TEXT INFORMATION RETRIEVAL SYSTEM FOR SILT'E LANGUAGE

ABDULMALIK JOHAR

A Thesis Submitted to

The Department of computing

School of electrical engineering and computing

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Information System

Office of Graduate Studies

Adama Science and Technology University

Adama

September 2017

TEXT INFORMATION RETRIEVAL SYSTEM FOR SILT'E LANGUAGE

ABDULMALIK JOHAR

ADVISOR: TILAHUN MELAK (PhD)

A Thesis Submitted to

The Department of computing

School of electrical engineering and computing

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Information System

Office of Graduate Studies

Adama Science and Technology University

Adama

September 2017

Approved by the Examining Board

We, the undersigned, members of the Board of Examiners of the final open defense by **ABDULMALIK JOHAR** have read and evaluated his thesis entitled "**TEXT INFORMATION RETRIEVAL SYSTEM FOR SILT'E LANGUAGE** " and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirement of the Degree of Information System

Dr. Tilahun Melak	_____	_____
Advisor	Signature	Date
_____	_____	_____
Chairperson	Signature	Date
_____	_____	_____
External Examiner	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date

DECLARATION

I hereby declare that this MSc Thesis is my original work and has not been presented for a degree in any other university, and all sources of material used for this thesis have been duly acknowledged.

Name: Abdulmalik Johar

Signature: _____

This MSc Thesis has been submitted for examination with my approval as thesis advisor.

Name: Dr. Tilahun Melak

Signature: _____

Date of submission: September 2017

ACKNOWLEDGEMENTS

I would like to express my gratitude and heartfelt thanks to my advisor, Dr. Tilahun Melak, for his guidance, encouragement and support throughout this research. Also I would like to express my deepest gratitude and heartfelt thanks to my instructor Mr.Bahiru Shifaw for his support. I would like to express my deepest thankfulness to Dr.Tibebe Beshah for his valuable feedbacks and support. My special thanks goes to my friends and classmates your support and advice has been worth considering starting from the beginning up to the completion of the program.

Finally, I would like to thank my parents, Sadiya Sani and Johar Mohammed for their support and prayers, and especially my wife Hidaya Serefa for the constant encouragement and caring during the time of my study.

TABLE OF CONTENTS PAGE

LIST OF TABLES

TABLE **PAGE**

LIST OF FIGURES

FIGURE **PAGE**

LIST OF ACRONYMS

ALS **Apache Lucene Solr**

BIM **Binary Independent Model**

BM **Best Match**

CLIR **Cross Language Information Retrieval**

Doc-Id **Document Id**

IR **Information Retrieval**

ISO **international standard organization**

MAP **Mean Average Precision**

NR **Non-Relevant**

Okapi **online keyword access public information**

PRP **probabilistic relevance principle**

R **Relevant**

STV **Semitic South Transverse**

VSM **Vector Space Model**

ABSTRACT

The main aim of an information retrieval system is to extract appropriate information from an enormous collection of data based on user's need. The basic concept of the information retrieval system is that when a user sends out a query, the system would try to generate a list of related documents ranked in order, according to their degree of relevance. Most of the present information retrieval systems assign numeric scores by weighting functions to certain documents and put them in rank based on the scores. The BM25 weighting function has been one of the most efficient and widely-used information retrieval weighting models in the past three decades. It has a good term weighting is based on three principles inverse document frequency, term frequency, and document length normalization. Digital unstructured Silt'e text documents increase from time to time. The growth of digital text information makes the utilization and access of the right information difficult. Thus, developing an information retrieval system for Silt'e language allows searching and retrieving relevant documents that satisfy information need of users. In this research, we design probabilistic information retrieval system for Silt'e language. The system has both indexing and searching part was created. In these modules, different text operations such as tokenization, stemming, stop word removal and synonym is included. For the experimenting purpose, we have used 134 Silt'e text documents and 10 queries were used to test the system. Which are collected from the sources of different government organization in Silt'e zone. We Use Apache solr for prototype development and the schema similarity for weighting the document used probabilistic weighting function, BM25. According to the experimentation, the performance of the system after stemming and synonym register 88% mean average precision. The result is promising to develop Silt'e text information retrieval systems and Silt'e search engines. The challenging task in the research was lack of standardized and well-prepared Silt'e text corpus and test queries which required conducting certain experimentation for evaluation of the proposed system and these will be future research directions in this area which contribute to the improvement of the system.

Keywords: - Information Retrieval, Probabilistic Model, Silt'e Language

CHAPTER ONE

1. Introduction

1.1. Background

Information retrieval is a set of activities that represent, store and ultimately obtain information based on a specific need [1]. IR deals with the representation, storage, organization of, and access to information items [2]. Nowadays, information is everywhere that helps people to make the correct decision at the right time. 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 [3]. Information retrieval (IR) is different to database retrieval. A database retrieval system simply retrieves all documents or objects that satisfy certain criteria, whereas an information retrieval system needs to assess the information needs of its users and rank the answer documents based on likely relevance [2]. Information retrieval systems are designed to facilitate access to stored information. In addition, they are concerned with the representation, storage, and organization of information items. The components of information retrieval system consist of a set of information items, a set of requests and a mechanism to determine which information items are most likely to meet the requirements of the requests. Basically, users of information state their information needs in the form of queries and submit their queries to the information retrieval system. Based on, the user queries the system output information item that is relevant to the user query. In another word, the items that have common entries in both the database and users query are returned to users. This happens when a matching exists between queries and information items. It is difficult to specify information that is needed using exact query terms. The most critical language issue for retrieval effectiveness is the term mismatch problem. The indexers and the users do not often use the same words. To attain a match between these two components, the items in the documents as well as in the query should be represented in some form [4].

There are various methodologies and techniques tried so as to make the effective and efficient way of retrieving documents from very large and unstructured corpus. Some of the scientific approaches to resolving the problems are ontology based IR, latent semantic indexing, query

expansion and reformulation, the statistical method for semantic indexing, and others [5]. Predicting relevant documents is one of the core issues in IR system, and IR models are responsible for determining the prediction of what is relevant and what is not [2]. A number of retrieval models have been proposed since the mid-1960s. They have evolved from specific models intended for use with the small structured document to recent models that have the strong theoretical basis and which are intended to accommodate a variety of full-text document types such as Boolean model, vector space model, probabilistic model, etc.

Among the Current models one is the Probabilistic model. The probabilistic model ranks the documents by estimating the probability of relevance of documents with respect to the query. The process of estimation is critical, and this is where the implementations vary from one another. This model is based on the notion that for a query, each document can be classified as relevant or non-relevant [1]. The probabilistic model is the oldest but also the hottest research area in IR. There can be a variety of sources of evidence that are used by the probabilistic retrieval methods. The most common one is the statistical distribution of the terms in both the relevant and non-relevant documents. The principle takes into account that there is uncertainty in the representation of the information need and the documents [6]. In general, probabilistic models avoid shifting uncertainties to users by providing solutions for computing relevance certainty, whereas the Boolean model searches the documents based on Boolean logic and classical set theory that uncertainties remain as uncertainties for users to deal with [7]. While in comparison to the vector space model whose documents are retrieved and ranked by the degree of similarity, probabilistic models are more interpretable and computable.

According to Baeza-Yates and Ribeiro-Neto, [2] BM25 was created as the result of a series of experiments on variations of the probabilistic model. A good term weighting is based on three principles inverse document frequency, term frequency, and document length normalization. The classic probabilistic model covers only the first of these principles. This reasoning led to a series of experiments with the Okapi system, which led to the BM25 ranking formula. The BM25 has been one of the most efficient and widely-used information retrieval weighting models in the past three decades. Unlike other probabilistic models, the BM25 could be computed without relevance information. Since BM25 almost outperforms classic vector model for general data collections, it has substituted the vector space model to be used as a baseline for comparison for decades.

Therefore in this thesis work we used BM25 similarity measure for design Silt'e text information retrieval.

1.2. Statement of the Problem

There are more than 80 languages in Ethiopia. Silt'e is one of the zonal languages in the southern Nations, Nationalities and peoples region of Ethiopia. According to the Central Statistics Authority (2007:80), the number of Silt'e speakers is 751,159. The census report reveals that 47,097 people or 6.28% of the total population are urban or semi-urban inhabitants [8] [9].

It is the South-Ethio-Semitic language family and uses Saba Ge'ez (Ethiopic script). A lot of peoples speak it in many parts of Ethiopia. Elementary schools in Silt'e zone give all subjects in Silt'e from grade one up to four. After grade four, it is given as one subject up to preparatory level and takes it as grade ten national exams. The Silt'e zone administration decided to use the langue as a working language since 2014 [10].

Digital unstructured Silt'e text documents in government organizations and schools increase from time to time in the Silt'e zone. The growth of digital text information makes the utilization and access of the right information difficult. Therefore, an information retrieval system needed to store, organize and helps to access Silt'e digital text information.

Silt'e IR systems have been developed using vector space model by Jemal.Z [11]. He score 81% MAP with 100 Silt'e document and without control synonym. However, the use of vector space model may not control uncertainty nature of IR system. As stated by Fuhr [12], the other problem of developing an IR system using vector space model is lack of handling uncertainty nature of IR. How exact is the representation of the document and query? How well is query match to documents? How relevant is the searching result to the query? Are the relevant documents uncertain? The most successful approach for coping with uncertain in IR is a probabilistic model. Probability theory seems to be the most natural way to quantify uncertainty. Thus in this study will an attempt to develop probabilistic (BM25 model) based Silt'e IR system to control uncertainty nature of the Silt'e IR system also consider synonyms of word in Silt'e language. Also compare the performance of the system with previous vector space model Silt'e IR system.

In the backdrop of the above discussion, we designed corpus-based Silt'e language information retrieval system prototype to overcome the stated problems or showcase how access to digital

resources in Silt'e language can be initiated. Hence, the research finds answers to the following research questions.

- What are the linguistic features of the Silt’e language?
- Which text operations and indexing procedures are suitable to prepare the Silt’e documents?
- Does probabilistic model enhance the effectiveness of the IR system in attempting to search relevant Silt’e documents?

1.3. Objectives

1.3.1. General Objective

The general objective of this studies is to design text information retrieval system for Silt’e language.

1.3.2. Specific Objectives

The specific objective of this study.

- To review a literature to get sufficient understanding of the concepts on probabilistic IR model.
- To investigate the semantic and syntactic structure of the Silt’e language
- To design a prototype probabilistic Silt’e IR system that can search for relevant documents from Silt’e corpus.
- To assess the overall performance of the system using IR effectiveness measures such as recall, precision, F-measure and MAP.
- Recommend future work instructions for in addition investigations

1.4. Scope and Limitation of the Study

The scope of this study is to develop a prototype IR system by way of making use of an Okapi BM25 method of the probabilistic model. And checking probabilistic model valuable in Silt’e language. The IR system is designed to effectively search from within Silt’e document corpus. We

take care of synonymy terms for morphologically complicated information retrieval systems. The system develops the following procedures of the IR system. Given Silt'e document corpus, text operations by means of tokenization, stop words removal, synonymy and stemming are applied. Inverted file indexing is used to organize content-bearing index terms Searching module is enabled to allow users search for relevant documents that satisfy their information need.

The main limitation to get on this study the system is tested using the limited amount of Silt'e text documents prepared by the researcher. This is due to lack of standard Silt'e corpus prepared for IR purpose. As a result of the time factor, limited corpus we use for evaluating the performance of the IR system developed in this study.

1.5. Significance of the Study

Applying current information retrieval technologies in local languages have several benefits. This technology to preserve the Silt'e language being dominated by other languages and improves the development of the language. From this study, the primary and immediate beneficiaries are Silt'e speakers retrieving text documents in Silt'e language efficiently, effectively. Finally, the findings of this study could provide information for those who are interested in making a further study in this similar area.

1.6. Organization of the Research

This thesis is organized into five chapters. The first chapter briefly discusses the background to the problem area and tells the problem, objectives of the study, scope and significance of the research.

Chapter two deals with literature review basic of Silt'e language, overview of information retrieval system, information retrieval models, and general process in information retrieval system and related works. Chapter three materials and methods used for Implementation of IR system using probabilistic model. Chapter four Experiment and discussion were presented. Chapter five provides conclusion and offers recommendations for future works.

CHAPTER TWO

2. Literature Review

2.1. Basic of Silt'e Language

Information retrieval research process contains many aspects. Among those aspects understanding the language basics are the most important one [13]. As a result, in this section, we discussed the Silt'e language basics clearly and precisely. To done this we referred researches conducted on the Silt'e language.

Silt'e is a Semitic language mainly spoken in Ethiopia Southern Nations, Nationalities and People's Region in the Silte Zone. Some Silt'e speakers settled in other parts of the country, particularly in Adama, Addis Ababa and Jimma. The Silt'e language is given the International Standard for Language (ISO 639_3) code of STV (Semitic South Transverse). The secondary level (L2) literacy of the language reaches 17%. The literacy has been increasing, because it is used as a medium of instruction in primary schools from grade 1-4 and taken as a subject from grade 5-12 [14].

2.1.1. Silt'e Dialects

The Silt'e language consisting of four dialects, these are Azernet-Berbere, Silti, Wuriro and Ulbareg. These four differences are determined by the topographical region. They are grouped into four dialect families but, the strong similarity exists among them [14].

2.1.2. The Writing System of Silt'e Language

The accurate period that the Ge'ez character set was used for Silt'e writing system is not known. However, since the 1980s, Silt'e has been written in the Ge'ez alphabet, or Ethiopic script, writing system, originally developed for the now extinct Ge'ez language and most known today in its use for Amharic and Tigrigna [14]. Portion of the Ge'ez (Ethiopic) alphabets and their phonetic order are shown in the following table.

1st order	2st order	3st order	4st order	5st order	6st order	7st order
ሀ ha	ሁ hu	ሂ hi	ሃ haa	ሄ he	ህ hi	ሆ ho

ለ le	ሉ lu	ሊ li	ላ la	ሌ le	ል li	ሎ lo
መ me	ሙ mu	ሚ mi	ማ ma	ሜ me	ም mi	ሞ mo
ረ re	ሩ ru	ሪ ri	ራ ra	ሬ re	ር ri	ሮ ro
ሰ se	ሱ su	ሲ si	ሳ sa	ሴ se	ስ si	ሶ so

Table 2.1 Phonetic order of Ge'ez alphabets

2.1.2.1. Silt’e Vowels

In the Ge'ez scripts only seven vowels are distinguished, but in written Silt'e these seven vowels are mapped onto ten , vowels, these are the five short vowels አ[a], ኦ[o], እ[i], ኡ[u], ኤ[e] and five long vowels ኣ[aa], ኦኦ[oo], ኢ[ii], ኡኡ[uu], ኤኤ[ee] [15].The following practical example illustrates Silt’e language vowels clearly.

