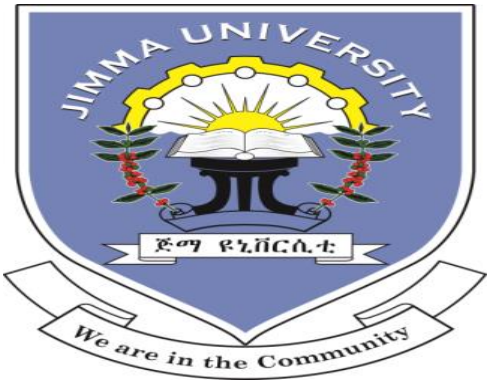

JIMMA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

DEPARTMENT OF INFORMATION SCIENCE

A PROBABILISTIC INFORMATION RETRIEVAL SYSTEM FOR AFAN OROMO TEXT

BY:

TOLESSA DESTA

NOVEMBER, 2018

JIMMA, ETHIOPIA

JIMMA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

DEPARTMENT OF INFORMATION SCIENCE

A PROBABILISTIC INFORMATION RETRIEVAL SYSTEM FOR AFAN OROMO TEXT

A Thesis Submitted to School of graduate Studies of Jimma University in Partial Fulfillment of the Requirements for Degree of Masters' of Science in Information Science (Information and Knowledge Management)

By:

TOLESSA DESTA

Principal Advisor: Million Meshesha (PhD)

Co-Advisor: Workineh Tesema (MSc.)

November, 2018

Jimma, Ethiopia

Advisor Name: Million Meshesha (PhD)[1] and Workineh Tessema (MSc.)[2]

Department: Information Science

University: Addis Ababa University[1] and Jimma University[2]

City: Addis Ababa[1] and Jimma[2]

Declaration and Certification

This is to certify that the thesis entitled **"A probabilistic Information Retrieval System for Afan Oromo Text"** submitted by **Tolessa Desta**, the MSc. (IKM) student of department of Information Science, Jimma University, for the award of Master's of Science in Information and Knowledge Management specialization, is a record of original work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the University and in my opinion the thesis has reached the standard needed for submission. The results embodied in the thesis have not been submitted to any other University or Institute for the award of any degree or diploma. I hereby declare that the student has incorporated the comments given during the mock defense to improve the work substantially.

Name	Title	Signature	Date
Birhanu Megersa(MSc.)	Chair person	_____	_____
Million Meshesha (PhD)	Principal Advisor	_____	_____
Workineh Tessema (MSc.)	Co-Advisor	_____	_____
Tibebe besha(PhD)	External Examiner	_____	_____
Takele Tadese (MSc.)	Internal Examiner	_____	_____

Dedication

This thesis work is dedicated to my family, especially to my lovely wife Ayantu Gebeyehu, my father Desta Ayele and my mother Birki Begna who have raised me to be the person I am today.

Declaration

I hereby declare that, this thesis is my original work. It has not been presented (submitted) to a partial requirement for a degree in any other university and all sources of material used for the study have been duly acknowledged.

_____

TOLESSA DESTA AYALE

November, 2018

This thesis has been submitted for examination with our approval as university advisors.

_____

Advisor: Million Meshesha (PhD)

_____

Co-Advisor:Workinek Tessema(MSc.)

November, 2018

Acknowledgement

First of all, I would like to thank my Almighty God for all ups and downs, for every success in my life, for giving me the wisdom and the strength I need to discharge my duty. Secondly, I would like to express my deepest appreciation to my Advisor Dr. Million Meshesha and my Co-advisor Mr. Workineh Tessema for their critical comments and their patience in helping me to complete my work. This study would not become real without them from beginning to end of the study.

My special thanks go to my brother, *Wakjira Desta* for his extreme support and encouragement in difficult times to become my life partner at times I needed most and also, for passing all those hardships I had to go through easily. God bless you for your encouragement in all side. I would also thank my family for their encouragement and support especially my caring wife *Ayantu* for her endless devotion, enthusiasm, and patience, my father *Desta Ayele* and my mother *Birki Begna*, thank you that you sent me to school. Again, I would also like to thank Worku Jimma (PhD Candidate) who has helped me on the grammar correction and giving suggestion during my study progress by giving his honorable time. Additionally, I would like to thank Mr. Merga Abera, Mr. Mulugeta Assefa, Mr. Tirate Kumera and Mr. Diriba Girma for being my good friends as advisors/brothers/project/lunch/tea/lab partners during our graduate study.

Lastly but not least, I offer my regards and blessings to all of those who supported me in any respect during the completion of the thesis as well as expressing my apology that I could not mention all and Wollega University for financial support.

Table of Contents **Page**

Dedication i
Declaration ii
Acknowledgement iii
List of Figures vii
List of Tables viii
List of Abbreviations and Acronyms ix
Abstract x
CHAPTER ONE 1
INTRODUCTION 1
1.1. Background of the Study 1
1.2. Statement of the Problem 3
1.3. Objective of the Study 5
1.3.1. General Objective 5
1.3.2. Specific Objectives 5
1.4. Scope and Limitation of the Study 5
1.5. Significance of the Study 6
1.6. Definition of Operational Terms 7
1.7. Organization of the Study 7
CHAPTER TWO 8
LITERATURE REVIEW AND RELATED WORKS 8
2.1. Overview of Information Retrieval 8
2.2. The Information Retrieval Process 11
2.2.1. The Indexing Process 12
2.2.2. Query Processing 13
2.2.3. The Matching Process 13
2.3. Information Retrieval models 14
2.3.1. The Boolean Model (BM) 15
2.3.2. The Vector space model 15
2.3.3. The Probabilistic Model 20

2.3.3.1. Binary Independent model (BIM) 21
2.3.3.2. Bayesian Networks Model 22
2.3.3.3. Bayesian Inference Network Model 23
2.3.3.4. Bayesian Belief Network Model 23
2.4. Query Operation 24
2.4.1. Relevance Feedback 24
2.4.2. Query Reformulation 25
2.5. IR Systems Evaluation 26
2.6. Overview of Afan Oromo 29
2.6.1. Historical Overview of Afan Oromo and its Writing System 29
2.6.2. Alphabets and Sounds (*Qubeelee fi Sagaleewwan*) 29
2.6.3. Word Morphology in Afan Oromo 31
2.6.3.1. Types of Morphemes in Afan Oromo 32
2.6.3.2. Word Segmentation 33
2.6.4. Afan Oromo Punctuation Marks 34
2.6.5. Short Forms of Compound Words 34
2.7. Related Works 35
2.7.1. IR Systems by Global Researches 35
2.7.2. IR Systems by local researches 39
CHAPTER THREE 44
METHODOLOGY 44
3.1. Study Design 44
3.2. Development Tools 44
3.3. Proposed solutions 44
3.0. Corpus Acquisition and Preparation 46
3.0.1. Tokenization 47
3.0.2. Normalization 48
3.0.3. Stop word removal 49
3.0.4. Stemming 50
3.4. Inverted index 53
3.5. Searching Using Probabilistic Model 53

3.6. Evaluation Method 56
CHAPTER FOUR 60
EXPERIMENTATION AND DISCUSSION 60
4.1. Corpus and Query Preparation 60
4.2. Indexing 62
4.3. Evaluation Method 64
4.4. Performance Evaluation 65
4.4.1. Relevance feedback 72
4.5. Results and Discussion 73
4.5.1. Findings and Challenges of the Study 76
CHAPTER FIVE 79
CONCLUSION AND RECOMMENDATION 79
5.1. Conclusion 79
5.2. Recommendation 81
References 82
Appendices 86

List of Figures

Figure 2. 1:- General Architecture of a Search Engine 10
Figure 2. 2:-Information retrieval processes 11
Figure 2. 3:-Taxonomy of IR models. 14
Figure 2. 4:-Inner product interpretation of two vectors 18
Figure 2. 5:- Bayesian network model for IR systems 22
Figure 2. 6:- Model for experiments in information retrieval systems 27
Figure 2. 7:-Syllable structure 31
Figure 3. 1:-A probabilistic Based Architecture of Afan Oromo Text Retrieval System 46
Figure 3. 2:- Python Code Fragment for Document Tokenization 48
Figure 3. 3:- Python Code Fragment for Document Normalization 49
Figure 3. 4:- Python Fragment Code that Removes Stop Words 50
Figure 3. 5:- Python Fragment Code of Stemmer 51
Figure 3. 6:-Architecture of Afan Oromo Stemmer 52
Figure 4. 1:-Sample Snapshot for Vocabulary File……..……...……63
Figure 4. 2:- Retrieved documents for a given query *'qulqullina barnootaa mirkaneessuu'* 66
Figure 4. 3:-Retrieved Documents after Relevance Feedback for *"qulqullina barnootaa mirkaneessuu"* 69
Figure 4. 4:- Precision/Recall curve before and after relevance feedback 72

List of Tables

Table 2. 1:- Experimental result of Robertson and Sparck Jones work 35
Table 2. 2:- 2X2 Conventional Contingency 36
Table 2. 3:-Assumptions-Principles Contingency Table. 39
Table 2. 4: -Summary of Related Works 42
Table 3. 1:-Term incidence contingency table 55
Table 3. 2: -Retrieved versus relevant documents 57
Table 4. 1:- Types of news article used for development of Afan Oromo IR system 60
Table 4. 2:-List of queries with their relevant judgments 61
Table 4. 3:-vocabulary file 63
Table 4. 4: - Posting File 63
Table 4. 5:- The Initial Performance of the System 67
Table 4. 6:- The Performance of the System after User Relevance Feedback 70
Table 4. 7:-Summarized result of the overall performance of Afan Oromo IR 74

List of Abbreviations and Acronyms

BIM	Binary Independent Model
BIR	Binary Independent Retrieval
BM	Boolean Model
BNM	Bayesian Network Model
CLIR	Cross-Lingual Information Retrieval
EBM	Extended Boolean Model
GVSM	Generalized Vector space model
IR	Information retrieval
IRS	Information Retrieval System
VSM	Vector Space Model

Abstract

This thesis presents a research work on a probabilistic information retrieval system for Afan Oromo text. The primary purpose of an information retrieval system is to retrieve all the relevant documents, which are relevant to the user query. Information retrieval is not being an optional technology; it is an important to everybody and mandatory to use. As considerable amount of information is being produced in Afan Oromo rapidly and continuously; experimenting on the applicability of information retrieval system for Afan Oromo is important. The main objective of this study is to design a prototype architecture of Afan Oromo text retrieval system based on probabilistic model in order to increase its effectiveness in retrieving relevant documents as per the users information need. A Probabilistic retrieval model that has the capability of reweighting query terms based on relevance feedback could be used and also the potential of the model was investigated. The study presents the design and implementation of a probabilistic model for Afan Oromo free-text-documents. Both indexing and searching modules were constructed. Text operations were applied in both modules. Then, the retrieval system was tested using two hundred (200) Afan Oromo free-text-documents and ten (10) queries. Other types of documents like video, images and audio were not included. The development platform used to develop the system prototype is Python 3.6.5 programming language. The experimental results show that probabilistic based IR system in Afan Oromo free-text-documents returned encouraging result. The system registered, after user relevance feedback, an average precision, recall and F-measure of 60%, 91.56% and 72.5% respectively. This result is achieved without controlling the problem of synonyms and polysemous of terms that exist in Afan Oromo text. Though the performance of the system is greatly affected by the word variants, the result obtained is encouraging. It can be concluded that; when the terms are added to the user query and user relevance feedback is applied; the performance of the retrieval system increases. It is recommended that further research works be done to see the retrieval effectiveness of Afan Oromo IR system using other probabilistic models like bayesian network, Bayesian belief network, and Bayesian inference network model.

Keywords: - Information retrieval system, binary independent model, probabilistic model, Afan Oromo text, information retrieval

CHAPTER ONE
INTRODUCTION

1.1. Background of the Study

Today's a vital asset is information. This asset should be stored and well organized to be accessed anytime and anywhere. To do this, Information Retrieval is a very interesting area. Information retrieval (IR) is concerned with the retrieval of information that is relevant to a user's information need. It is an activity of obtaining relevant documents based on user needs from collection of retrieved documents. Information retrieval System is a system which has a capable of storing, maintaining from a system and retrieving of information. To conclude that, the main objective of IR system is to retrieve all the documents which are relevant to a user query while retrieving as a few non-relevant documents as possible (Majumder, 2009).

IR is a well-established field in information science, which addresses the problems associated with retrieval of documents. The goal of any IR system is to respond to user-requested information by providing reference documents that meet the desired criteria (Manning, 2009). In the age of information today, people use the Internet day and night to fulfill their various needs of information. As the written information becomes large in size and digital documents easily available electronically, it would be difficult to retrieve relevant documents among the accumulated document collections. This exponential growth of information records of all kinds' results in the problem of information explosion (Christopher, 2009). The need to store and retrieve written information became increasingly important over centuries, especially with inventions like paper and the printing press. Soon after computers were invented, people realized that they could be used for storing and retrieving large amounts of information (Kocabas, 2011).

Nowadays, there are different IR models for determining what is relevant and what is not. They have evolved from specific models intended for use with small structured document to recent models that have strong theoretical basis and a variety of full text document types like Boolean model, Vector Space Model (VSM) and probabilistic model (Baeza-Yates, 1999).

According to Atalay (2014), citing Robertson (1977), noted that the current models handle documents with complex internal structure and most of them incorporate a relevance feedback component that can improve performance. Among the Current models, Probabilistic model is the most common. It works based on the probability ranking principle. As stated by Robertson (1977), "If a reference retrieval systems response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data".

Experimental evidences show that other models like VSM and its variant models such as, Extended Boolean Model (EBM) and Generalized Vector Space Model (GVSM) are not attempted to define uncertainty in IR system (Crestan, 2001). They do not have relevance feedback and term reweighting mechanism by them-selves to do with the external realities of users. There are different IR methods which have a probabilistic basis. The most widely used ones are binary independent model (BIM) and Bayesian network model (BNM). BIM works based on representation of queries and documents with relevance feedback data (Crestan, 2001).

Probabilistic methods are one of the currently best method in IR (Manning, 2008). Predicting relevant documents is one of the core issues in IR system. Probabilistic IR models are based on the probabilistic ranking principle. BIM is the most and first influential model used in IR system. As the name implies, that the index terms exist independently in the documents and we can then assign binary values to these index terms. The terms in the document are distributed independently (Neto, 1999).

Boolean model and VSM are important in the history of IR, and then probabilistic model came in to take the dominant role in IR system. It ranks documents in decreasing order of probability of relevance to the information need, and classifies them to either relevant or non-relevant groups. The motivation of this study is not only comparatively good at solving traditional IR system problems of the unclearness of users' information needs, but also helps to overcome obstacles embedded in the nature of document relevancy, being inherently uncertain (Zhu, 2016).

Since IR system is dealing with free-text-document, there is a need to apply text operations, such as tokenization, normalization, stop word removal and stemming. Stemming is the process of reducing morphological bounding of a given word to their stem, root or base. Stemmers are categorized into three subcategories by their stemming method. Dictionary-based stemmers, statistical-based stemmers and affix removal stemmers (Sharifloo, 2008).

The purpose of this study is to investigate probability of relevance for Afan Oromo text retrieval system by using BIM and affix removal stemming algorithms. In this case, a probabilistic approach can improve the uncertainty of the retrieval system for users query and effectiveness of the system, because it overcomes the limitations of VSM present in the language. The motivation behind this probabilistic is to allow and decide whether the document is relevant or non-relevant to the users query by handling uncertainty in the retrieval process.

1.2. Statement of the Problem

There are more than 80 languages in Ethiopia (Tsegaye, 2013). Afan Oromo is one of the widely spoken languages in Ethiopia with large number of speakers under Cushitic family. Currently, it is an official language of Oromia national regional state and spoken by 40 million Oromo's within Ethiopia. In addition, the language is also spoken in Somalia, Kenya, Uganda, Tanzania and Djibouti (Omnigton, 2013).

The language has become the official language of Oromia regional state which contains the highest language speakers in Ethiopia. It is a medium instructional language of the region starting from elementary to University level. As a result, newspapers, journal, references and other governmental documents are compiled using this language.

In this information age, information is highly needed than anything else, but finding and obtaining the right information needs a technology to support the language of users. Since a lot of information is available on the web, it is difficult to get the right information at the right time from the right source. The people who use Afan Oromo documents are increasing from time to time. But they may not get a system that searches for relevant documents that satisfy their information need. From this point of view, having an information retrieval system that works for Afan Oromo users is necessary.

Today as technology improves, several forms of information are in use everywhere in the World. Books, journal, articles and other documents can be accessed electronically. The issue has become one of storing and accessing this pervasive information in an effective and efficient manner using an IR system. Developing an IR system that enables searching and retrieving relevant documents written in Afan Oromo is a vital tool (Debela, 2010).

As Afan Oromo has a large number of speakers in Ethiopia, a huge amount of information is released by this language per day. Those speakers may want to browse by their own language; since, they can easily build their query by their own language. When users lack to clearly define their information need, it is difficult to find and get relevant documents. Using probabilistic model reduces difficulty of finding relevant documents and uncertainty. Therefore, the absence of retrieval system by Afan Oromo brings a lot of challenges to browse their information need.

Technology has a great role for the development of one language because it links the speakers and users of that language with easy system to access information in their daily activities. The fact that initiated this study is also enabling development of Afan Oromo to grow with the support of information technology. IR is not being optional technology; it is something that is very important to everybody and mandatory to use (Melkamu, 2017).

In addition to the above facts, most of documents available on the Internet which could be relevant to the users. This wealth of information which is available on the Internet should be accessed to all users specifically for the society of Afan Oromo speakers. To make this wealth of information available for Afan Oromo speakers society, the documents available on the web used by those users should be resolved in order to get the relevant documents. To solve this issue, a probabilistic model which used to solve problems for accessing relevant information is an important tool.

The development of probabilistic text retrieval for Afan Oromo is crucial in order to get relevant documents and to handle uncertainty that exist in IR systems. Most of the time short queries have an ambiguity for the users to retrieve the most top relevant documents. And also, previous research done does not have the capable of re-weighting query terms based on relevance feedback used. Hence, this study tries to fill such a gap in the language.

The major problem of retrieving texts for Afan Oromo is its effectiveness to identify relevant documents for the satisfaction of the users query. In order to increase the effectiveness of the relevant documents to satisfy the users query, IR techniques is crucial. Hence, to minimize the challenge of getting relevant information, probabilistic IR system is necessary.

Therefore, the aim of this study is to investigate statistical approach for Afan Oromo Text Retrieval System. More specifically, it aims to develop and test a probabilistic model of IR system for Afan Oromo text retrieval with a better accuracy level.

To this end, the study attempts to explore and answer the following research questions:

- What is the effect of the stemming on the performance of Afan Oromo probabilistic IR?
- What is the contribution of incorporating probabilistic model for Afan Oromo texts in improving the overall coverage of relevant documents for a given query?

1.3. Objective of the Study

1.3.1. General Objective

The general objective of this study is to design a probabilistic Afan Oromo IR system in order to increase its effectiveness in retrieving relevant documents as per the users' information need.

1.3.2. Specific Objectives

In order to achieve the general objective of the study, the following specific objectives were accomplished throughout the study:-

- ✓ To review related literature so as to identify suitable techniques and methods
- ✓ To index documents for speeding up searching for relevant documents
- ✓ To design an architecture of the prototype based on a probabilistic Afan Oromo IR
- ✓ To test and evaluate the effectiveness of the prototype

1.4. Scope and Limitation of the Study

This study focused on design of an IR system that searches effectively with in Afan Oromo text corpus. And also, it uses a probabilistic IR model to retrieve text corpus. The approaches corpus used for this study is health, education, religion, social, economy, culture, sport, politics and justice. Therefore, the scope of this study is restricted to Afan Oromo queries to retrieve relevant documents written in Afan Oromo for the users query. Text operations were done for the purpose

of tokenization, stop word detection, normalization and stemming. The study was limited to text retrieval excluding video, image, graphics and audio retrieval. These text documents was organized from different sources, such as Oromia Broadcasting Network, *Bariisaa* news, *Kallacha Oromiyaa* news, Voice of America, Bible chapters, Internet, Oromia Culture and Tourism bureau, online educational resources. Due to the absence of standard corpus, we had to prepare corpus for the experimentation which is unannotated free-document-texts. However, the amount of corpus prepared for this study is relatively small and requires further development; because of non-existence of the required corpus.

Limited corpus and queries was used for evaluating the performance of the IR system developed in the study as a result of time factor. It takes a lot of time to prepare relevance judgment for queries and corpus with many documents.

1.5. Significance of the Study

The output of the study is to help those who search for Afan Oromo documents to satisfy the user's information need. These users can be anyone who can formulate queries in Afan Oromo and understand the content of documents that are returned for the given query. The major contribution of the study is to develop a probabilistic IR for Afan Oromo speakers and capable of using IR systems to obtain their information need by their native language. Users enter their query in their native language and the system retrieves relevant documents in their own. Therefore, the beneficiaries of this study includes: individuals, schools, researchers, academic staffs, administrators, leaders, teachers, lecturers, and other researchers who know and speak Afan Oromo. This work also put stone for the future researcher to improve the better performance of Afan Oromo IR system by applying the rest of probabilistic models. This study also enables Afan Oromo speakers retrieving text documents in Afan Oromo efficiently and effectively.

1.6. Definition of Operational Terms

Information retrieval: - deals with the representation, storage, organization of, and access to information items.

Afan Oromo: - One of the major languages that are widely spoken and used in Ethiopia.

Query reformulation: - mechanism used to enhance the performance of the retrieval system

Information retrieval system: - a system which has a capable of storing, maintaining from a system and retrieving of information

Relevance feedback: - a mechanism of engaging users or system in retrieval process so as to improve the final result of the IR system.

1.7. Organization of the Study

This study is organized in to five chapters including appendices and references. The first chapter discusses introduction that provides background information, the problem statement, the research objective, research question, scope and limitations and significance of the study.

The second chapter reviews the literature covering the various theories/philosophies, techniques and methods of information retrieval system and various information retrieval models especially probabilistic model. In addition, the historical background of Afan Oromo language and its writing system are covered.

The third chapter presents the technique implemented, the architecture adopted and the algorithm used in this study. The experimental settings, test results interpretations and the findings of the experiment are presented in chapter four.

Finally, chapter five presents conclusion drawn from the findings of the study and recommendations that should be considered in future researches for designing an applicable Afan Oromo IR system.

CHAPTER TWO
LITERATURE REVIEW AND RELATED WORKS

This chapter is concerned with the basic concepts of information retrieval and the review of literature. It also covers some general background of information retrieval, IR models and different descriptions. Also an overview of the language structure is explained. General overview of probabilistic IR, its relation with IR and approaches were presented. Finally, review of related works on this probabilistic IR model is also covered.

2.1. Overview of Information Retrieval

An information retrieval is a very wide-ranging area of study, with the main aim of searching information of relevant documents from large corpus that satisfies information needs by the users. Since IR research involves language dependent process, in this study the researcher attempt to review the literature articles of Afan Oromo morphology works done, to enable us to design prototype of a probabilistic IR system for Afan Oromo text (Zaman, 2010).

In old days, people have become aware about the consequences of archiving and finding information. With arrive of computers, storing the huge amount of information become possible and finding the useful information become necessary. For this purpose, information retrieval becomes a very important research. This information retrieval concerned with searching and retrieving of information from a huge collection of documents (Deep, 2017).

According to Hiemstra (2009), IR technology is a combination of experiments and theory. Experiments are required to assess how the technology deals with the rapid growth of information and documents, and theoretical models help researchers avoid deductive reasoning during such experiments.

