negation in Afaan Oromo

Negation in Afaan Oromo seems similar as English ‘*hin*’ with ‘don’t’ it is a way of changing positive statements in the opposite one such as ‘I love study’ in to ‘I don’t love study’. In most cases negation is formed by adding prefix ‘*hin*’ to the verb like ‘don’t’ in English [24]

2.6. Challenges in Afaan Oromo Text Retrieval Systems

Researchers who had engaged in developing Afaan Oromo IR system indicate many of challenges they faced. As the language is young and under progress of study, obtaining the required resources such as, an effective stemmer, word net, and the complete stop-word list are major challenges [19].

The polysemy and synonymy problems are another challenges observed due to the nature of the language. Another challenges indicated by the researchers are lack of different materials available which prepared in Afaan Oromo such as documents and literatures that can helps to prepare large amount of corpus [24].

2.7. Related works

There are attempts of developing IR system for local language attempts that includes mono-lingual and Cross Lingual Information Retrieval (CLIR). The researcher goes through various local IR system developed by different researchers presented as follows.

2.8. IR System for Local Language

The Afaan Oromo text retrieval system is done by Gezahegn [24]. In this study the researcher used Vector Space Model to represent the documents. For this study 100 corpus is prepared from different sources and the experiment is done using 9(nine) queries.

The experiment shows that the performance is on the average 57.5% precision and 64% recall. The researcher also faced some challenges in handling the problems caused by synonymy and polysemy. Weakness of the stemmer algorithm observed to all word variants, and ambiguity of words in the language is too due to the nature of the language.

According to the researcher the performance of the system can be increased if stemming algorithm is re-designed and the researcher suggested that the query expansion technique.

Most of local language IR systems are designed for Amharic language. Amharic text retrieval is developed by Tewodros [30] using latent semantic indexing (LSI) strategy with singular value decomposition. His work mainly focuses on indexing process, to solve the problem found in exact term match retrieval system.

N-gram-based automatic indexing for Amharic text retrieval is developed by Betelihem [31]. This research mainly conducted to solve the problem of not having standard stemming procedure and stop word list for Amharic language. Thus, to solve such problem a method called n-gram automatic indexing has been developed.

Amharic search engine is designed and developed by Tessema and Solomon [32], which retrieve web documents written in Amharic language. Their systems simply match patterns without considering the feature of Amharic language that affect the retrieval performance.

Another Amharic IR system is developed by Alemayehu and Abiy [33] by integrating query expansion techniques to enhance the performance of the system. The researchers are conducted to solve the problem of polysemous and synonymous exists in Amharic text.

In conclusion, most of works are attempted to design Amharic IR system using various approaches. As Amharic language uses its own letter and has different grammatical rules unlike Afaan Oromo, the researcher focused on the Probabilistic model which used in Amharic language because Afaan Oromo uses Latin letter (Qubee) and has its own grammatical rules.

2.8.1. Probabilistic IR System for Amharic Language

Probabilistic based Amharic IR system is done by Amanuel [34]. In this study the researcher developed full probabilistic model to retrieve relevant document among the corpus prepared by the researcher which is about 300 documents each comprises different issues and queries about 10 are presented to test the designed system. Relevance feedback technique is used which helps the user to provide relevance judge.

Queries are categorized under easy level with very few synonymous terms in the document collection and under difficult level with several synonymous terms in the corpus. Using those queries the system is tested and resulted the average precision 42% and recall 88% respectively for easy level queries. The performance of the system for difficult level queries are 30% precision and 67% recall.

After users provide relevance feedback, the average percentage of precision is increased by 52% and recall is decreased by 26% for easy level queries and precision increase by 26% and recall is decreased by 25% for difficult level queries. Thus, the performance of the system increased and scored 10% and 8% improvement of F-measure for easy and difficult level queries respectively.

As it is observed from evaluation results of the system with difficult queries, the system performance decreased due to unable of handling the synonymies and polysemy so that the researcher left the future work for other researchers.

2.8.2. CLIR IR System for Local Languages

Afaan Oromo-English CLIR A Corpus Based Approach

In this study, the researcher Daniel [21] developed a corpus-based CLIR system which uses word-based query translation method for Afaan Oromo-English languages. This approach requires the availability of parallel documents hence such documents are collected and organized from different sources such as Bible chapters, legal and some other available religious documents.

Evaluation of the system is conducted both in monolingual and bilingual retrievals. In the monolingual run, the Afaan Oromo queries are given to the system and Afaan Oromo documents are retrieved and in the bilingual run of the Afaan Oromo queries are given to the system after being translated into English to retrieve English documents. For the bilingual run

translation of Afaan Oromo queries into their English equivalent is done by using bilingual dictionary constructed from the collected parallel corpora.

Recall and precision measurements result shows with the maximum average precision value of 0.421 and 0.304 are obtained for both Afaan Oromo and English documents respectively. In the second phase of experimentation, there was a result obtained than the first one in terms of some progress. Maximum average precision value of 0.468 and 0.316 are obtained for the Afaan Oromo and English documents respectively. In general, with the use of large and cleaned parallel Afaan Oromo-English document collections, it is possible to develop CLIR for the language pairs.

Afaan Oromo-Amharic CLIR: A Dictionary-Based Query Translation Approach

Afaan Oromo-Amharic CLIR work is done by Walelegn [25]. This study is intended to develop an IR system which uses Afaan Oromo query to retrieve relevant documents written in Amharic language. In order to cross the language barrier between the queries and document the researcher used bilingual dictionary query translation method which constructed from published dictionary.

The researcher prepared corpus from different internet sources, for the testing purpose 15 Amharic queries were prepared, and system performance evaluation was conducted on both Amharic monolingual and Afaan Oromo-Amharic CLIR retrievals. Amharic monolingual run was carried out for the comparison purpose. The researcher found encouraging results which implies that the first experiment retrieval performance gives a maximum average precision of 0.450 (79.3% of Amharic monolingual run). In the second experiment the researcher found out that significantly improved retrieval performance by a maximum average precision of 0.498 (87.8% of Amharic monolingual run). The main challenging tasks in the study was untranslatable search terms due to the lack of consideration of the stemmer algorithm to handle all word variants, limited dictionary coverage, and failure to identify phrasal words.

Corpus-Based Afaan Oromo-English CLIR is another research work done by [22]. The researchers aimed at developing Afaan Oromo-English CLIR system which enables Afaan Oromo native speakers to access and retrieve the vast online information sources that are available in English by writing queries using their own (native) language.

CHAPTER THREE

DESIGN AND DEVELOPMENT

In this study an attempt is done to develop a probabilistic Afaan Oromo IR system. Towards implementing it, the architectural view of the system is constructed based on the algorithm made for each of different tasks that carried out in IR system.

3.1. System Architecture

An IR system has two subsystems; indexing and searching. Indexing is an offline process of creating index file (inverted file for this study) from collected corpus. Searching is an online process of matching the user query with terms in inverted file[2]. The system architecture designed for this study is presented as follows in figure 3.1.