- ä → a: አለፈ alafa 'he passed'
- u → u, uu: ሙት mut 'death', muut 'thing'
- i →
  - ✓ ii: ኢን iin 'eye'
  - ✓ word-final i: መሪ mari 'friend'
  - ✓ i ending a noun stem: መሪከ marika 'his friend'
  - ✓ impersonal perfect verb i suffix: ባሊ baali 'people said'; በባሊም babaalim 'even if people said'
- a → aa: ጋራሽ gaaraaš 'your (f.) house'
- e → e, ee: ኤፌ eeffe 'he covered'
- ə →
- (except as above): እንግር ingir 'foot'
- consonant not followed by a vowel: አስሮሽት asroošt 'twelve'
- o → o, oo: ቆጬ k'oč'e 'tortoise', k'ooč'e 'he cut'

2.1.2.2. Silt’e Consonants

Silt'e language has twenty-six consonants which are similar with other Ethiopian Semitic languages [15]. There are ejective consonants in conjunction with plain voiced and voiceless consonants and these can be reduplicated after a short vowel.

ሀ | **h** |	ለ | **l** |	መ | **m** |	ረ | **r** |	ሰ | **s** |	ሸ | **s** |	ቀ | **q** |	በ | **b** |	ተ | **t** |	ቸ | **c** |	ነ | **n** |	ኘ | **n** |	አ | **a** |
ከ | **k** |	ወ | **w** |	ዘ | **z** |	ዠ | **z** |	የ | **y** |	ደ | **d** |	ጀ | **j** |	ገ | **g** |	ጠ | **c** |	ጨ | **c** |	ጰ | **p** |	ፈ | **f** |	ፐ | **p** |

Table 2.2 Silt’e consonants

2.1.3. Silt’e Grammar

In linguistic, a grammar deals with the set of fundamental rules governing the arrangement of words in any specified natural language [14]. Like other natural languages, Silt'e has its own grammatical rules and syntax. Mostly used grammatical indicators in Silt'e language are gender, number, definiteness, personal pronouns, adjectives, adverbs, prepositions, and negations. The detail descriptions of these indicators are presented as follow.

- **Gender**

In Silt'e language gender is the syntactical category with sex. It is determined in a natural manner; hence peoples and animals assign gender based on their sexual characteristics, but there are a few words available to express gender for inanimateness. For example, gender for astronomical elements ወሪ(warii) ‘moon’ define the masculine gender and አይሪቱ(ayiriite) ‘sun’ define the feminine gender [15].

- **Number**

Silt'e language has formal means to express differences of quantity. Grammatically it uses the three numbering techniques to express a number of nouns and pronouns; these are singular, plural and

paucity. Among those only paucities is marked the rest are unmarked. As an example, ሲን[siin] refer the single or large number of cups, but ሲንቸ[siinca] mention a small number of cups. Noun paucity in Silt'e formed by suffixation of -ቸ [ca] [15]. The following table shows some noun paucity.

Noun	Suffixation	paucity	Meaning
ጣይ[tayii]	-ቸ[ca]	ጣይቸ tayiica]	'sheep'
እንዳች[indaac]	-ቸ[ca]	እንዳችቸ[indaaciica]	'women'
ከራብ[karaab]	-ቸ[ca]	ከራብቸ[karaabcaa]	'oxen'
ባሊቅ[baliiq]	-ቸ[ca]	ባሊቅቸ[baliiqcaa]	'Elderly men'

Table 2.3 Paucity formation using suffixation

The other way of forming noun paucity is by reduplicating the last consonant of the noun. The following table shows this process.

Noun	Noun paucity with reduplication	meaning
ገኛ[ganaa]	ገኛኛ[gananoo]	'horses'
ደረሰ[darasaa]	ደረሳሶ[darasasoo]	'religious student'
ገረጃ[garajaa]	ገረጃጆ[garajajoo]	'young females'

Table 2.4 Paucity formation using consonant reduplication

Definiteness

Definiteness in Silt'e language is a semantic feature of nouns identifying entities that are recognizable in a certain context. There are two definite suffixes -ኡ [ii] for masculine and -ቱ [te] for feminine. Instead of definiteness function, they also play the numbering role. Look at, the

grammatical expression ዌሴቴ[wesetee] this expression suggest about a single enset plant and ኡመቲኢ[uumatii] denote a large number of peoples in place of its definiteness [15].

Personal Pronoun

Personal pronouns of the Silt'e language are showed in the following table.

First person		Second person		Third person	
Singular	Plural	Singular	Plural	Singular	Plural
ኢሄ[ihee] 'I'	ኢኘ[inaa] 'We'	አሽ[aash](f) አተ[ataa](m) 'You'	አቱም[atuum] 'You'	ኡሀ[uuha](m) 'He' ኢሸ[iisha](f) 'She'	ኡሁን[uuhun] 'They'

Table 2.5 Silt'e personal pronouns

Personal pronouns depicted in the table 2.6 also have gender markers. These are defined in the following table.

First person		Second person		Third person	
Singular	Plural	Singular	Plural	Singular	Plural
ኤ[**ee**] Example ማጤኤ 'My Brother'	ነ[**na**] Example ማጠነ 'Our Brother'	አ|አሀ[**aa|ahaa**](**m**) Example ማጣአ|ማጣአሀ 'Your Brother' አሽ[**ash**](**f**) Example ማጣሽ	አሙ[**ammu**] Example ማጣአሙ 'Your Brother'	ከ[**ka**](**m**) Example ማጠከ 'His Brother' ሸ[**sa**](**f**) ማጠሸ 'Her Brother'	ኒሙ[**niimu**] Example ማጠኒሙ 'Their Brother'

Table 2.6 Silt'e gender marker for possessive markers

Adjective

Adjective in Silt'e used to describe characteristics of nouns. By doing this it modifies the meaning of a noun, so it has important function for daily communication.

Sample adjectives: በሬዳ [bareeda] 'beautyfull' በሬዶ [bareedoo] 'so beautyfull'

ኢማን [iimaan] ‘faith’ ኢማንቶ [bareedoo] ‘so faithy’

ሸባል [siibal] ‘sung’ ሸባልቶ [bareedoo] ‘singer’

In Silt’e language two techniques are used to form adjectives, these are inflectional and derivational adjectives [15].

Adverb

The adverb is part of communication in Silt'e language. It expresses manner, place, time, frequency and level of certainty. Some Silt'e adverbs are listed in the following table.

Adverb of frequency	Adverb of manner	Adverb of place	Adverb of time
ሁለም ነግ ‘always’	ፈየኮ ‘very’	ሂኔ ‘here’	በዞፍ ‘later’
ሀደደ ነግ ‘sometimes ’	ኮሞ ‘fast’	ሀኔ ‘there’	ሀውጇ ‘today’
ሲረም ‘never’	ሀዴኘ ‘together’	ሁለምኤት ‘everywhere’	ጌሰ ‘tomorrow’
አለፈ አለፋኔ ‘occasionally’	ለብቾ ‘alone’		ሴስተ ‘after tomorrow’

Table 2.7 Sample Silt’e adverbs

Silt’e nouns can end in any consonant or semivowels subject to the general phonological constraints of the language or one of the short vowels አ[a], ኦ[o], እ[i], ኡ[u], ኤ[e] [15]. It can be derived from verbs [15]. Morphophonemic changes of nouns with prefix and suffix are given in the following table.

Prefix morphophonemic	Suffix morphophonemic
Cአ + ወ → Cኦ Example: -በ[baa] + ወሪ[waarii] ‘month’ → ቦሪ[boorii] ‘in a month’ Cአ + እ →Cኢ Example:	Final vowel of a word is lengthened:- 1. Before acc.suffix -ነ[-na]. Example: ደቺ[daci] ‘land’→ ደቺይነ[daciina] ‘our land’ 2. before pragmatic marker -ሞ[-m] and -ወ[-w] 3. before masculine definite article

-ቢ[bii] + እንግር[ingiir]‘foot’ → ቢንግር[biingiir] ‘in a foot’	Example: ጨሎ[cuulo] →ጨሎይ ‘cuulooy’

Table 2.8 Noun morphology with prefix and suffix

- **Verb**

Silt’e verbs are two types. These are the original verb and derived verb. The derived verb obtained by inflection and derivation. The original verbs have from two to four radicals. Sample original verbs are shown in the following table.

Two radical	Three radical	Four radical
ሄደ[haade] ‘go’	ሀረሰ[haraasa] ‘plough ’	አዋለከ[awaalaka] ‘speak’

Table 2.9 Sample Silt’e verbs

The final letter of a verb can be of four kind consonant only, consonant followed by ኣ[a], non-palatal consonant followed by ኣ[a] and palatalized to ኤ[e] [15].

2.2. Overview of IR system

Information retrieval refers to the extraction of user-specified information from documents and files, ranging from books to online blogs, journals, and academic articles [3] The primary objective of IR is quickly and precisely retrieving from a collection a subset of information related to the user‘s interests [16]. According to Christopher D.Manning. Information retrieval (IR) is the process of looking for material (usually documents) of an unstructured nature that satisfies an information need of the user from within large collections (usually stored on computers). IR can also cover other kinds of data and information beyond textual document; it could be an image, multimedia or other data. The main concept in IR is finding relevant resources on the net that satisfies user information needs. A computerized system which can perform information retrieval system is called information retrieval system [3].

The goal of any IR system is to respond to user-requested information by providing reference documents that meet the desired criteria. Different factors determine whether or not a document is relevant to a user‘s query. One of these factors is the documents themselves the scope of their content, how they are written, etc. The user is also an important factor (e.g. user knowledge, the reason for the search, etc.) [17]. Because a document‘s relevance to a user‘s interest is subjective and depends solely on the user‘s judgment, IR systems may not be 100 percent accurate. What an IR system can do, however, is propose methods that can estimate the efficiency of the results and whether they meet the user‘s information needs to be based on his or her query [17]. Relevancy is difficult to measure. Many factors can determine a result‘s relevance, but one of the most important factors is user satisfaction. If the user likes a document, the document will be considered relevant; if not, it will be considered irrelevant. Therefore, researchers are constantly trying to find new methods for defining the relevance of documents to a user‘s query. There are many theories related to document relevance called formal models [18].IR models differ from one another based on the various ways they compute the weight of matching documents. Each IR model serves some purposes better than others. Despite these differences, every IR system should maintain three basic operations: document representation, query formulation, and a process for matching queries with documents [18]. Figure 2.1 illustrates the basic operation of IR. The square shapes represent data, and the oval shapes represent processes.

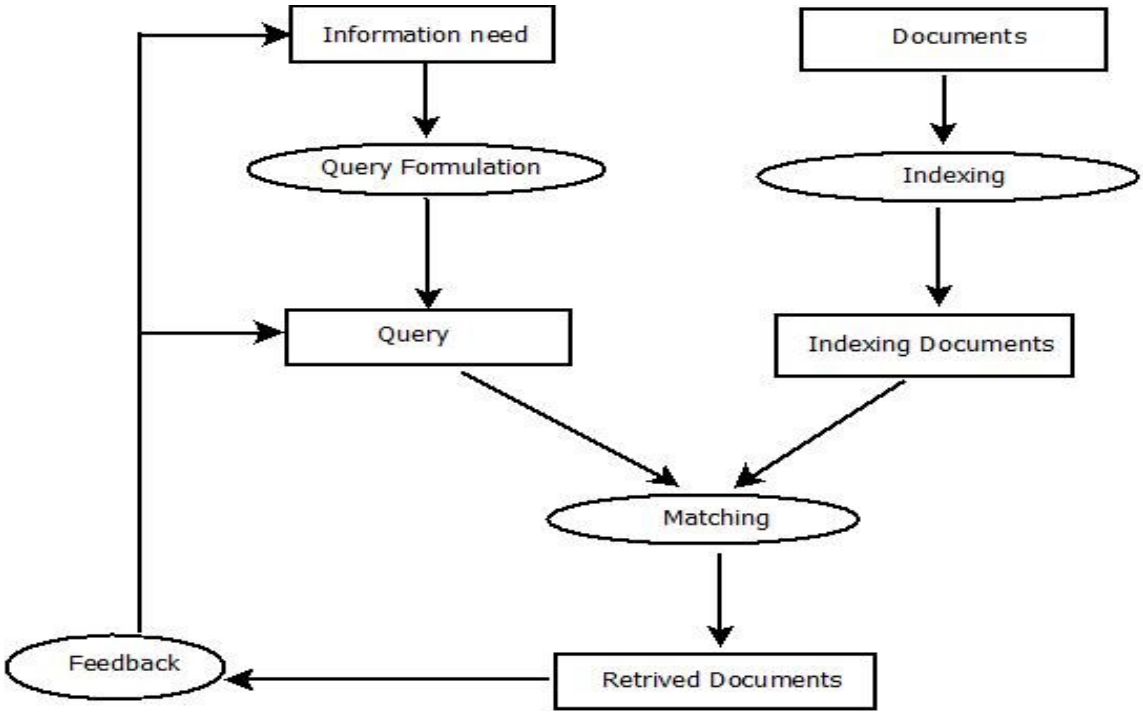

Figure 2.1 General information retrieval process

[18]

2.2.1. The Indexing Process

Representing the documents is usually called the indexing process. The process takes place off-line, that is, the end user of the information retrieval system is not directly involved. The indexing process results in a representation of the document [18]. Indexing is the process of extracting terms from the document in a collection and stores it in specified file format. The main aim of an indexing is to speed up the searching operation. The major steps it involves are preparing documents to index, tokenize, and remove stop-words, stemming, term weighting and store indexed file [3].

According to, D. Hiemstra [18], 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. An index file consists of records, called index entries. Index files are much smaller than the original file (because it is recommended to use keyword terms than full text). The

usual unit for indexing is the word called Index term. Index terms are used to look up records in a file [2].

2.2.2. Query Processing

Users do search just for a need of information. The process of representing their information need is often referred to as the query formulation process. The resulting representation is the query. In a broad sense, query formulation might denote the complete interactive dialogue between system and user, leading not only to a suitable query but possibly also to the user better understanding his/her information need [18].

Once the document is indexed and ready for retrieval process, the next step is to translate the information need of user into query language provided by the system. This process involves a serious of steps [19]. First, the user specifies his/her information needs using the natural language (e.g. Silt'e, English, Amharic, etc.) supported by the IR system. Second, the system transforms the query into the logical format by applying text operation, which is also used when the document was indexed.

2.2.3. The Matching Process

The comparison of the query against the document representations is called the matching process. The matching process usually results in a ranked list of documents. Users will walk down this document list in search of the information they need. Ranked retrieval will hopefully put the relevant documents towards the top of the ranked list, minimizing the time the user has to invest in reading them [18]. The result of the matching is a ranked list of documents according to their likelihood of relevance [19]. Baeza-Yates et al [2] 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 [18], the theory behind ranking algorithms is a critical part of information retrieval system. They attempt to display documents in decreasing order based on their similarity score with the query. Most of the time documents that are considered as relevant to users get the biggest score and displayed at the top of the retrieved list. Thus, IR models guide the process of matching and ranking relevant documents.

2.3. Information Retrieval Models

Modeling in IR is a complex process aimed at producing a ranking function [2]. To search relevant documents, you need a retrieval model. A retrieval model is a framework for defining the retrieval process by using mathematical concepts. It provides a blueprint for determining documents relevant to a user need, along with reasoning of why a document ranks above another. The model you choose depends a lot on your information retrieval need [1]. In this section, we give a brief description of well-known information retrieval models.