IR is the discipline that deals with retrieval of unstructured data, especially textual documents, in response to a query or topic statement. The need for effective methods of automated IR has grown in importance. IR focuses on retrieving documents based on the content of their unstructured components. An IR request may specify desired characteristics of both the structured and unstructured components of the documents to be retrieved (Zaman, 2010).

Melkamu (2017), citing Baeza-Yates and Ribeiro-Neto (1999) states that, “Information retrieval (IR) deals with the representation, storage, organization, and access to information items”. The representation and organization of the documents should provide the user with easy access to the information in which he/she is interested.

Information retrieval is a key technology that has been made in the history of humankind. It is the key technology behind search engines and an everyday technology for many web users. An IR system has been developed for serving users purpose of; good reading and magazine articles for an assignments; finding educational material for a learning objective; finding facts for decision making etc. However, the main problem is to retrieve what is useful and leaving what is not or in other word developing a perfect retrieval system. Perfect retrieval system does not exist, because relevant judgment is based on the subjective opinion of the users. What is relevant for one user may not be relevant for other user (Hiemstra, 2009).

Manning (2009) defined, information retrieval concerned with the domain of information searching, for both structured and unstructured information. The unstructured information searching of a text is searched from document corpus and the web. This IR includes various process and techniques.

The whole IR system includes three main subsystems: indexing, processing as well as searching and ranking (Zaman, 2010). **Indexing**: - is an offline process of extracting index terms from document collection and organize them using indexing structure to speed up searching (Gezahegn, 2012). Inverted file is the most common for text retrieval of this indexing structure. This structure is composed of vocabulary and term occurrences. The vocabulary is the set of all words in the text. For each word in the vocabulary, a list of all the text positions where the word appears is stored. The set of all those lists is called occurrences which is a language dependent process.

During Processing: - case normalization, stop word removal, stemming, lemmatization are applied on users query. In the case of textual retrieval, query terms are generally pre-processed by the same algorithms used to select the index objects. Additional query processing (e.g., query expansion) requires the use of external resources such as thesauri or taxonomies.

During Searching and Ranking: - user queries are matched against information items. As a result of this operation, a set of potential information items is returned in response to user needs. The ranking step aims to predict how relevant the items are comparatively to each other.

According to Chou (2010), Search Engine is a program which searches the database, gathers and reports the information with processing searching files using keywords specified. The key words found are returned and collated into the user information.

Since search engine is within the field of IR, understanding about the concept of information retrieval is necessary. Before internet was born, IR was just index searching. For example, searching authors, title, and subjects in library card catalogs or computers (Lam, 2001).

Search engines are special software tools that are designed to help people find information stored on the Web. Early search engines held an index of a few hundred thousand pages and documents, and received one or two thousand inquiries each day. Today, a top search engine will index millions of pages, and respond to tens of millions of queries per day (Tesfaye, 2010).

According to Sergey (n.d), search engine is a web program which is available over the internet that searches documents and files for keywords. It is divided into crawler (spider) and indexing parts. A crawler is a program that retrieves web pages. After the crawling processed indexing process is started in database.

A general architecture of a search engine is shown in Figure 2.1 below.

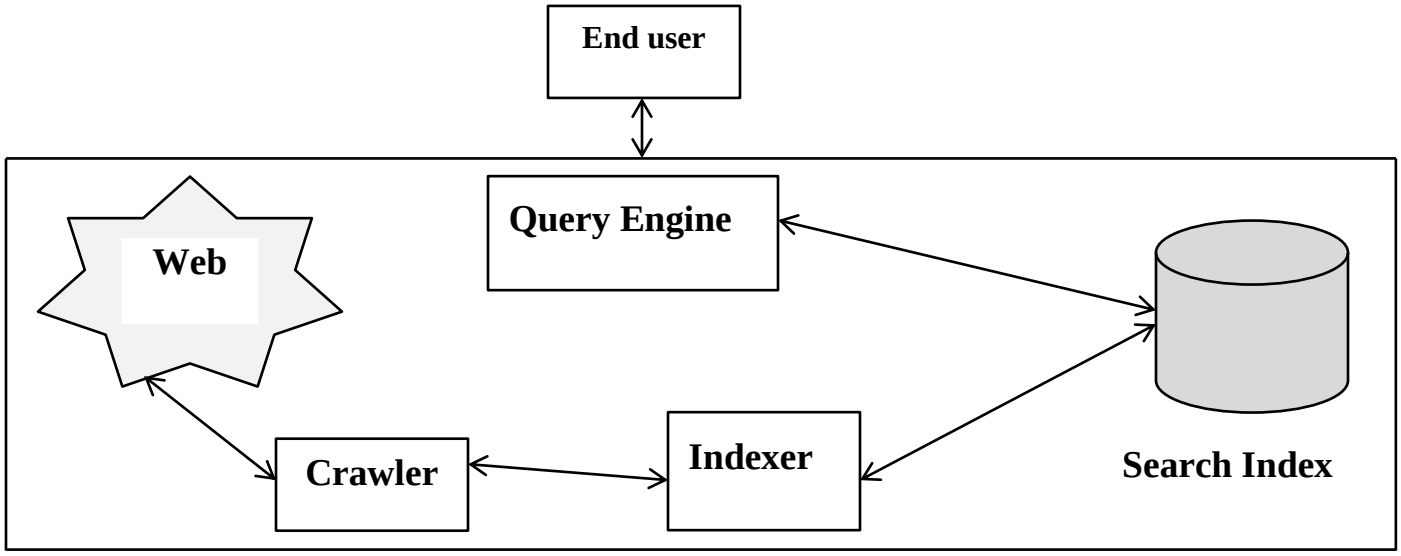

Figure 2. 1:- General Architecture of a Search Engine (Source: Tesfaye, 2010)

There is a difference between IR and searching the Web. IR allows access to whole documents, whereas, search engines do not. The reason is that it is too expensive to store all the Web pages locally and too slow to access remotely on other Web servers (Lam, 2001).

2.2. The Information Retrieval Process

The process that starts with the user need and ends up with retrieving documents is considered to be the core of the IR system. There are three basic processes of an information retrieval system has to support: the representation of the content of the documents, the representation of the user's information need, and the comparison of the two representations (Hiemstra, 2009).

The most widespread applications of IR are the ones dealing with textual data. As textual IR deals with document sources and questions, both expressed in natural language, a number of textual operations take place "on top" of the classic retrieval steps.

1. The user need is specified via the user interface, in the form of a textual query
2. The query *q* is parsed and transformed by a set of textual operations; the same operations have been previously applied to the contents indexed by the IR system
3. Query operations further transform the preprocessed query into a system-level representation (Ceri, 2013). The processes are visualized in Figure 2.2. In the figure, squared boxes represent data and rounded boxes represent processes.

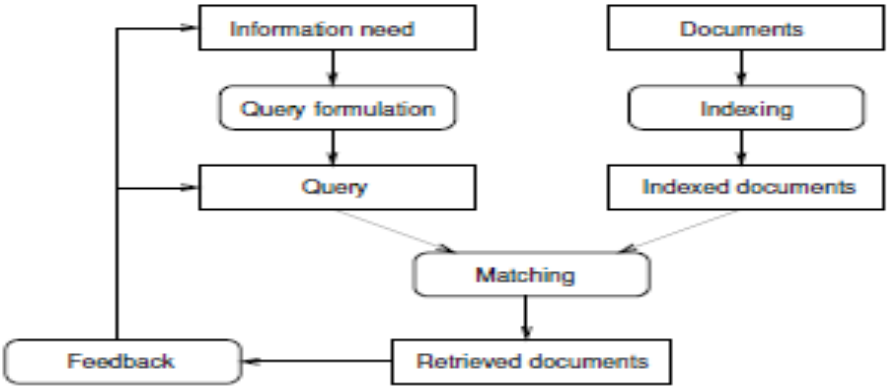

Figure 2. 2:-Information retrieval processes (Source: Hiemstra, 2009)

Representing the documents is usually called the indexing process. The process takes place off-line, that is, the end user of the information retrieval system is not directly involved. The indexing process results in a representation of the document (Hiemstra, 2009).

2.2.1. The Indexing Process

The indexing process consists of three basic steps: defining the data source, transforming document content to generate a logical view, and building an index of the text on the logical view. In particular, data source definition is usually done by a database manager module which specifies the documents, the operations to be performed on them, the content structure, and what elements of a document can be retrieved (Ceri, 2013). According to Hiemstra (2009), indexing process is an arrangement of index terms to permit fast searching and reading memory space requirement used to speed up access to desired information from document collection as per users query such that it enhances efficiency in terms of time for retrieval. Relevant documents are searched and retrieved quickly. Index file usually has index terms in a sorted order.

There are several index structures used for generating index terms; such as, sequential file, inverted file, suffix tree, suffix array and signature file. Sequential file is an indexing structure, which access elements of record in a predetermined ordered sequence. The records are arranged serially one after another in lexicographic order on the value of some key field. Inverted file stores a map from content to its locations in a database file. Suffix tree and suffix array process the suffixes of a given string to allow particular quick implementation string operations. Signature file works based on hash coded. It is a word oriented index structure (Neto, 1999).

The most popular indexing structure is inverted file, which is also adopted in this research. Inverted file stores a map from content to its locations in a database file. Inverted file is a mechanism for indexing a text collection so as to make the searching task fast. There are two elements involving in building the inverted file (Baeza-Yates et al., 1999): the vocabulary and the occurrence. The vocabulary file is the set of index terms in the text collection and it is organized by terms. The vocabulary file stores all of the keywords that appear in any of the documents in lexicographical order and for each word a pointer to posting file. The occurrence contains one record per term, listing all the text locations where the words occur and frequency of each term in a document (Kocabas, et al., 2011).

Document Operations

Documents are essential objects in IR systems and their numerous operations are considered crucial in information retrieval process. In many types of IR systems, documents added to a database must be given distinctive identifiers, parsed into their component fields, and those fields sliced into field identifiers and index terms (Sanjay, 2011).

2.2.2. Query Processing

Users do search just for a need of information in this digital era. The process of representing their information need is often referred to as the query formulation process. The resulting representation this query reformulation is the query. Query formulation might denote the complete interactive dialogue between system and user, leading not only to a suitable query but also to the user better understanding his/her information need: This is denoted by the feedback process in Figure 2.2 (Hiemstra, 2009).

2.2.3. The Matching Process

The method for determining the degree of relevance of the user's query with respect to the document representation is the matching process. In most practical cases, this process is expected to produce a ranked list of documents, where relevant documents appear at the top of the ranked list, in order to minimize the time spent by users in identifying relevant information (Ceri, 2013).

According to Baeza-Yates et al. (1999) stated that, one central problem of any information retrieval system is predicting which documents are relevant and which are not. The ranking algorithms are used for such decision. According to Hiemstra (2009), the theory behind ranking algorithms is a crucial part of IR system. They attempt to display documents in decreasing order based on their similarity score with the query. Documents that are considered as relevant to users will be displayed at the top of the retrieved list. Thus, IR models guide the process of matching and ranking relevant documents. The three classic models of Information retrieval: the Boolean model, the Vector space model and the probabilistic model are often used to accomplish these tasks. These models are briefly discussed below in the next section 2.3.

2.3. Information Retrieval models

The goal of IR is to provide users with those documents that will satisfy their information need. Make the IR to be efficient, the documents are typically transformed into a suitable representation. Now such type of information is retrieve efficiently with the help of IR models (Sanjay, 2011).

IR model is the mechanism of predicting and explain the need of the user given the query to retrieve relevance documents from the collection. IR models serves as blueprint so as to develop applicable IR system. In addition to that, IR models guide the matching process to retrieve a ranked list of relevant document given a query (Hiemstra, 2009).

IR models are categorized into two approaches. Those are semantic and statistical approach. Semantic approaches models like, latent semantic indexing and neural network try to work on syntactic and semantic analysis. They attempt to implement some degree of understanding the natural language text that users provide. Statistical approaches such as, VSM and probabilistic model attempts to retrieve documents that are highly ranked in terms of statistical measure (Greengrass, 2011).

The three most widely used information retrieval model that bases on statistical approaches are: Boolean model, vector space model, and probabilistic model (Singhal, 2002).

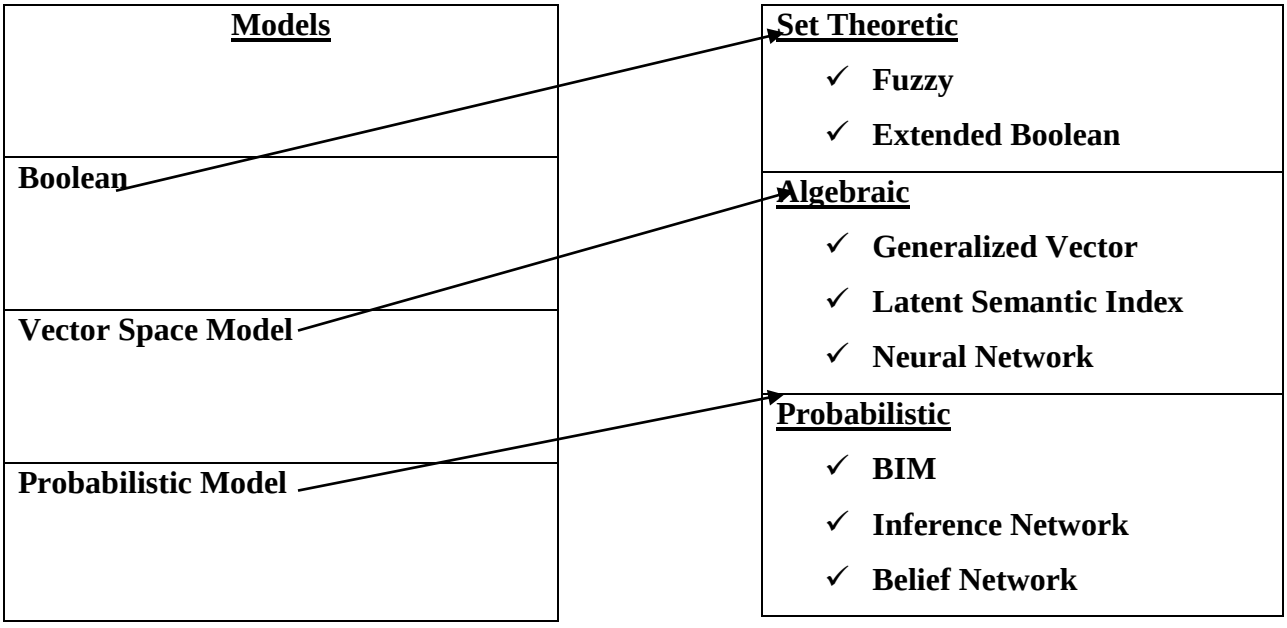

Figure 2. 3:-Taxonomy of IR models (Source: Neto, 1999).

2.3.1.The Boolean Model (BM)

The BM uses set theory and has three components AND, OR and NOT. For query formulation, it has one major drawback: a Boolean system is failed to rank the result list of retrieved documents. In the Boolean model, all documents are associated with a set of distinct words or key-words and User Queries are also represented by expressions of keywords separated by AND, OR, or NOT. The retrieval function of BM takes a document as either relevant or irrelevant (Raman, 2012).

The Boolean model is the first model of information retrieval and probably also the most criticized model. An advantage of the Boolean model is that it gives users a sense of control over the system. It is immediately clear why a document has been retrieved given a query.

According to the Boolean model a document is either relevant or non-relevant with respect to a particular query; there is no notion of grading. This implies that the similarity of a document dj to a query q is binary, i.e., similarity (dj;q) ε {0, 1}. The similarity of a document dj to a query q is defined in equation 2.1.

$$sim_{dj,q} = \begin{cases} 1 & \text{,if document satisfies the Boolean query} \\ 0 & \text{, otherwise} \end{cases} \quad \text{.....(2.1)}$$

2.3.2. The Vector space model

Vector space model is also term vector model which represents the documents and the queries as a vector in a multidimensional space. The term-specific weights in the query and document vectors are computed by the products of local and global parameters which is term frequency-inverse document frequency (tf-idf) model. In VSM, if q_i is the given input search term and D_i is the collection of document, and then the vector matrix is given as follows:

	Q1	q2	q3	…	qn
D1	d11	d12	q13	…	d1n
D2	d21	d22	q23	…	d2n
.	…	…	…	…	…
.	…	…	…	…	…
Dn	dn1	dn2	qn3	…	dnn

Where i ranges from 0 to n.

Search term may be of single word, keywords, query or longer phrases. If the chosen terms are words, then the number of words in the vocabulary decides the dimensionality of the vector. In other words, the numbers of distinct words occur in the corpus.

The VSM is an algebraic model used for IR (Singh, 2015). It represent natural language document in a formal manner by the use of vectors in a multidimensional space. The VSM is a way of representing documents through the words that they contain. The concepts behind VSM are that by placing terms, documents, and queries in a term-document space. The VSM allows decisions to be made which documents are similar to each other and to queries (Singh, 2015).

The VSM procedure can be divided in to three stages. The first stage is the document indexing where content bearing terms are extracted from the document text. The second stage is the weighting of the indexed terms to enhance retrieval of document relevant to the user. The last stage ranks the document with respect to the query according to a similarity measure.

The similarity function between a document vector Di and query Q is depicted in equation 2.2 as follows (Manning, 2008):-

$$\boldsymbol{Cosine\theta = sim(Q, D_i) = \frac{\sum_{i=1}^{v} w_{Q,j} \; x \; w_{i,j}}{\sqrt{\sum_{j=1}^{v} w_{Q,j}^2} \; x \; \sqrt{\sum_{j=1}^{v} w_{i,j}^2}}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \mathbf{(2.2)}$$

Where $w_{Q,j}$ is the weight of term j in the query Q, and is defined in similar way as $w_{i,j}$ (that is, $tf_{Q,j}$ ×idf_j). The term weighting scheme plays an important role for similarity measure. The weight of term in document vector can be determined using Tf×Idf method. The weight of term is measured how often the term j occurs in the document i using term frequency tfi,j and also the number of documents containing term j in the whole document collection using the document frequency dfj. The weight of a term j in the document i is presented in equation 2.3:

$$\mathbf{w_{i,j} = tf_{i,j} x \; idf = tf_{i,j} x \; \log \frac{D}{df_j}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \mathbf{(2.3)}$$

Where, D is the number of documents in the document collection and IDF stands for inverse document frequency.

The vector space model has been widely used in the traditional IR field and most web search engines. Both documents and queries are represented by vectors, which are sets of terms with associated weights. A vector similarity function, such as the inner product, can be used to

compute the similarity between a document and a query. We assume each document and each query are represented by a term frequency vector d = {x_1, x_2,…,x_n) and q = {y_1,y_2,…,y_n) respectively, where n is the total number of terms or the size of vocabulary and x_i,y_i are the frequencies of term t_i in d and q respectively.

TFxIDF is an efficient and simple algorithm for matching words in a query to documents that are relevant to that query. TFxIDF returns documents that are highly relevant to a particular query. If a user were to input a query for a particular topic, TFxIDF can find documents that contain relevant information on the query. Furthermore, encoding TFxIDF is straightforward, making it ideal for forming the basis for more complicated algorithms and query retrieval systems. Despite its strength, TFxIDF has its limitations. In terms of synonyms, notice that TFxIDF does not make the jump to the relationship between words (Berger et al, 2000).

The IR systems are using tf*idf weighting technique: *wij = tf(i,j) * idf(i)*. Search engines and information retrieval systems use this weighting technique often. It can be used for filtering stop-words and have application in text summarization and classification. The good thing about tf-idf is that it relies on both term frequency (tf) and inverse document frequency (idf). This makes simple to reduce rank of common terms throughout the whole document corpus, but increase in rank of terms exist in fewer documents more frequently (Christopher, 2009).

The ***term frequency*** (***tf***) is simply the number of times a given term appears in that document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term count regardless of the actual importance of that term in the document) to give a measure of importance of the term within particular document.

$$\boldsymbol{Tfij = \frac{fij}{\max\{fij\}}} \quad \ldots \quad (\mathbf{2.4})$$

The ***inverse document* frequency *(idf)*** is measure whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, then taking the logarithm and quotient. *Higher idf* value is obtained for rare terms whereas lower value for common terms. It is mainly used to discriminate importance of term throughout the collection.

$$\boldsymbol{Idf = log2\left(\frac{N}{dfi}\right)} \quad \ldots \quad (\mathbf{2.5})$$

Document frequency (*df*) — number of documents containing the given term. The more a term *t* occurs ***throughout*** all documents, the more poorly *t* discriminates between documents. The less frequently a term appears in the whole collection, the more discriminating it is.

Then the tf****idf*** is ***product*** of *tf* and *idf*.

$$Tf * idf = tfij * log2\left(\frac{N}{dfi}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (2.6)$$

By its definition, tf-idf is metric that multiplies the two quantities tf and idf. Here tf provides a direct estimation of the occurrence probability of a term when it is normalized by the total frequency in the document collection depending on the scope of the calculation. The normalization factor is common for all the terms in the scope, and thus can be omitted. On the other hand, idf can be interpreted as 'the amount of information' in conventional information theory, given as the log of the inverse probability. Bearing these in mind, a component of textual data such as a document or a term, the significance of the component is expressed as a product of the probability that it occurs and the amount of information that it represents (Aizawa, 2000).

According to Joho (2007), Vector space model has four main similarity measures. These four techniques are: - the inner product, cosine similarity, dice similarity and Jaccard similarity.

A. Inner Product

Inner product is the first technique in the vector space model. This technique considered as a base for other techniques. In a vector space, inner product is a way to multiply vectors together. The multiplication of such vectors forms the scalar between them.

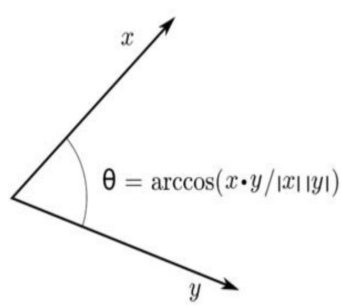

Figure 2. 4:-Inner product interpretation of two vectors

$$Q.D = \sum_{i=1}^{nv} w_{i,Q}\ x\ w_{i,D} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.7)$$

B. Cosine similarity

Cosine is used as a measure to find the similarity between two vectors. This is done by finding the cosine of the angel between them. So, if the inner product is used to find the distance between two vectors, the cosine is used to find the angel between these vectors.

$$\boldsymbol{Cosine}(\boldsymbol{Q},\boldsymbol{D}) = \frac{\sum_{i=1}^{nv} \boldsymbol{wi}, \boldsymbol{Q}\ \boldsymbol{x}\ \boldsymbol{wi}.\boldsymbol{D}}{\sqrt{\sum_{i=1}^{nv} \boldsymbol{w2i}, \boldsymbol{Q}\ \boldsymbol{x}}\sqrt{\sum_{i=1}^{nv} \boldsymbol{w2i}, \boldsymbol{D}}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots . (\mathbf{2.8})$$

C. Dice Similarity

Dice measurement is used like Jaccard to find the similarity between two vectors but gives twice the weight to agreements.

$$\boldsymbol{Dice}(\boldsymbol{Q},\boldsymbol{D}) = \frac{\mathbf{2}\sum_{i=1}^{nv} \boldsymbol{wi}, \boldsymbol{Qxwi}, \boldsymbol{D}}{\sum_{i=1}^{nv} \boldsymbol{w2i}, \boldsymbol{Q} + \sum_{i=1}^{nv} \boldsymbol{w2i}, \boldsymbol{D}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{2.9})$$

D. Jaccard Similarity

Also known as Jaccard index, the Jaccard similarity coefficient is a statistical measure of similarity between sample sets (Jacob, 2008).