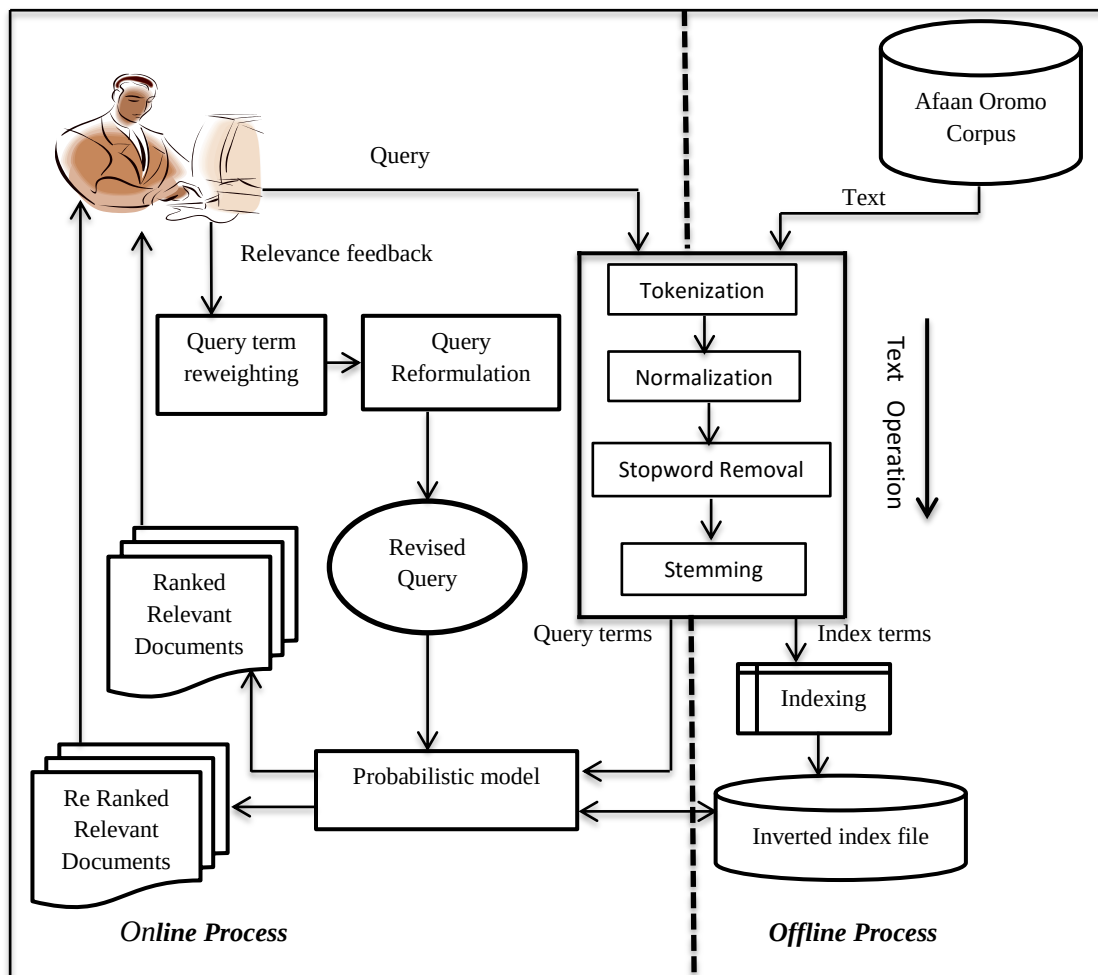


Figure 3.1. A System Architecture for Probabilistic Afaan Oromo IR System

Before indexing the documents text preprocessing such as tokenization, normalization, stop-word removal, and stemming are done to identify index terms. All these text preprocessing tasks are done using an algorithms developed.

3.1.1. Text Preprocessing Algorithms

Tokenization: Tokenization is used to identify words from the text documents. Words are mostly separated by space. Since special characters (such as punctuation marks are not part of words), they are removed during tokenization. Punctuation marks in Afaan Oromo are almost similar to that of English except apostrophe (‘) which is considered as part of the word in Afaan Oromo. E.g. “*Sa’a*” to mean “*cow*”. The rest of punctuation marks are: ? : " ! | / ? @ # * ~ \$ % ^ & () { } < > [] _ + = - , . " ... \ ; - _ + £ are removed by an algorithm designed[24]. Algorithm 3.1 presents tokenization algorithm for Afaan Oromo texts [24].

```
For file in corpus
    Define word delimiter to space
    Read files
    For file in read
        If there is word delimiter
            Put each terms as separate token
```

Algorithm 3.1 Tokenizing Afaan Oromo text

Normalization: The case of letters is usually not important for the identification of index terms. As a result, the normalization step converts all the text in to lower case. E.g. “ABBAA” to “abbaa” or in English “FATHER” to “father” [7].

Stopwords Elimination: words frequently appear in the documents are not important to represent the document[1], these frequent words in documents include *preposition*(e.g. Gara/towards, eega /since); *conjunctions*(fi/and, yoo/if). They are considered as stopwords for removal from the list of content bearing terms for representing documents and queries [21]. List of stopwrds are prepared separately using *stopw.txt* [24] file which is accessed by the python code to find the match in the query and documents to be removed. List of stopwords are presented in Appendix I. The algorithm detecting stopwords from Afaan Oromo text is presented in the following Algorithm 3.2.

```

Open and Read stopwordslist File
while end of stopwordslist
    If term matches a stopword
        Delete term
    Else
        Return nonstopwordlist

```

Algorithm 3.2 Afaan Oromo stopwords remover

Stemming: This is the process of removing inflectional and derivational affixes. After completed stemming process the word is set in to root term as it is discussed in section 2.2.1 in detail.

3.2. Indexing Afaan Oromo Documents

After text operation is completed on document in the corpus prepared, building inverted index is done. The reason why the researcher prefer to use inverted file among different types of indexing method is that the number of pointers is smaller than positions and also all the occurrences of a word inside a single block are collapsed to one reference, and also its simplicity of use[10].

3.2.1. Inverted Index File

An inverted index is an index into a set of texts of the words in the texts. In this study the index is accessed by probabilistic based search method. Inverted index file is divided in to vocabulary file and posting file. Vocabulary file contains each index entry with their document frequency and collection frequency. Posting file provides document id and frequency of occurrence of terms in the documents with the locations with the terms occur within the text document[2]. The logical view for inverted index file is presented in the following figure 3.2.

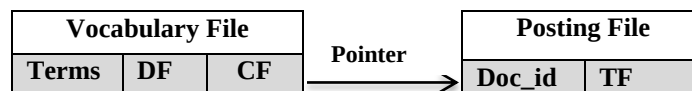


Figure 3.2 vocabulary file and posting list 1

3.3. Probabilistic Model for Searching

In probabilistic based IR process, there are different tasks which are carried out sequentially. Basically, the probabilistic document retrieval process can be divided into two main processes: the initial and after relevance feedback. The following figure 3.3 shows how different tasks are carried out sequentially in probabilistic based IR process at initial and after relevance feedback.