2.3.1. Boolean Retrieval Model

The Boolean retrieval model was used by the earliest search engines and is still in use today. It is also known as exact-match retrieval since documents are retrieved if they exactly match the query specification, and otherwise are not retrieved. Although this defines a very simple form of ranking, Boolean retrieval is not generally described as a ranking algorithm. This is because the Boolean retrieval model assumes that all documents in the retrieved set are equivalent in terms of relevance, in addition to the assumption that relevance is binary. The name Boolean comes from the fact that there only two possible outcomes for query evaluation (TRUE and FALSE) and because the query is usually specified using operators from Boolean logic (AND, OR, NOT) [20]. Boolean model is the oldest model of information retrieval. In the Boolean, there are three basic logical operators AND, OR and NOT. AND is the logical product, OR is the logical sum and NOT is the logical difference. AND is used to group set of terms into single query/statement. For example ‘Information and Technology' is two term query combined by ‘AND'. In such case, the only document indexed with both terms will be retrieved. If terms in the user query are linked by operator OR, documented with either of terms or all terms will be retrieved. For example, if the query is Information OR Technology, the document containing Information, or Technology, or Information Technology will be retrieved [14].

Drawbacks of the Boolean Model. Retrieval based on binary decision criteria with no notion of partial matching No ranking of the documents is provided (absence of a grading scale). Information need has to be translated into a Boolean expression, which most users find awkward. The Boolean queries formulated by the users are most often too simplistic. The model frequently returns either too few or too many documents in response to a user query [2].

2.3.2. Vector Space Model

The vector space model (VSM) is a retrieval model that can be used to rank the indexed document with respect to a user query. The query response contains the documents, along with scores assigned to it, which signifies the degree of similarity with the user query. The document with the highest score is supposed to be the best match [1]. Any text can then be represented by a vector in this two-dimensional space. To assign a numeric score to a document retrieved by a query, the model measures the similarity between the query and vector created by vocabulary in the index versus documents in the corpus. The similarity between two vectors is once again not inherent in the model. Typically, the angle between two vectors is used as a measure of divergence between the vectors. The similarity can be measured using cosine similarity or dot product or others [2].

Boolean matching and binary weights is too limiting the vector model proposes a framework in which partial matching is possible this is accomplished by assigning non-binary weights to index terms in queries and in documents term weights are used to compute a degree of similarity between a query and each document the documents are ranked in decreasing order of their degree of similarity [2].

For the vector model:

- The weight $w_{i,j}$ associated with a pair (k_i, d_j) is positive and non-binary
- The index terms are assumed to be all mutually independent
- They are represented as unit vectors of a t- dimensional space (t is the total number of index terms)
- The representations of document d_j and query q are t-dimensional vectors given by

$$\overrightarrow{d_j} = (w1_j, w2_j, \ldots, wt_j) \qquad \textit{Equation 2-1}$$

$$\vec{q} = (w1_q, w2_q, \ldots, wt_q) \qquad \textit{Equation 2-2}$$

The vector space model proposes to evaluate the degree of similarity of the document dj with regard to the query q as the correlation between the vectors d_j and q. This correlation can be quantified by the cosine of the angle between these two vectors as

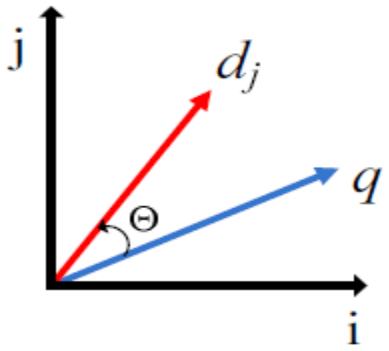

Figure 2.2 Similarity between a document d_j and a query q [2]

$$\cos(\boldsymbol{\theta}) = \frac{\vec{d_j} \cdot \vec{q}}{|\vec{d_j}| \times |\vec{q}|}$$ *Equation 2-3*

$$\text{sim}(d_j, q) = \frac{\sum_{t=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w^2{}_{i,j}} \times \sqrt{\sum_{j=1}^{t} w^2{}_{i,q}}}$$ *Equation 2-4*

Generally The Vector 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 the relevant document. The final step is ranking the documents to show the best matching with respect to the provided query by the user.

The main advantages of the vector model are:

1. term-weighting improves the quality of the answer set
2. partial matching allows retrieval of docs that approximate the query conditions
3. cosine ranking formula sorts documents according to a degree of
4. the similarity to the query document length normalization is naturally built-in into the ranking

Although it is the resilient and most popular ranking strategy nowadays, 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 [2].

2.3.3. Probabilistic Model

The probabilistic model in IR was developed following the basic probabilistic theory. The probability of a document D being relevant to the user's query Q is represented by

$$P(R|Q,D) = \frac{P(D|R,Q) * P(R|Q)}{P(D|Q)} \qquad \textit{Equation 2-5}$$

In general, probabilistic models avoid shifting uncertainties to users by providing solutions for computing relevance certainty, whereas the Boolean model searches the documents based on Boolean logic and classical set theory that uncertainties remain as uncertainties for users to deal with [7]. While in comparison to the vector space model whose documents are retrieved and ranked by the degree of similarity, probabilistic models are more interpretable and computable.

2.3.3.1. Binary Independence Model

The binary independence retrieval model [3] is the model that has traditionally been used with PRP, and it develops the idea with precise assumptions shared by most of other probabilistic models. The "binary" here introduces the idea equivalent to Boolean, where the documents and queries are both represented in binary term incidence vectors. And the "independence" here suggests that the terms occurring in the documents share no association between each other, and thus are modelled independently.

More specifically, we model the probability that a document is relevant to the query with the term incidence vector $P(R|\vec{x},\vec{q})$ with the application of Bayes theorem, the probability of relevance is:

$$P(R=1|\vec{x},\vec{q}) = \frac{P\,(\vec{x}|R=1,\vec{q}) * P(R=1|\vec{q})}{P(\vec{x}|\vec{q})} \qquad \textit{Equation 2-6}$$

$$P(R=0|\vec{x},\vec{q}) = \frac{P\,(\vec{x}|R=0,\vec{q}) * P(R=0|\vec{q})}{P(\vec{x}|\vec{q})} \qquad \textit{Equation 2-7}$$

Here we denote $P(\vec{x}|R=1,\vec{q})$ *as* the probability of a relevant document being retrieved while denoting $P(\vec{x}|R=0,\vec{q})$ as an irrelevant, besides, the $\vec{x}$ the document representation. According to the classic probabilistic theory, since a document is either relevant or non-relevant to a query, we must have that:

$$P(R=1|\vec{x},\vec{q}) + P(R=0|\vec{x},\vec{q}) = 1 \qquad \textit{Equation 2-8}$$

In order to derive a ranking function for query terms, we set O as a constant for the given query, and thus,

$$O(R|\vec{x},\vec{q}) = \frac{P(R=1|\vec{x},\vec{q})}{P(R=0|\vec{x},\vec{q})} = \frac{\frac{P(R=1|\vec{q}) * P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|\vec{q})}}{\frac{P(R=0|\vec{q}) * P(\vec{x}|R=0,\vec{q})}{P(\vec{x}|\vec{q})}} = \frac{P(R=1|\vec{q})}{P(R=0|\vec{q})} \times \frac{P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|R=0,\vec{q})} \qquad \textit{Equation 2-9}$$

2.3.3.2.Bayesian Networks Model

According to Baeza-Yates et al [2] Bayesian networks are directed acyclic graphs (DAGs) in which the nodes represent random variables the arcs portray causal relationships between these variables the strengths of these causal influences are expressed by conditional probabilities The parents of a node are those judged to be direct causes for it This causal relationship is represented by a link directed from each parent node to the child node The roots of the network are the nodes without parents.

Let:

x_i be a node in a Bayesian network G

Γx_i be the set of parent nodes of x_i

The influence of Γx_i on x_i can be specified by any set of functions $F_i(x_i, \Gamma x_i)$ that satisfy

$$\sum_{\forall x_i} F_i(x_i, \Gamma x_i) = 1 \qquad \textit{Equation 2-10}$$

$$0 \leq F_i(x_i, \Gamma x_i) \leq 1$$

Where x_i also refers to the states of the random variable associated to the node x_i

A Bayesian network for a joint probability distribution $P(x_1, x_2, x_3, x_4, x_4, x_5)$

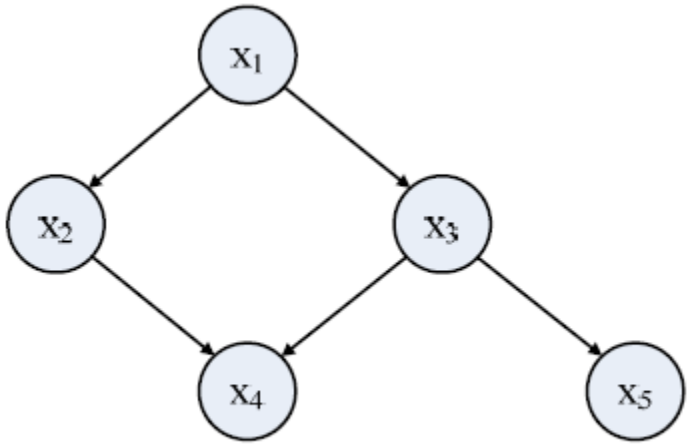

Figure 2.3 Bayesian network for a joint probability distribution [2]

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

2.3.3.2.1. Bayesian Inference Network Model

According to Baeza-Yates and Ribeiro-Neto [2], there are two traditional schools of thought in probability which are based on the frequentist view and the epistemologist view. The frequentist views probability as a statistical notion related to the laws of chance. The epistemologist views probability as a degree of belief whose specification might be devoid of statistical experimentation. Bayesian Inference Network model is built up on epistemologist view of the information retrieval problem. It associates random variables with the index terms, the documents and the user queries. The model computes $P(I|D)$, the probability that a user's information need (I) is satisfied given a particular document (D). This probability can be computed separately for each document in a collection. The documents can then be ranked by probability, from highest probability of satisfying the user's need to lowest [2].

2.3.3.2.2. Bayesian Belief Network Model

According to Baeza-Yates and Ribeiro-Neto [2], stated in Bayesian belief network the user's query q is modelled as a network node to its associated random variable. Whenever q completely covers the concept space C the variable is set to 1. Therefore belief network computes the probability of q, (i.e.$P(q)$) by the degree of coverage of the space C by q.

Document d_j is modelled as the network node to its associated binary random variable. If d_j completely covers the concept space C the variable set to 1. To compute the probability of d_j ($P(d_j)$), compute the degree of coverage of the space C by dj. In belief network documents and the user query modelled as subsets of index terms. Each subset is interpreted as the concept in the concept space C [2].

The ranking principle in belief network expressed as, the degree of coverage provided to the concept d_j by the concept q. $P(d_j|q)$ is adopted as the rank of the document d_j with respect to the query q [2].

In general, probabilistic models attempt to capture the information retrieval problem within a probabilistic framework. Unfortunately, the probabilistic model has got its own drawbacks. First, the probability theory analysis takes much more time and efforts, and it offers unnecessary theoretical burden on the researcher. Second, the probabilistic model needs to guess the initial separation of documents into relevant and non-relevant sets. Third, the model does not take into account the frequency with which an index term occurs inside a document. In another word, all weights are binary [2] [21].

However, the probabilistic model has several potential advantages [2] [21]. The first, advantage is the expectation of retrieval effectiveness that is near to optimal relative to the evidence used is high. Second, it has less reliance on traditional trial and error retrieval experiments. Third, each document's probability of relevance estimate can be reported to the user in ranked output. It would presumably be easier for most users to understand and base their stopping behaviour (i.e., when they stop looking at lower ranking documents).

2.3.3.3. 2-Poisson Model

In 1974, Bookstein and Swanson [22] first proposed a 2-Poisson indexing model for term frequencies to suggest some ideal ways of incorporating certain variables into the probabilistic models. Robertson and Walker in 1994, further studied the topic and proposed that within-document term frequency, document length and the within query term frequency are the three major variables concerned for the weighting function [14].

$$W(\underline{X}) = \log \frac{P(x|R)P(\underline{0}\,|\overline{R})}{P(\underline{x}|\overline{R})P(\underline{0}\,|R)} \qquad \textit{Equation 2-11}$$

Where $\underline{X}$ is the vector of information about the document, R denotes the relevance, $\overline{R}$ is the non-relevance, and $\underline{0}$ is the reference vector representing the zero-weighted document.

The 2-Poisson model is constructed based on the assumption that by determining which one of the two Poisson distributions the selected term should belong to (choose one in between), it will be possible to decide whether modelled should be assigned to a document. The two Poisson distributions which mentioned above are modelled in the distribution of within document frequencies for relevant documents, and the distribution of within document frequencies for non-relevant documents with a different mean. The problem with this two-Poisson model is that it needs probabilities conditioned on relevance. According to Robertson and Sparck Christopher D [3], their final formula has too many parameters which no data could practically fit into them.

$$p(TF_i = tf|rel) = p_{i1}E_{i1}(tf) + (1 - p_{i1})E_{i0}(tf)\ (1) \qquad \textit{Equation 2-12}$$

Where E_{i1} denotes the eliteness in Robertson's paper, referring to a relevant term, and the E_{i0} referring to the irrelevance of certain term. Then the probability function (1) leads to an equation for the term weighting of an elite term:

$$W_i^{elite} = \log \frac{\big(p_1E_1(tf) + (1 - p_1)E_0(tf)\big)\big(p_0E_1(0) + (1 - p_0)E_0(0)\big)}{\big(p_1E_1(0) + (1 - p_1)E_0(0)\big)\big(p_0E_1(tf) + (1 - p_0)E_0(tf)\big)} \qquad \textit{Equation 2-13}$$

Where E and $\overline{E}$ refer to the eliteness and non-eliteness respectively. Eliteness could be interpreted as a form of aboutness: if the term is elite in the document, the document is considered to be about the concept.

The n-Poisson model is an extension of the 2-Poisson model to the n-dimensional case, assuming that there are **n** classes of documents in which the term **t** appears with different frequencies.

2.3.3.4. BM25 Model

According to Baeza-Yates and Ribeiro-Neto, [2] BM25 was created as the result of a series of experiments on variations of the probabilistic model. A good term weighting is based on three principles inverse document frequency, term frequency, and document length normalization. The classic probabilistic model covers only the first of these principles. This reasoning led to a series of experiments with the Okapi system, which led to the BM25 ranking formula.

Initially, the Okapi system used the BM1 formula in probabilistic model as a ranking formula when non-relevance information is given

$$\mathrm{sim}(d_j, q) \sim \sum_{k_i \in q \Lambda k_i \in d_j} \log \frac{N - n_i + 0.5}{n_i + 0.5} \qquad \textit{Equation 2-14}$$

The first idea for improving the ranking was to introduce a term-frequency factor $F_{i,j}$ in the BM1 formula this factor, after some changes, evolved to become

$$F_{i,j} = S_1 \times \frac{f_{i,j}}{K_1 + f_{i,j}} \qquad \textit{Equation 2-15}$$

Where $f_{i,j}$ is the frequency of term k_i within document d_j K_1 is a constant setup experimentally for each collection S_1 is a scaling constant, normally set to K_1 = (K_1 + 1) If K_1 = 0, this whole factor becomes equal to 1 and bears no effect on the ranking

By adding the document length normalization principle into the formula above, this equation is transformed to,

$$F'_{i.j} = S_1 \times \frac{f_{i,j}}{\frac{K_1 \times \text{len}(d_j)}{avg\_doclen} + f_{i,j}}$$ *Equation 2-16*

Where $len(d_j)$ is the document length of d_j and avg_doclen denotes the average document length of the document collection.