Jaccard Similarity measure is another measure for calculating the similarity in the queries and documents. In this measure, the index starts with a minimum value of 0 (completely dissimilar) and goes to a maximum value of 1 (completely similar).

$$\boldsymbol{Jaccard}(\boldsymbol{Q},\boldsymbol{D}) = \frac{\sum_{i=1}^{nv} \boldsymbol{wi}, \boldsymbol{Q}\ \boldsymbol{x}\ \boldsymbol{wi}, \boldsymbol{D}}{\sum_{i=1}^{nv} \boldsymbol{w2i}, \boldsymbol{Q} + \sum_{i=1}^{nv} \boldsymbol{w2i}, \boldsymbol{D} - \sum_{i=1}^{nv} \boldsymbol{wi}, \boldsymbol{Q}\ \boldsymbol{x}\ \boldsymbol{wi}, \boldsymbol{D}} \ldots\ldots\ldots\ldots (\mathbf{2.10})$$

$$\boldsymbol{sim}(\boldsymbol{di},\boldsymbol{q}) = \boldsymbol{cos\theta} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots .. (\mathbf{2.11})$$

Generally, the whole VSM involves three main procedures. The first is *indexing* of the document in the way that only content bearing terms represent the document. The second is *weighting* the indexed terms to enhance retrieval of relevant document. The final step is ranking the documents to show best matching with respect to the provided query by user.

The main advantages of the vector model are:

1. Its term-weighting scheme improves retrieval performance;

2. Its partial matching strategy allows retrieval of documents that approximate the query conditions; and

3. Its cosine ranking formula sorts the documents according to their degree of similarity to the query

Although, it is the resilient and most popular ranking strategy now days, theoretically the vector model has the disadvantage that index terms are assumed to be mutually independent and also it is computationally expensive since it measures the similarity between each document and the query (Baeza-Yates and Ribeiro-Neto, 1999).

2.3.3. The Probabilistic Model

A major difference between information retrieval (IR) systems and other kinds of information systems is the intrinsic uncertainty of IR. Probabilistic information retrieval is the estimation of the probability of relevance that a document d_i will be judged relevant by the user with respect to query q. which is expressed as, $P(R|q,d_i)$, where, R is the set of relevant document. Typically, in probabilistic model, based on the query of user the documents are divided in to two parts. The first contain relevant documents and the second contain non-relevant (irrelevant) documents. However, the probability of any document is relevant or irrelevant with respect to user query is initially unknown. Therefore, the probabilistic model needs to guess at the beginning of searching process. The user then observe the first retrieved documents and gives feedback for the system by selecting relevant documents as relevant and irrelevant documents as irrelevant. By collecting relevance feedback data from a few documents, the model then can be applied in order to estimate the probability of relevance for the remaining documents in the collection. This process iteratively applied to improve the performance of the system so as to retrieve relevant documents which satisfies users need (Neto, 1999).

In probabilistic model, the order in which documents are presented to the user is to rank documents by their estimated probability of relevance with respect to the user information need. The principle behind this assumption is called, probability ranking principle (PRP) (Neto, 1999).

The most crucial or essential function of the probabilistic model is its initiate to rank documents by their probability of relevance given a user's query. Both documents and user queries are represented by vectors d and q these are binary vectors, each binary vector component show whether a given document attribute/component or term occurs in the document or query, or not. Instead of probabilities, for the probabilistic model the index term weight variables are all binary that is, $w_{ti}^{d}\{0, 1\}$, $w_{qi}\{0, 1\}$ (Drumond, 2009).

IR is an uncertain process: the criteria by which the information need is converted into a query, documents are converted into index terms, and query/document terms match far from exact. A probabilistic IR model attempts to represent the probability of relevance of a document given a query; in particular, it computes the similarity coefficient between queries and documents as the probability that a document *dj* will be relevant to a query *q (*Tunkelang, 2009*).*

The measure of probability defined as query q containing t_1, t_2…, t_n words and document d is defined as presented in equation 2.12.

$$\boldsymbol{score(q, di|t1, t2, tn) = product(p(R|t1) + p(R|ti) + p(R|tn} \ldots\ldots\ldots\ldots\ldots\ldots \boldsymbol{(2.12)}$$

According to Bansal (2012), probabilistic model is more efficient, effective and faster than VSM with the less complexity of index and positioning in the VSM. The results possess simplicity, flexibility, performance and formalism of the probabilistic model is more as compared to the VSM. Several retrieval models have been developed, which has a probabilistic basis. The most used and recent developed methods of probabilistic model are, Binary Independent Model, Bayesian Network Model, Inference Network Model and Belief Network Model.

2.3.3.1. Binary Independent model (BIM)

As the name implies, this model assumes that the index terms exist independently in the documents and we can then assign binary values to these index terms. For a further illustration of this model, consider a document *Dk* in a collection, is represented by a binary vector *t = (t1, t2, t3,…,tu)* where *u* represents total number of terms in the collection, *ti=1* indicates the presence of the i^{th} index term and *ti=0* indicates its absence (Indrawan,1998).

In the BIR model, as in most other probabilistic IR models, we seek to estimate the probability that a specific document *dm* will be judged relevant with respect to a specific query *qk.* In order to estimate this probability (denoted as *P (R\qk, dm))* in the following, we regard the distribution of terms within the documents of the collection. The basic assumption is that terms are distributed differently within relevant and non-relevant documents.

2.3.3.2. Bayesian Networks Model

Bayesian network models were first introduced in IR by Turtle and Croft (Turtle, 1990). In their model, index terms, documents and user queries are seen as events and are represented as nodes in a Bayesian network. The model takes the viewpoint that the observation of a document induces belief on its set of index terms, and that specification of such terms induces belief in a user query or information need. This model was shown to perform better than traditional probabilistic models for the task of document ranking.

A Bayesian network is a directed acyclic graph (DAG) whereby a node represents a proposition or an event and an arc represents a direct cause-effect dependency between two propositions or events. To represent the knowledge contained in document *d* and the knowledge represented by query *q*, a Bayesian network model uses two separate networks (Indrawn, 1998).

Document network and query network represent documents in the collection and the user query respectively. The two networks are combined when a retrieval process is performed as shown in figure 2.5.

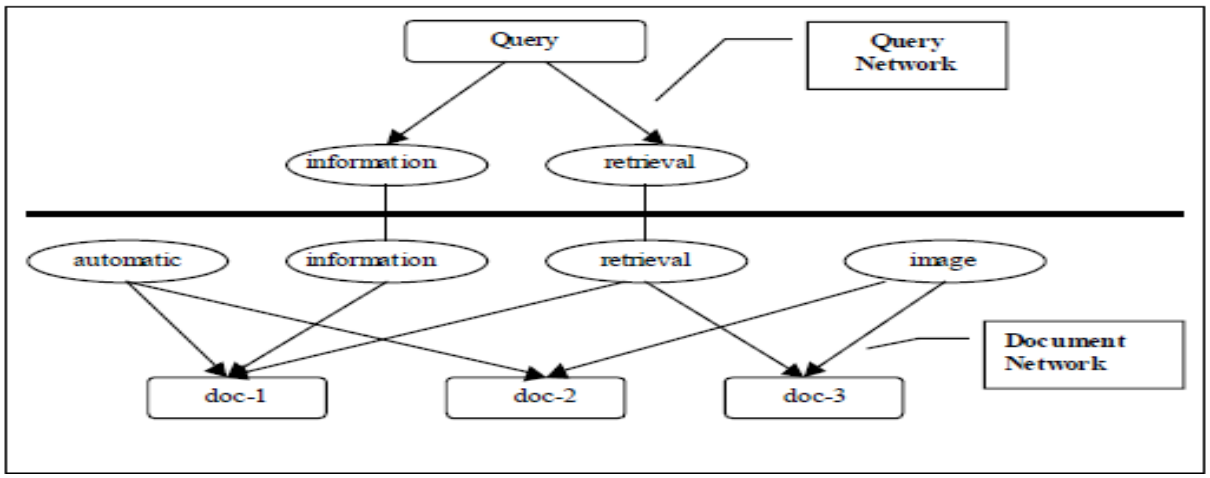

Figure 2. 5:- Bayesian network model for IR systems (Source: Indrawn, 1998).

The document and query networks are similar except that the document network is established at the creation of the database collection and remains the same unless new documents are added to or obsolete documents are deleted from the collection. The query network, on the other hand exists only for the duration of the user's query and is very dynamic in the sense that the query network changes for different queries as opposed to the document network which remains the same for different queries (Indrawn, 1998).

2.3.3.3. Bayesian Inference Network Model

IR is an inference reasoning process in which we estimate the probability that a user's information need, expressed as one or more queries. Therefore, network representation can be used to model this technique. The inference network has the ability to perform a ranking given many sources of evidence by performing a combination of confidence. The inference network is used to model documents, the contents of documents, and the query given by a user. The inference network consists of two sub-networks: the document network that is produced during indexing and the query network that is produced from the query text during retrieval process (Andrew, 2002).

2.3.3.4. Bayesian Belief Network Model

Bayesian Belief Networks (also known as Belief Networks, Causal Probabilistic Networks, Causal Nets, Graphical Probability Networks, and Probabilistic Cause- Effect Models) are an emerging modeling approach of artificial intelligence (AI) research that aims to provide a decision-support framework for problems involving uncertainty, complexity and probabilistic reasoning. The approach is based on conceptualizing a model domain (or system) of interest as a graph (i.e. network) of connected nodes and linkages. In the graph, nodes represent important domain variables and a link from one node to another represents a dependency relationship between the corresponding variables. To provide quantitative description of the dependency links, Bayesian Belief Networks (BBNs) utilize probabilistic relations, rather than deterministic expressions (Wooldridge, 2003).

Bayesian belief network is the use of Bayesian calculus to determine the probabilities of each node from the predetermined conditional and prior probabilities. However, probabilistic model have several potential advantages. First, the expectation of retrieval effectiveness that is near to optimal relative to the evidence used is high. Second, it has less reliance on traditional trial and error retrieval experiments. Third, each documents probability of relevance estimate can be reported to the user in ranked output. It would presumably be easier for most users to understand and base their stopping behavior (Greengrass, 2011).

2.4. Query Operation

Without detailed knowledge of the collection make-up and of the retrieval environment, most users find it difficult to formulate queries which are well designed for retrieval purposes. In fact, the users might need to spend large amounts of time reformulating their queries to accomplish effective retrieval. This difficulty suggests that the first query formulation should be treated as an initial attempt to retrieve relevant information. Query reformulation involves two basic steps: expanding the original query with new terms and reweighting the terms in the expanded query. For the query operations to happen relevance feedback and query reformulation is necessary. Relevance feedback enables to identify relevant document retrieved and query reformulation enables to expand the original query with the new terms and reweight the terms in the expanded query to retrieve relevant documents which satisfy user's information need (Hiemstra, 2009).

2.4.1. Relevance Feedback

In probabilistic model, the terms are treated as evidence that a document is relevant to a query. Given the assumption of term independence, the probability of a relevant document is computed as a product of the probabilities of each term in the document matching a term in the query. The probabilistic model is well suited for relevance feedback because it is necessary to know how many relevant documents exist for a query to compute the term weights (Christopher, 2007).

There are two relevance feedback mechanisms: user relevance feedback and pseudo relevance feedback. User relevance feedback is used to improve the final result of the IR system by involving the users in relevance feedback during the retrieval process. First, the user provides a query based on the IR system. Second, the user marks some returned documents as relevant or non-relevant. Third, the system computes a better representation of the information need based on the user feedback. Lastly, the system displays a revised set of retrieval results. Pseudo relevance feedback provides a method for automatic local analysis and it automates the manual part (Manning, 2008).

The process of relevance feedback is usually presented as an activity cycle, an information retrieval system presents a uses with a set of retrieval documents, the user only concern with that are relevant and the system will uses this information to produce an improved version of the query. The improved query is then used to retrieve a new set of documents for the make known of user. This whole process is known as iteration of Relevance Feedback (RF). RF can be

positive, negative or both. Positive RF only brings relevant documents into play and negative RF makes only use of irrelevant documents; any effective RF algorithms include a "positive" component (Rijsbergen, 2004).

On the other hand Pseudo relevance feedback, also known as blind relevance feedback, provides a method for automatic local analysis. It automates the manual part of relevance feedback, so that the user gets improved retrieval performance without an extended interaction. The method is perform normal retrieval to find an initial set of most relevant documents, then it assumes that the top k ranked documents are relevant, and finally, the system displays a revised set of retrieval results (Renata, 2013).

Relevance feedback shields the user from the details of the query reformulation process because all the users has to provide is a relevance judgment on documents and breaks down the whole searching task into a sequence of small steps which are easier to grasp (Baeza-Yates and Ribeiro-Neto, 2011).

2.4.2. Query Reformulation

Query reformulation is a mechanism used to enhance the performance of the retrieval system by using two different methods called query expansion and term reweighting. Query expansion technique is a process of adding a new term from relevant documents. There are two types of query expansion strategies: global analysis and local analysis. Global analysis strategy examines all documents in the collection so as to expand query. Local analysis examine only documents retrieved automatically for a given query q to determine query expansion (Neto, 1999).

The study of query reformulation give the idea of different approach and expand the query by using synonyms of words, and searching for the synonyms various morphological forms of words by stemming. The reweighting terms in the original query include: Simple use of co-occurrence data: Use of document classification, an automatically derived thesaurus, Query reformulations based on query log mining (Josh, 2013).

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, probabilistic term reweighting etc. Rocchio algorithm in one the most widely used algorithm designed for VSM. It finds a query vector which increases similarity with relevant document while decreases similarity with non-relevant documents (Manning, 2009).

Probabilistic relevance weights can be estimated from the relevant and non-relevant documents retrieved in an initial search and can be used in the next iteration of the search to improve retrieval performance. Such investigations test the benefits of modifying query statements and relevance weights in a feedback process. It attempts to predict the probability that a given document will be relevant to a given query (Manning, 2009). The similarity of document dj to a query q can be expressed as:-

$$sim(dj, q \propto \sum_{i=1}^{t} (wi, q)(wi, j) \left(\log \frac{p\left(\frac{ki}{R}\right)}{1 - p\left(\frac{ki}{r}\right)} + \frac{\log_P \left(\frac{ki}{\check{R}}\right)}{1 - p\left(\frac{ki}{\check{R}}\right)} \right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2.13$$

Where, P(ki|R) express the probability of getting term ki in the relevant documents of R and P(ki|$\check{R}$) represent the probability of getting term ki in the non-relevant documents of R . However, initially equation 2.13 is not used because of the probabilities of P (ki/R) and P(ki/R) are unknown. For the initial search (when there are no retrieved documents yet), two assumptions often made include: (a) P(ki/R) is constant for all terms ki (typically 0.5) and (b) the term probability distribution P(ki/R) can be approximated by the distribution in the whole collection.

2.5. IR Systems Evaluation

At the heart of IR, evaluation is the concept of *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. IR systems can be evaluated with respect to efficiency (operational issues like cost, time factor, space, etc.) as well as effectiveness (how well the retrieved documents satisfy the user's request) (Tesfaye, 2010).

The purpose of evaluation is to assess retrieval effectiveness against some standards of expected performance. For IR evaluations, a reasonably large set of documents is collected, and the relevance of each document to each query is judged. In practice, the thoroughness of relevance judgments will vary. A given system's performance will be reported in terms of *recall* and *precision*: recall indicates what percentage of all the relevant documents were retrieved at a given point; precision indicates what percentage of the documents retrieved were relevant. As recall increases to 100%, precision will decrease correspondingly (Broglio, n.d).

In IR system evaluation, the two common measure of system performance are efficiency and effectiveness. Efficiency is the time and space used by the system in retrieval process. To be called efficient system, the retrieval and indexing time of the system should be shorter and the space used in indexing file should be smaller (Neto, 1999).

In information retrieval experiments, the recall and precision levels are obtained by performing several retrievals on the test collection using the supplied queries. A test collection in information retrieval experiments comprises: *A set of documents, A set of queries, A set of relevance judgements* (Indrawn, 1998).. Each of these query-document sets in the test collection is used during experiments. The interaction of these sets in an information retrieval experiment is depicted by figure 2.6 below.

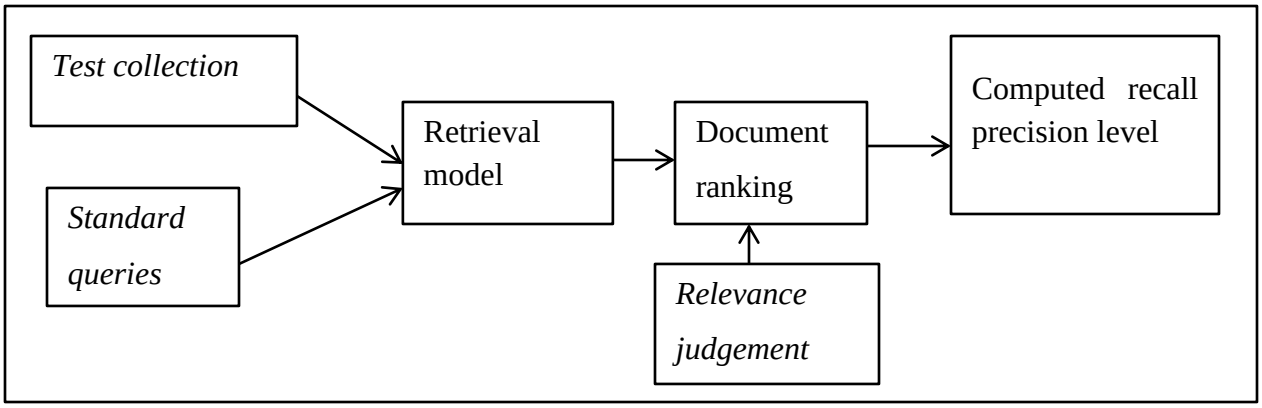

Figure 2. 6:- Model for experiments in information retrieval systems (Indrawn, 1998).

According to Baeza-Yates et al. (1999), while Precision (P) measures the ability to retrieve top-ranked documents that are mostly relevant (see equation 2.14). Recall (R) measures the ability of the search to find all of the relevant items in the corpus (see equation 2.15).

$$\boldsymbol{P = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ \text{documents retrieved}}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{2.14})$$

$$\boldsymbol{R = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ relevant\ documents}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{2.15})$$

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

$$\boldsymbol{F = \frac{2PR}{P+R} = \frac{2}{\frac{1}{P}+\frac{1}{R}}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{2.16})$$

A generalization of the F-measure is the E-measure, which allows emphasis on precision over recall or vice-versa. The value of the parameter β controls this trade-off: if β = 1 precision and recall are weighted equally (E=F), if β < 1 precision weights more, and if β > 1 recall weights more (Atalay, 2014).

$$\boldsymbol{E = \frac{[1+\beta^2]PR}{\beta^2 P+R} = \frac{[1+\beta^2]}{\frac{\beta^2}{R}+\frac{1}{P}}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{2.17})$$

Usually precision is more important than recall in IR systems, if the user is looking for an answer to a query, not for all the possible answers. Recall can be important when a user needs to know all the relevant information on a topic. A system can increase precision by decreasing recall and vice-versa; there is a precision-recall tradeoff (for example, recall can be increased by simply retrieving more documents, but the precision will go down, since many retrieved documents will not be relevant). Precision-recall curves can be used to compare two IR systems for all values of precision and recall. The *interpolated average precision* computes precision at fixed recall intervals (11 points), to allow fair average over all the queries in the test set at the same recall levels. This measure is in use lately in evaluating IR systems (Baeza-Yates et al., 1999).

2.6. Overview of Afan Oromo

In this subtopic, the basic structure of Afan Oromo is presented in order to understand the nature of the language. The language's nature like to what extent the language is spoken, application area of the language how words are formed, morphological nature of the language and other important features of the language that is specifically important is discussed under this section.

2.6.1. Historical Overview of Afan Oromo and its Writing System

Ethiopia is one of the multilingual countries. It constitutes more than 80 ethnic groups with diversified linguistic backgrounds (Assefa, 2005). The country comprises the Afro-Asiatic super family (Cushitic, Semitic, Omotic and Nilotic). As indicated in section 1.1, Afan Oromo belongs to an East Cushitic language family of the Afro-Asiati language super family and the most widely spoken in Ethiopia. It has around 40 million speakers, 50% of the total population of the country, native speakers and the most populous language of Ethiopia (Tilahun, 1992).

It is widely used as both written and spoken language in Ethiopia and neighboring countries like Kenya and Somalia (Kula et al., 2008). Currently Afan Oromo is an official language of Oromia Regional State (which is the largest region in Ethiopia) and used as an instructional media for primary and junior secondary schools of the region. Currently Afan Oromo is a language of research, administration, political and social interaction. The language is also academic language in Ethiopia universities like Jimma University, Addis Ababa University, Ambo University, etc. and Oromia regional state in primary schools. Currently newspapers, news, online education, magazines, journals, books, videos, pictures and entertainment Medias are increasingly published in this language (Dejene, 2015).

2.6.2. Alphabets and Sounds (*Qubeelee fi Sagaleewwan*)

Afan Oromo is phonetic language in which its characters sound is the same in every word in contrast to English language. Afan Oromo was written with either the Ge'ez script or the Latin alphabet until the 1970s. Starting from 1974-1991 writing any piece of writing using any script by Afan Oromo language was protected to be official. Since 1991 Latin alphabet is used as official alphabet of Oromo Language (Ager, 2012).

Some of the Afan Oromo language specific features are having one or two vowels in between consonants convey different meanings which are called as '*Jecha dheeraa'* and '*jecha*

gabaabaa' depending on the number of vowel letters used. In the language, if there are more than two vowels next to each other a glottal stop consonant or which is called '*hudhaa*' is used to make the word to be meaningful otherwise it is not allowed to have more than two vowels next to each other. And also in the word formation it is not allowed to have more than two similar consonant letters next to each other. For instance:

R + a + f + u + u = Rafuu to mean 'to sleep'

R + a + a + f + u + u = Raafuu to mean 'cabbage'

In general, in Afan Oromo using '*jecha dheeraa'* and '*jecha gabaabaa'* rely on vowels. A word which has two similar consonant letters has different meaning with that of one consonant letter. A word which has two similar consonant letters is named as '*Jecha jabaa*' and single consonant is called '*Jecha laafaa'*. For instance: *H + a + r + e = Hare* to mean 'he clean'

H + a + r + r + e = Harre to mean 'we clean'

This word which has two similar consonant letters and one consonant letter has an impact in information retrieval. This is considered as spelling error if we write *harre* instead of *hare*; the meaning can be automatically changed.

A. Vowels-*Dubbachiiftuu*

Afan Oromo vowels are similar to that of English but sound differently. There are five vowels **a, e, o, u** and **i**. All vowels pronounced in the similar way throughout every Afan Oromo literature. These vowels pronounced in sharp and clear fashion.

A: aanaa,abbaa,haadha,adaadaa

E: eebba, eeboo, eegumsa, waaree, maree

I: ardii, sadii, ilaali, dhiiga, dhiifama

O: Oromoo, odaa, obboleessa, onkoloolessa

U: ulfina,ulfa,guddina,guutuu,utaalcha

B.* Consonants- *Dubbifamaa

Most Afan Oromo constants do not differ greatly from Italian, but there are some exceptions and few special combinations.

i. The consonant "g" has a hard sound. *Gaarii, gadibayi, gargaari.*

ii. The combinations NY and DH have a hard sound. e.g *Nyaadhu, Dhugi*.