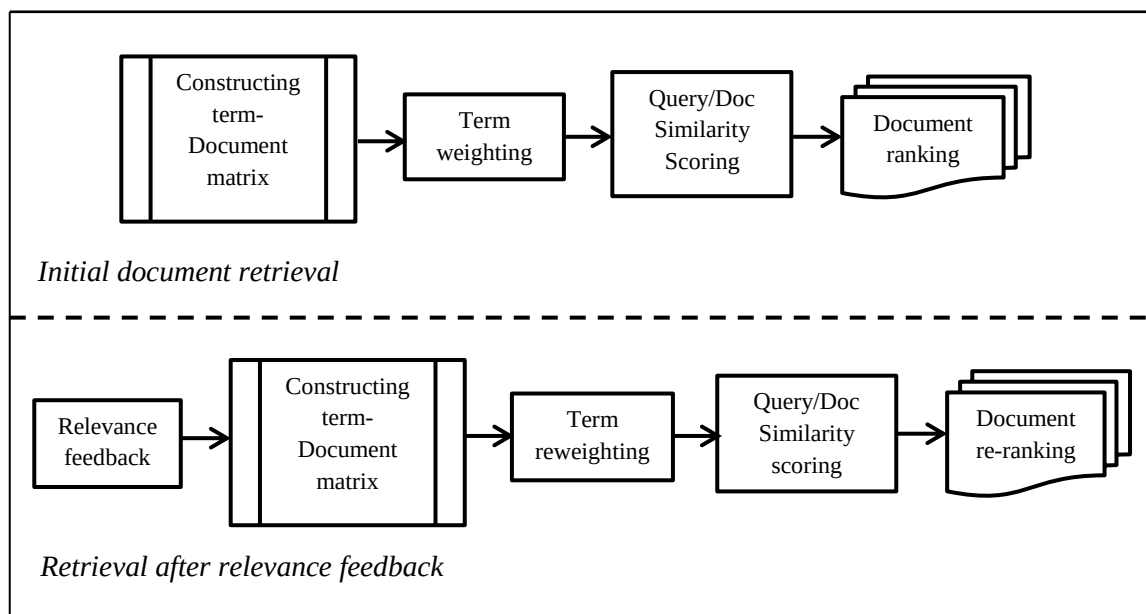


Figure 3.3 A Probabilistic based document retrieval process

As it is illustrated in the above figure 3.3 at initial document retrieval process, constructing term-document matrix is the first task which used to weight each of the query terms against each of the documents; then query-document similarity scoring is done in which document ranking is done in descending order based on the weight of documents.

After the first document retrieval process completed and presented to the user, the user provide the relevance feedback to the system and then the system reformulate the query by calculating term reweighting which also used to measure the document-query similarity score. Finally, documents are re-ranked and presented back to the user in descending order. This process continues until the user satisfies his/her information need.

In this study Binary Independence Model (BIM) is used to design a Probabilistic Afaan Oromo IR System. This model is constructed based on the presence or absence of independently distributed terms in relevant and non-relevant documents. Moreover, it can work with

Probabilistic Ranking Principles (PRP) which introduce some simple assumptions that makes estimation using the probability function $p(R/d, q)$ that indicates the fundamentals of probability concept for uncertainty theories[22].

In BIM there are three significant steps of computing term probabilities[2]. The first step is computing the terms even though there is no document retrieved at initial attempt. The second step is computing the terms after documents are retrieved while the user's relevance feedback is provided. The third step is computing the terms after feedback is provided [2].

At the beginning, an attempt of retrieving relevant document only indexed terms are known because they are already prepared in offline process but still the document to be retrieved are unknown rather than guessing them. There are two assumptions at initial guess:

- i. Assumption 1: $p(k_i/R) = 0.5$, the value is constant for all index terms of k_i .
- ii. Assumption 2: The distribution of index terms among non-relevant documents can be approximated by the distribution of index terms among all documents in the collection using the following equation 3.1.

$$p(t|\bar{R}) = \log \frac{N-n+0.5}{n+0.5} \quad (3.1)$$

Where N is the number of documents in the collection and n is the number of documents which contain the term.

After term weighting process completed at initial attempt, term-document similarity scoring is applied using the following equation 3.2, and finally document ranking is done in descending order based on the documents weight.

$$sim(d_j, q) = d_j \cdot q = \sum_{i=1}^n w_i \cdot q_i \quad (3.2)$$

After initial guess is done and retrieve documents, the result presented to the user and then, the user assess and label the documents as relevant and irrelevant. In this study the user relevance feedback is used which is discussed in (section 2.4.1). Once the system receives the feedback from the user, term reweighting technique is applied in which query is reformulated using the concept of term incidence contingency[1][2]. The following table 3.1 presents contingency table of relevance judgment.

	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 Contingency table of Relevance Judgments

Here N is the total number of documents in the collection, n is the total number of documents that contain term k_i , R is the total number of relevant documents retrieved, and r is the total number of relevant documents retrieved that contain term k_i . [16].

After identifying relevant documents and non-relevant documents for a given query, the next step is estimating the probability of finding term (t_i) in relevance document and the probability of finding term (t_i) in non-relevant document [1] using equation 2.16 and equation 2.17 respectively:

$$w = \log \frac{r(N-n-R+r)}{(R-r)(n-r)} \quad (3.3)$$

This formula for w obviously breaks down if ($r = R$) or zero ($r = 0$). Similarly, w breaks down if ($n-r = N-R$) or zero ($n = r$). Statistical theory has been used to justify modifying the formulas for both cases to avoid these singularities by adding a constant c to the numerator and one to the denominator, where $c = 0.5$ or $c = n/N$ [2]. For this study equation 3.4 is used.

$$Relevance\ Weighting\ (Rw) = \log \frac{(r+0.5)(N-n-R+r+0.5)}{(R-r+0.5)(n-r+0.5)} \quad (3.4)$$

As the term reweighting process completed, term similarity document scoring is done using the same equation 3.2 which already used at initial document retrieval process. Finally, document re-ranking is done in descending order based on the weight of the documents.

3.4. Evaluating IR System Performance

After completing IR system prototype testing, evaluation for system performance is the next task to be carried out. Thus, IR systems require the evaluation of how precise is the answer set

[10]. There are two principal aspects to measuring IR system performance: *efficiency* and *effectiveness*. Efficiency may be measured in terms of *time* and *space*, where time is for *response time or latency*, space is for the storage space measured by byte of disk and memory required to store the index and other data structures[3].

Effectiveness is more difficult to measure than efficiency, since it depends on human judgment. The key idea behind measuring effectiveness is the notion of *relevance*. The fundamental goal of relevance ranking is expressed in terms of *Probabilistic Ranking Principle* (PRP) [3].

If an IR system's response to each query is a ranking of the documents in the collection in order of decreasing probability of relevance, then the overall effectiveness of the system to its users will be maximized [3].

At the heart of IR system evaluation there is the concept which is “relevance”. Relevance is an inherently subjective concept in the sense that satisfaction of human needs is the ultimate goal, and hence the judgment of human users as to how well retrieved documents satisfy their needs is the ultimate criterion of relevance[23].

The reason why carrying out IR system evaluation is that every system needs to pass such criteria's for assurance of its *validation* and *verification*. The process of evaluating IR system answers number of questions like whether the system is right or not and is it the right system or not based on the objectives of intended IR system. IR System evaluation also helps to compare with another IR system by identifying its strength and weakness which makes it important for the improvement of the retrieval performance of the system[2].

In order to measure IR system performance several techniques are use; Such as: *precision*, *recall*, *F-measure*, *E-measure*, *MAP* (*Mean average precision*), and *R-measure*, etc. In this study the three widely used techniques *precision*, *recall* and *F-measure* are used[16].

Precision and *recall* are the most frequently used and still they are the dominant approach to evaluate the performance of IR systems. These are first defined for the simple case where an IR system returns a set of documents for a query[1].

Precision (*P*) is the fraction of retrieved documents that are relevant.

$$Precision = \frac{\#(relevant\ documents\ retrieved)}{\#(retrieved\ documents)} = p(relevant|retrieved) \quad (3.5)$$

Recall (R) is the fraction of relevant documents that are retrieved.

$$Recall = \frac{\#(relevant\ documents\ retrieved)}{\#(relevant\ documents)} = R(retrieved|relevant) \quad (3.6)$$

Among the documents retrieved, the answer set are first sorted according to a degree of relevance. Then the user examines this ranked list starting from the top document. In this situation, the recall and precision measures are varying as the user proceeds with the user's examination of the answer set. Thus, proper evaluation plots a *precision* versus *recall* by interpolating using 11-standard recall levels.

The advantage of having the two numbers for *precision* and *recall* is that one is more important than the other in many circumstances. Nevertheless, the two quantities clearly tradeoff against one another: which can always get a *recall* of 1 (but very low precision) by retrieving all documents for all queries. *Recall* is a non-decreasing function of the number of documents retrieved. On the other hand, in an IR system, *precision* usually decreases as the number of documents retrieved is increased[1].