Aside from that, the BM1 function also used a correction factor G_j, q which depends purely on the document length and query length,

$$G_j, q \ = K_2 \times \text{len}(q) \times \frac{avg_{doclen} - \text{len}(d_j)}{avg_{doclen} + \text{len}(d_j)}$$ *Equation 2-17*

Where K_2another constant is applied and $\text{len}(d_j)$is the length of the query.

Thus, the last step is to apply the third factor to the term frequencies within queries,

$$F_j, q \ = S_3 \times \frac{f_{i,q}}{K_3 + f_{i,q}}$$ *Equation 2-18*

Where S_3 again, is the scaling constant related to K_3, which is a constant as well. And $f_{i,q}$ is the frequency of term K_i within query q.

With the integration of the three factors introduced above, the BM1 function was led to BM 11, and BM15 respectively, where $K_i\left[q, d_j\right]$ is a short notation for $k \in q \text{^} k_i \in d_j$.

BM15:

$$sim\left(d_{j,q}\right) \sim G_2 + \sum_{k \in q \text{^} k_i \in d_j} \frac{s_1 f_{i,j}}{\left(k_1 f_{i,j}\right)} \times \frac{s_3 f_{i,j}}{\left(k_3 f_{i,j}\right)} + \log \frac{N - n_i + 0.5}{n_i + 0.5}$$ *Equation 2-19*

BM11:

$$sim\left(d_{j,q}\right) \sim G_2 + \sum_{k \in q \text{^} k_i \in d_j} \frac{s_1 f_{i,j}}{\frac{k_1 \times len(d_j)}{avg\_doclen} + f_{i,j}} \times \frac{s_3 f_{i,j}}{\left(k_3 f_{i,j}\right)} + \log \frac{N - n_i + 0.5}{n_i + 0.5}$$ *Equation 2-20*

The widely used BM25 ranking formula we use today is structured by combining the BM11 and BM15 ranking formulae, mainly the term frequency factors

$$B_{i,j} = \frac{(K_1 + 1) f_{i,j}}{K_1 \left[(1 - b) + b \frac{len(d_j)}{avg\_doclen} \right] + f_{i,j}} \qquad \textit{Equation 2-21}$$

Where **b** is a constant with values in the interval [0, 1] If **b** = 0, it reduces to the BM15 term frequency factor If **b** = 1, it reduces to the BM11 term frequency factor for values of **b** between 0 and 1, the equation provides a combination of BM11 with BM15. And thus, the BM25 is presented as follows:

$$sim_{BM25(d_{j,q})} \sim \sum_{k_i [q, d_j]} B_{i,j} \times \log \left(\frac{N - n_i + 0.5}{n_i + 0.5} \right) \qquad \textit{Equation 2-22}$$

where K_1 and **b** are empirical constants K_1 = 1 works well with real collections **b** should be kept closer to 1 to emphasize the document length normalization effect present in the BM11 formula, For instance, **b = 0.75** is a reasonable assumption Constants values can be fine-tuned for particular collections through proper experimentation.

The BM25 has been one of the most efficient and widely-used information retrieval weighting models in the past three decades. Unlike other probabilistic models, the BM25 could be computed without relevance information. Since BM25 almost outperforms classic vector model for general data collections, it has substituted the vector space model to be used as a baseline for comparison for decades [23].

2.4. Related Works

2.4.1. IR Systems for International Languages

Understanding the role of information retrieval technology in the current information age, different researchers work on many aspects of information retrieval systems for different languages that can solve the information accessing problems on those languages and make the retrieving process more efficient.

There are different issues to be considered in designing information retrieval systems such as the test collection, collection indexing approach, retrieval techniques, and overall performance evaluation. Performance evaluation is crucial at many stages in information retrieval system improvement. At the end of the development process, it is significant to show that the final retrieval system achieves an acceptable level of performance and that it represents a significant improvement over present retrieval systems [24].

Improvement in probabilistic information retrieval model - rewarding terms with high relative term frequency by Runjie Zhu [23] underline on the result of the integration of relative term frequency weighting and the term frequency normalization based on document length, and its application to the classic probabilistic weighting function of BM25. He tries to propose the relative term frequency to be integrated into traditional probabilistic models.

Search queries in an information retrieval system for Arabic language texts by Zainab, M. Albujasim [25] implementation and evaluation of the search functionality using the Vector Space Model with several weighting schemes. She uses the ISRI stemming algorithms as the essential stemming technique and the common Arabic stop word list for building an inverted index for Arabic-language documents. She evaluates the implementation on a corpus containing selected technical papers published in Arabic-language journals. The average recall is only 10%. The proposed IR system has a relatively high recall, between 40% and 60% without stemming and between 70% and 90% with stemming.

2.4.2. IR Systems for Ethiopian Languages

In Ethiopia, there are more than 86 languages. These languages are grouped in different language families and each family has its own scripting and grammar style. More documents are available in organizations, schools and on the Internet that is prepared using local languages. Supporting the target users to access and organize the enormous and widespread amount of information is becoming a primary issue. Information retrieval systems ability to retrieve relevant documents became critical due to the existence scripting difference exists in local languages [11].

Understanding the role of information retrieval technology in the current information age, different researchers work on many aspects of information retrieval systems for different Ethiopian

languages that can solve the information accessing problems on those languages and make the retrieving process more efficient.

Gezehagn works on Afaan Oromo language in the objective of designing information retrieval system that can enable to search for applicable Afaan Oromo text corpus using Vector Space Model. The methods utilized in his study had been the literature review, corpus preparation, and python 2.7.3 programming language to design the prototype. According to the experimentation, he made the system registered 0.575 precision and 0.6264 recall which is promising to design Afaan Oromo information retrieval system that searches within large corpus [26].

The other research work on Amharic text retrieval was done by Wordofa [27]. He investigated semantic indexing and document clustering techniques to improve the performance of Amharic information retrieval. His experiment show semantic indexing has improved the performance of Amharic information retrieval system from 60% to 66% F-measure. Moreover, semantic indexing and document clustering reduced the computational cost of the system by 1/k* Terms Reduction Ratio factor, where K is the number of the cluster.

Probabilistic Information Retrieval System for the Amharic Language developed by Amanuel H [28] observed in previous works using vector space model of Amharic Information Retrieval system. Tried to design a probabilistic based information retrieval system for the Amharic language based on the probabilistic model. To test the prototype system developed, 300 Amharic document corpus. Additionally, 10 test queries were selected by the researcher to test the performance of the system. In designing the IR system, Binary Independent Model (BIM) model become decided on and applied. System evaluation becomes primarily based at the F-measure and registered on the average of 73% F-measure without controlling those issue about synonyms Furthermore polysemy terms that exist in Amharic text.

A Probabilistic Information Retrieval System for Tigrinya evolved via Atalay [29] to test the prototype system evolved, 300 Tigrinya documents and 10 queries had been used to check the technique. The prototype is developed utilizing Python 3.2.2 programming language. The system registered, after stemming and pseudo-significance feedback, an average precision 69.1%, recalls 90%, and F- measure 74.4%. This result is achieved without controlling the issue of synonyms and polysemy of terms that exist in Tigrinya textual content.

Jemal Z. developed Silt'e Text Information Retrieval System Based on Vector Space Model. He utilized Apache Lucene Solr to design the prototype. To test the prototype system developed 100 Silt'e archive corpus. Those experimentations performed using the prototype that was developed without invoking stemming algorithm scored 81.4% mean average precision. This end result is accomplished without controlling the issue of synonyms and polysemy of phrases that exist in Silt'e textual document [11].

As we have mentioned within the above pieces of literature a number of works attempted to design information retrieval system using the statistical technique for local languages. Near designing information retrieval systems for those languages, cross-lingual information retrieval systems have been also advanced.

A corpus based approach for Afaan Oromo-English Cross-Lingual Information Retrieval (CLIR) examined by (Bekele, 2011) [30]. The result about he acquired might have been a maximum Average precision about 0. 468 Furthermore 0. 316 to Afaan Oromo What's more English respectively. Despite the fact that that corpus-based approach is impacted by size Also personal satisfaction of the corpus, that effect got might have been empowering on creating Afaan Oromo-English CLIR by utilizing this approach.

To design text retrieval machine for a specific language study of automated indexing strategies play essential point for that language. Stemming algorithms are among the techniques in automated indexing. To Silt'e dialect Muzeyn K. [15] 2012 tried to design stemming algorithm. To evaluate the stemmer developed in this study, test data of size 1486 words were selected randomly from the sample texts. The experiment showed that the stemmer performs with the accuracy of 85.71 % and reduce dictionary size by 34.99% for stems.

Developing information retrieval system has a great impact on the accessing of information in a particular language. We have seen in the literature there are different information retrieval systems designing approaches and each model has its own strength and limitation. To enhance the correctness and efficiency of the retrieved information from the corpus we have to select appropriate information retrieval system development approaches.

CHAPTER THREE

3. Materials and Methods

3.1. Silt'e IR System Design and Architecture

In most information retrieval systems, the scheme architecture contains two main parts. These are indexing part and searching part. In this research, we design Probabilistic Silt'e information retrieval system architecture that has the two parts indexing and searching. Indexing is an offline process used to facilitate access to preferred information from document corpus accurately and efficiently as per users query. Searching is an online process used to scan document collection so as to find relevant documents that satisfy users need.

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

The indexing part consists of several processes. These are corpus preparation process, document analyzes like tokenization, stop word removing, stemming, synonymy and finally indexing the corpus.

The other system architecture part is searching. This part also passes sequential process. Here user provides a query to the system the system takes the query and performs preprocessing operations like tokenization, stop word removal, stemming, synonyms and calculating similarity of each term. Once all these tasks were completed similarity measure between query and documents performed to return a search result. The technique applied here is BM25 similarity measure. Finally, the ranked documents are returned based on relevance to the query.

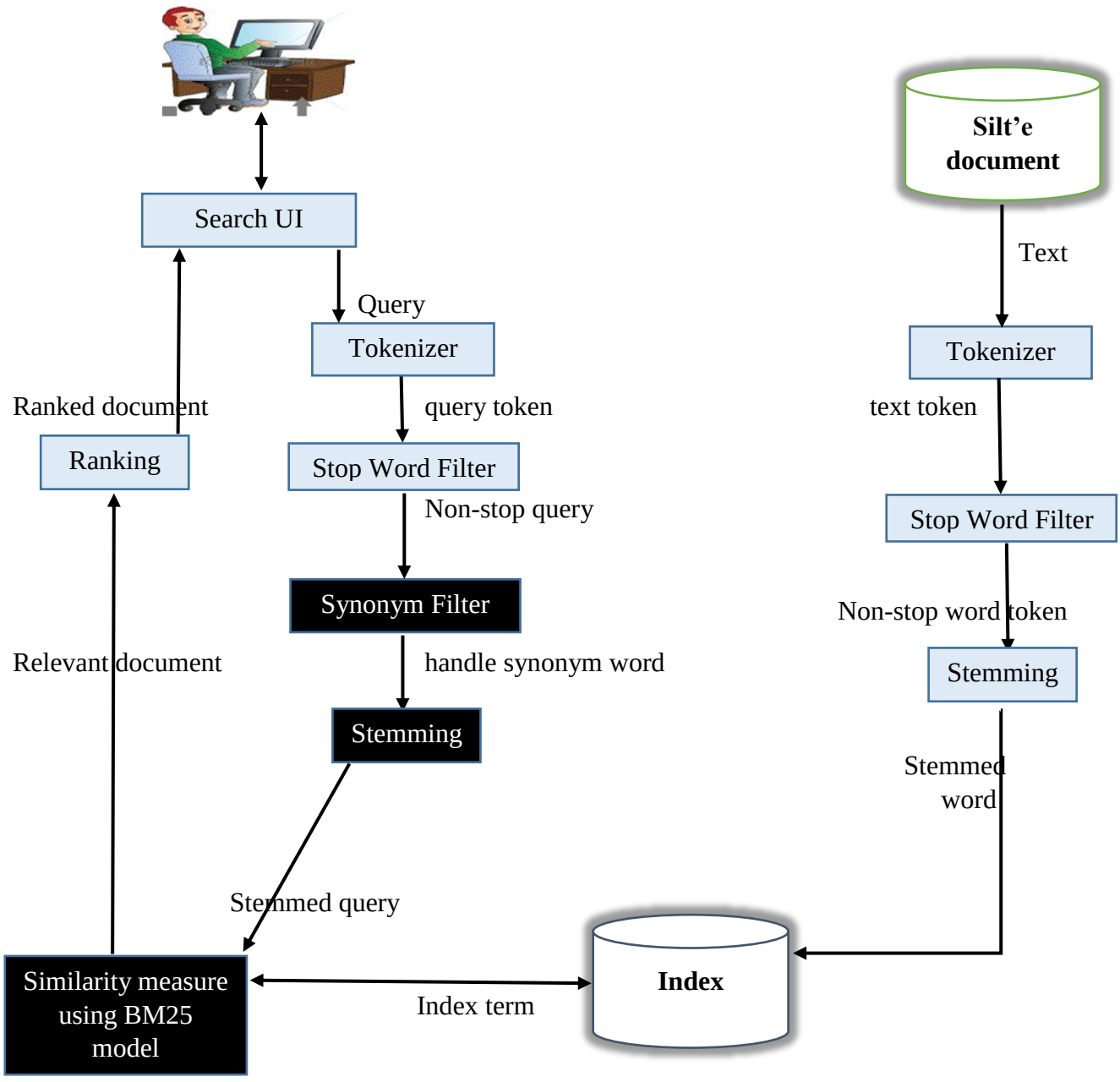

Figure 3.1 information retrieval system architecture [18].

3.2. Prototype Development Tool

We use Apache Solr version 6.4.1 to develop Silt'e information retrieval system prototype. Apache Solr is highly scalable search platform built using Apache Lucene. It is written in Java and runs as a stand-alone server [1]. Simple commands like start and stop being used to start and stop it. It is extensively used for indexing a large collection of documents and supporting full-text search. Solr

attract peoples to use for their search system because it is the most widely used search solutions in different domains. Some attractiveness features of Solr are:-

- **Highly scalable and fault-tolerant**: - we can add or remove computing capacity to Solr, just by adding or removing replicas of your instance as needed.
- **Enterprise ready**: - many organizations in the different domain trusted on Solr for their search requirement.
- **Full-text search**: - it provides all the matching capabilities needed and autocomplete.
- **Flexible and extensible**: - we can modify and customize components by extending the Java classes and adding the configuration for the created class in the appropriate file.
- **Easy configuration**:- as Solr is configuration based platform it is easy to configure for our work

3.2.1. Schema Design in Solr

Schema design in Solr is a description of document structure that is the precondition for indexing and searching because a search result in solr is a document. It is a basic condition to develop information retrieving systems in solr. After structuring the data a series of text analysis process are applied to control the matching actions. The Extensible Markup Language (XML) format is used to define the schema and field type outline. The following XML definition describes a general layout of a given schema design structure in solr [1].

```
<schema name="Silte" version="1.6">

Fields are defined here…

</schema>
```

The name and version are attributes, where name specifies name of the schema and version define version number for the schema syntax and semantics. Current schema syntax and semantic version for solr-6.4.1 that we use in our research are 1.6. Schema definition for Silt'e text information retrieval system prototype is presented in the appendix VI.