C. Double Consonants –*Dubbifamaa Dachaa*

All Afan Oromo consonants except the combination consonants *ch, dh, ny, ph, ts* and *sh* have double consonant combinations if the syllable is stressed. Failure to make this distinction results in miscommunication. For instance: *marga, dalga, gamna, olaantummaa, onkoloolessa, Malaammmaltummaa, walqixxummaa, walabummaa* (Melkamu, 2017).

D. Stress and syllables –*Qubee Jabaataa fi Birsaga*

Afan Oromo words do have stress or emphasis, which is placed within its syllables. Afan Oromo words are pronounced with the stress on the last syllable. For example: *hanna, ganna, sabboonaa, qabbaneessuu.* Stresses in Afan Oromo have double same consonants in the words.

In Afan Oromo, syllables have two parts. These are start and center. The center has two parts. These are pillar and closed. See its structure below:

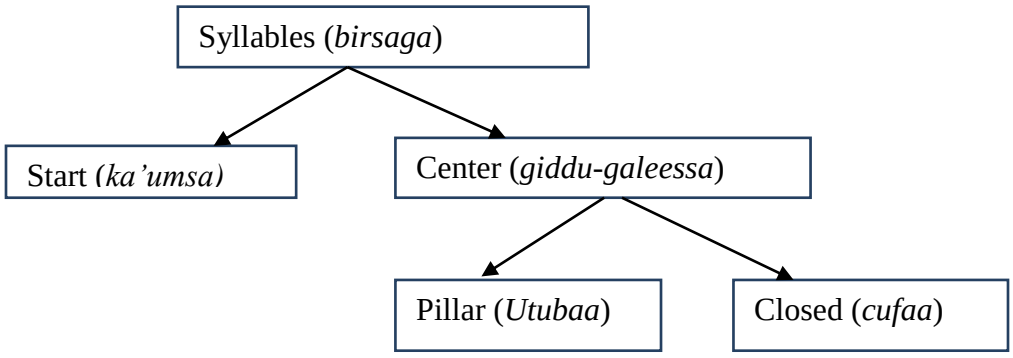

Figure 2. 7:-Syllable structure

In this structure the pillar is a vowel in the syllable. For example the word *fardaan* has two syllables such as *far* and *daan*. For this **f** :(*ka'umsa* or start), **a** :(*utubaa* or pillar) and **r** :(*cufaa* or closed).

2.6.3. Word Morphology in Afan Oromo

The word morphology has different meaning in different field of study. As it is stated in different literature, morphology in linguistic is classified in to inflectional and derivational morphology. Inflectional morphology describes the word variants that can be formed from the same stem word. It is the process by which affixes are combined with the root word to indicate basic

grammatical classification like tense or plurality. Inflectional morphology is the process of adding some meaning to the existing words. Different word forms are formed from one root word to indicate person, numbers, and gender, tense or case (Abebe, 2010).

In Afan Oromo language every word contains one or more morphemes (‘*dhamjechoota*’). These morphemes are categorized as free and bound morphemes. Free morphemes can stand as a word and they can constitute words by themselves but a bound morpheme does not occur as a word on its own. They are parts of words occur combined to free morphemes. For example in English word ‘cats’ the word ‘cat’ is free morpheme because it can stand alone and the character ‘-S’ is a bound morpheme because it does not stand alone like that of the unbound (free) morpheme.

2.6.3.1. Types of Morphemes in Afan Oromo

There are two categories of morphemes: free and bound morphemes. Free morpheme can stand as a word on its own whereas bound morpheme does not occur as a word on its own. In Afan Oromo roots are bound as they cannot occur on their own like “*dhug-”* (drink) and “*beek-“* (know), which are pronounceable only when other completing affixes are added to them.

Like the root, an affix is also a morpheme that cannot occur independently. It is attached in some manner to the root, which serves as a base. These affixes are of three types-prefix, suffix and infix. The first and the second types of affixes occur at the beginning and at the end of a root respectively in form a word. In *beekumsa* (knowlwdge), for instance, *-umsa* is a suffix and *beek-* (know) is a stem. An infix is a morpheme that is inserted within another morpheme. Like English, Afan Oromo does not have infixes as far as I could ascertain from the existing literature (Debela, 2010). The stemmer didn`t consider words formed by duplication of some characters at all. But Afan Oromo is rich in this kind of word formation. Most of the adjectives form the plural by reduplication of the first syllable. For example words like, *jajjabaa (stron-plural), gaggabaabaa (short-plural)* are formed from *-jabaa* and *-gabaabaa* by duplicating the first syllabus, respectively. Similarly an affix is also a bounded morpheme that cannot occur independently.

According to Kekeba et al (2008), it is possible to categorize suffixes in Afan Oromo into three basic groups: derivational, inflectional, and attached suffixes. Attached suffixes are particles or postpositions like arra, -bira, -irra, -itti, -dha, -f, etc. that are attached to stem/root words. For

instance the word “adunyaarratti” (in the world) is formed from a stem, i.e. adunyaa and two attachment suffixes, i.e. -irra + -itti.

Inflectional suffixes are a combination of word stem with grammatical/syntactic morphemes, usually resulting in a word of the same class as the original stem. These suffixes include plural noun markers such as oota, (e.g. nama + -oota = namoota i.e. person + -s = persons); -lee (e.g. jabbi + -lee = jabbilee, i.e. calf + -es = calves) and -wwan (e.g. indaaqqoo + -wwan = indaaqqoowwan, i.e. chicken + -s = chickens). Inflectional suffixes for other forms of nouns as well as adjectives and verbs are affixed to the stem words in similar manners though they are sometimes more complicated by requiring certain modifications in the stem. Derivational suffixes enable a new word, often with a different grammatical category, to be built from stem/root other words. For example, the stem verb qabuu + -eenyaa be-comes qabeenyaa which is noun while the adjective gowwaa +-ummaa becomes gowwummaa, which is also another Afan Oromo noun. Based on our current observations the most common order/sequence of Afan Oromo suffixes (right to left) is derivational, inflectional and attached suffixes. Thus, the stemmer is expected to remove from the right end first all the possible attached suffixes, then inflectional suffixes and finally derivational suffixes.

2.6.3.2. Word Segmentation

The word is the smallest unit of a language which is called in Afan Oromo “jecha”. 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 (Meyer, 2008). Afan Oromo is one of Cushitic family that uses Latin script for textual purpose and it uses white space character to separate words from each other’s. For example, “*Gammachuun Jimmaatii dhufe*”. In this sentence the word “*Gammachuun”,” Jimmaatii”* and *“dhufe*” are separated from each other.

Word and Sentence boundaries

In Afan Oromo, like in other languages, the blank character (space) shows the end of one word. Moreover, parenthesis, brackets, quotes are being used to show a word boundary. Furthermore, sentence boundaries punctuations are almost similar to English language i.e. a sentence may end with a period (.), a question mark (?), or an exclamation mark (!) (Getachew, 2014).

2.6.4. Afan Oromo Punctuation Marks

Punctuation marks used in both Afan Oromo and English languages are the same and used for the same purpose with the exception of apostrophe. Apostrophe mark (‘) in English shows possession but in Afan Oromo it is used in writing to represent a glitch (called *hudhaa*) sound. It plays an important role in the Afan 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(Daniel, 2011).

2.6.5. Short Forms of Compound Words

Afan Oromo compound words are written in different format. Mostly space and hyphens are used to separate them. When the hyphen is used the two words are treated as one word. For example; *“sar-diida”,”gar-malee”,”wal-faana”,”jal-bultii”* are compound words in separated by hyphens. However, when they are separated by space their meaning differ. For instance, “*saree diidaa”, “garaa malee”*. Short form representation of compound words is common in Afan Oromo similar to other languages like Amharic and English. In such representation, the use of forward slash (/) is much common although a dot or period (.) can also be used alternatively. In Afan Oromo documents, either of the short or long form of compound words may exist. If such words exist in query terms in information retrieval, there needs to be a way to handle the situation so that both forms could be retrieved as alternative for the query. But, this can create problems in IR (Tesfaye, 2010).

Example:

Afan Oromo (short form)	**Afan Oromo (long form)**	**English**
I/gaafatamaa	***Itti gaafatamaa***	Boss/Chief/Head
B/M	***Bulchiinsa Magaalaa***	City Administrator
W.A.	***Waldaa Aksiyoonaa***	Share Company
K.K.F	***Kan Kana Fakkaatan***	etc./ and so on

2.7. Related Works

Nowadays, information is the most powerful that enable as to be global people to share a common resources with the world. As a result designing IR system for Afan Oromo is becoming the daily work of researchers today. Hence, a number of studies have conducted on the probabilistic IR system which results several body of literature on the topic. The following sub sections summarize the review of some related works at international and local levels.

2.7.1. IR Systems by Global Researches

Robertson and Jones (1999), developed the well-known classical probabilistic model called Binary Independent Retrieval model so as to estimate the probability of relevance for the given query from document corpus, i.e. (p(R=r |D, Q) or p(R= r |D,Q)).

Probabilistic retrieval provides formal models for incorporating user or system relevance feedback. This is possible based on the presence or absence of independently distributed terms in relevant and non-relevant documents, is the BIM. In test collections for which exhaustive relevance evaluations provide complete inventories of relevant and non-relevant documents for a set of queries, term relevance weights can be computed definitively. Outcomes of such investigations establish the optimal performance of the retrieval model (Robertson, 1976).

When we back to the probabilistic model background, Maron and Kuhns (1960) first suggested the probabilistic retrieval model. The basic idea is to rank the documents in a collection based on their probability of being relevant to the current information need. This is expressed as P(r/N), or the probability that the information need is met given document N. A user's information need is something internal to the user and cannot be expressed exactly to the system, so this probability must be estimated using the terms supplied by the user in a query. The estimation is simplified using Bayes theorem to rewrite the probability as; prN =$prNprpN$ where r= relevant, N=total documents in the corpus.

Table 2. 1:- Experimental result of Robertson and Sparck Jones work

Weighting function	Precision	Recall
F0	29%	90%
F1	50%	90%
F2	60%	90%
F3	66%	80%
F4	70%	80%

As the result indicates, the ordering principle O2 is correct and O1 is incorrect. The performance also shows that F3 and F4 performed consistently better than F1 and F2. The performance of the system improves when information about the occurrences of terms in relevant documents is added to information about their simple document incidence. Specifically, relevance weights give a better performance than simple term matching (Robertson, 1976).

Several attempts have been also made to improve the binary independent representation. For instance, Shaw (1995) tries with the concept of term-relevance computations and perfect retrieval performance on the CF database by modified the convention computing equations for binary independent (BI) term relevance weights. First the author tries to see and interpret the BI relevance weights in terms of the conventional 'contingency' table 2.2.

Table 2. 2:- 2X2 Conventional Contingency

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

According to Shaw (1995) the relevance weight of term k, denoted wk, can be derived from a 2x2 contingency table in which the number of relevant and non-relevant documents including or excluding term k is represented symbolically. The number of documents in the four cells and marginal totals of table 2.2 is expressed in terms of four variables: the total number of documents (N), the total number of relevant documents (R), the number of documents in which term k appears (n), and the number of relevant documents in which term k appears (r).

"The probability term k appears in a relevant document, denoted by Pk, and the probability term k appears in a non-relevant document, denoted by uk, are defined respectively by $rRand$ $n−rN−R.$*The probability term k appears in a relevant document, defined by* pk $1−pkand$ *the probability term k appears in a non-relevant document, defined by*$uk1−ukcan$ *be written in terms of the empirical frequencies and are given respectively by* $rR−randn−rN−R−n+r.$ *If the probability is high that term k appears in relevant documents and low that term k appears in non-relevant documents, the presence of term k can discriminate the few relevant documents from the many non-relevant documents in a large document collection, which is the*

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

$wk=logepk1-pkuk1-uk$…….. (2.18)

Values of the term relevance function appear in the range $-\infty \leq wk \leq +\infty$. *When the odds term k appears in a relevant document are equal to the odds term k appears in a non-relevant document, wk = 0. A positive value of the term relevance function (wk> 0) indicates the odds favor term k appearing in a relevant document, and a negative value (wk< 0) indicates the odds favor term k appearing in a non-relevant document".* The relevance function, subject to BIM assumptions, can also be derived from a formal model based on Bayesian probability theory. In the formal model, the logarithm of the odds ratio causes the value of documents to be an additive function of relevance weights (wk). Although the meaning of relevance weights is conceptually ; computations of P*k*, u*k*, and w*k* can present difficulties, even with prior knowledge of all relevant documents; P*k* is undefined if R=0, uk is undefined if *N-R=0,* and w*k* is undefined if either P*k* or u*k* equals one or zero. Statistical theory has been invoked to resolve the problem of undefined values; leading to computing formulas.

pk=r+cR+1…….. (2.19)

uk=$n-r+cN-R+1$…… (2.20)

Shaw (1995) states that "*The theory demonstrates that the logarithm of* $\boldsymbol{r+cR-r+c}$ *is an unbiased estimate of the logarithm of* $\boldsymbol{pk1-pk}$ *and that the logarithm of* $\boldsymbol{n-r+cN-R-n+r+c}$ *is an unbiased estimate of the logarithm of* $\boldsymbol{uk1-uk}$ *when c = 0.5. Statistical theory allows one half to be added to each of the four cells of table 2.3 to guard against the effect of small cell frequencies on certain statistical calculations and causes one to be added to the marginal totals of the table. Consequently, the conventional computing formulas for pk and uk are defined by equation 2.19 and equation 2.20, with c=0.5. The conventional computing formulas do not allow wk to be undefined when r is equal to R or zero, or when n-r is equal to N-R or zero. Statistical theory does not, however, insure that equation 2.19 and equation 2.20, with c=0.5, provide unbiased estimates of pk and uk, or that subsequent computations of wk are meaningful in the present context*".

In 1976, Stephen Robertson and Karen Spärck-Jones proposed a probabilistic model for information retrieval under the following assumptions and principles:

Independence Assumptions:

I1 – the distribution of terms in relevant documents is independent and their distribution in all documents is independent.

I2 – the distribution of terms in relevant documents is independent and their distribution in non-relevant documents is independent.

Ordering Principles:

O1 – Probable relevance is based *only* on the presence of query terms in the documents.

O2 – Probable relevance is based on *both* the presence and absence of query terms in the documents.

I1 states that the presence of a term in a relevant document does not impact the presence of other terms in the same document or its presence in other relevant documents. I1 says nothing about the distribution of terms in non-relevant documents.

I2 extends I1 to non-relevant documents by stating that the presence of a term in a non-relevant document does not impact the presence of other terms in the same document or its presence in other non-relevant documents. Since documents are either relevant or non-relevant to a query, this is why I2 is more realistic than I1.

O1 indicates that documents should be ranked only if they contain all of the terms specified in a query. It is an AND approach. It says nothing about the absence of query terms in the documents.

O2 takes O1 a little further and states that we should consider both the presence and absence of query terms. It is an OR approach. So for a query consisting of two terms t1 and t2, documents mentioning both terms should rank higher than those mentioning one or none of these terms.

To implement O2, a system using an inverted index has to identify all terms present and not present in a document. To avoid exhaustively tracking the inverted index, we can assign zero probability of relevance to documents lacking of all query terms. Adopting this strategy implies that we have some evidence of non-relevance. It also has the effect of artificially converting O2-based weights to presence-only O1 weights. This makes O2 more practical than O1.

Table 2. 3:-Assumptions-Principles Contingency Table.

Independence Assumptions			
		I1	I2
Ordering	O1	F1	F2
Principles	O2	F3	F4

In Table, F1-F4 is weighting functions. According to Robertson and Spärck-Jones (1976), I2 is more realistic than I1 while O2 is correct and O1 is incorrect. The model then predicts that F4 is likely to yield the best results and is therefore the best match.

2.7.2. IR Systems by local researches

The review also has been done to find out work done for local languages in Ethiopia. When we come to the local researches done, a number of IR studies have been conducted so far for Amharic language, Tigrigna language and Tigrigna-Amharic CLIR language in probabilistic model which had been developed by Amanuel in 2012, by Atalay in 2014 and by Tsegaye in 2013 respectively. But nothing is found for Afan Oromo IR study on the probabilistic model. Information explosion in electronic text written in Afan Oromo caused an increasing need of designing effective and efficient IR system for Afan Oromo texts. A number of IR systems developed so far for retrieving Afan Oromo texts.

The research by Amanuel (2012) conducted on Probabilistic IR system for Amharic language was to experiments on the context. The main objective of the study was to experiment and design the effectiveness of probabilistic IR model for searching relevant documents from Amharic text corpus. The scope of the study was limited to only to develop a prototype IR system by applying a BIM for Amharic language. The news articles contain seven clusters of news items which were covers politics, sport, accident, education, health, tourism and justice. To test the prototype system developed, three hundred (300) Amharic News articles were used as a document corpus. All news articles were obtained from the web site of Walta Information Center. Additionally, ten (10) test queries were selected by the researcher to test the performance of the system. The problem explored in the study is IR systems based on VSM developed for Amharic language so far has not registered a better performance. The study was conducted by python programming language as an implementation tools. The basic statistical measures like recall, precision and F-measure were used as a testing procedures. As the experimental result show, probabilistic based

Amharic IR system register a better performance and score on the average 73% F-measure. This is a promising result to design an applicable IR system if polysemous and synonymous nature of Amharic words is controlled with the help of thesaurus and co-occurrence analysis. Finding standard corpus, ontology based stemming algorithm, building hybrid system of VSM and probabilistic models are the main concluding remarks forwarded by the researcher to get good performance.

According to Gezahegn (2012), Afan Oromo text retrieval system is conducted for the first time by using VSM. The problem statement to develop the system was even if the cross-lingual information retrieval (CLIR) was developed by Kekeba, et al in 2007 and Bekele in 2011 the users of the language are not satisfied yet, because the user's needs to get relevant information by searching in their language. The main objective of this study is to come up with an IR system that can enable to search for relevant Afan Oromo text corpus. It wasn't compared to any other IR models. Hence, it need further study to figure out best model that works for Afan Oromo retrieval system. Text document corpus was prepared by the researcher. Various techniques of text pre-processing including tokenization, normalization, stop word removal and stemming were used for both document indexing and query text. The performance result obtained were 57.5 % precision and 62.64 % recall respectively. Additional, the adopted stemmer was the main challenge by being incapable to handle word variants and the nature of synonymy and polysemy affect the performance of the system. Stemming algorithm was used for the study. Ontology, having thesaurus system, query reformulation, integrating video, audio, graphics and pictures were the concluding remarks forwarded by the researcher to get a good performance.

Atalay (2014) studied a probabilistic IR system for Tigrinya using BIM. Three hundred (300) Tigrinya documents and 10 queries were used to test the approach. The researcher presents the design and prototype implementation of the probabilistic model for Tigrinya documents. Rule based stemming algorithm and both indexing and searching modules were used and constructed for the study respectively. An average precision 69.1%, recalls 90%, and F- measure 74.4% were registered. Finding standard corpus, hybrid of rule based and dictionary based Tigrinya stemming algorithm or, ontology based stemming algorithm, synonym and polysemy terms are the major recommendations remarks given by the researcher.

Tsegaye (2013) developed probabilistic Tigrigna-Amharic CLIR. His work mainly focuses on an attempt to develop Tigrigna-Amharic CLIR system which enables Tigrigna native speakers to access and retrieve the online information sources that are available in Tigrigna and Amharic by writing queries using their own (native) language.

Dictionary based approach were conducted for the study and machine readable dictionary were used. Probabilistic IR model were employed in addition to the construction of both indexing and searching module. Those modules include: tokenization, normalization, stemming and stop word removal for both Tigrigna and Amharic languages are included. The system registered an average recall of 84% and 93%, an average precision of 75% and 64%, and average F-measure of 79% and 73% for Tigrigna and Amharic languages respectively. He concluded that the performance of the system obtained was encouraging given the limited size of the MRD used.

The research by Melkamu (2017) stated that, Query expansion is widely used technique for improving information retrieval effectiveness. Even if Gezahegn (2012) attemted to solve problems by developing IR system that can enable to search for relevant Afan Oromo text corpus, the system cannot search effectively for relevant Afan Oromo text corpus; because of query expansion. The main objective of this study is to apply query expansion for enhance the performance of Afan Oromo text retrieval system. The ideas of original Lesk algorithm were used for word sense disambiguation. Lexical resource like WordNet were constructed as reference for identifying the senses and meaning of the user's query using word sense disambiguation by semantic similarity measure. The result obtained were 56% F-measure which improves the performance by 5% from original query. Finding a standard corpus, combination of synsets to gloss and gloss to gloss for sense disambiguation, scholars to compare and contrast VSM with the performance of probabilistic model are the concluding remarks forwarded by the researcher .

Although there is no research done on probabilistic IR system for Afan Oromo text, Gezahegn (2012) mentioned the gap and recommendation of comparing any other IR models with VSM to figure out best model that works for Afan Oromo retrieval system. Also Melkamu (2017) recommends that any researchers can compare and contrast the performance of the VSM with probabilistic retrieval model. Starting from Gezahegn's and Melkamu's research gap and recommendations, the researcher developed a Probabilistic IR system for Afan Oromo text.

Table 2. 4: -Summary of Related Works

No	Title	Author	Method/Approach	Result
1	Probabilistic Tigrigna-Amharic CLIR	Tsegaye Semere	Probabilistic model(Binary Independent Model)	Average F-measure of 0.79 and 0.73 for Tigrigna and Amharic is obtained respectively.
2	Afan text retrieval system	Gezahegn Gutema	Vector space model	0.575 *precision* and 0.6264 recall were registered
3	Query expansion for Afan Oromo using Word Net	Melkamu Abetu	Vector space model	On average 95% recall, 41% precision and 56% F-score were registered
4	Probabilistic information retrieval system for Tigrinya	Atalay Luel	Probabilistic model(Binary Independent Model)	The system registered an average precision of 69.1%, recall 90.0%, and F-measure 74.4%.
5	Probabilistic IR system for Amharic language	Amanuel Hirpa	Probabilistic model(Binary Independent Model)	The score registered on the average 73% F-measure.
6	Afan Oromo search engine	Tesfaye Guta	Vector space model	An average precision of 93% were obtained
7	Towards the Sense Disambiguation of Afan Oromoo Words Using Hybrid Approach (Unsupervised Machine Learning and Rule Based)	Workineh Tessema	Hybrid (Unsupervised Machine Learning and Rule Based) ,vector space model	Accuracy of 70% in Unsupervised Machine learning and 81.1% in Hybrid Approach

Based on the above table 2.4, even if several IR systems have been developed, most of works were attempted to design an Afan Oromo IR system using vector space model. However, the use of vector space model may not control uncertainty nature of IR system. Thus, in this study an attempt is made to develop probabilistic based Afan Oromo IR system to enhance the performance of the Afan Oromo IR system. Vector space model does not re-formulate a query unless it is integrated with other modules. Gezahegn (2012) attempted Afan Oromo text retrieval based on VSM. But, still this could not give satisfactory performance for the peoples of Afan Oromo speakers as retrieving Afan Oromo documents by own query. Depend on an attempted gap and recommendation, the researcher developed Afan Oromo text retrieval based a probabilistic model that enhance the retrieval performance. Hence, query reformulation and relevance feedback improves the performance of the system.

CHAPTER THREE

METHODOLOGY

This chapter contains the detail methodology of the study such as proposed solutions, architecture of the system; algorithms and evaluation method were used. It was conducted in order to figure out the way to implement a probabilistic IR system for Afan Oromo text.

3.1. Study Design

Methodology is a way to systematically solve the research problem. Research designs are the plans and procedure that cover the decision from broad assumption to detailed methods of data collection. In this study, experimental research design was selected. Experimental research, which is also called empirical research or cause and effect method is a data-based research coming up with conclusions which are capable of being verified with observation or experiments. Experimental research is appropriate when proof is sought that certain variables affect other variables in some way. The experimental approach involves identifying potential methods, and implementing and testing iteratively.