A single measure that trades off *precision* versus recall is the *F measure*, which is the weighted harmonic mean of *precision* and *recall* [1]:

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad \text{where } \beta^2 = \frac{1-\alpha}{\alpha} \quad (3.7)$$

Where $\alpha \in [0, 1]$ and thus $\beta^2 \in [0, \infty]$. The default *balanced F measure* equally weights *precision* and *recall*, which means making $\alpha = 1/2$ or $\beta = 1$. It is commonly written as *F1*, which is short for $F_{\beta=1}$, even though the formulation in terms of α more transparently exhibits the *F measure* as a weighted harmonic mean. When using $\beta = 1$, the formula on the right simplifies to[1]:

$$F_{\beta=1} = \frac{2PR}{P+R} \quad (3.8)$$

Recall, *precision*, and the *F measure* are inherently measures between 0 and 1, but they are also very commonly written as percentages, on a scale between 0 and 100[4].

The reason why used a harmonic mean rather than the simpler average (arithmetic mean) is that it can always get 100% recall by just returning all documents, and therefore it can always get a 50% arithmetic mean by the same process. This strongly suggests that the arithmetic mean is an unsuitable measure to use[4].

For more than one query, each query a distinct *precision* versus *recall* curve is generated. To evaluate the retrieval performance of an algorithm over all test queries, the average of the precision figures at each recall level is calculated as follows[2, 4].

$$\bar{p}(r) = \sum_{i=1}^{N_q} \frac{p_i(r)}{N_q} \quad (3.9)$$

Where $\bar{p}(r)$ is the average precision at the recall level r , N_q is the number of queries used, and $P_i(r)$ is the precision at recall level r for the i -th query[4].

CHAPTER FOUR

EXPERIMENTATION AND ANALYSIS

4.1. Data for Experimentation

Due to scarcity of resources available in Afaan Oromo, the researcher is limited in collecting small amount of documents. However, in order to test the designed probabilistic Afaan Oromo IR system and analyze the results found, Afaan Oromo corpus and queries with various topics are prepared.

4.1.1. Document Preparation

The prepared corpus contains about 116 documents, each of the documents are covering different issues, predominantly public news, educational, cultural, historical, business, entertainments, sport, and politics. All these collected documents are saved under common file extension file .txt (Notepad) which is suitable to write and read from using python code. All collected documents with their content categories and their numbers are presented in the following table 4.1.

N_o	Document topics	N_o of documents
1	Educational	20
2	Cultural	19
3	Historical	11
4	Religious	13
5	Entertainments	14
6	Politics	24
7	Health	15
Total		116

Table 4.1 Number of documents with their categories

4.1.2. Query Preparation

For the purpose of evaluating the designed IR system 10 queries are prepared. All these queries are covering different issues and various topics considering the documents in the corpus. Using the prepared queries is very important to identify which documents are relevant for selected query and which are not relevant and answers the question that how much the system retrieves

the relevant documents as it is intended for. List of the queries used for testing this IR system are presented in the following table 4.2.

No	Query	Topic
1	Waaqeffannaan, Madda Qulqulloominaa	Religion
2	Ayyaana kanaan urjiilee jirraan	Culture
3	Itiyoophiyaa, fi Sudaan Kibbaa	Politics
4	Bulchiinsa gaarii mootummaa Itiyoophiyaa	Politics
5	Godaansa Ummata Oromoo fi Misooma	History
6	manneen amantii Oromoo	Religion
7	Carraan Biyyoota Hiyyeeyyii	Politics
8	Waldaan Misooma Oromiyaa	Social
9	baajata Manneen mootummaa Itiyoophiyaa	Politics
10	Lakkoofssii Waldootii amantii oromo	Religion

Table 4.2 List of Queries with their categories

4.2. Overview IR System Prototype

The probabilistic based IR system in this research has two modules: indexing and searching. In both of these modules, text preprocessing is applied for identifying content-bearing terms in the document collection and query terms. Text processing involves different steps, such as tokenization, normalization, stopword removal, and stemming.

Tokenization

Tokenization is the task of chopping a sequence of characters into pieces called *tokens*; at the same time throwing away certain characters such as punctuation (see section 2.2.1).

Afaan Oromo language mainly uses space as a word. There are also special characters and punctuation mark that are frequently used in writing. During tokenization both space and punctuation marks are used to identify a word. Code 4.1 presented fragmented python code for tokenization.

```

token=text.split()
p=codecs.open("Punctuation.txt",'r')
Punctuation = p.read()
for pc in Punctuation:
    if str.endswith(pc):
        str=str.rstrip(pc)
    if str.startswith(pc):
        str=str.lstrip(pc)
for j in Punctuation:
    for i in range(1,len(str)):
        if j in Punctuation and j in str:
            str=str.strip(j)
return str

```

Code 4.1 fragmented python code for tokenization and punctuation remove

Normalization

Once a list of words are identified, they are further preprocessed to normalize any superficial difference among words in which character cases are set in to uniform cases usually lower case. Different word to word combinations using hyphen are also separated and changed in to same format[1]. Fragmented text normalization python code is presented in the following code 4.2.

```

text=open(".\corpus").read()
txt=text.lower().split()

```

Code 4.2 fragmented python code for converting text to lower case

Stop-word Removal

In order to detect and remove the stop-words, a stop-word list is prepared. The stop list contains common words that are frequently occurring in Afaan oromo documents. Finally, given list of words, they are checked for stop-word using the python code given below in Code 4.3.

```

stop_words=codecs.open("stopw.txt",'r')
n=stop_words.read()
s=n.split()
stop=0
for i in range(0,len(s)):
    if (index) in s[i]:
        stop=1
        if stop==0:
            indexterms2.append (index)

```

Code 4.3 fragmented python code for stop-word removal

Stemming

In this study, developing stemming algorithm was very challenging due to the nature of Afaan Oromo since some of root terms in this language are end with vowel letters which are defined as suffixes in another terms. Over stemming and under stemming are the major problems which make the root terms to lose their meaning. In this study the Afaan Oromo stemmer developed by Debela [27] is adopted, as presented in the following python Code 4.4.

```

Def thestemmer(token):
    stem=[]
    stm=[]
    if token.endswith('ittii'):
        if count_m(stem)>=1:
            stem=token.replace('ittii', ' ')
            return stem
        else:
            return token
    elif token.endswith('ttii'):
        if count_m(stem)>=1:
            stem=token.replace('ttii', ' ')
            return stem
        else:
            return token
    elif token.endswith('tii'):

```

Code 4.4 fragmented python code for stemming Afaan Oromo terms

Indexing Afaan Oromo Documents

Inverted index file is constructed for Afaan Oromo document collection. Text preprocessing which include tokenization, normalization, stopword removal, and stemming are done for inverted index.

Inverted index file comprises two files, Vocabulary file and posting list. The vocabulary file contains term, document frequency, and collection frequency. The posting list file contains document id and term frequency. The two separated files are connected via the pointer which used to refer the documents that contain particular terms in the query. The following figure 4.1 shows sample inverted index file.

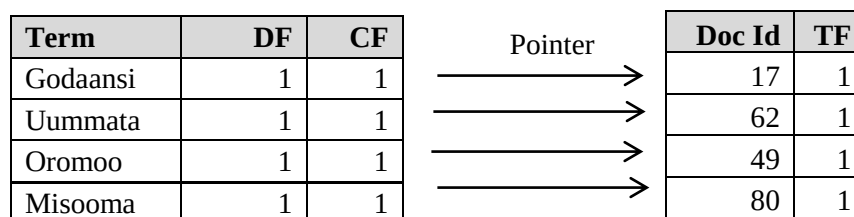


Figure 4.1 Sample for inverted index file (Vocabulary and Posting list)

Probabilistic Information Retrieval

Probabilistic based IR system starts with preparing term-document matrix that enables the system to weight the query terms in the documents based on their presence or absence in the documents using the BIM concept. There are 29 documents retrieved among 116 documents in collection for the query number 5 in the query list. However, the following sample term-document matrix is prepared only for the top 5 ranked documents retrieved for the given query. The following table 4.3 presents a sample to show that how term-document matrix is prepared by the designed IR system at initial retrieval.