3.3. Data Pre-Processing and Corpus Preparation for Silt'e Document Indexing

In information retrieval system research is collecting documents needed for evaluation of the system also used for indexing [3]. These are all available text documents on different topics. We collected Silt'e text documents for this research. School text books which were collected from Silt'e Zone Education Bureau and other government organizational bureau takes the highest portion of the corpus because of other types of documents like a news article, reports were not accessed enough during our data collection time. The collection consists of 134 documents on different topics.

3.3.1. Tokenization

Tokenizer accepts the text stream, processes the characters, and produces a sequence of tokens. It can break the text stream based on characters such as whitespace or symbols [1]. Consider the statement "ሉላሉሌ ዱንጇ ሉባምቸነ ‹‹ፉገ››፣ ‹‹፣አረሺ››፣ ‹‹ቡደ››፣›› the tokens after implementing standard tokenizer splitting the sentence are "ሉላሉሌ", " ዱንጇ", "ሉባምቸነ","ፉገ","አረሺ" and "ቡደ". Therefore the purpose of tokenization is to break down a document collection into tokens so that it is easy to match with tokens in the query. As a result, it increases the relevancy of a document to a query.

```
public class StandardTokenizerFactory extends TokenizerFactory {

        private final int maxTokenLength;

public StandardTokenizerFactory(Map<String,String> args) {

        super(args);

maxTokenLength = getInt(args,
"maxTokenLength",StandardAnalyzer.DEFAULT_MAX_TOKEN_LENGTH);

        if (!args.isEmpty()) {

        throw new IllegalArgumentException("Unknown parameters: " + args);

        }
```

```
}

@Override

public StandardTokenizer create(AttributeFactory factory) {

StandardTokenizer tokenizer = new StandardTokenizer(factory);

tokenizer.setMaxTokenLength(maxTokenLength);

return tokenizer;

}

}
```

Algorithm 3-1 Tokenization [1]

3.3.2. Stop Word Removal

Not all terms found in the document are equally important to represent documents they exist in. Some terms are most common in documents. Therefore, removing those terms, which are not used to identify some portions of the document collection, is important. Such terms are removed based on two methods. The first method is to remove high frequent terms by counting the number of occurrences (frequency). The second method is using stop word list for the language [31]**.** In this research, we used the second method. The stop word list is adopted from Muzeyn K. [15].below shows sample stop word list used in this research.

አብተቴእብታይ	ወይ	ቢትላይም	ገፌ
እብተቴ	ታሌ	አሎነ	ሀነግነ
አብታይ	ናር	ሂነሚ	እነይ
አብተቴ	ናርት	በሁነትንሙ	ሱርዋ
እሊ	ናሩ	ሁሞተኒመዋ	ገናሚ

አሊ	ብቾ	አታይ	ሂንኩምንገ
እነይ	ለገነ	ሀነይ	ኡሀ
አነይ	ገነ	ሁኖ	ያሸ
እቲ	ሁልምከ	ገና	ዬዬ
አቲ	ለሳድባድከ	የልስ	በዬ
እቢ	ለሰባድሽ	ሉላሉሌ	ዋ
አቢ	ለሰባድኑም	በሉሌ	ለገነገናም

Table 3.1 sample stop word list

```
public void inform(ResourceLoader loader) throws IOException {
        if (stopWordFiles != null) {
        if (FORMAT_WORDSET.equalsIgnoreCase(format)) {
        stopWords = getWordSet(loader, stopWordFiles, ignoreCase);
        } else if (FORMAT_SNOWBALL.equalsIgnoreCase(format)) {
        stopWords = getSnowballWordSet(loader, stopWordFiles, ignoreCase);
        } else {
throw new IllegalArgumentException("Unknown 'format' specified for 'words' file: " + format);
        }
        } else {
        if (null != format) {
throw new IllegalArgumentException("'format' cannot be specified w/o an explicit 'words' file: " +
format);
        }
        stopWords = new CharArraySet(StopAnalyzer.ENGLISH_STOP_WORDS_SET,
ignoreCase);
        }
        }
```

Algorithm 3-2 Stop word removal [1]

3.3.3. Word Stemming

Stemming is the process of converting words to their base form in order to match different tenses, moods, and other inflictions of a word. The base word is also called a stem. It's important to note that in information retrieval, the purpose of stemming is to match different inflections on a word [1]. Stemming is the language dependent process in a similar way to other natural language processing techniques. It is often removing inflectional and derivational morphology. E.g. በስልጤ, የስልጤኛ, የስልጤ → ስልጤ. Stemming has both advantage and disadvantage. The advantage is it helps us to handle problems related to inflectional and derivational morphology. That makes words with similar stem/root word to retrieve together. This increases effectiveness IR system. Stemming has disadvantage sometimes some terms might be over stemmed, this changes the meaning of the terms in the document; different terms might be reduced to the same stem, which still enforces the system as to retrieve non-relevant documents [26] for controlling over the stemming problem we use *KeywordMarkerFilter* class.

Like other Semitic languages, Silt'e experiences complex morphological structure that resulted in very large variants of a word [24]. M.Kedir developed the Silt'e stemmer that involves the removal of both prefix and suffix but not standard Silt'e stemming algorithm support by solr so in this research we create stemming dictionary using the list of suffix and prefix from M.kedir [24].

3.3.4. Synonymy and Polysemy

Any text retrieval system must overcome the fundamental difficulty that the presence or absence of a word is insufficient to determine relevance. This is due to two basic problems of natural language: synonymy and polysemy. Synonymy refers to the fact that a single underlying concept or idea can be represented by many different terms or combinations of terms (e.g. "car" and "automobile" often refer to the same class of objects). Polysemy refers to the fact that a single term can refer to more than one underlying concept or idea (e.g. "car" may be an automobile or the head of a LISP cons cell). Because of synonymy, it is difficult to realize that two documents describe the same topic when they use different vocabulary, leading to relevant documents being rejected (false negatives). Because of polysemy, it is difficult to realize that two documents that use some of the same terms describe different topics, leading to the retrieval of unwanted documents (false

positives) [3]. In this study, we apply synonymy filtering factor for considering the problem of synonymy terms that exist in Silt'e text.

3.4. Searching Using the BM25 Model

Apache solr has different alternative for scoring the document. The alternatives VSM, BM25 DFR Similarity. Among this alternative we select BM25 similarity measures.

BM25 Similarity is the most widely used ranking algorithm among the alternative implementations provided by Lucene. This similarity is based on the Okapi BM25 algorithm, where BM stands for best matching and is a probabilistic retrieval model. It can produce better results than TF-IDF-based ranking for indexes containing small documents [1].

According to Konrad Beiske [32], BM25 model is similar to the VSM, in the sense that it takes a bag-of-words approach and considers term frequency and inverse document frequency to compute and summarize the score. But it has some prominent differences. Two of the most important differences between these algorithms are as follows:-

- ***Saturation***: In the VSM, the normalized term frequency of a document grows linearly with the growing term count and has no upper limit. In the BM25-based model, the normalized value grows nonlinearly, and after a point does not grow with the growing term count. This point is the saturation point for term frequency in BM25.

- ***Field length****:* BM25 considers the fact that longer documents have more terms, and so the possibility of higher term frequency is also more. As a term can have a high frequency due to the long length and might not be more relevant to the user query

The retrieval documents are ranked in an ordered list according to their probabilities of relevance to the query. Also, based on the within document term frequency and the query term frequency of the given search term, the search term is assigned with a weight using the weighting function in BM25.

$$w = \log \frac{(1 + doccount - docfreq + 0.5)}{docfreq + 0.5} * \frac{freq * (k_1 + 1)}{freq + k_1 * \left(1 - b + b * \frac{fieldlen}{avgfieldlen}\right)} * boost$$ Equation 3-1

Where $doccount$ the number of indexed documents in the collection, $docfreq$ represents the number of documents which contain a specific term, $freq$ is the within-document term frequency, $avgfieldlen$ is the average document length, the $fieldlen$ is document length, k_1 is optional parameter controls the saturation level of the term frequency. The default value is 1.2. A higher value delays the saturation, while a lower value leads to early saturation, b is optional parameter controls the degree of effect of the document length in normalizing the term frequency. The default value is 0.75. A higher value increases the effect of normalization, while a lower value reduces its effect

3.5. IR System Evaluation

A core issue in IR research is the evaluation of IR systems. The standard approach to information retrieval system evaluation revolves around the notion of relevant and non-relevant documents. With respect to a user information need, a document in the test collection is given a binary classification as either relevant or non-relevant [3]. In this study, the three widely used techniques precision, recall, and F-measure are used to measure the effectiveness of the IR system designed.

Precision and recall are two important measures for evaluating the effectiveness of the information retrieval system. Precision is the fraction of retrieved documents that are relevant, and recall is the fraction of relevant documents that are retrieved [1]. Precision and recall can be computed using this formula:

$$Precision = \frac{Relevant \cap Retrieved}{Retrieved}$$ Equation 3-2

$$Recall = \frac{Relevant \cap Retrieved}{Relevant}$$ Equation 3-3

Precision is a measure of quality, whereas recall is a measure of quantity. So, high recall means that an algorithm returned most of the relevant results, and high precision means that an algorithm returned more relevant results than irrelevant [1].

F-measure can be used as a single measurement for the system, as it combines both precision and recall to calculate the result. It computes harmonic mean of precision and recall as follows:-

$$F\_measure = \frac{2 \times Precision \times Recal}{Precision + Recall} \qquad \textit{Equation 3-4}$$

CHAPTER FOUR

4. Result and Discussion

In this chapter, we present the experiments conducted using the architecture designed in chapter three, and the findings from the experiment with some detailed discussion. In this research, an attempt has been made to design a probabilistic information retrieval system for Silt'e language. The system has both the indexing and searching parts.

4.1. Test Document Preparation

The researcher has utilized different sources of text for the development of the probabilistic information retrieval system for Silt'e language and the experiments. So far there is no standard established test collection for Silt'e information retrieval testing. Experiments in this study are based on sets of documents and queries set up by the researcher.

In Ethiopia several information retrieval system developed in different language. Most of the researcher uses 100 to 300 text documents for testing purpose in deferent cluster. A total of 134 Silt'e documents were used as a document corpus to test the prototype system. The documents are obtained from Silt'e zone different government sector.

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

	Types of Documents	Number of Documents
1	Health related	28
2	Education related	30
3	Social related	26
4	Politics related	24
5	Art related	26
Total		134

Table 4.1 Corpus used for the development of IR system

4.2. Query Preparation

For this research, we prepared ten free text queries. We tried to make the queries well representative and inclusive to test the designed prototype. These queries are marked across each document as either relevant or irrelevant to make relevance evaluation as in appendix II for each document. The main importance of having identified queries is to evaluate the performance of the system [2]. These queries are selected subjectively by the research after reviewing content each document manually. For the performance evaluation purpose, the selected queries are given in table 4.2.

No	Query
ነ	የሰብ ወልድ ተረዘቆት
2	የልባስ ጡሃርነት ቂሮት
3	በደም ኢትላለፋነን ነቶ
4	ስልጤዋ ያየር ንባረትከ
5	የስልጤ አፍ
6	ያበሮስ ቅጨ ሃለት
7	የቶጵያ አፍች ሩክቦኒሙ
8	ያፍ አበሮስ
9	በሸርቅ አዘርነት በርበሬ
ነ0	ለባድ ወልድ

Table 4.2 selected queries for system evaluation

4.3. Description of the Prototype System

In developing the prototype of probabilistic based IR system for Silt'e language, several components are integrated together as described in Figure 4.1. Those components describe the major activities done in the probabilistic retrieval model development. This system has to index and searching. In this part, we discuss the indexing process and searching the document.

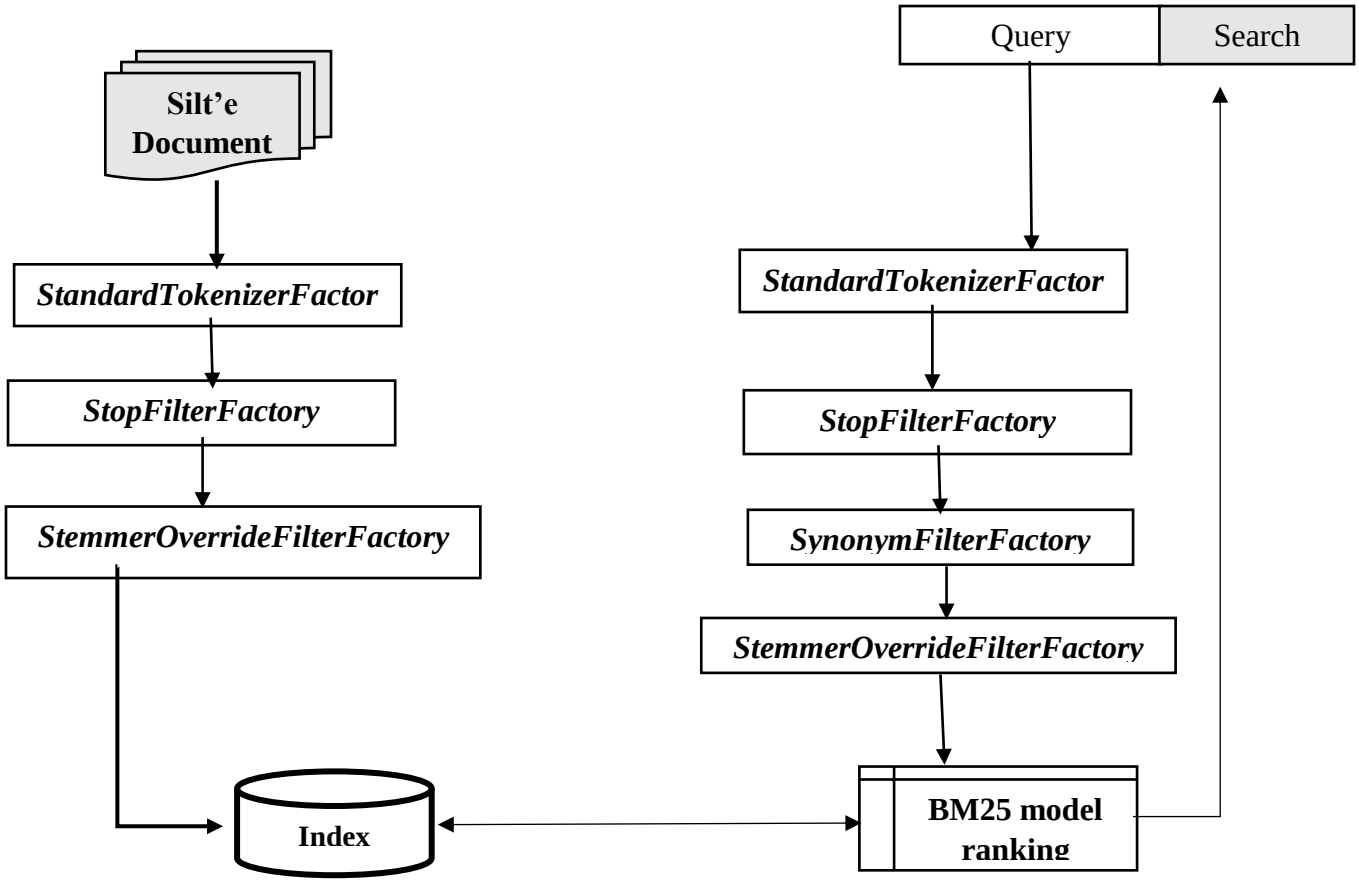

Figure 4.1 Silt'e information retrieval system architecture

4.3.1. Document Preprocessing and Indexing

In this study preprocessing the document that involves tokenizing, stop word removal, synonymy, stemming Silt'e document, and creating inverted indexing file structure.