3.2. Development Tools

To perform this experiment, python programming language is used for developing the system of Afan Oromo text retrieval. Python is dynamic programming language that is used in a wide variety of application domains. It is strong, involves natural expression of procedural code, modular, dynamic data types, and embeddable with in applications as scripting interface (Chun, 2006). Hence, Python 3.6.5 is used to develop and implement Afan Oromo IR system as a programming language. Designing IR system is more of text processing for generating index and query terms so as to apply indexing and searching. We used UTF-8 encoding for our corpus file and queries. The Python environment (version 3 and above) supports UTF_8 encoding. Hence, Documents are stored as text files using UTF_8 encoding.

3.3. Proposed solutions

The probabilistic model attempts to address the uncertainty problem in IR through the formal methods of probability theory. Unlike in the VSM, in this model the document ranking is based on the probability of the relevance of documents and the query submitted by the user.

In indexing part which was depicted in figure 3.1, text preprocessing like tokenization, normalization, stop word removal and stemming techniques were done. One can view an index inverted file as a list of words were each word is followed by the identifier of every text that contains the word. The number of occurrences of each word in a text is also stored in this structure. Hence, the major step in creating an inverted index file is:-

1. Collecting the documents and read all documents to be indexed
2. Tokenizing the document collected
3. Normalizing the tokenized documents in similar case
4. Remove the stop word list from the documents collected
5. Change all terms from the documents collected into their root (stem) words
6. Identify list of tokens to be indexed and create inverted index file which includes vocabulary files and posting files. This vocabulary files and posting files was discussed in table 4.3 and table 4.4 below briefly.

Generally, this study followed an experimental study in a sense that build different algorithms and test them until the required level of performance is achieved. Development of IR system involves various techniques and methods. As depicted in figure 3.1, the Afan Oromo text retrieval system takes both documents and queries as input. The query was processed to identify terms using which searching is done to identify and retrieve relevant documents.

In the searching part, a similar text preprocessing (tokenization, normalization, stop word removal and stemming) techniques is followed just as in the indexing part. Then, probability of relevance based on binary independent model techniques used to retrieve from inverted index file and rank relevant documents accordingly. After ranking the relevant retrieved documents, the users gave a feedback by reformulating the query and restart the search for improved results. Again based on binary independent model, searching systems from the inverted index file can be applied. The users query also expanded to enhance relevant documents retrieval to satisfy information need of the users. Then, the documents are re-ranked in decreasing order based on their probability of relevance. The user reformulates the query by adding new terms in order to enhance the performance of the system and to satisfy the user information need. Based on the reformulated query, the system searches the documents from inverted index file and give a result in decreasing order again. The system gave an option for the user to search relevant documents

up to the user can be satisfied. If the user is satisfied to the obtained documents, the system exit and gave an acknowledgement for the users. Hence,an activity of a cycle could be applied in the architecture designed. The above descriptions are depicted in figure 3.1 below.

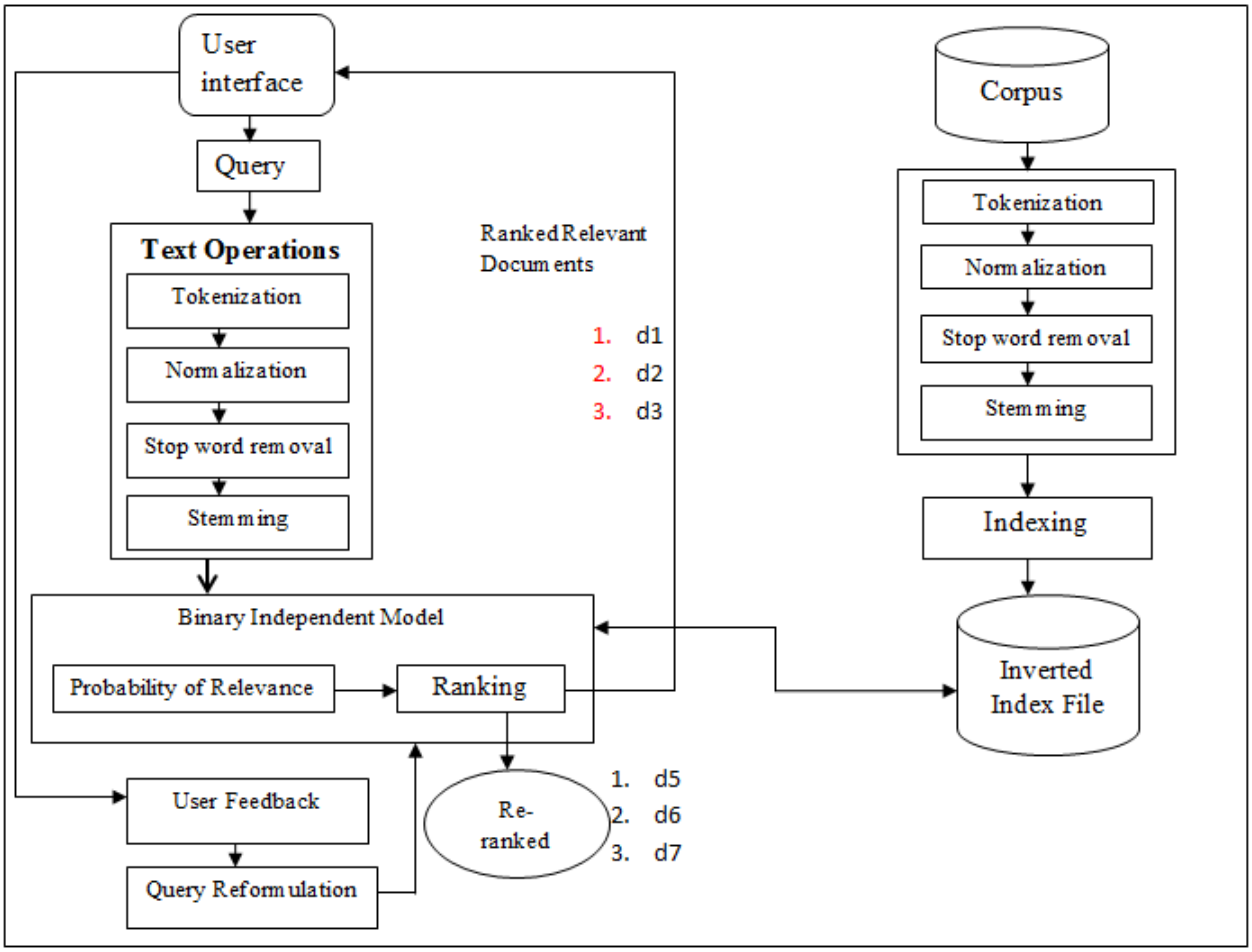

Figure 3. 1:-A probabilistic Based Architecture of Afan Oromo Text Retrieval System (Baeta-Yates et al., 1999)

Again, as shown in the figure 3.1 above, IR process starts with the specifying the problem (user need), then this user need is transformed to some query. The system searches from the inverted index file for such query and get back the result to specify the relevancy of the document to his/her need.

3.0. Corpus Acquisition and Preparation

Since there is not a standard document corpus prepared for the thesis, the corpus of Afan Oromo documents were selected from different sources for the experimentation. In IR system, the corpus is needed for training of the system. Corpus is a large collection of texts. It is a body of

written material upon which is a free raw of texts. Document is collected from different news articles and other online resources, including Oromia Broadcasting Network (OBN), Voice of America (VOA), different websites publishing magazines, newspapers, educational books and fictions to make the corpus variety. For the sake of this study, two hundred (200) documents with average size of 6 MB and 10 testing queries was collected from the puplic via questions (in appendix 5) from random native speakers of the language for training and testing evaluation of the system respectively. To this end, we collected the user query from 20 individuals randomly and selected the 10 most frequently user query in addition to reviewing content of the documents manually. Newspapers are considered as consisting different issues of the community such as social, political, economic, sport, educational, justice, religion and health issues. They are a potential source for collecting corpus, which is not biased to specific issue. This Heterogeneity of the data set help to evaluate the system more generic. Therefore, the reason to widen the coverage of the source is to make it relatively balanced, representative of a language variety and avoid bias. Each document file is saved under common folder using .txt format.

For speeding up searching the document corpus was indexed using inverted index structure. To this end, text operations was applied for identifying content-bearing terms with the help of tokenization, stop word detection, normalization and stemming processes. Given Afan Oromo text corpus, the IR system organize them using index file to enhance searching. The first step is tokenization of the text words to identify stream of tokens (or terms).

Next, text is normalized in order to bring together similar word written with different punctuation marks and variation cases (UPPER, lower or mixed). The normalized token is checked again as it is not stop word. This is followed by removing stop words from the corpus. Content bearing terms (nonstop words) are stemmed. For all stemmed tokens their weight calculated and then inverted index file was constructed.

3.0.1. Tokenization

The corpus which is a set of sentences first tokenized into words. Since, Afan Oromo uses Latin alphabet the sentences can split using similar word boundary detection techniques like the use white space in English. And also, punctuation marks in all the documents will removed except apostrophe (‘) that considered as part of character which is called (“*hudhaa”*) in Afan Oromo. For example, in the word *“re’ee*” (goat), the apostrophe is used to show that the vowels are

produced independently. Thus, the word "*re'ee*" has to be treated as a single token in the tokenization process. Apostrophe can represent as "h" character. For instance, *'ja'a'* which means six written as '*jaha'*. All punctuation marks are converted to space and space is used as a word separation.

Tokenization is useful both in linguistics (where it is a form of text segmentation), and in computer science, where it forms part of lexical analysis. It is the process of splitting on white spaces and throwing away punctuation characters and tokenizes the text, turning each document into a list of tokens (Manning *et al*., 2008). For example, if the original document is: *"Oromiyaan hayyootaa fi beektota hedduu qabdi.*" The tokenized output would be like: "*Oromiyaan", "hayyootaa", "fi*" *„beektota", "hedduu", "qabdi".*

```
Characters= "/=-.,!#$%^&*-();:-\n\t\\\"?!{}[]_ - <>/0123456789/"
def tokenization (document):
        terms =document.split():
        return [term. strip (characters) for term in terms]
```

Figure 3. 2:- Python Code Fragment for Document Tokenization

From the above tokenized documents, below activities are accomplished. Those are as text follows: first, the content of file is read line by line. Second, split them by space in to list of words. Third, check whether the word within the list contains punctuation marks, control characters or special characters of Afan Oromo; if any exist within the word replace it with space. This step continues until end of line is reached.

3.0.2. Normalization

Normalization involves process of handling problem related with variation of cases (UPPER CASE, or lower case or Mixed Cases). So the good way to handle this problem is converting the whole document in to similar case. Often convert to lowercase everything, since most of the time users use lowercase regardless of correct capitalization. For instance, *Kumarraa vs kumarraa, Galma vs galma, FUULA vs fuula.* Hence, all terms in the documents are normalized to understandable as lower case format.

```
for files in os.listdir(“.”):
        text=open(files,”r”)
        rd=text.read()
        string=rd.lower()  #Normalization
```

Figure 3. 3:- Python Code Fragment for Document Normalization

The above fragment code used to normalize documents. Each and every term in the document is converted in to similar case format which is into lower case. After normalize terms, stop words are removed. In the step of stop word removal, the term in the document were checked as it is not part of stop word list. If the word is in the list of stop word, it can be ignored and those terms not appear in the stop word list, they are considered as of stemmed list.

Punctuation marks are usually attached to the words which precede them is also the case before forward to stemming. Removal of punctuation marks is crucial to prepare data for the indexing process. This is because the same words attached to a punctuation mark and not attached to a punctuation mark are considered as different words. For example the word “*haannaa”* and “*haannaa?”* are different words when they are indexed unless the question mark (?) that is attached to the latter is removed.

3.0.3. Stop word removal

After tokenizing our document corpus, we were removed the stop words, since they are no effect on the meaning of words. Not all terms found in the document are equally important to represent documents they exist in. Some terms are common in most documents. Therefore, removing those terms, which are not used to identify some portions of the document collection, is important. For instance, words such as (‘*sana’*, ‘*kana’*), conjunctions (“*’fi’*, ‘*akkasumas’*, ‘*kana malees’*).

According to Greengrass (2000), few terms occur frequently, a medium number of terms occur with medium frequency and many terms with very low frequency. This shows that writers use limited vocabulary throughout the whole document.

```
for files in os.listdir(“.”):
        text=open(files,”r”)
        rd=text.read().split()
        string=rd.lower()  #Normalization
        stopword=open(stoplist.txt,”r”)
        stoplist=stopword.read()
        st=stoplist.lower()
        for i in string:
                if i not in st:#stopwordlist removal
                        keep i
                else:
                        remove i
```

Figure 3. 4:- Python Fragment Code that Removes Stop Words

Afan Oromo stop word list is saved in text file ‘stoplist.txt’. The algorithm reads the files and saves it on variable. Then tokenize and normalize the terms which are available in the files and check the terms whether different from the terms of stop words. Terms not stop word are forwarded to the stemming function.

3.0.4. Stemming

Stemming is a technique to remove affixes from a word, ending up with the stem. It is a normalization step that reduces the morphological variants of words to a common form usually called a stem by the removal of affixes (Manning, 2008). On the other hand, stemming is also used to reduce the dictionary size (i.e. the number of distinct terms used in representing a set of documents). The smaller the dictionary sizes the smaller storage space and processing time required (Manning, 2009).

In Afan Oromo writing system, words are morphologically variants. Morphological variant words have similar semantic interpretations. In IR system, those words considered as equivalent words. Therefore, words have to be reduced to their root using stemming technique. These Stemming techniques are language dependent. Therefore, every language needs to have language specific stemming technique. There are many word variants/affixes. To conflate them into stem word, stemming technique/ algorithm developed by Debela (2010) were used. He developed the stemmer that involves the removal of both prefixes and suffixes using hybrid approach.

In Afan Oromo text, large numbers of suffixes can attach to the end of the words. We architect the retrieval of Afan Oromo text from the system based on the affix removal stemming algorithms. This affix removal includes prefix and suffix of the language, but most text have suffix (Debela, 2010). Hence, for the stemming purpose, we focused on suffix removal algorithm.

```
def thestemmer(terms)>=1:
        if terms.endswith("dhaaf"):
        stem=terms.replace("dhaaf","")
                return stem
        elif terms.endswith("ittii"):
        stem=terms.replace("ittii","")
                return stem
        elif terms.endswith("ummaa"):
        stem=terms.replace("ummaa","")
                return stem
        elif terms.startswith(wal):
        stem=terms.replace("wal","")
                return stem
        elif terms.endswith("oota"):
        stem=terms.replace("oota","")
                return stem
                …
        else:
                return terms
```

Figure 3. 5:- Python Fragment Code of Stemmer

The stemming code creates stemmed words depending on a rule based stemming algorithm (affix removal algorithms). It takes list of index terms and check if the length is greater than or equal to one or not. If the length of the word is less than one, the word is returned without further processing. If the length of the word is greater than or equal to one, the code iterates on word and check characters if they matched with one of the suffix found in suffix list. If the character is matched, the stemmed word is returned. It follows similar procedure to remove prefix. One of the challenges faced when implementing stemming on Afan Oromo words is the algorithm adopted from Debela (2010). Sometimes, the algorithm over stem, words that are inflected using prefix. This results the stem to become too short word. For instance, the word *"hawaasummaa"* meaning 'sociability' and *"hawaa"* meaning 'universe' are over stemmed to *"haw"*.

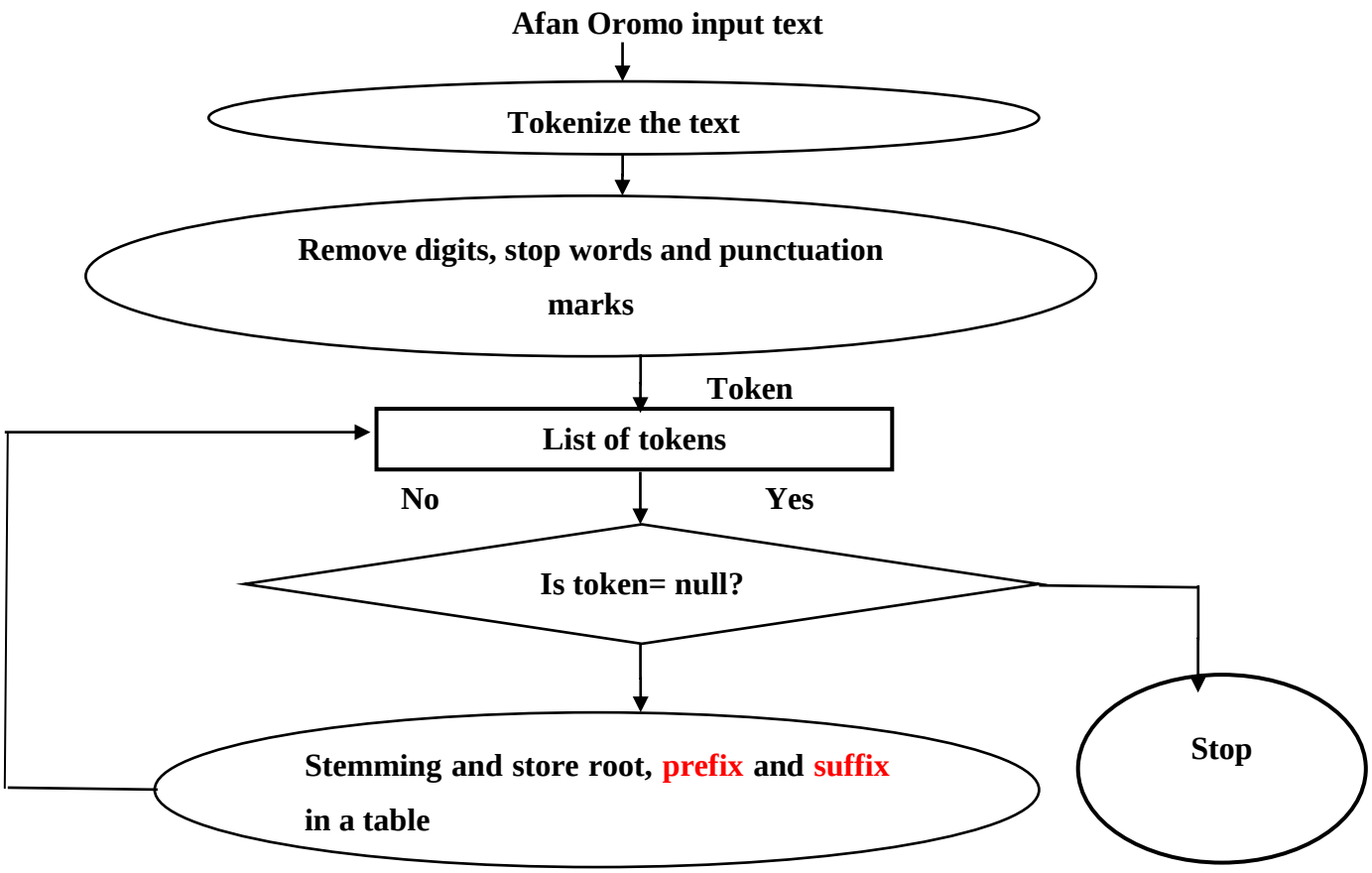

Figure 3. 6:-Architecture of Afan Oromo Stemmer

3.4. Inverted index

The major concept in information retrieval is how the documents/corpus are going to be represented in information retrieval system. This logical representation of documents using its content bearing words is inverted file. An Inverted index always maps back from terms to the parts of a document where they occur. Inverted index, or sometimes inverted file, has become the standard term in information retrieval. The basic idea of an inverted index is a dictionary of terms (sometimes also referred to as a vocabulary or lexicon) (Christopher,2009). So as to have fast retrieval time, we need to build the index in well organized and structured manner.

An inverted index is an optimized data structure that can be used for information retrieval. The basic idea for building an inverted index is to keep a dictionary of the unique terms in the collection. For each term in the collection, we maintain a list of documents (by document IDs) in which the term occurs as well as a number for the term‘s frequency in the specified document. This list is called a posting list. The posting list is stored in the secondary storage, while the dictionary is stored in main memory (Manning, 2008).

As shown in figure 3.1 above, the indexing part of the corpus contains vocabulary file and posting file. The text should undergo several preprocessing operations like tokenization,normalization,stop word removal and stemming before it can be stored in an inverted index. The inverted file allows an IR system to quickly determine what documents contain a given set of words, and how often each word appears in the document. Other information can also be stored in the inverted file such as the location of each word in the text (Heinz, 2003).

3.5. Searching Using Probabilistic Model

The retrieval module is responsible for two main tasks. The first one is the indexing of document collection so as to identify which words are more representative of a given document than the others. Searching is another main task of the retrieval module (Baeza-Yates et al., 1999). In this study, search results were provided for the user in Afan Oromo text. Since this study is based on probabilistic model, probability of relevance between query and document is used.

As briefly described in section 2.3.3 and depicted in figure 3.1, searching for this study was done for the language runs. From the figure, Afan Oromo queries were sent to the search module to look for Afan Oromo documents judged to be relevant for a given queries. During this searching

process if terms in the queries match with any of index terms, then the document identification numbers of the document that contains those terms are returned. Thus, during searching the matching between the index terms and query terms is needed to increase the performance of an IR system by relating different variants of a word.

Because of its capability of handling the uncertain nature of information retrieval, the BIM is used to design probabilistic Afan Oromo IR system. This is because, according to Greengrass (2000), the first step in most of probabilistic methods is to make some simplifying assumption. Thus, BIM is the model that has been used with the probabilistic ranking principle by introducing some simple assumptions which makes estimating the probability function P (R|d, q) practical.

The probabilistic model that attempts to simulate the uncertainty nature of an IR system guides the searching process. Binary independent probabilistic IR model is adopted to search the relevant documents from Afan Oromo corpus. The feedback process is also directly related to the derivation of new weights for query terms and the term re-weighting is optimal under the assumptions of term independence. In addition, it is the first model that has been used in several researches because of its clear and simple mathematical and theoretical assumptions (Baeza-Yates, 1999).

3.5.1. BIM Algorithm

In BIM, there are three steps to compute term probability. The first step compute terms when there is no retrieved document at initial stage. The second step compute terms after documents are retrieved and feedback is provided by the user. The third step compute terms when partial feedback is given (Neto, 1999). The assumptions made in this step are:

• P (ki|R) is constant for all index terms k (usually, it is equal to 0.5)

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

These two assumptions will give as;

$$\boldsymbol{P(k_i|R) = 0.5 \ and \ P(k_i|R) = log\left(\frac{N-n_i+0.5}{n_i+0.5}\right)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (\mathbf{3.1})$$

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

Using this initial guess, documents are retrieved which contain query terms and provide an initial probabilistic ranking. After documents are retrieved, the user looks at the retrieved documents and marks them as relevant and non-relevant. The system then uses this feedback to refine the description of the answer set. At this stage, initial ranking is shown and more discriminating information about terms is available this will allow more accurate estimation. Therefore, relevant documents retrieved should be improved using probabilistic relevance weighting technique. This technique uses the concept in term incidence contingency (Fuhr, 1999).