	Godaansi	Uummata	Oromoo	Misooma
Doc017	0	1	1	0
Doc062	1	1	1	0
Doc049	0	1	0	1
Doc080	0	0	1	1
Doc084	0	0	0	1
W_{ij}	1.88	1.51	1.51	1.51

Table 4.3 sample for term-document matrix at initial retrieval

After term-document matrix is prepared, the next task is calculating weight of the terms in each of the documents presented. Accordingly, using equation 3.1, term weighting is calculated and assigned the result for each of the terms in W_{ij} as it is appeared in the above table 4.3.

After completing term weighting calculation, document-query similarity scoring is calculated using equation 3.2; afterwards, document ranking is done in decreasing order as it is shown in following table 4.4.

Doc rank	Doc weight
Doc062	4.90
Doc049	3.02
Doc017	3.02
Doc080	3.02
Doc084	1.51

Table 4.4 Sample for retrieved document in descending order

As the initial document retrieval process completed and displayed to the user, the user start examining the result obtained and labeled them that some of the documents are relevant. Using equation 3.4, term reweighting is calculated and the result is presented as follows in table 4.5.

	Godaansi	Uummata	Oromoo	Misooma	Relevance
Doc062	1	1	1	0	R
Doc084	0	0	0	1	NR
Doc017	0	1	1	0	NR
Doc080	0	0	1	1	NR
Doc049	0	1	0	1	NR
W_{ij}	0.96	0.25	0.25	0.25	

Table 4.5 term reweighting result after relevance feedback

As the term reweighting is completed, the query-document similarity scoring is calculated using equation 3.2 and then documents are re-ranked. These activities are repeatedly performed for two iterative processes for each of the ten queries prepared by the three different users.

4.3. Testing IR System Prototype

In order to identify the relevant documents for each of the query sets, document versus query matrix that shows relevance judgment is prepared manually which is presented in Appendix II. As the system is use probabilistic model, it allows examining the initial result and then provides relevance feedback to re-rank relevant documents which are labeled by the user as relevant documents.

For the initial retrieval, the three different users presented and issued the queries to the system. User A used the first three queries (Q1, Q2, Q3), user B used the next three queries (Q4, Q5, Q6) and user C used the last four queries (Q7, Q8, Q9, Q10) at the threshold ≥ 0.0 . Accordingly, the retrieval performance evaluation result for precision, recall and F-measure are presented in the following table 4.6.

No	Query	Relevant	Retrieved	Relevant Retrieved	Precision	Recall	F-Measure
1	Waaqeffannaan Madda Qulqulloominaa	8	5	4	0.80	0.50	0.62
2	Ayyaana kanaan urjiilee jirraan	17	8	6	0.75	0.35	0.48
3	Itiyoophiyaa, fi Sudaan Kibbaa	4	2	2	1.00	0.50	0.67
4	Bulchiinsa gaarii mootummaa Itiyoophiyaa	4	2	2	1.00	0.50	0.67
5	Godaansa Ummata Oromoo fi Misooma	25	5	4	0.80	0.16	0.27
6	manneen amantii Oromoo	34	4	3	0.75	0.09	0.16
7	Carraan Biyyoota Hiyyeeyyii	4	3	2	0.67	0.50	0.57
8	Waldaan Misooma Oromiyaa	4	2	2	1.00	0.50	0.67
9	baajata Manneen mootummaa Itiyoophiyaa	4	2	2	1.00	0.50	0.67
10	Lakkoofssii Waldootii amantii oromo	13	7	6	0.86	0.46	0.55
Average					0.86	0.41	0.53

Table 4.6 performance evaluation result at initial retrieval

As the performance evaluation result shows at initial retrieval in the above table 4.6 using Mean Average Precision (MAP), the precision average result is 86%, recall 41% and F-measure average 53%. The average precision result indicates that the relevant documents out of retrieved documents are about 86%. The average recall result indicates 41% of relevant documents are retrieved out of the relevant documents judged manually. F-measure result is 53% this indicates that the accuracy of the system retrieval.

The following graph in figure 4.2 illustrates precision vs. recall result using 11-standard interpolated recall level.

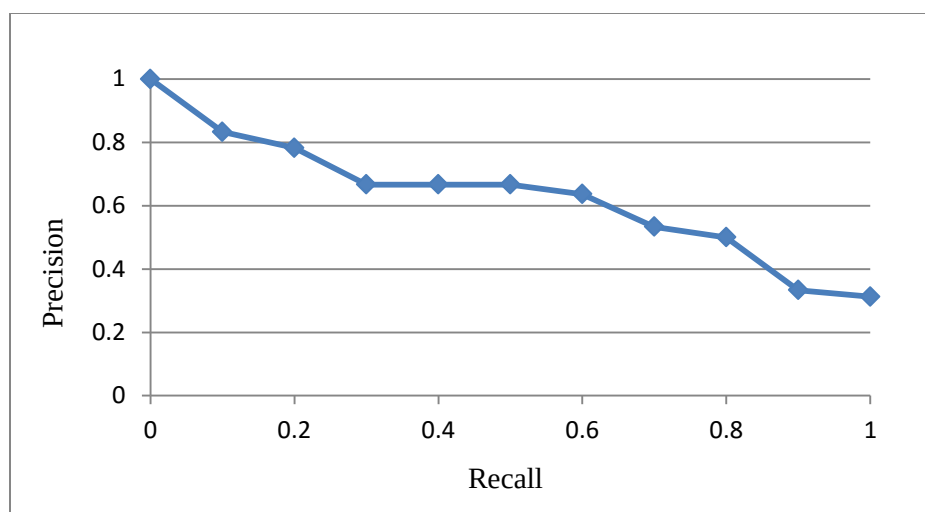


Figure 4.2 Precision-Recall graph at initial retrieval

The retrieval performance of the system is improved in the second document retrieval process after the system consider the user's relevance feedback. (In this study, two times iterative process are done). In query reformulation, terms are re-weighted, documents are re-ranked by the system. The following table 4.7 presents *precision*, *recall* and *F-measure* results after relevance feedback is provided.

No	Query	Relevant	Retrieved	Relevant Retrieved	Precision	Recall	F-Measure
1	Waaqeffannaan Madda Qulqulloominaa	8	6	5	0.83	0.63	0.71
2	Ayyaana kanaan urjiilee jirraan	17	15	8	0.53	0.47	0.50
3	Itiyoophiyaa, fi Sudaan Kibbaa	4	4	3	0.75	0.75	0.75
4	Bulchiinsa gaarii mootummaa Itiyoophiyaa	4	4	3	0.75	0.75	0.75
5	Godaansa Ummata Oromoo fi Misooma	25	23	18	0.78	0.72	0.75
6	manneen amantii Oromoo	34	32	20	0.63	0.59	0.61
7	Carraan Biyyoota Hiyyeeyyii	4	4	3	0.75	0.75	0.75
8	Waldaan Misooma Oromiyaa	4	3	3	1.00	0.75	0.86
9	baajata Manneen mootummaa Itiyoophiyaa	4	3	2	0.67	0.50	0.57
10	Lakkoofssii Waldootii amantii oromo	13	11	7	0.64	0.54	0.58
Average					0.73	0.64	0.68

Table 4.7 performance evaluation result after relevance feedback

After relevance feedback is provided to the system, the result presented in the above table 4.7 shows an average precision result scored 73%. The average result for recall is 64% which increased by 23% as it is compared to the initial retrieval average result shown previously.