4.3.1.1. Tokenizing Silt’e Text

To break down Silt'e text stream into tokens we used StandardTokenizerFactory that is implemented in solr. Different tokenizers are provided by solr to use as needed, but we choose StandardTokenizerFactory because it is advanced and general purpose tokenizer. It is implemented based on Unicode Text Segmentation algorithm. While splitting a text stream it looks for white space, punctuations and identifies for sentence boundary [1]. We defined StandardTokenizerFactory in the Silt’e information retrieval system prototype schema as follow

</analyzer>

<tokenizer class="solr.StandardTokenizerFactory"/>

</analyzer>

The following sample Silt’e text that consists of Silt’e characters, punctuation marks such as **- :: ፣ /** and white space was prepared to see effect of the selected tokenizer.

ሀታይ ሰፍ-ቃውል ፣ኡንቂት ፣ሄሰ/ጌሰ/ ፣ ሰንፈሮ፣ ሳለመቻቹ ::

ST	text	ሀታይ	ሰፍ	ቃውል	ኡንቂት	ሄሰ	ጌሰ
	raw_bytes	[e1 88 85 e1 89 b3 e1 8b ad]	[e1 88 b5 e1 8a 93]	[e1 89 83 e1 8b 8d e1 88 8d]	[e1 8a a1 e1 8a 95 e1 89 82 e1 89 b5]	[e1 88 84 e1 88 b6]	[e1 8c 8c e1 89 a6]
	start	0	4	7	12	18	21
	end	3	6	10	16	20	23
	positionLength	1	1	1	1	1	1
	type	<ALPHANUM>	<ALPHANUM>	<ALPHANUM>	<ALPHANUM>	<ALPHANUM>	<ALPHANUM>
	position	1	2	3	4	5	6

Figure 4.2 Tokenized Silt’e texts

4.3.1.2. Stop-Words removal

After tokenization, the next step in the preprocessing stage is Stop word removal. Avoiding these stop-words from Silt’e text is helpful in the ranking of documents. To remove stop-words from a text stream solr provides a number of TokenFilters. These TokenFilters are used to process tokens

that are the output of the tokenizer. The Solr's TokenFilter we choose in this work is StopFilterFactory and define it next to tokenizer during designing the Silt'e text information retrieval system prototype schema structure. In the schema design, we used it as follow.

</analyzer>

<filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt" />

</analyzer>

To see the token filter we prepared sample Silt'e text that contains some Silt'e stop-words.

የገረድ ወልድ በለ ወልደ በጩኼት ተጪኖቲ ግነ አዬኘ ለቲንዛዙ ምካትቸ ቲደወሳቲ ቲፈት እመጨን

In Figure 4.2 shows the result for the above token after implementing *solr.StopFilterFactory*

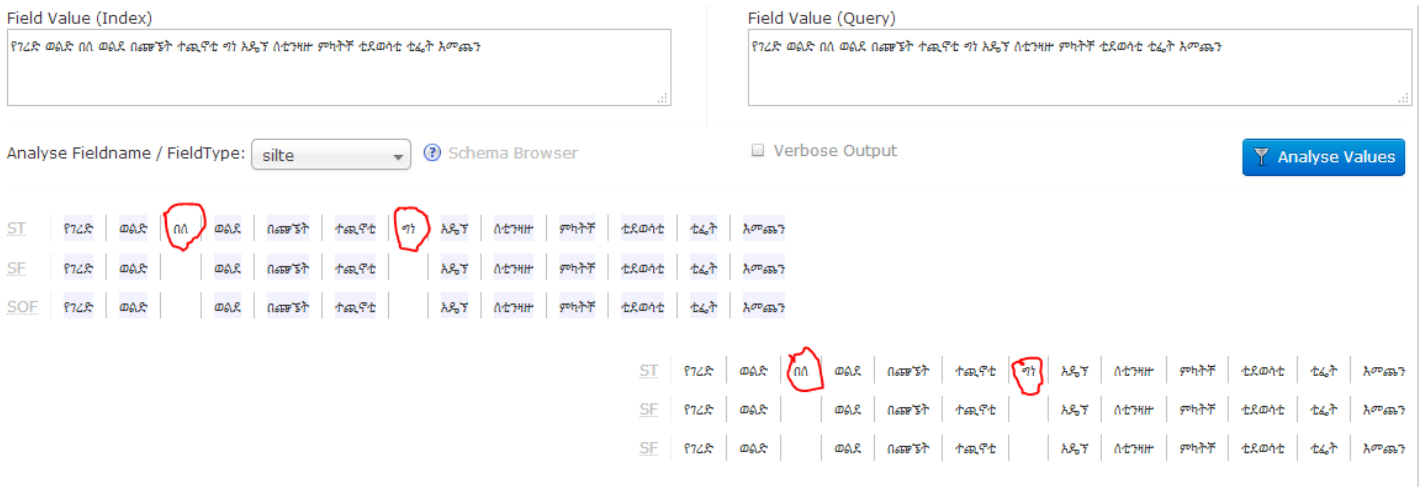

Figure 4.3 Removing Silt'e stop-word analysis result

4.3.1.3. Synonymy Filter

This filter does synonym mapping. Each token is looked up in the list of synonyms and if a match is found, then the synonym is emitted in place of the token. The position value of the new tokens are set such they all occur at the same position as the original token [33]. In this prototype, we use

solr.SynonymFilterFactory for filter synonym word in Silt'e language. In the schema design, we used it as follow

</analyzer>

<filter class="solr.SynonymFilterFactory" expand="true" ignoreCase="true"

synonyms="synonyms.txt"/>

</analyzer>

To see the token filter we prepared sample Silt'e text that contains some Silt'e Synonymy words.

ያበሮስ ቅጨ ሃለት

Figure 4.3 shows the result for the above token after implementing *solr.SynonymFilterFactory.*

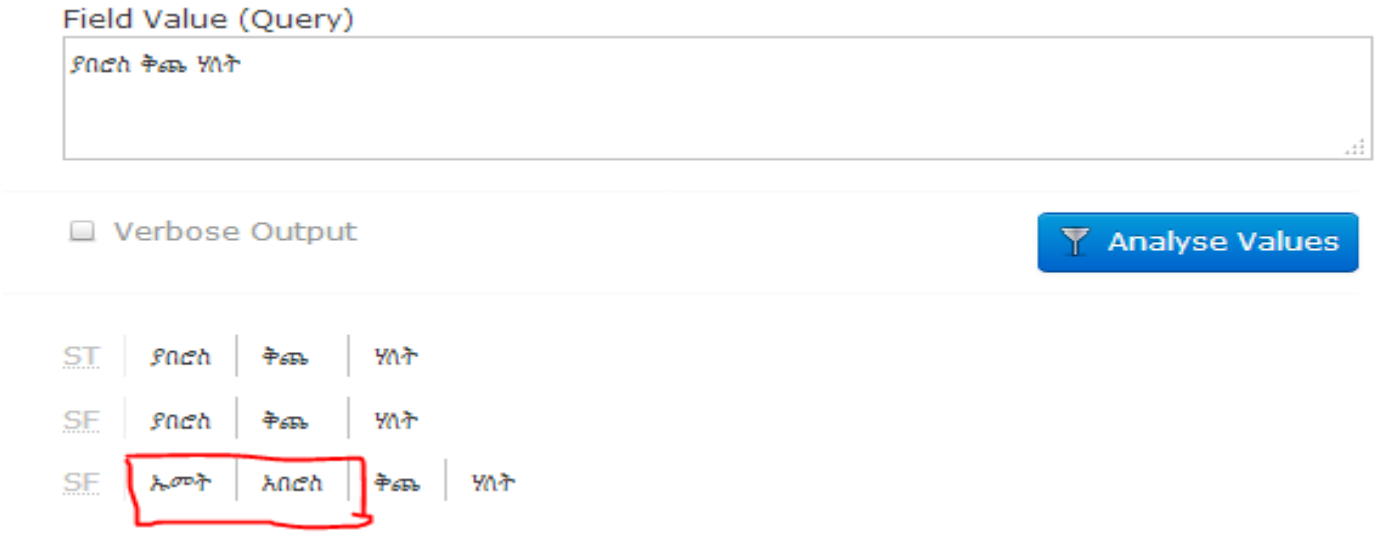

Figure 4.4 Synonymy Filter analyses result

4.3.1.4. Stemming

We used *StemmerOverrideFilter* for stemming Silt'e language token. It overrides the stemming done by the configured stemmer, with the stem mapped to the word in the stemming override file [1]. The following is a sample field-Type configuration in the schema.

< analyzer >

<filter class="solr.StemmerOverrideFilterFactory" dictionary="stemdict.txt" />

</analyzer>

Figure 5.3 shows the analysis sample word stemming. The given token convert to the root word.

Figure 4.5 word stemming

4.3.1.5.Indexing Silt'e Document

A Solr index can accept data from many different sources, including XML files, comma-separated value (CSV) files, data extracted from tables in a database, and files in common file formats such as Microsoft Word or PDF [1]. In this thesis, we post 134 document in .pdf and .docx format using solr posting tool bin\post. Then the inverted index is created. These indexed terms are used for matching and ranking in search requests. Figure 5.5 show sample posting using cmd and solr posting tool.

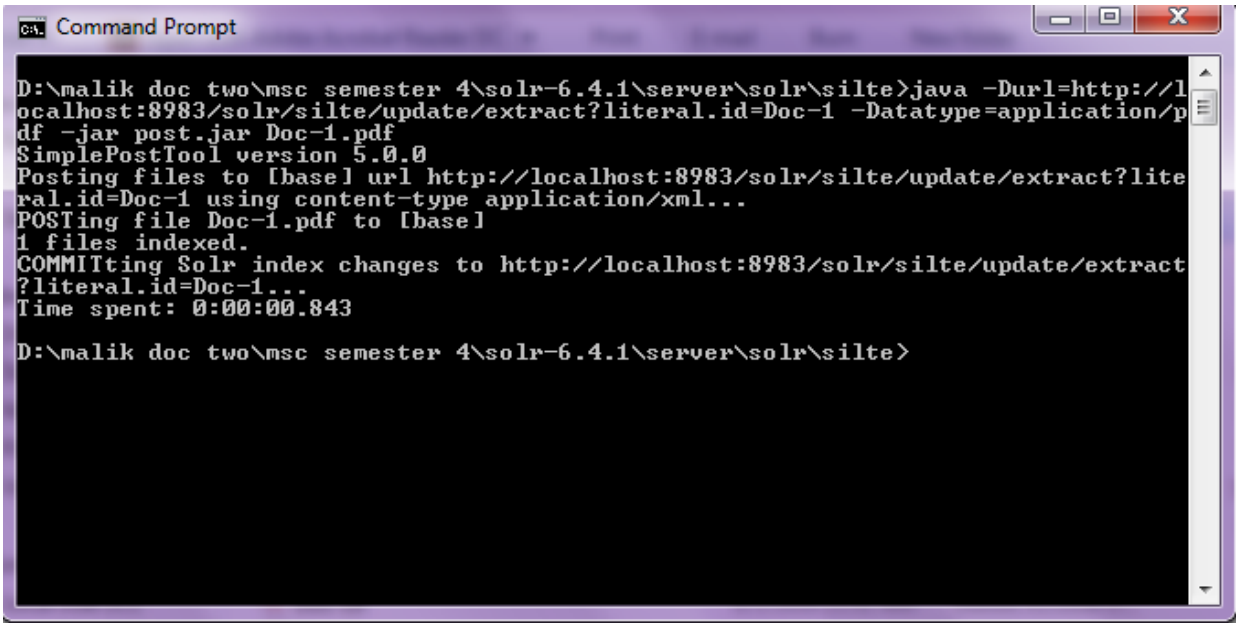

Figure 4.6 indexing document

4.4. Searching for Silt'e Documents

User interaction with information retrieval system is needed to access the required information. To satisfy user-system interaction, we designed an interface in the prototype. It has only query input search box to access a search result. The following screenshot show sample search result for the query "የቶጵያ አፍች ሩክቦኒሙ"