Table 3. 1:-Term incidence contingency table

	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

Where,

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

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

$$\boldsymbol{P}(\boldsymbol{ti}|\boldsymbol{R}) = \left(\frac{\boldsymbol{r}}{\boldsymbol{R}}\right) \ldots\ldots\ldots (\mathbf{3.2})$$

$$\boldsymbol{P}(\boldsymbol{ti}|\boldsymbol{R}) = \left(\frac{\boldsymbol{n}-\boldsymbol{r}}{\boldsymbol{N}-\boldsymbol{R}}\right) \ldots\ldots\ldots (\mathbf{3.3})$$

As Robertson (1976) noted, the above equations can be rewritten to compute term presence weighting function as;

$$w = log \frac{r(N-n|-R+r)}{(R-r)(n-r)} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (3.4)$$

However, Robertson (1976) noted different assumptions lead to a different formula for computing term weighting. They argue "in practice users may find themselves in the situation where, even if they know some relevant documents are retrieved, they wish to continue searching". They assume that "users may not found all the relevant documents that would satisfy their need". Therefore, the record in the center of the contingency table (i.e. N – n – R + r) may not be taken as absolute. The estimation of document relevance when considering new items has to allow for uncertainty. This estimation adds 0.5 to all the central record and it derives a specific term relevance weighting formula; - Atalay (2014), citing Robertson (1976)

Relevance Weighting

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

3.6. Evaluation Method

As it was described in section 2.5, IR evaluation is highly related to the concept of relevance. Relevance is the degree of correspondence between retrieved documents and information need of users. Since individuals need is varying, relevance is subjective. Often peoples look relevance differently. What is relevant to somebody might be irrelevant to others. Nature of user query and document collection affects relevance. Additionally relevance depends on individual's personal needs, preference subject, level of knowledge, specialization, language (Greengrass, 2000).

The experimentation for evaluating the effectiveness of the system is done by using two hundred (200) text documents and ten (10) queries were selected. Queries were formulated and relevance judgment is prepared to construct document-query-matrix that shows all relevant documents for each test queries. To assess the effectiveness of the proposed system (i.e., the quality of its search results) the most frequent and basic statistical measures, recall, precision and F-measure are used. Precision is the fraction of the documents retrieved that are relevant to the users" information need; recall is the fraction of the documents that are relevant to the query that is successfully retrieved and F-measure is the mean of precision and recall (Greengrass, 2013).

Specifically, in this work the interpolated precision value at **11 standard recall levels** is used to draw precision-recall curve in order to evaluate retrieval effectiveness of the system.

Based on the concept of relevance, there are several techniques of measures of IR performance available, such as, precision and recall, F-measure, E-measure, MAP (Mean average precision), R-measure. In this study, the three widely used techniques precision, recall, and F-measure was used to measure the effectiveness of the IR system designed (Greengrass, 2013).

Recall is percentage of relevant documents retrieved from the database in response to users query, whereas precision is percentage of retrieved documents that are relevant to the query and F-measure is a single measure that trades-off precision versus recall. Also, there is always trade-off between precision and recall. If every document in the collection is retrieved, it is obvious that all relevant documents are retrieved, so that recall will be higher (Greengrass, 2000).

The recall, precision and F-measure can be calculated using equation 3.6, 3.7 and 3.8 respectively using information from table 3.2

Table 3. 2: -Retrieved versus relevant documents

	Relevant	Not relevant
Retrieved	A	B
Not retrieved	C	D

Collection size = A + B + C +D

Relevant = A + C

Retrieved = A + B Therefore,

$$\mathbf{Recall} = \frac{|\mathbf{Relevant}| \cap |\mathbf{Retrieved}|}{\mathbf{Relevant}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{3.6})$$

$$\boldsymbol{Precision} = \frac{|\boldsymbol{Relevant}| \cap |\boldsymbol{Retrieved}|}{\boldsymbol{Retrieved}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{3.7})$$

$$\boldsymbol{F - Measure} = \frac{\mathbf{2} * \boldsymbol{Recall} * \boldsymbol{Precision}}{\boldsymbol{Recall} + \boldsymbol{Precision}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{3.8})$$

The above formula for precision and recall assume that, all documents retrieved is examined by user manually. Thus, the retrieved document (A + B) cannot be presented for the user at once. Rather, the retrieved documents are presented according to their degree of relevance as per the user query. Then, the user examines the ranked documents starting from the top. This kind of examination of documents by the user leads the recall and precision measures to vary. Therefore, for appropriate evaluation of recall and precision, plotting a precision versus recall curve is necessary (Neto, 1999). To draw precision-recall curve, for example let documents retrieved by the system is Dr Where Dr = {d3, d33, d9, d1, d10, d11, d14, d7, d44, d49, d50, d53, d55, d77, d133} in ranked order. Assume Rq contain a set of relevant documents for the query. Where, Rq = {d1, d3, d7, d10, d14, d33, d44, d49, d55, d133}. After the recall- precision curve is constructed, based on the original recall and precision may result in saw tooth curve. However, when constricting the curve based on the original recall and precession may result in saw tooth curve, there is a need to smooth the curve using interpolation technique.

The recall levels for each query might be different from the standard recall levels, which is difficult to compare performance across queries. Therefore, interpolation procedure is necessary. As a result, a single precision value for each query can be used that takes a precision at some recall level for each single query. The interpolated precision versus recall curve is shown as follows; Let rj, j E {0, 1,2, ... , 10}, be a reference to the jth standard recall level. Then, P(rj) = max rj ≤ r≤rj+1 P(r), “which states that the interpolated precision at the jth standard recall level is the maximum known precision at any recall level between the jth recall level and the (j + l)th recall level.

In general, precision and recall have been used widely so as to evaluate information retrieval system performance. However, they measure two different aspects of the system and thus they are inversely relative. If recall of a system is improved then the precision is reduced. The reason behind this is that, when attempt is made to include many of the relevant documents, irrelevant documents more and more exist in the answer set. On the other hand, if precision of a system is improved then the recall is reduced. This is because; there are retrieved relevant documents among the whole relevant documents found in the corpus. Achieving both precision and recall 100% is ideal, but impossible (Baeza-Yates et al., 1999).

Several problems have been distinguished. First, to make appropriate estimation of maximum recall for a query, it needs deep knowledge of all the documents in the collection. Second, even if many situations consider the use of a single measure, which combines both, recall and precision capture different aspects of the set of retrieved documents. One of the methods developed to alleviate the above recall and precision problems is the F-measure (Baeza-Yates et al, 1999).

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

CHAPTER FOUR
EXPERIMENTATION AND DISCUSSION

This chapter presents the experiments conducted and findings of the study. In this study, an attempt has been made to design a probabilistic IR system for Afan Oromo text. The system has both indexing and searching parts as in Figure 3.1. Inverted file indexing structure is used to organize documents so as to speed up searching. The main objective of this study is to design Afan Oromo IR system using probabilistic approach by adding user feedback to improve the performance of the system. To achieve this objective, we have shown the result by experiment. Finally, the result of the experimentation has also discussed.

4.1.Corpus and Query Preparation

As presented in section 3.0, the lack of standard corpus has led the researcher to explore the use of free-text-documents to perform probabilistic text retrieval system. Hence, finding a large size and standard corpus for Afan Oromo text is one of the challenges faced in this study.

As shown in table 4.1 below, the news articles contain nine clusters of news, which are health, education, Religion, social, economy, culture, sport, politics, and justice. The state-of-the-art in the area of text processing indicates that, there is no any developed standard corpus for Afan Oromo text. For the purpose of this study, a corpus with two hundred (200) short documents were used. Hence, experiments in this study were based on sets of documents and queries set up by the researcher.

Table 4. 1:- Types of news article used for development of Afan Oromo IR system

No	Types of news	Number of documents
1	Health	15
2	Education	20
3	Religion	20
4	Social	25
5	Economy	25
6	Culture	30
7	Sport	30
8	Politics	25
9	Justice	10
Total		200

Each news articles are saved under common file folder using .txt format in notepad, which is supported by most programming languages. Additionally, 10 test queries were selected by the researcher to test the performance of the system after collecting user query from the puplic via questions from twenty individuals of the native speakers and reviewing the corpus collected. These queries are marked across each document-query-matrix as either relevant or irrelevant to make relevance evaluation as in appendix one (1) for each document. Subjective relevance judgment is also done for identifying which document is relevant for a given test query. However, unlike vector space model, which obtain frequency of each query terms in the documents, probabilistic model obtain the absence and presence of the query terms from the documents. In addition to that, queries are weighted two times (i.e. before and after relevance feedback is given) using the probabilistic weighting assumption as presented in section 3.4. Having an identified query is very important in order to evaluate the performance of the system. Based on the selected Afan Oromo corpus, appropriate queries which are able to describe the documents were selected subjectively by the researcher after reviewing the content of each article manually and collecting questions from 20 individuals randomly. For this performance evaluation purpose, the selected queries are given in table 4.2. Table 4.2 shows an example of document-query-matrix constructed for weighting query before relevance feedback.

Table 4. 2:-List of queries with their relevant judgments

No	Queries	Relevant	Non-relevant	Relevant documents retrieved
1	*Rakkoo Fayyaa Maatii*	15	185	15
2	*Qulqullina Barnootaa Mirkaneessuu*	20	180	17
3	*Bu'uura amantii ilma namaa*	20	180	16
4	*Hirmaannaa Ummataa*	25	175	16
5	*Misooma biyyaa*	25	175	17
6	*Meeshaa Aadaa Hawaasa Oromoo*	30	170	29
7	*Ispoortii atileetiksii itoophiyaa*	30	170	30
8	*Aangoo mootummaa*	25	175	22
9	*Qaamolee haqaa naannoo keenyaa*	10	190	7
10	*Shaakala dorgommii atileetiksii kilaboota kubbaa miilaa*	30	170	25

From the table 4.2 above, the relevance judgment of each document is identified to be forwarded for each query terms. For each query terms there are at least seven relevant documents and a maximum of 30 relevant documents as presented in table 4.2.

4.2. Indexing

The prototype system development begins with constructing inverted file or vocabulary file, which is usually called indexing. Once the inverted file is constructed, the query is supplied to the system. The system has been built using python version IDLE 3.6.5. In the preprocessing stage, the study addresses tokenization, normalization, stop word removal and word stem (stemming) of Afan Oromo documents. After preprocessing stage, the indexing is done. Indexing involves text pre-processing and creating inverted file structure which includes vocabulary file and posting file. Both searching and indexing are built up of different sub components.

The indexing process is implemented to construct inverted index. As discussed in section 2.2.1, when implementing inverted index, there are several tasks need to be done. Such as tokenization, normalization, stop word removal and stemming. Also, as discussed in section 3.0 in briefly, the first step in constructing inverted index is generating index terms from document collection. In this step, the first task is tokenizing terms. The inverted file has two separates files vocabulary and posting file; the vocabulary file contains Terms, Doc frequency and Collection frequency and illustrated in table 4.3 below or figure 4.1 and posting file contains documents ID, Term frequency and terms location which is also illustrated in the table 4.4 below. That depicts structure of inverted file (vocabulary file and posting file). The document frequency and collection frequency is cross referenced to posting file.

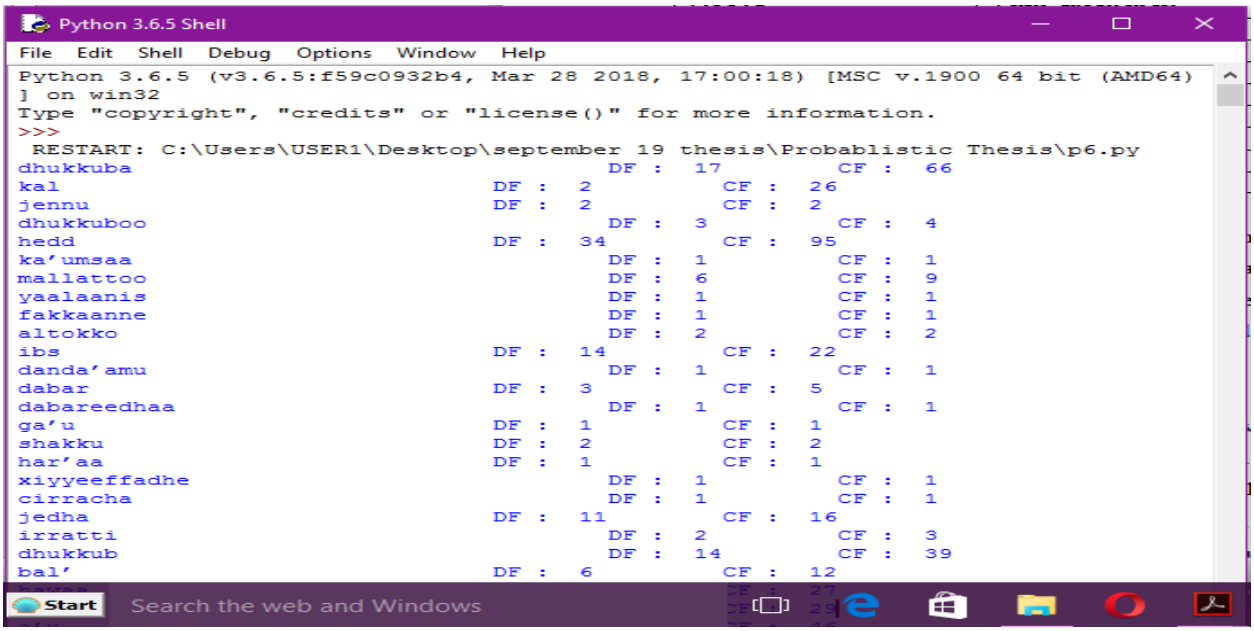

Figure 4. 1:-Sample Snapshot for Vocabulary File

From the above figure 4.1, terms are retrieved with their document frequency (DF) and collection frequency (CF).

Table 4. 3:-vocabulary file

Vocabulary file		
Terms	Doc frequency	Collection frequency
Dhukkuboota	2	3
Mallattoo	2	4
Fakkaannee	1	2
Hawaa	3	3

Table 4. 4: - Posting File

posting file		
Doc id	Term frequency	Term location
d1	1	79
d15	2	4079, 5746
d98	3	9356, 1443, 4816
d138	1	4577
d145	2	2722, 3210
d40	1	3433
d55	1	9985
d160	1	2220

The searching component is written with a python code which helps to implement probabilistic model. Query text pre-processing is done in similar way to indexing part. After the corpus is indexed, the probability of relevance is computed between the documents and query. The user information needs are formulated using combinations of terms. In searching step, as indexing, all preprocessing were done. Each query term should pass through each of these processes as it was elaborated in Figure 3.1 on the architecture of Afan Oromo text retrieval system. In this step, based on their probability of relevance, ranking is done.

As shown in table 4.3 and table 4.4, the vocabulary file which is created in indexing step holds non-stop word terms in the document with its collection frequency and document frequency. And also, Posting file holds terms, name of each document holding the term (document id), term frequency and positions of the terms in the documents. So, the search algorithm brings together these things in order to calculate probability of relevance and rank the documents. For instance, the term "*dhukkuboota*" is available in two documents such as in d1 and d15. This term occurs one times in d1 and two times in d15 as presented in table 4.3. Hence, the collection frequency of this term is three with document frequency of two. As can be seen from table 4.4 posting file contains docid, term frequency and location of the term. Therefore, the term "*dhukkuboota*" is available in document d1 and d15 with one term frequency in d1 and two term frequency in d15 that is located at 79, 4079, and 5746 respectively. It is basic to find the relevant documents from the collection by calculating their probability between query and documents in the collection.

4.3.Evaluation Method

As described in section 2.5 and section 3.5, Evaluation of IR system is highly related to the relevance concept. Relevance is the degree of correspondence between retrieved documents and users information needed. The users often look relevance from different sides. What is relevant to some users might be irrelevant to other users. Nature of user query and document collection affects relevance. Additionally relevance depends on individuals personal needs, preference subject, level of knowledge, specialization, language, etc. (Christopher, 2009).

Recall is ratio of relevant items retrieved to all relevant items in the corpus. There is always trade-off between precision and recall. If every document in the collection is retrieved, it is obvious that all relevant documents are retrieved, so that recall is higher. In contrary when only little proportion of the retrieved document is relevant to given query, retrieving everything

reduces precision (even to zero). The higher score in both recall and precision means the higher the performance of the system (Christopher, 2009).

There are different techniques for measuring effectiveness of IR system. In many IR systems, recall and precision are considered as basic measures for retrieval effectiveness (Manning et al., 2009). Recall measures the ability of a retrieval system to find out relevant documents. It considers how many percentages of relevant documents are correctly retrieved by the system. Using recall alone is not enough when measuring the effectiveness of a retrieval system since retrieved documents contains both relevant and irrelevant documents (Hull, 1996). Therefore, precision which measures the ability of a retrieval system to find out only relevant documents is also essential (Baeza, 2004). In this study, Interpolated average precision is also used.

4.4. Performance Evaluation

A result of an experiment is conducted with ten (10) queries and two hundred (200) corpus documents to retrieve relevant documents. The relevance judgments were prepared to construct document-query-matrix that shows all relevant documents for each test query prepared as shown in appendix one (1). The performance of the prototype system is evaluated before and after user relevance feedback of a text. Each test query is measured by precision, recall and F-measure. As stated earlier, the experiment has two phases. The first phase is retrieving relevant documents before relevance feedback while the second phase again retrieves after user relevance feedback by adding terms to the user query until the user is satisfied. The results are given in figure 4.2 below. Since it's not feasible to list all of the results of each query here, we calculated average recall, precision and F-measure. Also interpolated average recall precision graph is used to show the effectiveness of the system at the 11 standard recall levels. Since the recall level for queries may vary from the standard recall levels, interpolation have been used.

The performance of the system is evaluated before and after relevance feedback using ten (10) queries. Based on the performance registered, an attempt has been made to compare the result of probabilistic based IR system for Afan Oromo with the previously done Afan Oromo IR system using vector space model. Using the information given in table 4.2, evaluation is done by measuring the recall, precision and F-measure for the initial performance of the system. The above table 4.2 presents, relevant documents retrieved from the Afan Oromo corpus for each test query.

The first step of this system is to get query from the user. Figure 4.2 presents a screen shoot which shows list of retrieved documents using the query '*qulqullina barnootaa mirkaneessuu*'.

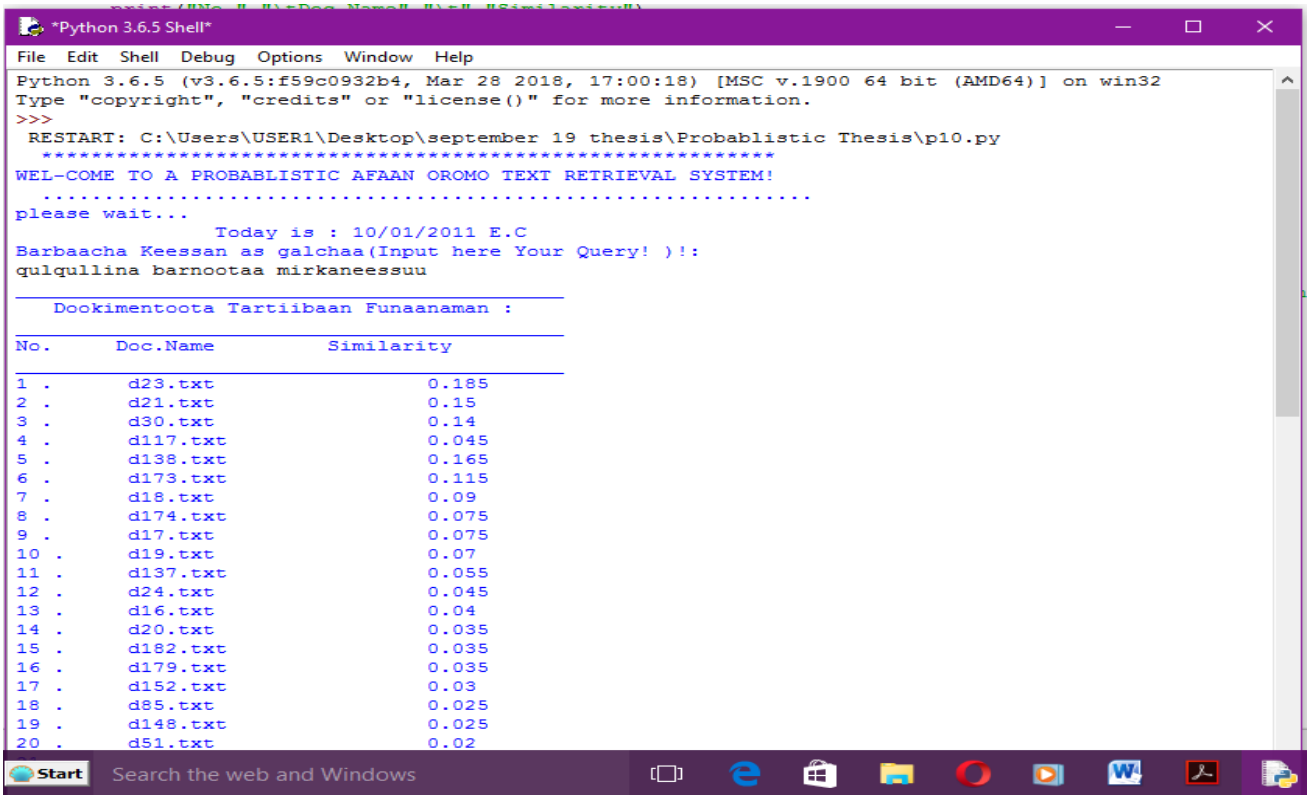

Figure 4. 2:- Retrieved documents for a given query *'qulqullina barnootaa mirkaneessuu'*

From the above figure 4.2, documents are retrieved by BIM based on probability of relevance for the users information need. For instance; as can be seen in figure 4.2, **d23.txt** is the first top ranked document retrieved based on the user query. From here, the users enter the query, and then the system retrieves the relevant documents for the users query. Even though all relevant documents are not seen on the snapshot figure, the most top twenty relevant documents retrieved are depicted on the figure. Accordingly, **d23.txt, d21.txt** and **d30.txt** are the first, second and third ranked relevant documents as depicted in figure 4.2 respectively.

The above figure 4.2 shows the result of this process. For the query '*qulqullina barnootaa mirkaneessuu'*, it retrieves 60 documents, out of them 17 documents were relevant. However, in the corpus, there were twenty (20) relevant documents for the query. Table 4.5 below shows, the effectiveness of a probabilistic Afan Oromo IR system based on 10 queries selected for the experiment.

Table 4. 5:- The Initial Performance of the System

No	Query	Retrieved	Relevant	Rel-retrieved	Recall	Precision	F-measure
1	*Rakkoo Fayyaa Maatii*	97	15	15	1	0.155	0.268
2	*Qulqullina Barnootaa Mirkaneessuu*	60	20	17	0.85	0.283	0.425
3	*Bu'uura amantii ilma namaa*	50	20	16	0.8	0.32	0.46
4	*Hirmaannaa Ummataa*	34	25	16	0.64	0.47	0.54
5	*Misooma biyyaa*	83	25	17	0.68	0.2	0.309
6	*Meeshaa Aadaa Hawaasa Oromoo*	84	30	29	0.97	0.345	0.51
7	*Ispoortii atileetiksii itoophiyaa*	82	30	30	1	0.375	0.545
8	*Aangoo mootummaa*	91	25	22	0.88	0.24	0.377
9	*Qaamolee haqaa naannoo keenyaa*	30	10	7	0.7	0.233	0.346
10	*Shaakala dorgommii atileetiksii kilaboota kubbaa miilaa*	29	30	25	0.83	0.86	0.845
Average					0.835	0.3481	0.4914

As table 4.5 shows, the retrieval result of the prototype on the average Precision, Recall and F-measure is 0.3481, 0.835 and 0.4914 respectively. The result shows the system retrieved most of the relevant documents in the collection out of the total relevant documents in the corpus. However, the result of the precision indicates that, the non-relevant documents retrieved are higher than the relevant documents retrieved. This is because; documents containing one of query terms but not-relevant were retrieved. For this reason, the result of precision is very low performance. And also the harmonic mean of recall and precision is low which indicates that the performance of the system is not satisfactory.