F-measure average result increased by 15% after relevance feedback. This result is obtained due to the increment of recall. The following graph in figure 4.3 illustrates the interpolated recall curve using 11-standard interpolation after relevance feedback.

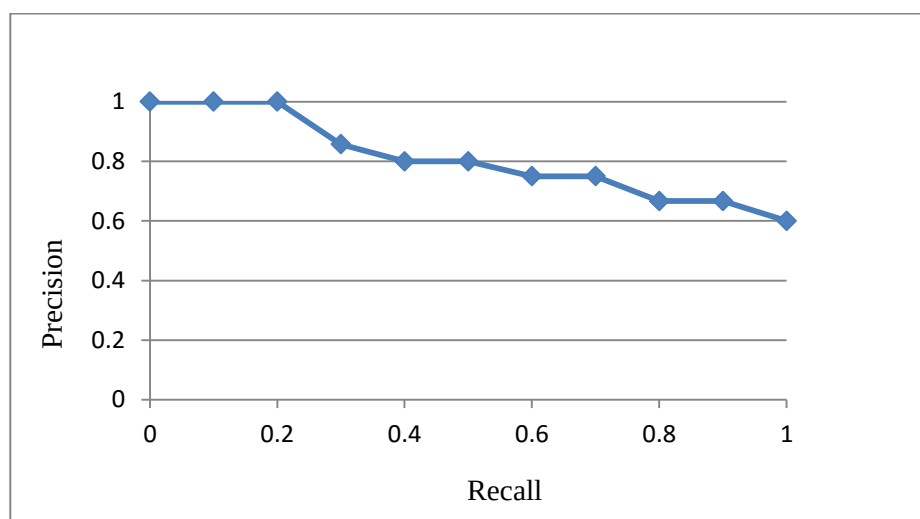


Figure 4.3 Precision-Recall graph after relevance feedback

4.4. Discussion

The result obtained from the designed probabilistic Afaan Oromo IR system can be said satisfactory and encouraged in terms of its higher recall result scored.

As the user provides relevance feedback, the system continues retrieving documents until the user satisfy his/her information need. In this process, query terms are reweighted that increase the ranks of un-retrieved documents which contain the reweighted terms, the system gives more weight for the terms found in the documents which are labeled as relevant by the user but still there is decrement in precision by 13% this is due to the increment of relevant retrieved documents.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1. Conclusion

The concept of information retrieval (IR) involves in tracing and recovering specific information from the data storage like internet or other information suppliers. Disseminating information in different form of multimedia in various languages becomes an important issue so that considering the demands and using information retrieval technologies becomes very crucial.

Afaan Oromo is the most widely spoken language in Ethiopia; in addition to that, the language is serving as medium of instruction in primary schools and working language for the regional state of Oromia. The language uses Latin letters (Qubee) and also the rules to use punctuations are almost similar to English language except apostrophe since it has special meaning in the language. As the language is widely spoken in overall Oromia, it is morphologically rich and highly inflected, polysemy and synonymy semantic behaviors are the major characteristics of the language which is already discussed in detail in section 2.5.

Considering the need for having IR system which accepts the query prepared in Afaan Oromo and retrieve documents in the same language, a Probabilistic Afaan Oromo IR system is developed using Binary Independence Model (BIM).

BIM is adopted as it is working by assuming the relevant document based on the query term occurrence in the document in which the model judges the relevance of documents if the term is present in the document otherwise irrelevant if the term is absent in the document rather than counting frequency of terms in each of the documents. In BIM absence of counting term frequency is considered as its main drawback comparing to VSM; however, probabilistic model has its own features like term reweighting technique as the user provides relevance feedback. The model reformulates the user query that enables the system to give more weight for relevant documents labeled by the user.

Term reweighting technique is applied in this study by which the user can review all retrieved documents and provide relevance feedback until the users satisfy their information needs. This is done for two iterative retrieval processes using 10 queries and 116 documents prepared.

Three different users participated for issuing the prepared queries for the purpose of testing the designed IR system. On average three queries are used by each of the three different users. Results for each attempt are evaluated using the most widely used evaluation techniques: precision, recall and F-measure. The average result obtained at initial retrieval is 86% precision, 41% recall and 53% F-measure. After relevance feedback 73% precision, 64% recall and 68% F-measure.

As the evaluation result shows, there is a better score in recall which is exceeded by 23% after relevance feedback. When the recall value gets higher score than previous one, it can be said that the system can retrieve the more relevant documents comparing to the initial retrieval. Based on the result found, this study can be said that it achieved its goal in terms of Applying the probabilistic model for Afaan Oromo language. However, there is still the weakness in the designed system which is caused by the nature of the intended language as it is morphologically rich, it is highly inflected. Handling this problem needs more effective stemming algorithm that can reduce the problem occurred due to under stemming and over stemming of the terms.

5.2. Recommendation and Future Work

The designed IR system performance can be optimized by avoiding some of the weaknesses observed in the process of testing the retrieval performance of the model. It is believed that identifying the problems help to continue further investigation in order to come up with better achievements. Therefore, the following points are identified and presented by the researcher.

- Under stemming and over stemming Afaan Oromo terms is a factor since the language uses highly inflected terms which makes a challenge in matching the terms as the result the system forced to ignore relevant document from retrieval.
- Due to time and resource constraints the researcher couldn't build large size of corpus therefore, at initial and after relevance feedback document retrieval process there is only 5.1 average number of documents are retrieved for each of the queries prepared. It is believed that as the size of corpus increase, it helps to identify the optimal performance of the system. Therefore, preparing large size of corpus is very essential.
- This days ontology is the main focusing area for modern information retrieval system; it is believed that the implementation of ontology based IR system for Afaan Oromo language text retrieval system can enhance the performance of the system than the model used in this thesis.
- There is an open area to develop the IR system for afaan Oromo language for the retrieval process of image, audio, video, and other graphics using different multimedia retrieval approaches whereas most IR systems are developed for text retrieval only so far.

Reference

- [1] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval* vol. 1: Cambridge university press Cambridge, 2008.
- [2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval* vol. 463: ACM press New York, 1999.
- [3] S. Büttcher, C. L. Clarke, and G. V. Cormack, *Information retrieval: Implementing and evaluating search engines*: Mit Press, 2010.
- [4] P. E. R. Ingwersen, *Information Retrieval Interaction:(click on title for download access via'Documents')*: Taylor Graham, 1992.
- [5] P. Pingali, K. K. Tune, and V. Varma, "Hindi, telugu, oromo, english CLIR evaluation," in *Evaluation of Multilingual and Multi-modal Information Retrieval*, ed: Springer, 2007, pp. 35-42.
- [6] K. K. Tune, V. Varma, and P. Pingali, "Evaluation of Oromo-English Cross-Language Information Retrieval," *Language Technologies Research Centre IIIT, Hyderabad India*, 2007.
- [7] M. Diriba, "An Automatic Sentence Parser for Oromo Language using Supervised Learning Technique," MSc Thesis at AAU, 2002.
- [8] C. Kothari, *Research methodology: Methods and techniques*: New Age International, 2004, ISBN-10: 8122436234 .
- [9] G. Van Rossum, "Python Programming Language," in *USENIX Annual Technical Conference*, 2007.
- [10] G. Salton and M. J. McGill, "Introduction to modern information retrieval," 1986.
- [11] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 275-281.
- [12] C. Monz and M. de Rijke, "Inverted index construction," *Christof Monz & Maarten de Rijke*, pp. 1-39, 2002.
- [13] D. Hiemstra, "Information retrieval models," *Information Retrieval: searching in the 21st Century*, pp. 2-19, 2009.
- [14] W. Gasarch, "A survey on private information retrieval," in *Bulletin of the EATCS*, 2004.
- [15] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, pp. 513-523, 1988.
- [16] E. Greengrass, "Information retrieval: A survey," 2000.
- [17] K. S. Jones, S. Walker, and S. Robertson, "A probabilistic model of information retrieval: Devepment and SStatus," *Department of Information Science, City University, London*, vol. 74, 1998.
- [18] H. Turtle and W. B. Croft, "Evaluation of an inference network-based retrieval model," *ACM Transactions on Information Systems (TOIS)*, vol. 9, pp. 187-222, 1991.
- [19] A. Nefa, "Long Vowels in Afaan Oromo: A Generative Approach," MA Thesis. School of Graduate Studies, Addis Ababa University, 1988.
- [20] T. Gamta, "Qube Afaan Oromo: Reasons for Choosing the Latin script for Developing an Oromo Alphabet," *Journal of Oromo Studies*, vol. 1, 1993.
- [21] B. Daniel, "Afaan Oromo-English Cross-Lingual Information Retrieval (CLIR): A Corpus Based Approach," 2011.
- [22] D. Hiemstra and S. E. Robertson, "Relevance feedback for best match term weighting algorithms in information retrieval," 2001.