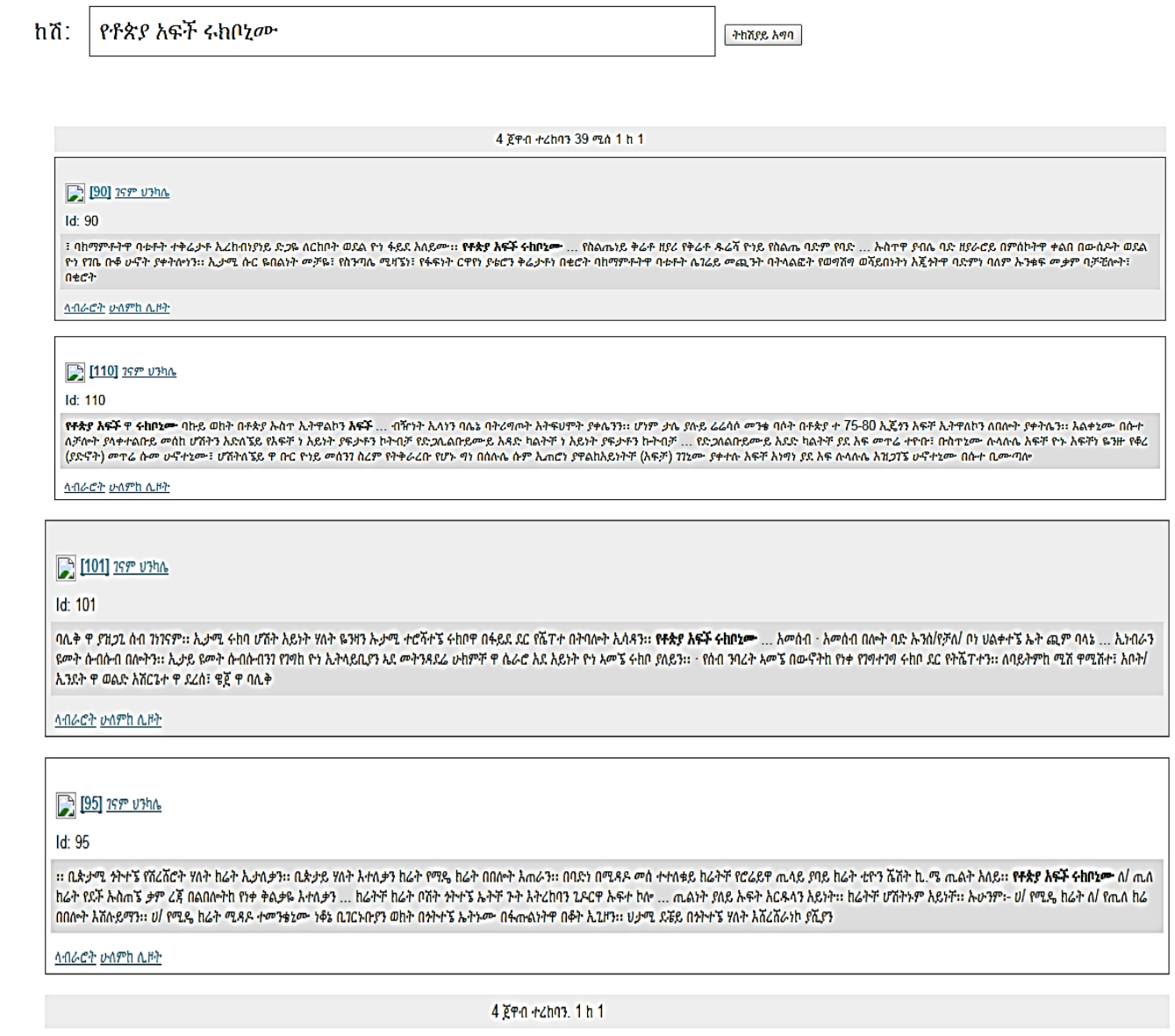

Figure 4.7 UI for retrieved document for a given query

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

Below show sample calculate the score of the document with the query "የቶጵያ አፍች ሩክቦኒሙ" using BM25 model.

```
5.2458105 = sum of:
  1.7837099 = max of:
    1.7837099 = weight(text:የቶጵያ in 86) [SchemaSimilarity], result of:
      1.7837099 = score(doc=86,freq=1.0 = termFreq=1.0),
 product of:
        0.5 = boost
        2.5612795 = idf, computed as log(1 + (docCount - docFreq + 0.5) /
(docFreq + 0.5)) from:
          10.0 = docFreq
          135.0 = docCount
        1.3928272 = tfNorm, computed as (freq * (k1 + 1)) / (freq + k1 * (1 -
b + b * fieldLength / avgFieldLength)) from:
          1.0 = termFreq=1.0
          1.2 = parameter k1
          0.75 = parameter b
          206.06667 = avgFieldLength
          64.0 = fieldLength
  1.0883209 = max of:
    1.0883209 = weight(Synonym(text:አዋልከ text:አፍ text:አፍች) in 86)
[SchemaSimilarity], result of:
      1.0883209 = score(doc=86,freq=1.0 = termFreq=1.0),
product of:
        0.5 = boost
        1.5627508 = idf, computed as log(1 + (docCount - docFreq + 0.5) /
(docFreq + 0.5)) from:
          28.0 = docFreq
          135.0 = docCount
        1.3928272 = tfNorm, computed as (freq * (k1 + 1)) / (freq + k1 * (1 -
b + b * fieldLength / avgFieldLength)) from:
          1.0 = termFreq=1.0
          1.2 = parameter k1
          0.75 = parameter b
          206.06667 = avgFieldLength
          64.0 = fieldLength
  2.3737795 = max of:
    2.3737795 = weight(text:ሩክቦኒሙ in 86) [SchemaSimilarity], result of:
      2.3737795 = score(doc=86,freq=1.0 = termFreq=1.0), product of:
        0.5 = boost
        3.4085774 = idf, computed as log(1 + (docCount - docFreq + 0.5) /
(docFreq + 0.5)) from:
          4.0 = docFreq
          135.0 = docCount
        1.3928272 = tfNorm, computed as (freq * (k1 + 1)) / (freq + k1 * (1 -
b + b * fieldLength / avgFieldLength)) from:
          1.0 = termFreq=1.0
          1.2 = parameter k1
          0.75 = parameter b
          206.06667 = avgFieldLength
          64.0 = fieldLength
```

As it, the above computation in the given query weight each token after filtering tokenization, stop-word removal, stemming and synonym using BM25 similarity measure. The score is the sum of the token after individually computing. For example, the weight of token የቶጵያ score is 1.78 and the weight of token አፍ score is 1.08 and the weight of token ሩክቦኒሙ score is 2.37. The total score of the query የቶጵያ አፍች ሩክቦኒሙ is the sum of three token 5.24. With average document length of 206.06 and document length of 64.0. And with the docFreq of "የቶጵያ" "አፍች" "ሩክቦኒሙ" 10,28,4 respectively in 134 document. For this computation we use the constant value of k1 and b is 1.2, 0.75 respectively.

4.5. Performance Evaluation

4.5.1. System Performance Testing

Once information retrieval system is designed and developed it is essential to carry out its evaluation. Evaluate an IR system is to measure how well the system meets the information needs of the users. Without appropriate retrieval evaluation, cannot determine how well the IR system is performing. There are different techniques of evaluation of the performance of a system based its goal. In fact, the primary goal of an IR system is to retrieve all the documents which are relevant to a user query while retrieving as few non-relevant documents as possible [2]. The objective of this research is to examine the effectiveness of probabilistic IR system using Silt'e language text corpus. Therefore, for this research is taken into consideration to determine the performance of the system. As a result, retrieval effectiveness of a system is evaluated on the given set of documents, queries and relevance judgments. The performance of the system is evaluated using ten queries.

The relevance judgments are prepared to construct document query matrix that shows all relevant documents for each test query prepared as shown in appendix II. Then, each test query is measured by precision, recall and F-measure, and then the average of each test query represents the performance registered by the system. Mean average precision (MAP) evaluate overall performance of the system. The performance of the prototype system is evaluated before and after stemming and synonym in two different ways.

In Table 4.3 show that List of relevant document from Silt'e document corpus for each test query evaluate manually for each document in the collection.

Nº	Query	Relevant documents
1	የሰብ ወልድ ተረዘቆት	Doc-30, Doc-50, Doc-51, Doc-72 Total =4
2	የልባስ ጡሃርነት ቂሮት	Doc-34, Doc-35, Doc-43, Doc-57 Total =4
3	በደም ኢትላለፉነን ነቶ	Doc-31, Doc-33, Doc-52, Doc-51 Total =4
4	ስልጤዋ ያየር ንባረትከ	Doc-11, Doc-20, Doc-28 Total =3
5	የስልጤ አፍ	Doc-19, Doc-72, Doc-114, Doc-112, Doc-111, Doc-117, Doc-115, Doc-113, Doc-116, Doc-129 Total =10
6	ያበሮስ ቅጨ ሃለት	Doc-16, Doc-18, Doc-57 Total =3
7	የቶጵያ አፍች ሩክቦኒሙ	Doc-90, Doc-95, Doc-101, Doc-110 Total =4
8	ያፍ አበሮስ	Doc-25, Doc-26, Doc-28, Doc-109, Doc-119, Doc-124, Doc-121, Doc-51, Doc-30, Doc-16, Doc-123, Doc-122, Doc-120, Doc-126 Total =14
9	በሸርቅ አዘርነት በርበሬ	Doc-89, Doc-1 Total =2
10	ለባድ ወልድ	Doc-15, Doc-27, Doc-52, Doc-66, Doc-22 Total =5

Table 4.3 List of relevant document from Silt'e document corpus for each test query

.

In Table 4.4 Compute a recall and precision pair for each query in the ranked list that contains a relevant document without stemming and synonyms. And compute MAP of the system.

Query															
Rank Position		1	2	3	4	5	6	7	8	9	10	11	12	13	Avg
የሰብ ወልድ ተረዘቆት	Doc_id	50	42	30	51										
	Relevancy	R	NR	R	R										
	Recall	0.3	0.25	0.5	0.75										0.50
	Precision	1.00	0.50	0.66	0.75										0.80

	F-Measure	0.7		0.66	0.85										0.62
የልባስ ጡሃርነት ቂሮት	Doc_id	43	52	34	57										
	Relevancy	R	NR	R	R										
	Recall	0.25	0.25	0.50	0.75										0.50
	Precision	1.00	0.50	0.67	0.75										0.81
	F-Measure														0.62
በደም ኢትላለፋነን ነቶ	Doc_id	33	44	52	55	31	51								
	Relevancy	R	NR	R	NR	R	R								
	Recall	0.25	0.25	0.50	0.50	0.75	1.00								0.63
	Precision	1.00	0.50	0.67	0.50	0.60	0.67								0.73
	FMeasure														0.67
ስልጤዋ ያየር ንባረትከ	Doc_id	28	29	11	20	13									
	Relevancy	R	NR	R	R	NR									
	Recall	0.33	0.33	0.67	1.00	1.00									0.67
	Precision	1.00	0.50	0.67	0.75	0.40									0.81
	F-Measure														0.73
የስልጤ አፍ	Doc_id	117	114	111	112	12	115	113	19	72					
	Relevancy	R	R	R	R	NR	R	R	R	R					
	Recall	0.10	0.20	0.30	0.40	0.40	0.50	0.60	0.70	0.80					0.45
	Precision	1.00	1.00	1.00	1.00	0.80	0.83	0.86	0.88	0.89					0.93
	F-Measure														0.61
ያበሮስ ቅጨ ሃለት	Doc_id	16	57	18											
	Relevancy	R	R	R											
	Recall	0.33	0.67	1.00											0.67
	Precision	1.00	1.00	1.00											1.00
	F-Measure														0.8
የቶጵያ አፍች ሩክቦኒሙ	Doc_id	90	110	101	95										
	Relevancy	R	R	R	R										
	Recall	0.25	0.50	0.75	1.00										0.63
	Precision	1.00	1.00	1.00	1.00										1.00
	F-Measure														0.77
ያፍ አበሮስ	Doc_id	28	124	25	125	29	123	120	26	122	134	126	119	109	
	Relevancy	R	R	R	NR	NR	R	R	R	R	NR	R	R	R	
	Recall	0.07	0.14	0.21	0.20	0.20	0.29	0.36	0.43	0.50	0.47	0.57	0.64	0.71	0.39
	Precision	1.00	1.00	1.00	0.75	0.60	0.67	0.71	0.75	0.78	0.70	0.73	0.75	0.77	0.82
	F-Measure														0.53

በሸርቅ አዘርነት በርበሬ	Doc_id	117	1												
	Relevancy	NR	R												
	Recall	0.00	0.50												0.50
	Precision	0.00	0.50												0.50
	F-Measure														0.5
ለባድ ወልድ	Doc_id	15	24	22	27										
	Relevancy	R	NR	R	R										
	Recall	0.20	0.20	0.40	0.60										0.40
	Precision	1.00	0.50	0.67	0.75										0.81
	F-Measure														0.53
Mean average precision (MAP)															0.82

Table 4.4 Mean average precision for top ranked document without stemming and synonym

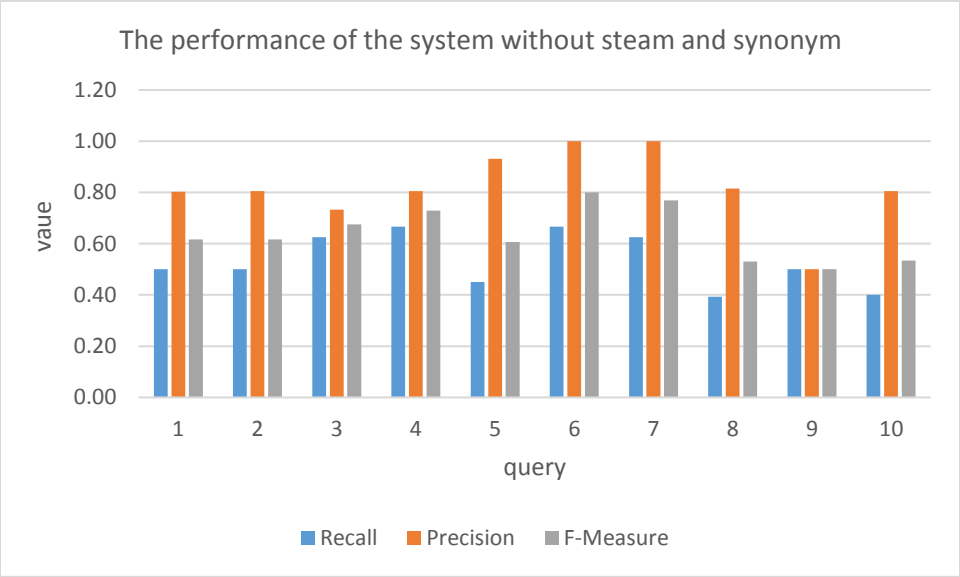

Figure 4.8 The average recall, f-measure and precision of the system without stemming and synonym

As it is observed from table 4.4. Mean Average Precision of the system without considering stemming and synonym is 0.82 (82%). This show the system 82% retrieved documents are relevant. On the other hand as it is observed from figure 4.7 the system achieve 100% average precision for the queries Q6 and Q7. For the remaining queries the system not scores 100% average precision. The average precision is computed as a function of relevant documents retrieved, but the average precision effect for the queries Q1, Q2, Q3, Q4, Q5, Q8, Q9 and Q10 shows that there are non-relevant documents are retrieved.

Query															
Rank Position		1	2	3	4	5	6	7	8	9	10	11	12	13	Avg
የሰብ ወልድ ተረዘቆት	Doc_id	50	42	72	30	51									
	Relevancy	R	NR	R	R	R									
	Recall	0.25	0.25	0.50	0.75	1.00									0.63
	Precision	1.00	0.50	0.67	0.75	0.80									0.80
	F-Measure	0.66		0.66	0.85										0.70
የልባስ ጡሃርነት ቂሮት	Doc_id	43	34	35	52	57									
	Relevancy	R	R	R	NR	R									
	Recall	0.25	0.50	0.75	0.75	1.00									0.63
	Precision	1.00	1.00	1.00	0.75	0.80									0.95
	F-Measure														0.75
በደም ኢትላለፋነን ነቶ	Doc_id	33	44	52	55	31	51								
	Relevancy	R	NR	R	NR	R	R								
	Recall	0.25	0.25	0.50	0.50	0.75	1.00								0.63
	Precision	1.00	0.50	0.67	0.50	0.60	0.67								0.73
	FMeasure														0.67
ስልጤዋ ያየር ንባረትከ	Doc_id	11	28	20	29	13									
	Relevancy	R	R	R	NR	NR									
	Recall	0.33	0.67	1.00	1.00	1.00									0.67
	Precision	1.00	1.00	1.00	0.75	0.60									1.00
	F-Measure														0.8
የስልጤ አፍ	Doc_id	117	114	12	111	112	115	90	113	19	72				
	Relevancy	R	R	NR	R	R	R	NR	R	R	R				
	Recall	0.10	0.20	0.20	0.30	0.40	0.50	0.50	0.60	0.70	0.80				0.45
	Precision	1.00	1.00	0.67	0.75	0.80	0.83	0.71	0.75	0.78	0.80				0.84
	F-Measure														0.59
ያበሮስ ቅጨ ሃለት	Doc_id	16	18	57											
	Relevancy	R	R	R											
	Recall	0.33	0.67	1.00											0.67
	Precision	1.00	1.00	1.00											1.00
	F-Measure														0.80
የቶጵያ አፍች ሩክቦኒሙ	Doc_id	90	110	101	95										
	Relevancy	R	R	R	R										
	Recall	0.25	0.50	0.75	1.00										0.63