In this study, the systems only retrieves 83.5% relevant documents for a given 10 queries. During the experimentation, there were irrelevant documents retrieved from the corpus and relevant documents not retrieved. For example: for query *'aangoo mootummaa'* which express the power of government, 91 documents were retrieved, from these retrieved documents, only 22 documents were relevant for the query, but there were 25 relevant documents in the corpus. The system retrieves 69 irrelevant documents because they contain query terms '*aangoo*' but the query terms have not the same meanings, those 69 query terms express the tem related to study.

Afan Oromo terms are highly inflected for number, genders, possession, plural, and conjunctions. There are many terms with the same meanings (synonyms) and a term which has many meanings (polysemous) is another reason for the result. For example, term '*aangoo*' have the meaning with '*taayitaa*' which is 'Authority'. In addition, spelling error is also another factor for being lower performance. For instance; for the word *'barnoota'*, if you miss n and write '*baroota'* it is completely changed. '*barnoota'* means education, but, '*baroota'* means years.

On the other hand, in probabilistic model, the initial guess of relevant document is based on Boolean expression. Thus, all terms that match one of user queries will be retrieved which increases the number of denominator used for calculating precision, thereby decreasing the percentage of precision. Therefore, in order to increase the performance of the system, the probabilistic model uses relevance feedback from the users so as to apply query terms reweighting and the weight of terms found in relevant documents and decrease the weight of terms found in non-relevant documents.

As a general, the performance of the evaluation result is low. But in the user relevance feedback, , the queries were allowed to be expanded and re-formulated to get a good result. The result of user relevance feedback for recall and precision was 0.9156 and 0.60 respectively. Since all queries may not exactly have the standard recall levels, we used interpolation to calculate the average recall precision to show the overall performance of the system across queries and the interpolated Average precision at the 11 standard recall levels. Not only in the first experimental result, but also in second experimental result, not all relevant documents were retrieved. The reason of this problem was presence of polysemy and synonyms, word variants.

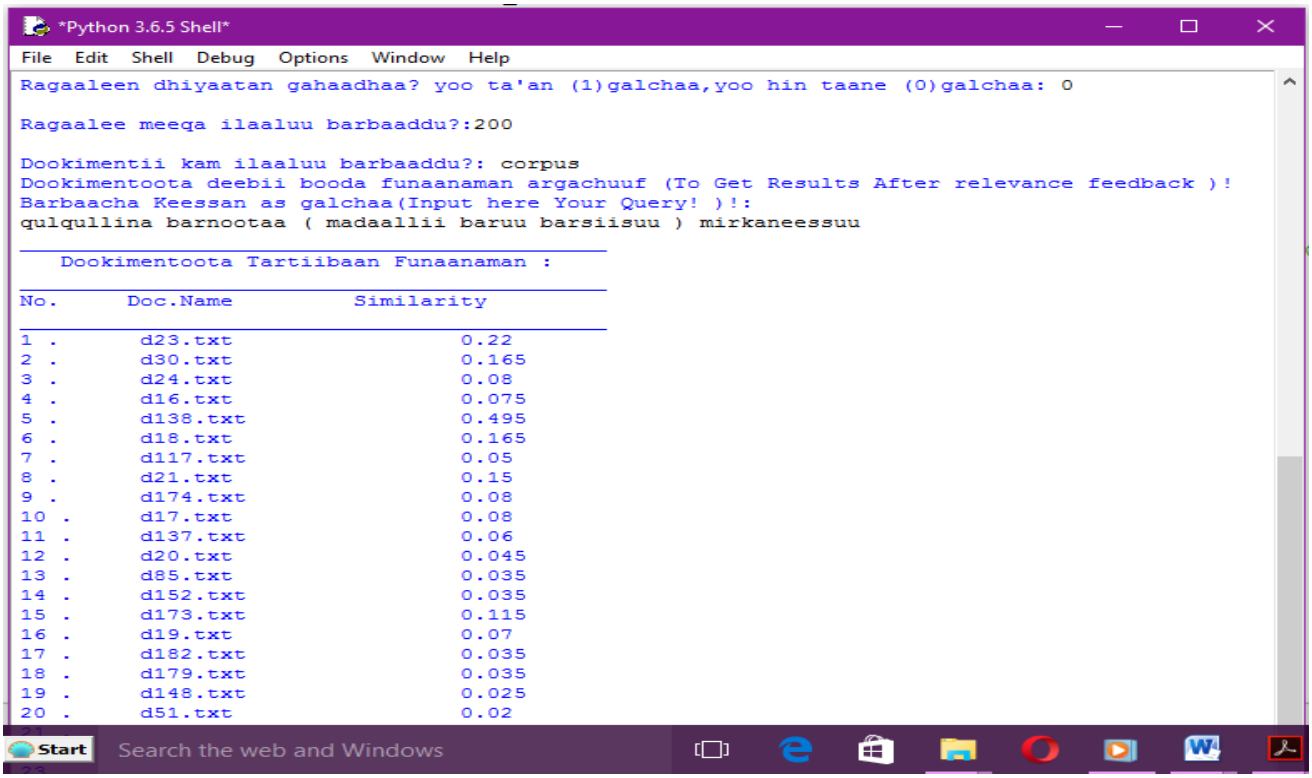

Figure 4. 3:-Retrieved Documents after Relevance Feedback for *"qulqullina barnootaa mirkaneessuu"*

As it was discussed earlier, the most top twenty relevant documents are retrieved and re-ranked based on the re-formulated query. From figure 4.3, **d23.txt, d30.txt** and **d24.txt** is the first, second and third relevant documents in decreasing order of their relevance to the user query respectively. In the figure 4.2, **d23.txt, d21.txt** and **d30.txt** is the first, second and third relevant documents. From this discussion, user relevance feedback, **d30.txt** is the third retrieved document. But in case of after relevance feedback, **d30.txt** is the second ranked relevant documents retrieved in ranking order in order to minimize the time spent by users. In probabilistic model, the ranking for relevant retrieved documents is based on estimation. It does not depend on the term weighting like term frequency and inverse document frequency.

Table 4. 6:- The Performance of the System after User Relevance Feedback

No	Query	Retrieved	Relevant	Rel-retrieved	Recall	Precision	F-measure
1	*Rakkoo Fayyaa Maatii*	25	15	14	0.933	0.56	0.7
2	*Qulqullina Barnootaa Mirkaneessuu*	31	20	19	0.95	0.613	0.7452
3	*Bu'uura amantii ilma namaa*	32	20	18	0.90	0.5625	0.693
4	*Hirmaannaa Ummataa*	41	25	22	0.88	0.537	0.667
5	*Misooma biyyaa*	44	25	22	0.88	0.5	0.64
6	*Meeshaa Aadaa Hawaasa Oromoo*	47	30	30	1	0.64	0.781
7	*Ispoortii atileetiksii itoophiyaa*	57	30	30	1	0.5264	0.69
8	*Aangoo mootummaa*	33	25	22	0.88	0.667	0.76
9	*Qaamolee haqaa naannoo keenyaa*	16	10	8	0.8	0.5	0.62
10	*Shaakala dorgommii atileetiksii kilaboota kubbaa miilaa*	32	30	28	0.933	0.875	0.903
Average					0.9156	0.60	0.725

As shown on the architecture of the prototype in figure 3.1, the user relevance feedback is tested. Table 4.5 and Table 4.6 depicts summery of recall and precision of the final result for Afan Oromo queries before and after relevance feedback that is applied.

By comparing table 4.5 and table 4.6, the performance of the system is increased with an improvement of 23.33% F-measure. In this stage, the system retrieved documents that are judged as relevant by the user and other documents nearer to the judged relevant documents. However, all of the documents judged as relevant by the user are not retrieved because; the weight of the query terms found in those documents is decreased.

Precision and recall are set-based measures (Atalay, 2014). That is, they evaluate the quality of an unordered set of retrieved documents. To evaluate ranked lists, precision can be plotted against recall after each retrieved document as shown in the figure 4.4. From figure 4.4 for before relevance feedback, the exact recall points are 0.6, 0.7, 0.8, 0.9 and 1. When a relevant document is not retrieved at all, its precision is assumed to be 0. Hence, the maximum precision value for figure 4.4 is 0.9. However, recall is difficult for large collections (web retrieval).

As it is discussed in the table 4.6 above, user relevance feedback changes and expands the original query term into its possibility of a term available. The result of this run showed better result of recall and precision. The interpolated average recall precision curve (graph) is given in figure 4.4 below. The results from the first and second experiment were discussed above. For ease of understanding and comparison, the retrieval effectiveness of the system in retrieving Afan Oromo documents of the two experiments is shown in figure 4.4 below.

The performance of the system is evaluated before and after relevance feedback for free-text-documents. As can be seen on the architecture of the prototype in figure 3.1, the system is designed with user relevance feedback and re-formulates the query. From table 4.6, the researcher used user relevance feedback to enhance the effectiveness of the system.

Obtaining the value of precision at standard recall levels for each available 10 queries is important so as to show the performance of the system. Thus, interpolation of precision/recall curve is done. Figure 4.4 shows the interpolated recall-precision curve to depict the performance of the designed prototype system before and after relevance feedback is given. All the documents that were retrieved by the system for a given query may not be relevant. Therefore, relevance of test documents for each test queries were determined during query preparation time, based on the user judgment.

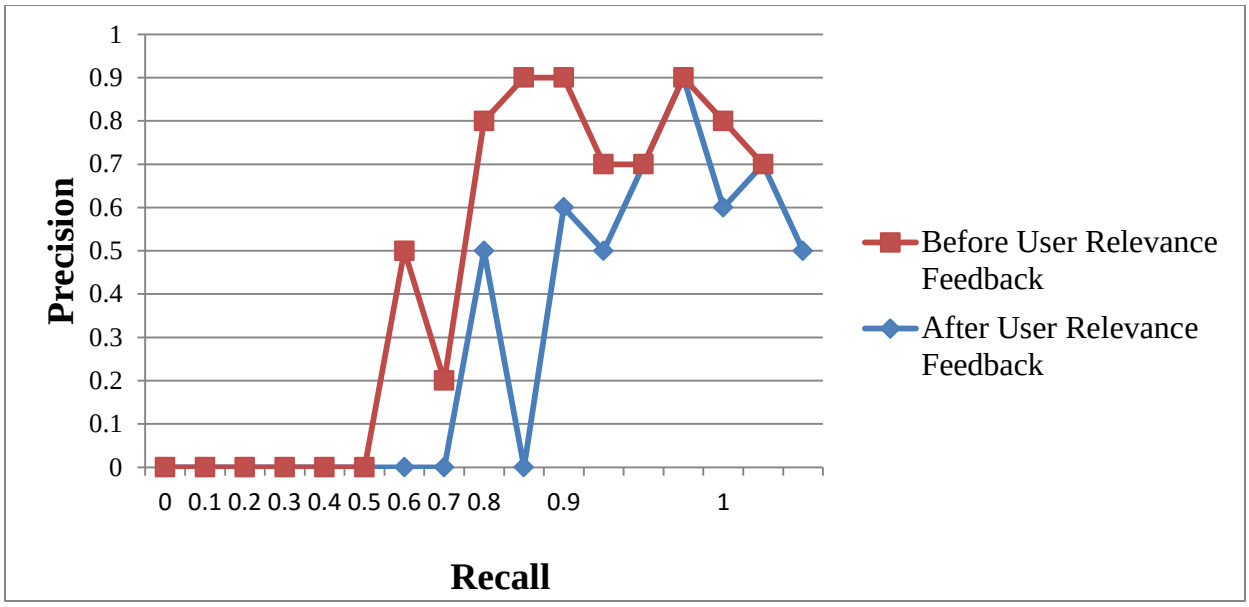

Figure 4. 4:- Precision/Recall curve before and after relevance feedback

As shown in the Figure 4.4 above, the performance of the system registers better performance when the user provides relevant feedback. Vertical and horizontal line of the figure 4.4 above shows that, the result of precision and recall respectively. The curve at the upper side of the graph in which recall and precision reaches maximum point indicates, the highest performance registered by the system. The maximum precision registered is 0.9, before and after relevance feedback at recall level of 0.8 and 0.9 respectively. From this point, the value of maximum precision is same before and after feedback. But the value of recall level is increased by 0.1. This represents the average performance registered by the system. From this figure, the minimum precision value before and after user relevance feedback is 0.2 and 0.5 respectively. Hence, the minimum value of precision after user relevance feedback is increased by 0.3.

4.4.1.Relevance feedback

The users tend to ask short queries, even when the information need is complex. Irrelevant documents are retrieved as answers because on the ambiguity of the natural language (words have multiple senses) (Baeza-Yates, 1999). If we know that some of retrieved documents were relevant to the query, terms from those documents can be added to the query in order to be able to retrieve more relevant documents. This is called relevance feedback. Often, it is not possible to ask the user to judge the relevance of the retrieved documents. It assumes the ten (10) retrieved documents are relevant and use the most important terms from them to expand the

query and document relevance is reweighted again then ranks in decreasing order. Retrieving all relevant documents or achieving a 100% level of recall has not been accomplished by existing information retrieval systems. The problem of limited recall has been recognised as the major difficulty in information retrieval systems. Probabilistic retrieval treats the relevant and non-relevant set equally in reweighting the query (Manning, 2008).

4.5. Results and Discussion

In the experiment, we presented on Afan Oromo text retrieval by probabilistic model approach by using ten queries that have a number of relevant documents in the data collections. We calculate the percentage of recall and precision for each output. We described the experiment conducted to compare the performance of the experiment. The obtained result shows that the result obtained enhances the system performance. The performance of the system is evaluated before and after user relevance feedback. Using the information given in table 4.5 and table 4.6, evaluations was done by measuring the recall, precision and F-measure of each test query with averages of them before and after user relevance feedback.

The results before relevance feedback as indicated in table 4.5 registered low performances with their registered value for recall, precision and F-measure of 83.5%, 34.81% and 49.14% respectively. The F-measure score registered 49.14% for Afan Oromo text, which indicates the performance of the system is not good. This is because; documents containing one of the query terms but that are not-relevant are retrieved. These documents are irrelevant because, the query term found in those documents does not express the meaning of the query with respect to other terms found in the query. On the other hand, in probabilistic model the initial guess of relevant document is based on Boolean expression. Thus, all terms that match one of user queries were retrieved which increases the number of denominator used for calculating precision, thereby decreasing the percentage of precision. Therefore, in order to increase the performance of the system, the probabilistic model uses user relevance feedback so as to apply query terms re-weighting in order to increase the weight of terms found in relevant documents and decrease the weight of terms found in non-relevant documents.

For each of the parameters precision, recall and F-measure, the user relevance feedback registered better value than that of before relevance feedback. Precision increases by 25.19%, recall increases by 8.06% and F-measure increases by 23.36%. The summary of this discussion is given in the table 4.7 below. The table below shows the difference of the result achieved before and after user relevance feedback for each parameter.

Table 4. 7:-Summarized result of the overall performance of Afan Oromo IR

Measures	Before relevance feedback	After relevance feedback	Difference
Recall	0.835	0.9156	0.0806
Precision	0.3481	0.60	0.2519
F-measure	0.4914	0.725	0.2336

The table 4.7 above shows, the system achieved high recall, low precision and F-measure. The reason for high recall and low precision is synonym and polysemy of words in the corpus. For example, the first retrieved results before relevance feedback and the retrieved results after relevance feedback, the recall of "*misooma biyyaa*" is 0.68 and 0.88 respectively. Recall (R) measures the ability of the search to find all of the relevant items in the corpus (Manning, 2008) whereas precision (P) measures the ability to retrieve top-ranked documents that are mostly relevant and F-measure (F) is the harmonic mean of the two measures that is Precision and Recall. As we observed from table 4.7, the natures of the model in using relevance feedback, after user relevance feedback of the results were improved.

As can be easily understood from the comparisons, before and after user relevance feedback scores of recall, precision and F-measure in table 4.5, after user relevance feedback has achieved the best performance with 91.56%, 60% and 72.5% for recall, precision and F-measure respectively. As a result, the second experiment is better than the first experiments. In general, the results found are encouraging. In both experiments, not all relevant documents were retrieved. The performance of information retrieval is highly depend on the size and quality of the corpus. Hence, one reason for low performance is the size and quality of the corpus used.

Generally, one of the major objectives for conducting our experiment was to identify and determine the effects of stemming on Afan Oromo IR system. As it can be easily observed from summarized statistics presented in table 4.7, our model which has a relevance feedback and re-formulates the user query has performed much better than the initial result. This implies the fact that, a user feedback that is designed in figure 3.1 is effective and useful in a development of Afan Oromo text retrieval system.

From table 4.6, the performance result is 0.9156(91.56%) for recall and which is increased by 0.0806(8.06%), 0.6(60%) for precision and it is increased by 0.2519(25.19%) and 0.725(72.5%) for F-measure and it is increased by 0.2336(23.36%) when it is compared with the initial retrieved documents before relevance feedback. This indicates that relevant documents in the collection were almost retrieved except some document. Finally, the harmonic mean of recall and precision is increased. This indicates that the performance of the system is increased after user relevance feedback because of the recall and precision was increased. The precision result is lower, because of morphological variation in word form and the way of Afan Oromo writing system. Afan Oromo can use *ummata* or *uummata* at any place in the words which means people. There are many different ways of Afan Oromo text writing. For example, in query 4('*hirmaannaa ummataa'*), the registered performance of recall and precision is 88% and 53.7 % respectively. The reason is writing the term *'ummataa'* in different ways which is people in English. In query 6('*meeshaa aadaa hawaasa Oromoo'*), the registered performance for recall and precision is 100% and 64% respectively. The reason is writing the term *'hawaasa'* in different ways which is social in English. It can be written as '*ummataa','uummataa*' and '*hawaasa*','*hawwaasa*'. Hence this creates an ambiguity and only a few documents were retrieved. It should be handled by semantic similarity. Therefore, when the total number of irrelevant documents retrieved is increased, the precision is decreased.

4.5.1. Findings and Challenges of the Study

The result obtained in the experiments indicates, IR system build based on probabilistic model registered better result than Afan Oromo IR system developed by Gezahegn (2012). The IR system build in this study has recorded 12.5% F-measure improvement compared to the better achievement in previous works that was gezahegn's work (2012). This is a promising result to design an applicable IR system if polysemous nature of Afan Oromo text is controlled with the help of thesaurus and improve performance of the system. Note: He tested using 100 documents and nine (9) user queries through vector space retrieval approach.

In a general, the work done has registered promising performance. The result of this study indicates that, the IR system for Afan Oromo text register encouraging performance with 60%, 91.56% and 72.5% for precision, recall, F-measure respectively. With this performance registered, there are a brief description of several challenges that the researcher encountered during the process of implementing the concepts. These challenges hampered the system not to register the better result.

The first challenge comes from the probabilistic model itself. In probabilistic model the initial guess is made based on Boolean expression. This results every documents that matches one of the term in query are retrieved. It is possible to limit the retrieved documents using threshold; however, from the experiment it is found that if the user enter less than or equal to two queries there is a high probability of having similar weight. This will result retrieving no documents or retrieving all documents containing those terms. In this case, the precision decreases highly and the recall becomes 1.0. As a number of relevant documents increase with same irrelevant documents retrieved, both recall and precision is high. But when the numbers of retrieved documents are increased with same relevant documents retrieved after relevant documents, no effect on recall, but precision highly decreases. When recall value equal to1, all relevant documents are retrieved while when the value of precision value is equal to 1, the number of relevant retrieved documents and the total retrieved documents are equal. If no documents are not retrieved, precision could be undefined while recall could be zero.

The second challenge was checking the performance of IR system for Afan Oromo text using the Afan Oromo stemmer developed by Debela (2010). Still here are problems with stemmer. The

stemmer is a hybrid based stemmer. It is unable to handle the over stemming and under stemming. In the presence of prefix is one of the potential reasons to produce meaningless words. For example, if we remove a prefix *wal-* from the word *waldaa* which means "church", is conflated to "*–daa"* which is no meaning; because of the prefix *"wal-"* matches one of the prefixes listed in the stemmer and this is understemming . Unlike other language like English, Afan Oromoo suffixes are not quite different from non-suffix endings. Suffixes like "*-aa"* as in the case of *"waldaa*" conflated to –d-because of removing prefix "*wal-* " and suffix "*-aa*" matches one of the prefixes and suffixes listed in stemmer algorithm. Under stemming occurs when too much of the term is removed and over stemming occurs when too little of the term removed. For instance, for the term "*barbaadanitti*" is conflated to "*barbaad-*" and "*bar-*"; this stemming term is overstemming. To this end, a researcher has found difficult words after stemming. From this point of view, there is no standard stemmer for Afan Oromo to be used for every research.

The third challenge was a problem of controlling polysemous and synonyms of terms. Like any other information retrieval model, the probabilistic model has not incorporated a mechanism to control synonym or polysemous terms. If this problem is not addressed well, it was bit difficult to get Afan Oromo relevant documents. Hence, the word in Afan Oromo text has polysemy and/or synonym words. In IR system unless there is a mechanism to control those kinds of words, the performance of the system highly decreases because relevant documents containing synonym word for the query term were not retrieved, while irrelevant documents that contains polysemy /synonym words are retrieved. For instance, for a query "*miidhaa*" meaning threat, a document contain word "*dhibee*" meaning disease which is appeared in "**d7.txt**" and "*dhukkuba* " meaning also disease which is available in "**d5.txt**" could not be retrieved unless it contain the query word itself "*miidhaa*". Those different words represent similar idea. The combination of the above results leads to decrease the performance of the system in both precision and recall. Both high recall and high precision are desirable in information retrieval systems. However, they are difficult to achieve simultaneously.

The fourth challenge was finding a large size and standard corpus for Afan Oromo text retrieval system. The state-of-the-art in the area of text processing indicates that, there is no any developed standard corpus for Afan Oromo. Thus, the researcher uses small size corpus from

different sources as discussed in section 3.0. This result is not only weakens the performance of the system; but also makes it difficult to compare the result obtained with several researches since there is different in test queries, document content and size used for testing.

The fifth challenge was retrieving irrelevant documents rather than relevant documents. Additionally, in a very rare case, the experiment result shows a relevant document not retrieved. This is because of synonymy and polysemy of terms. For instance the Query ("*misooma biyyaa'*) which is *(*country's development) is one of the queries which registered lower precision in initial performance. In this case some relevant documents are not retrieved. Example document "**d65.txt'** is relevant for this query but not retrieved. This happens because this query is not directly available at document " **d65.txt", but** this document contains the term (*guddina dinagdee ummataa'*) meaning (economic growth of the people) in which they are synonym. Thus, there are such problems in other documents also. This problem can handle by considering synonymous and polysemy words.

As a general, as shown in table 4.6, after user relevance feedback on initially retrieved documents as relevant and non-relevant, the average percentage of precision, recall and F-measure is increased by 25.19%, 8.06% and 23.36% respectively. By comparing the previous study conducted by Gezahegn (2012), the performance obtained in this study is better than Gezahegn's study. The average percentage of precision and recall is increased by 2.5% and 28.92% respectively. This result is a promising result as compared with the previous study with a good achievement.

CHAPTER FIVE
CONCLUSION AND RECOMMENDATION

5.1. Conclusion

Text retrieval system is very important for retrieval of textual documents. The study attempts to develop a probabilistic IR system for Afan Oromo text. The developed prototype has two modules: indexing and searching. The indexing part of the work involves tokenization, normalization, stop word removal and stemming. The stemmer is adopted from (Debela, 2010).