- [23] G. Salton and C. Buckley, "Improving retrieval performance by relevance feedback," *Readings in information retrieval*, vol. 24, pp. 355-363, 1997.
- [24] G. Gezahegn, "Afaan OromoText Retrieval System", Thesis ,AAU, 2012.
- [25] A. Walelign, “ Afaan Oromo – Amharic Cross-Lingual Information Retrieval: a Dictionary- Based Query Translation Approach”, Thesis, University of Gondor,2015.
- [26] Ceri, S; Bozzon, A; Brambilla, M.; Della Valle, E.; Fraternali, P.; Quarteroni, "Web Information Retrieval" S. 2013, xiv, 284 p. 113 illus., 3 illus. ISBN:
- [27] T. Debela "Designing Stemmer for Afaan Oromo Text: A Hybrid Approach", Thesis, AAU, 2010.
- [28] "<http://www.asindexing.org/about-indexing/history-of-information-retrieval/>" "access date 3/28/16, time 02:23pm
- [29] Farach, Martin (1997), "*Optimal Suffix Tree Construction with Large Alphabets*" (PDF), 38th IEEE Symposium on Foundations of Computer Science (FOCS '97), pp. 137–143.
- [30] Tewodros H., "Amharic text retrieval: An experiment using latent semantic indexing(LSI) with singular value decomposition (SVD)" , MSc Thesis, Addis Ababa University, School of Information science, Ethiopia, pp. 1-113, 2003.
- [31] Betelihem M., "N-gram-based automatic indexing for Amharic text", MSc Thesis, Addis Ababa University School of Information science, Ethiopia, pp. 1-101, 2002.
- [32] Tesema M. and Solomon A. , "Design and Implementation of Amharic Search Engine", fifth international conference on signal image technology and internet based systems, vol 3(1) , pp. 318-325, 2009.
- [33] Abiy Z. , Alemayehu W. , Daniel T. , Melese G. and Yilma S. , "Introduction to Research Methods", 1st Edition, Graduate studies and Research Office of Addis Ababa University, 2009.
- [34] Amanuel H. , "Probabilistic Based Amharic Information Retrieval System" MSc Thesis, AAU, 2012

Appendix I: List of Afaan Oromo Stop-words [24]

Aadde	Ammo	Dabalatees	Egaa	hoggaa	Ishiitti	kanaafi
Aanee	Ammo	Dacha	Eger	hogguu	Isii	kanaafuu
Aanoo	Ana	danda'a	Egeris	hoo	Isiin	kanaan
Achi	Ani	Dannuu	Ennaa	ibsanitti	Isin	kanaatti
Achitte	As	Darbee	Erga	ibse	isini	kanarraa
achumaanis	Asi	Dhaan	ergasii	immoo	isini	kanarratti
Adaraa	Asirratti	dhi'ootti	Ergii	ini	isiniif	kanneen
Addaddan	Asitti	Diida	Fi	innaa	isiniin	karaa
Addan	Asuma	Diidaa	Fullee	inni	isinirraa	kee
Addana	Asumatti	Dugda	fuuldura	irra	isinitti	keenna
Afur	Ati	Dur	fuundura	irraa	ittaanee	keenya
Agarsiisoo	Attam	Dura	Gaafa	irraan	itti	keesaa
Akka	Attamitti	Duraan	gaafana	isa	ittifufe	keessa
Akkam	baay'ee	Dureen	gaafas	isaa	ittifufu	keessa
Akkam	baay'inaan	Durii	gaafasuma	isaaf	itumallee	keessaa
Akkan	baay'isee	Durumasaa	gajjallaa	isaan	ituu	keessaan
Akkana	Bada	Duuba	Gama	isaani	ituullee	keessan
Akkas	Bara	Duubaan	Gara	isaanii	jala	keessatti
Akkasitti	Barana	Duukaa	Garaa	isaanirraa	jalaa	kiyya
Akkuma	Bayyee	Duwaadhaaf	gararraa	isaanis	jara	koo
Ala	Bira	Duwwaa	Garas	isaanitti	jechaan	kun
Alaalatti	booda	Eda	Garuu	isaatiin	jechuu	kuni
Alana	booddee	Edana	Giddu	isarraa	jechuun	kunneen
Alas	borii	Eega	gidduu	isatti	jira	kunneenis
Alatti	boroo	Eegana	Gubba	isee	jiran	lafa
Alla	Boru	Eegasii	gubbaa	iseen	jiru	lama
Altokko	borumtaa	Eessa	gulana	ishee	kan	malee
Amma	boruu	Eessaa	hamma	ishii	kana	malee
Ammaa	bukkee	Eessuma	Hanga	ishiif	kanaa	manaa
Ammaan	caalaatti	Naa	Henna	ishiin	kanaaf	maqaa
Ammo	cookkoo	Naaf	Hin	ishiirraa	kanaafi	moo
Si	tawullee	Yookanimmo	si'a	teella	yookiin	taa'eera
Sii	teenya	Yookiis	naannoo	siif	teessan	yookinimoo
Narraa	Siin	Tiyya	Yoom	natti	silaa	tokko
Ni	Silaa	Tokkoo	Nu	simmoo	tokkoorratti	nu'i
Sinitti	tokkotti	Nurraa	Siqee	tun	tuni	nutti
Nuu	Sun	waa'ee	Nuuf	Suni	waa'een	nuun
Waan	Nuy	ta'an	waayee	obbo	ta'an	waggaa
Odoo	taasuu	Wajjiin	Ofii	ta'e	wajjin	oggaa
Tahullee	Wal	Ol	Tan	Warra	otoo	tanaaf
Osoo	Tana	Yammuu	yemmuu	Otumallee	tanaafi	yeroo
Otuu	tanaafuu	Ykn	otuullee	Tarii	yommii	qaba
ta'ullee	yommu	Saaniif	ta'uu	Yommuu	sadii	ta'uutiin
Yoo	sana	ta'uyyu	yookaan			