Probabilistic model is used with probability of relevance based and evaluate the performance of Afan Oromo IR system. The hypothesis was enabling accessing unstructured Afan Oromo free-text-documents from the system using Afan Oromo queries and increasing the performance of the system by using user relevance feedback. The result obtained showed that a significant improvement over the previous runs. The fact that, this is relatively good improvements due to the enhancement of the performance and refinements of model.

Two hundred (200) different textual documents were used for doing the experimentation. And also, 10 queries were prepared to evaluate the performance of the prototype. Those queries were prepared subjectively by reviewing contents of each selected documents manually and collecting terms from twenty individuals randomly for each approaches.

In this study, the Binary Independent Model (BIM) was chosen and implemented. At first step when the search component initiated, the system generates the first ranked list of relevant documents then using terms from the initial guess made the system also searches again using user relevance feedback.

Finally, based on the user relevance feedback, the system improves its performance. This leads us to conclude that, user relevance feedback is useful for the improvement of an IR system. However, as the researcher observed from literatures, since probabilistic model used Boolean expression for initial guess of relevant document, it does not consider the importance of the document based on the frequency of the terms in the document. Because of this, sometimes those documents having query terms with highest frequency than others could be ranked lately. In this case, users faced with the problem of having to choose the appropriate words that are also used

in the relevant documents. Hence, poor result could be displayed when the system retrieve documents after user relevance feedback.

The stemming technique significantly increases the number of documents that match a user‘s query. A preprocessing techniques is used, in which the corpus were preprocessed using the tasks such as tokenization, case normalization, stop word removal, stemming, and indexing allows us to have the same standard between query terms and index terms.

In this study, an attempt has been made to design and develop a probabilistic IR system with a view to enable Afan Oromo speakers. Therefore, the model is used to search and retrieve relevant documents written in their own native language queries. The low performance achieved for the Afan Oromo document retrieval was because of spelling error that affects the accuracy of the retrieved document. According to the evaluation of the experiments, the results obtained shows that the prototype system registers 60%, 91.56% and 72.5% for precision, recall and F-measure respectively. As the experiments have shown, probabilistic model has improved the performance of Afan Oromo information retrieval system from 60% to 72.5% F-measure. From this result, the overall result is an encouraging figure with an improvement of 12.5% F-measure from the previous study.

5.2. Recommendation

IR technology is an ever-growing area of research, especially with regard to Afan Oromo text and the unique challenges presented by the complex characteristics of the language. The area is just at a beginning level on the probabilistic model. Even though, the result of the experiment is a promising result, based on the findings of the experiment we recommend, the following points as a future research area to improve the effectiveness and efficiency of the system:-

- ✓ One of the main problems faced to enhance the performance of Afan Oromo IR system in this study is the existence of synonym and polysemy terms. The researcher recommended that integrating mechanisms of controlling synonyms/ polysemy terms in the probabilistic model to enhance precision and recall of the system.
- ✓ Probabilistic model make the initial guess based on Boolean expression, which inhibit to know important words to represent a document and, accordingly may not retrieve relevant documents that contain large number of terms found in a given query. Hence, there is a need to build hybrid system that uses VSM to guess relevant documents for user query using non-binary weighting technique and then use probabilistic relevance feedback to improve the performance of the system.
- ✓ The performance of Afan Oromo IR system depends on the size of the corpus. Due to this, preparing of that corpus and using for training and testing activities needs standardization, which is not done currently. Thus, developing standards of a corpus is open research for Afan Oromo and other local languages.
- ✓ The stemming algorithm used in this study is the best algorithm developed so far for Afan Oromo. However, it frequently over stems word variants greatly affecting the performance of the system. Therefore, future work need to consider designing ontology based stemming algorithm that conflates based on meaning understanding.
- ✓ This research is only for text document. Other types of documents like images, video and audio were not included. It is recommended to study for those types of documents.
- ✓ Since the capability of the stemmer developed so far was not found to be efficient. Hence, an efficient stemmer should be developed.
- ✓ Further study is needed to figure out the effectiveness of the system by comparing the other probabilistic models like Bayesian network, Bayesian belief network and Bayesian inference network model.

References

Abebe Abeshu (2010). Automatic Morphological synthesizer for Afan Oromo, Department of Computer Science, Master's thesis, Addis Ababa University, Addis Ababa.

Ager, S. (2012). Oromo language, available:www.sas.upenn.edu/Africanstudies/Hornet/Afan Oromo 19777. Html. Accessed: October 12, 2016.

Aizawa, A. (2000). The feature quantity: an information-theoretic perspective of tfidf-like measures. *In Proceedings of the 23rd ACM SIGIR conference on research and development in information retrieval* (pp. 104–111).

Amanuel Hirpa Madessa (2012). Probabilistic Information Retrieval System for Amharic Language" MSc Thesis, School of Information Science, Addis Ababa University.

Andrew G. (2002). Video retrieval using an MPEG-7 based inference network. *In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM New York, NY, USA.

Assefa W/Mariam (2005). Developing Morphological Analysis for Afan Oromo Text, Master's Thesis, School of Graduate studies, Addis Ababa University.

Atalay Luel (2014). A Probabilistic Information Retrieval System for Tigrinya. MSc Thesis, School of Information Science, Addis Ababa University, Ethiopia.

B. R. RibeiroNeto (1999). Modern Information Retrieval, 2nd Edition, Addison Wesley Longman Publishers, New York, USA.

Bansal, S. (2012). Comparison between the Probabilistic and Vector Space Model for Spam Filtering. *International Journal of Computational Intelligence Techniques.*Available online at http://www.bioinfopublication.org/jouarchive.php?opt=&jouid=BPJ0000221

Berger, A et al (2000). Bridging the Lexical Chasm: Statistical Approaches to Answer Finding. *In Proc. Int. Conf. Research and Development in Information Retrieval, 192-199.*

Broglio, J. et al. (n.d). Technical Issues in Building an Information Retrieval System for Chinese.

C. D. Manning, et al. (2008). Introduction to information retrieval, Cambridge University Press.

C. J. Van Rijsbergen (2004). The Geometry of Information Retrieval. Cambridge, U.K.:

Ceri, et al. (2013). Web Information Retrieval. Available at: www.springer.com/series/5258

DebelaTesfaye (2010). Designing a Stemmer for Afan Oromo Text: A Hybrid Approach "MSc Thesis, school of information science, Addis Ababa University, Ethiopia.

Dejene Hundessa (2015). Definition Question Answering System for Afan Oromo Language. MSc. Thesis, school of information science, Addis Ababa University, Ethiopia.

Ed Greengrass (2000). Information Retrieval: A Survey, 30 November.

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

F. Crestani and M. Lalmas (2001). Logic and uncertainty in information retrieval, in Lectures on information retrieval, ed: Springer, pp. 179-206, London, UK.

F.Silva, R. Girardi, and L. Drumond (2009). An IR Model for the Web. *IEEE International Conference on information technology*, Brazil.

Getachew Rabirra (2014). Furtuu: Seerluga Afan Oromoo, Finfinnee Oromiyaa press.

Gezehagn Gutema (2012). Afan Oromo text retrieval system. MSc Thesis, Addis Ababa University, Addis Ababa, Ethiopia.

H.S. Christopher D.Manning, PrabhakarRaghavan (2009). An Introduction to Information Retrieval, 1st Edition, Cambridge University Press, Cambridge England.

H.R. Turtle, W.B. Croft (1990). Inference networks for document retrieval, in: J.-L. Vidick (Ed.), SIGIR'90, 13th International Conference on Research and Development in Information Retrieval, Brussels, Proceedings, ACM Press, 1990, pp. 1–24.

Hiemstra, D. (2009). "Information retrieval models," Information Retrieval: searching in the 21st Century, (pp. 1-19).

Indrawan, M. (1998). A Framework for Information Retrieval Based On Bayesian Networks.

Joho,J. and M. Sanderson(2007). Document frequency and term specificity. In *theRecherched'InformationAssiste par Ordinateur Conference (RIAO)*.

Jacob Bank and Benjamin Cole (2008). Calculating the jaccard similarity coefficient with map reduce for entity pairs in wikipedia.

Josh (2013). Techniques of Query Reformulation in Information Retrieval. *Journal of Information, Knowledge and Research in Computer Engineering.*

K.S. S.E.Robertson (1976). Relevance Weighting of Search Terms. *Journal of the American Society for Information Science,* 27(4):129-146.

Kocabas, et al. (2011). Investigation of Luhn's claim on information retrieval. *Turkish Journal of Electrical Engineering & Computer Sciences,* pp. 993-1004.

Kula, K., Varma, V. and Pingali, P. (2008). Evaluation of Oromo-English Cross-Language Technologies Research Center. Information Retrieval IIIT, Hyderabad, India.

Lam, S. (2001). The Overview of Web Search Engines.

Majumder, P. (2009). Indian Language Information Retrieval.

MandefroLegesse (2012). Named Entity Recognition for Afan Oromo, Master‟s Thesis, School of Graduate studies, Addis Ababa University.

Manning, C. D., Raghavan P. and Schutze H. (2009). An Introduction to Information Retrieval, Online Edition. Cambridge University Press, Cambridge, England.

Melkamu Abetu (2017). Query Expansion for Afan Oromo Information Retrieval Based On Wordnet. Msc Thesis, school of graduate studies, Haramaya University, Haramaya.

Meyer, C. (2008). On Improving Natural Language Processing through Phrase-based and one-to one Syntactic Algorithm, Msc. Thesis, Kansas State University Manhatan, Kansas.

N. Fuhr (1992). Probabilistic Models in Information Retrieval. *The Computer Journal.*

Omnigton (2013). The online encyclopedia of writing systems and language. Accessed from http://www.omniglot.com/writing/Oromo.htm, May 10.

R. Baeza-Yates et al. (1999). Modern information retrieval," ACM press New York, USA.

Robertson, S. E. (1977). The probability ranking principle in IR. *Journal of Documentation*.

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

Salton, G. and McGill, M. (1983). Introduction to Modern Information Retrieval, McGraw-Hill, New York.

Sanjay, et al. (2011). Information Retrieval Evaluative Model FTICT 2011. *Proceedings of the 2011, International conference on "Future Trend in Information & Communication Technology, Ghaziabad, India, Feb -2011*.

Schatz (1997). Information retrieval in digital libraries: Bringing search to the Net. *Journal of Science*, 275 (5298): 327-334.

Sergey Brin and Lawrence Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine,". [Online].Available: http://infolab.stanford.edu/~backrub/google.html

Sharifloo, Amir Azim, and MehrnoushShamsfard (2008). A Bottom Up approach to Persian Stemming." In*IJCNLP*, pp. 583-588.

Singh, et al. (2015). Vector Space Model: An Information Retrieval System. *International Journal of Advanced Engineering Research and Studies.*

S. Heinz (2003). Efficient single-pass index construction for text databases,‖ *Journal of the American Society for*, vol. 54, no. 8, pp. 713-729.

Singhal, A. (2002). Modern Information Retrieval: A Brief Overview.

Sneha Deep (2017). A Review Of Information Retrieval System Using Relevance Feedback Algorithm. *International Journal of Scientific Research Engineering & Technology (IJSRET), ISSN 2278 – 0882*

Tesfaye Guta (2010). Afan Oromo Search Engine. Department of Computer Science, MSc. Thesis, Addis Ababa University.

TilahunGamta (1992). The Oromo language and the latin alphabet, Journal of Oromo Studies .http://www.africa.upenn.edu/Hornet/Afan_Oromo_19777.html last visited on Friday, October 31, 2014. http://ethnomed.org/culture/Oromo/Oromo-alphabets-and-sounds sagaleewani-fi-loqoda/, last visited on Friday, October 31, 2014.

Tsegaye Semere (2013). Probabilistic Tigrigna-Amharic Cross Language Information Retrieval (ClIR). MSc Thesis, Addis Ababa University, Addis Ababa, Ethiopia.

Tunkelang (2009). Faceted Search. Synthesis Lectures on Information Concepts, Retrieval, and Services.

Wen-Jen Yu, Shrane Koung Chou(2010). A Bibliometric Study of Search Engine Literature in the SSCI Database. *Journal of Software*. Vol5, No 12 (2010), 1317-1322.

W. J. Chun (2006). Core python programming", Prentice Hall PTR, 2006.

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

Wooldridge,S. (2003). Bayesian Belief Networks.

WorkinehTesemaGudisa (2015). Towards the Sense Disambiguation of Afan OromooWords Using Hybrid Approach (Unsupervised Machine Learning and Rule Based).MSc Thesis, school of graduate studies, Jimma University, Jimma.

Zaman, A. (2010). Study of Document Retrieval Using Latent Semantic Indexing (LSI) on a Very Large Data Set.

Zhu, R. (2016). Improvement in Probabilistic Information Retrieval Model - Rewarding Terms with High Relative Term Frequency.

Zhou (n.d). Performance Comparison of Language Models for Information Retrieval.

Appendices

Appendix 1: Document-query matrix used for Relevance Judgment where R-Relevant, NR-non relevant

Docs	Rakkoo Fayyaa Maatii	Qulqullina Barnootaa Mirkaneessuu	Bu'uura amantii ilma namaa	Hirmaannaa Ummataa	Misooma biyyaa	Meeshaa Aadaa Hawaasa Oromoo	Ispoortii atileetiksii itoophiyaa	Aangoo mootummaa	Qaamolee haqaa naannoo keenyaa	Shaakala dorgommii atileetiksii i kilaboota kubbaa miilaa
d1	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d2	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d3	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d4	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d5	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d6	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d7	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d8	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d9	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d10	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d11	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d12	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d13	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d14	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d15	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
d16	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d17	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d18	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d19	NR	R N	NR	NR	NR	NR	NR	NR	NR	NR
d20	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d21	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d22	NR	R	NR	NR	NR	NR	NR	NR	NR	NR

d23	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d24	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d25	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d26	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d27	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d28	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d29	NR	R	NR	NR	NR	NR	NR	NR	NR	V
d30	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d31	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d32	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d33	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d34	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d35	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d36	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d37	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d38	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d39	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d40	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d41	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d42	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d43	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d44	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d45	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d46	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d47	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d48	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d49	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d50	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d51	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d52	NR	NR	NR	R	NR	NR	NR	NR	NR	NR

d53	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d54	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d55	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d56	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d57	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d58	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d59	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d60	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d61	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d62	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d63	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d64	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d65	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d66	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d67	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d68	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d69	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d70	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d71	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d72	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d73	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d74	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d75	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d76	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d77	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d78	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d79	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d80	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d81	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d82	NR	NR	NR	NR	NR	R	NR	NR	NR	NR

d83	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d84	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d85	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d86	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d87	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d88	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d89	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d90	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d91	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d92	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d93	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d94	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d95	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d96	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d97	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d98	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d99	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d100	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d101	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d102	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d103	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d104	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d105	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d106	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d107	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d108	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d109	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d110	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d111	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d112	NR	NR	NR	NR	NR	NR	R	NR	NR	NR

d113	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d114	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d115	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d116	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d117	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d118	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d119	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d120	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d121	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d122	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d123	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d124	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d125	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d126	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d127	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d128	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d129	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d130	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d131	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d132	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d133	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d134	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d135	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d136	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
d137	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
d138	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
d139	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
d140	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
d141	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
d142	NR	NR	NR	NR	NR	NR	NR	NR	R	NR

d143	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
d144	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d145	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d146	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d147	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d148	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d149	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d150	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d151	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d152	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d153	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d154	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d155	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
d156	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d157	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d158	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
d159	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
d160	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
d161	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
d162	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
d163	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d164	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d165	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d166	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d167	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
d168	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d169	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d170	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
d171	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d172	NR	R	NR	NR	NR	NR	NR	NR	NR	NR

d173	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d174	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d175	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
d176	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d177	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d178	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d179	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d180	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d181	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d182	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d183	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d184	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d185	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
d186	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d187	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d188	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
d189	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d190	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d191	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d192	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d193	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d194	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d195	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d 196	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
d197	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
d198	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d199	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
d200	NR	NR	NR	NR	NR	NR	R	NR	NR	NR

Appendix 2: Afan Oromo stop word list

aanee	aga	achi	achuma	adda	addaatti
afoo	agarsiisoo	akkasumas	al	ala	alatti
alla	akka	akkam	akkamii	akkamiitu	akkana
akkataa	akkataan	akkas	akkasitti	akkuma	akksumas
amma	ammaa	ammo	ala	alatti	alla
an	ana	anee	ani	asham	asitti
attam	ati	awu	bira	biraa	biratti
biro	biroon	biros	biyyam	booda	booddee
bukkee	cinaa	abalatees	dhaa	dhaan	dudduuba
dugda	duuba	dura	duraa	eega	eegana
eegasii	ennaa	eenuu	eenyu	eennu	eenyufaa
eenyuun	eennuun	eenyuuf	eennuuf	eenyufaa	eenyufaadhaan
eenyufaaf	eenyufaarraa	eenyufatti	eenyurra	eennurraa	eenyurraa
eennurraa	eenyurratti	eennuratti	eenyuuf	eenyuree	eennuree
eenyutti	eennutti	eenyum	eessa	ega	eessatti
eessarraa	eessaaf	eessaan	erga	ergii	f
faa	faallaa	fagaatee	faati	faatidhaa	faatiidhaa
fakkaatanitti	fakkaatti	fakkaatu	fakkaattu	fakkeenyaa	fakkeenyaaf
fedhetuu	fi	fk	fkf	fkn	fknf
fuuldura	fundura	fullee	fuullee	gaa	gad
gadi	gaditti	gahaa	gajjallaa	gala	galan
galani	galuu	gama	gamma	gar	gara
garam	garamiin	garamitti	garana	garas	gararraa
gararree	garii	garjalee	garuu	giddu	gidduu
godhe	gootee	gubbaa	haa	haga	hagam
hamma	hammam	hammamiif	hammamiin	hammamtu	hanga
henna	himantu	hin	hinjira	hinjiru	hinjirtu
hinjirtan	hinjiru	hoggaa	hoo	hunda	hunduma
idda	iddoo	if	illee	immoo	inni
irra	irraa	irraan	irratti	irrattii	isa

isaa	isaaf	isaan	isaanii	isaaniif	isaaniitiin
isaaniis	isaanniis	isaaniirratti	isarraa	isaanirraa	isatti
ishe	ishee	isee	ishii	ishiif	isiidhaa
ishiidhaa	isii	isiin	ishiifaa	ishiin	isin
isinii	isiniif	isiniin	isiniis	isinirraa	ittaanee
itti	ittillee	ittumallee	ittiin	ituu	ituullee
jala	jara	jechaan	jechoota	jechuu	jechuun
jedhu	jedhus	jira	jiraadha	jiraadhe	jiraanne
jiraate	jiraattee	jiraatta	jiraatti	jiraanna	jiraatan
jiraattan	jiran	jiranii	jiranirra	jiranu	jirre
jirta	jirti	jirra	jirtan	jirtanu	jirtu
jiru	kaa	ka'e	ka'en	kam	kamfaa
kami	kamidha	kamifaadha	kamiin	kamiinu	kamiinuu
kamiif	kaminiyyuu	kamirraa	kamitti	kamttuu	kamitu
kamuma	kamuu	kamirrattu	kamiyyuu	kan	kana
kanaa	kanaan	kanaaf	kanaafuu	kanaafi	kanaafiis
kanaatti	kanarra	kanarratti	kanisaanii	kanishii	kankee
kankeeny	kankoo	kanneen	kanneeni	kanarraa	karaa
kee	keenna	keenya	keessa	keessan	keessatti
keeysa	kiyya	kkf	koo	kun	qunnama
kuni	kunis	kunoo	kunneen	kanneeni	laata
lafa	lama	maal	maalfaa	maali	maalif
maaliin	maalirraa	maalirratti	maaliree	maalittuu	maaltu
maaluma	maalumaaf	malee	manna	maqaa	mee
meeqa	meeqaan	meeqaaf	meeqarraa	meeqatti	meeqatu
meerre	meerreree	miti	moo	na	naa
naaf	naan	nan	naannoo	natti	nu
nuhi	nu'i	nurraa	nuu	nuuf	nuun
nuti	nuyi	obboo	odoo	of	ofii
ofirraa	ofiif	ofiin	oftiin	ofitti	ofuma
ofumaa	ofumaaf	ofumaanirrattii	ofumatti	oggaa	ol

oli	olkaa'i	oo	osoo	otoo	otumalle	
otuu	otuullee	saaniif	sadii	san	sani	
sana	saanii	sanaa	sanaan	sanaas	saniif	
saniin	si	sii	siif	siin	silaa	
simmoo	sinitti	siqee	sirraa	sirrii	sirritti	
sirrumatti	sitti	suma	sun	suni	suniin	
sunneen	ta'a	takka	takkaan	ta'an	ta'anii	
ta'aniifi	ta'ani	ta'e	ta'etti	ta’uun	taane	ta’anii
ta’aniif	taa'ee	taa'een	taa'eetti	taata	taati	taana
taatan	ta'u	ta'uu	ta'uun	ta'uuf	taatu	taanu
taatan	tahullee	tana	tanaaf	tanaafi	tanaafuu	ta'ullee
ta'uyyuu	tawullee	tansaa	tam	tamiif	tamiin	
tamuma	tan	tana	tanisii	tanneen	tanneeni	tantee
tanteenya	tee	teenya	teessan	teeysa	tiyya	tuqa
tuqan	tuquu	tokko	tokkoo	tokkootti	tokkos	too
tun	tuni	tunoo	turan	ture	turee	turre
turte	turtan	utuu	waan	waa'ee	waggaa	wajjin
wal	walakkaa	walii	waliif	waliin	waliis	walitti
walirratti	walirra	walirraa	waljidduu	walleenuu	walqabatan	
walqabatanis	walqunnaman	walumaaf	walumaan	wahii	waanta	
waantoota	waayee	warra	woo	wojiin	wolbira	xiqqaa
xiqqaatuu	yammuu	yemmuu	yennaa	yeroo	yommii	ykn
yommuu	yoo	yookaan	yookiin	yookiinimmoo	yoom	
yoomiif	yoomiree	yoomuma	yoomittuu	yoos		

Appendix 3: Afan Oromo Numbering System

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**20**	**30**	**40**
50	**60**	**70**	**80**	**90**	**100**							

Appendix 4: Afan Oromo Punctuation mark list

,	**.**	**:**	**?**	**;**		**!**

Appendix 5:- Question for Collection of User Query

The questions are prepared and converted to Afan Oromo text to make suitable to collect the user query test from public.

Saala: _____umurii:_____

Sadarkaa barnootaa:_____ kutaa 1-5:_____kutaaa 6-12:_____

Barataa koolleejjii:_____Barataa yuuniversiitii:_____

Barsiisaa: _____ Qotee bulaa:_____

Kan biro: _____

1. Jechoota lamaa hanga torbaa (2-7 words) kan of keessaa qaban kan naannookeetti yeroo baay'ee fayyadamtu kudhan (10) barreessi . Jechoota kana yeroo barreessitan

   1ffaan:- waa'ee fayyaa kan ibsu

   2ffaan:-waa'ee barnootaa

   3ffaan:-waa'ee amantii

   4ffaan:-waa'ee hawaasumma

   5ffaan:-waa'ee Dinagdee

   6ffaan:-waa'ee aadaa

   7ffaan:-waa'ee spoortii bakka lamatti barreessaa

   8ffaan:-waa'ee siyaasaa

   9ffaan:-waa'ee haqummaa

Lakkoofsa	Jechoota (words)	Baay'ina jechichaa	Waa'ee isaa
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			