Appendix II: Document-Query Relevance Matrix

	Q-1	Q-2	Q-3	Q-4	Q-5	Q-6	Q-7	Q-8	Q-9	Q-10
Doc001	NR	R	NR	NR	R	R	NR	NR	NR	NR
Doc002	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Doc003	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Doc004	NR	NR	NR	NR	R	NR	NR	NR	NR	R
Doc005	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc006	NR	R	NR	NR	R	R	NR	NR	NR	NR
Doc007	R	R	NR	R	R	NR	NR	NR	NR	NR
Doc008	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc009	R	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc010	R	R	NR	NR	R	NR	NR	NR	NR	NR
Doc011	NR	R	NR	NR	R	NR	NR	NR	NR	NR
Doc012	NR	R	NR	NR	R	NR	NR	NR	NR	NR
Doc013	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc014	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc015	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc016	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Doc017	NR	R	NR	NR	R	R	NR	NR	NR	NR
Doc018	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc019	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Doc020	R	NR	NR	NR	NR	R	NR	NR	NR	R
Doc021	R	R	NR	NR	R	R	NR	NR	NR	R
Doc022	R	R	NR	NR	R	R	NR	NR	NR	R
Doc023	R	NR	NR	NR	R	NR	NR	NR	NR	R
Doc024	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc025	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc026	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc027	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc028	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc029	NR	NR	NR	NR	R	R	NR	NR	NR	NR
Doc030	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc031	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc032	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Doc033	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc034	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc035	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc036	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Doc037	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc038	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc039	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc040	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Doc041	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Doc042	NR	R	NR	NR	R	R	NR	NR	NR	NR
Doc043	NR	R	NR	NR	NR	R	NR	NR	NR	NR
Doc044	NR	R	NR	NR	R	R	NR	NR	NR	NR

Doc045	NR	NR	NR	NR	NR	R	R	NR	NR	NR
Doc046	NR	NR	R	R	NR	R	NR	NR	NR	NR
Doc047	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Doc048	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc049	NR	NR	NR	NR	NR	NR	R	R	NR	NR
Doc050	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc051	NR	NR	NR	NR	NR	R	R	NR	NR	NR
Doc052	NR	NR	NR	NR	R	R	NR	NR	NR	NR
Doc053	NR	NR	NR	NR	R	R	NR	NR	NR	NR
Doc054	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Doc055	NR	NR	R	R	NR	NR	NR	NR	NR	NR
Doc056	NR	NR	R	R	NR	NR	NR	NR	NR	NR
Doc057	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc058	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc059	NR	R	NR	NR	NR	R	NR	NR	NR	NR
Doc060	NR	R	NR	NR	R	R	NR	NR	NR	R
Doc061	NR	R	NR	NR	NR	R	NR	NR	NR	R
Doc062	NR	NR	NR	NR	R	R	NR	NR	NR	NR
Doc063	NR	NR	NR	NR	R	NR	NR	NR	NR	R
Doc064	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc065	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Doc066	NR	NR	NR	NR	R		NR	NR	NR	R
Doc067	NR	NR	NR	NR	R	R	NR	R	R	R
Doc068	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc069	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc070	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc071	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc072	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc073	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc074	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc075	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc076	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc077	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc078	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc079	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Doc080	NR	NR	NR	NR	R	R	NR	R	NR	NR
Doc081	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Doc082	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Doc083	NR	NR	NR	NR	R	R	NR	NR	NR	NR
Doc084	NR	NR	NR	NR	NR	R	R	R	R	NR
Doc085	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Doc086	NR	NR	NR	NR	NR	R	R	NR	NR	NR
Doc087	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc088	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc089	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Doc090	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc091	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR

Doc092	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc093	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Doc094	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Doc095	NR	NR	NR	NR	NR	R	NR	NR	NR	R
Doc096	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Doc097	NR	NR	NR	NR	NR	R	NR	R	NR	NR
Doc098	NR	NR	NR	NR	NR	R	NR	NR	R	NR
Doc099	NR	NR	NR	NR	NR	R	NR	NR	R	R
Doc100	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Doc101	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc102	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc103	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc104	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc105	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc106	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc107	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Doc108	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Doc109	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Doc110	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Doc111	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc112	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc113	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc114	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc115	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc116	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR

Appendix III Python Code for Indexing

```
def indexTerm(ldocid):
    fileterms=[]
    List2 = []
    List3 = []
    L7 = []
    List4 = []#store terms in each document in sorted order??
    List22 = []#store unique terms from the document
    indexterm=[]
    cff = {}#store terms and its collection frequency in a dictinay
    tff = {}#store terms and its term frequency in a dictinay
    dff = {}#store terms and its document frequency in a dictinay
    posFile=[]#store docid and tf of terms
    docID = 0#store docid
    L9 = []#L9 store term, docid and tf for all document
    L10=[]#L10 stor the docid and tf
    stop=open("stopw.txt",'r')
    stopw=stop.read()
    # the loop that reads the id of each document together
    for docid in ldocid:
        docID = docID + 1
        List5 = []
        aList1 = []#store the key of dictionary which is unique index
terms as a list
        aList2 = []#store the value(tf)
        aList3 = []#store the tf for further purpose
        res = []
        L8 = []#store term,docid and tf
        #fileterms store terms in each documents
        fileterms=fileReader(docid)
        for term in fileterms:
            List1 = []#store every index in every docs terms as a
list
            L6 = []#have terms
            indexterms=[]#indexterms[] store each terms in the
documents that are stemmed and punc removed
            for index in term.split():
                if index.isalpha():
                    stop_words=open("stopw.txt",'r')
                    n=stop_words.read()
                    s=n.split()
                    stop=0
                    for i in range(0,len(s)):
                        if thestemmer(punctuationremove(index))in
s[i]:
                            stop=1
                        if stop==0:
                            if len(thestemmer(punctuationremove(index)))
> 4:
indexterms.append(thestemmer(punctuationremove(index)))
                            for k in indexterms:
                                List1.append(k)
                            List4 = sorted(List4 + List1)
                            List22 = list(set(List4))
                            List5 = sorted(List5 + List1)
            #term frequency count loop
            for tf in List5:
```

```

        try:
            tff[tf] = tff[tf] + 1
        except:
            tff[tf] = 1
        tfff = sorted(tff.items())
    for keyValue in range(len(tfff)):
        aList2.append(tfff[keyValue][0])#index terms
        aList1.append(tfff[keyValue][1])#tf of terms
    for kk in aList1:
        aList3.append(kk)#aList3 store tf for futher process
purpose
        posFile.append((docID, kk))
    ListDict = dict(zip(aList2,aList3))
    for key in ListDict:
        L6.append(key)#have terms
        L8.append((key, docID, ListDict[key]))#store term,docid
and tf for a single doc
        L7 = sorted(L7 + L6)
        tff = {}
        List3 = sorted(List3 + List2)
        L9 = sorted(list(L9 + L8))#L9 store term, docid and tf for
all document
    #the loop count the collection frequency of terms
    for cf in List4:
        try:
            cff[cf] = cff[cf] + 1
        except:
            cff[cf] = 1
        cfreq = sorted(cff.items())

    #the loop count the document frequency of terms
    for df in L7:
        try:
            dff[df] = dff[df] + 1
        except:
            dff[df] = 1
        dfreq2 = sorted(dff.items())
    y=[]#store terms by extracting from cf
    for kk2 in range(len(cfreq)):
        y.append(cfreq[kk2][0])

    TDFCF = []#store index term df and cf as a single list
    for k1 in range(len(dfreq2) and len(cfreq)):
        TDFCF.append((dfreq2[k1][0], dfreq2[k1][1], cfreq[k1][1]))
    L33 = []
    ID = 0
    L44=[]
    for j in ldocid:
        ID = ID + 1
        f=open(j, 'r')
        f2=f.read()
        dic={}
        for term in y:
            if term not in dic:
                found=f2.find(term)
                while found > -1:
                    if term not in dic:

```

Declaration

This thesis is my original work has not been submitted as a partial requirement for a degree in any University

Solomon Regassa Guda

July, 2016

The thesis has been submitted for examination with my approval as University advisor.

Million Meshesha (PhD)

July, 2016