	Precision	1.00	1.00	1.00	1.00										1.00
	F-Measure														0.77
ያፍ አበሮስ	Doc_id	28	124	25	125	29	123	120	26	122	134	126	119	109	
	Relevancy	R	R	R	NR	NR	R	R	R	R	NR	R	R	R	
	Recall	0.07	0.14	0.21	0.20	0.20	0.29	0.36	0.43	0.50	0.47	0.57	0.64	0.71	0.39
	Precision	1.00	1.00	1.00	0.75	0.60	0.67	0.71	0.75	0.78	0.70	0.73	0.75	0.77	0.82
	F-Measure														0.53
በሸርቅ አዘርነት በርበሬ	Doc_id	89	117	1											
	Relevancy	R	NR	R											
	Recall	0.33	0.33	0.67											0.50
	Precision	1.00	0.50	0.67											0.83
	F-Measure														0.63
ለባድ ወልድ	Doc_id	15	24	22	27										
	Relevancy	R	NR	R	R										
	Recall	0.20	0.20	0.40	0.60										0.40
	Precision	1.00	0.50	0.67	0.75										0.81
	F-Measure														0.53
Mean average precision (MAP)															0.88

Table 4.5 Mean average precision for top ranked document with stemming and synonym

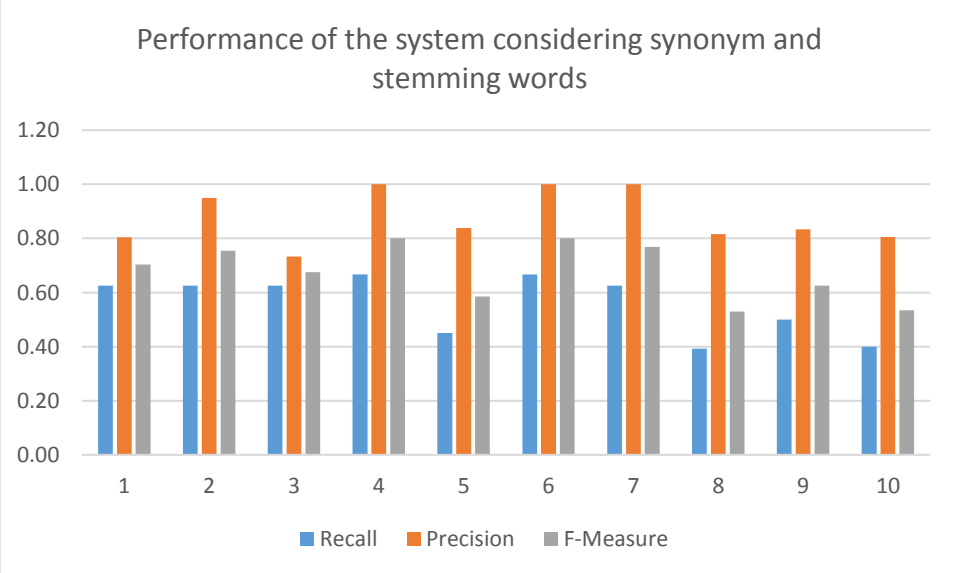

Figure 4.9 The average recall, f-measure and precision of the system with stemming and synonym

As shown in table 4.5 Mean Average Precision of the system with stemming and synonym is 0.88 (88%). This show the system 88% retrieved documents are relevant. Compere to the system without stemming and synonym the MAP increase by 0.06(6%) On the other hand it is observed from figure 4.8 the system achieve 100% average precision for the queries Q4, Q6 and Q7. For the remaining queries in the system not scores 100% average precision. The average precision result for the queries Q1, Q2, Q3, Q5, Q8, Q9 and Q10 shows that there are non-relevant documents are retrieved

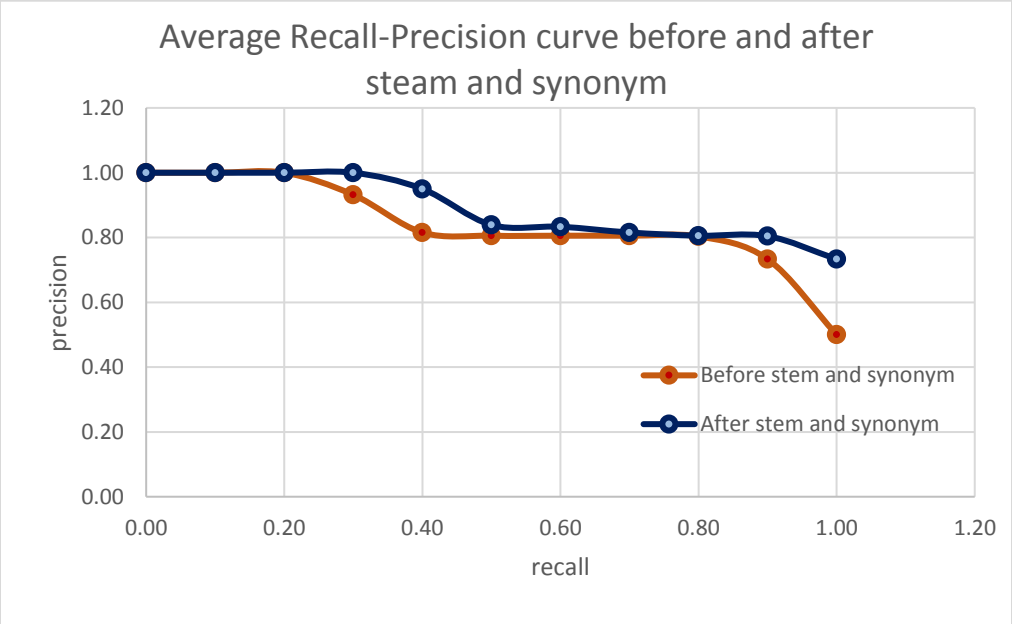

Figure 4.10 Average Recall-Precision curve before and after steam and synonym

Finding the value of precision at standard recall levels for each available 10 queries is important to show the performance of the system. Figure 4.9 shows the interpolated recall-precision curve at standard recall level to show the performance of the designed prototype system before and, after stemming and synonym. As it is observed from Figure 4.9 the system after stem and synonym register better performance than the system without steam and synonym. The curve closest to the upper right-hand corner of the graph indicates the best performance that means the system after stem and synonym

4.5.2. User Acceptance Testing

The aim is to make sure how well system prototype is performing on the eyes of users to assure that the system is acceptable and usable by them. To perform system wise inspection that means

usability and friendliness of the prototype we selected 20 peoples with different background. Due to time factor we limit the number of involved peoples to test the system wise evaluation. We discussed the details of the evaluation in the following table.

Number of peoples involved	Background of the peoples	Inspected elements	Response result
13	Have computing background in the field of Computer Science, Information System and Information Technology	Terminologies, coloring, layout, input and output formats used in the prototype interface	58.3% of the involved peoples provide positive response on the inspected elements. 41.7% of the involved peoples accept the terminology, layout and input/output formats, but they were not satisfied with the coloring of the interface.
7	Understand the language used in the system, but not have computing background	Easiness to use the system and satisfaction with the retrieved result	58.5% satisfied by both easiness and retrieved result. 41.5% satisfied by the retrieved result, but dissatisfied with the easiness because the Silt'e language uses Ethiopic character and

			they are not enough skilled to type Ethiopic characters from the keyboard

Table 4.6 Summary of User Acceptance Testing

4.6. Discussion

As discussed earlier, the system performance was calculated as 82% mean average precision without stemming and synonym and 88% mean average precision with stemming and synonym. The system result compare to previous work in Silt'e language better performance and promising result for developing text information retrieval in Silt'e language.

However several challenge to achieve promising performance. The reason is that in Silt'e text documents the same set of words are used to express different concepts in different documents. For example consider the words in the query "የስልጤ አፍ". Documents retrieved for this query describe about the background of the Silt'e language in the relevant documents, but the same set of words are used in the non-relevant documents to describe other things. When we examine the Silt'e text information sources that we collected for this study, also different scripting style is used for the same term in different documents. For example the words ንትተኜ, ንትተኘ and ንትለኜ have the same meaning with different scripting style. This situation has impact on the retrieved result.

In general, the testing and evaluation result of the prototype accomplished the objectives of the study. However, additional study is needed to bring a complete implementation and use of large corpus to check the system performance.

CHAPTER FIVE

5. Conclusion and Recommendation

5.1. Conclusion

Text retrieval system is very important for retrieval of textual documents. This study tries to develop probabilistic IR system for Silt'e language. The developed prototype has two parts: indexing and searching. The indexing part of the work involves tokenization, stop word removal, synonym and stemming using Apache solr filtering factor. For stemming purpose used stemming dictionary supported by solr. The system developed has also searching components. The main parts are query pre-processing, similarity measurement and ranking the document. The retrieval model used for this study is one of the probabilistic model BM25. The experimentation performed using the prototype that was developed scored 88% mean average precision. This is a promising result to design an applicable IR system for Silt'e language. Apache Solr allows us to add our own module to process Silt'e text.

Generally, the study shows promising results in developing text information retrieval system for Silt'e language. However, further exploration and study should be done to improve and produce a better performance by increasing the corpus size in the language and avoiding limitations on the Silt'e scripting system.

5.2. Recommendation and Future Works

Information retrieval research on Silt'e language is in its infant stage. The results of our work indicate the possibility of designing and developing efficient Silt'e text information retrieval systems. However, the following recommendations could provide the system with a better performance and bring it to the real world applications.

For practice

- For development of the language and for organizing, storing and accessed digital unstructured Silt'e text document necessary use information retrieval system in different sector of Silt'e zone. We recommend the people and other concerned body to use

information retrieval system for development of the language and increase the working activity.

For Future Researcher

- In this study, we considered only the Silt'e text information documents. To develop fully functional Silt'e information retrieval system, we endorse considering different document types in the future study.
- Finding a standard corpus and test queries with relevance judgment for testing the designed system is one of the challenges faced in this research. Therefore, future research needs to consider the development of standard Silt'e corpus that can be used by researchers to evaluate progress made in designing Silt'e IR systems. Thus, the future researchers need to give an emphasis on the development of corpus for Silt'e to reach an applicable information retrieval.
- No standard Silt'e stemming algorithm supported by Apache solr. We recommend implementing and adding standard Silt'e stemmer into Apache solr for enhancing the performance of the system.
- CLIR is one of information retrieval filed. In Silt'e language no work in this file. We recommend CLIR system for Silt'e language to another language.
- Relevance feedback is a mechanism of engaging users or system in retrieval process so as to improve the final result of the IR system so we recommend future researcher consider relevance feedback to improve system performance.

References

[1] D. Shahi, Apache Solr: A Practical Approach to Enterprise Search, Apress, 2015.

[2] Baeza-Yates, Ribeiro-Neto,, Modern Information Retrieval, New York, USA: Addison-Wesley-Longman, 1999.

[3] H.S. Christopher D.Manning, Prabhakar Raghavan, An Introduction to Information Retrieval, Cambridge England: Cambridge University Press, 2008..

[4] D. G. Uma, "Wordnet and Ontology Based Query Expansion for Semantic information retrieval in sports domain," *Journal of Computer Science,* vol. 11, no. 2, pp. 361-371, 2014.

[5] B. Y. Ricardo, "Modern Information Retrieval the concepts and technology behind search," *Pearson Education Limited,* vol. 2, 2011.

[6] J. Picard and R. Haenni, "Modeling Information Retrieval with Probabilistic Argumentation Systems," *BCS-IRSG Annual Colloquium on IR,* vol. 14, no. 7, pp. 10-25, 1998.

[7] S. Wartik, "Boolean Operations," in *Information Retrieval.*, ISBN, 1992, p. 264–292.

[8] Simons, Gary F. and Charles D. Fennig, "Ethnologue," 2017. [Online]. Available: http://www.ethnologue.com.. [Accessed 8 6 20017].

[9] M. P. Lewis, "Ethnologue," [Online]. Available: https://www.ethnologue.com/language/stv. [Accessed 26 4 2017].

[10] T. G. NISRANE, "LANGUAGE PLANNING AND POLICY IN THE SILT'E ZONE," ADDIS ABABA UNIVERSITY , ADDIS ABABA, JUNE 2015.

[11] J. Z, "SILT'E TEXT INFORMATION RETRIEVAL SYSTEM BASED ON VECTOR SPACE MODEL," Adama Science and Technology University,, Adama, 2016.

[12] N. Fuhr, "Probabilistic Models in Information Retrieval," *The Computer Journal,* vol. 35, no. 3, pp. 243-255, 1992.

[13] M. Vallez and R. Pedraza-Jimenez., "Natural Language Processing in Textual Information Retrieval and Related Topics," *"Hipertext.net",* no. 5, 2007.

[14] Lewis, M. Paul, Gary F. Simons, and Charles D. Fennig , "Ethnologue: Languages of the World," 2015. [Online]. Available: http://www.ethnologue.com/18/. [Accessed 26 4 2017].

[15] K. M, "Designinig a Stemming Algorithm for Silt'e," Addis Ababa University, Addis Ababa , 2012.

[16] Baldi, Pierre, Paolo Frasconi, and Padhraic Smyth., "Modeling the Internet and the Web," *Probabilistic methods and algorithms ,* 2003.

[17] V. V. a. S. M. W. Raghavan, "A critical analysis of vector space model for information retrieval," *Journal of the American Society for information Science,* vol. 37, no. 5, pp. 279-287, 1986.

[18] Hiemstra, Djoerd. , "information retrieval models," *Information Retrieval:searching in the 21st Century,* pp. 1-19, 2009.

[19] S. M. Katz, "Distribution of content words and phrases in text and language modeling," *Natural Language Engineering,* vol. 2, pp. 15-59, 1996..

[20] W. B. C. D. M. T. Strohman, Search Engines Information Retrieval in Practice, Pearson Education, 2015.

[21] E. Greengrass, "Information Retrieval: A Survey," [Online]. Available: www.csee.umbc.edu/cadip/readings/IR.report.120600.book.pdf. [Accessed 23 4 2017].

[22] A. B. a. D. Swanson, "Probabilistic Models for Automatic Indexing," *Journal of the American Society for Information Science,* vol. 25, p. 312–318, 1974.

[23] R. ZHU, "Improvement in probabilistic information retrieval model - rewarding terms with high relative term frequency," York University, Toronto, Canada, 2016.

[24] &. T. Z. Keneilwe, "Evaluation Of Information Retrieval Systems," *International Journal of Computer Science & Information Technology,* vol. 4, no. 3, 2012.

[25] Z. M. Albujasim, "Search Queries in an Information Retrieval System," University of Kentucky, 2014.

[26] G. G, " Afaan Oromo Text Retrieval System," Addis Ababa University, Addis Ababa, 2012.

[27] W. M. ., "Semantic Indexing and Document Clustering for Amharic Information RetrievaL," Addis Ababa University, Addis Ababa, 2013.

[28] A. H, "Probabilistic Information Retrieval System for Amharic Language," Addis Ababa University, Addis Ababa , 2012.

[29] A. L, " A Probabilistic Information Retrieval System for Tigrinya," Addis Ababa University, Addis Ababa, 2014.

[30] B. D, "Afaan Oromo-English Cross-Lingual Information Retrieval : A CORPUS BASED APPROACH," Addis Ababa University, Addis Ababa, 2011.

[31] C.J. Rijsbergen, Information Retrieval, 2, Ed., U.S.A: Butterworth-Heinemann publisher, 1979.

[32] K. Beiske, "found-similarity-in-elasticsearch," 26 November 2013. [Online]. Available: https://www.elastic.co/blog/found-similarity-in-elasticsearch. [Accessed 15 june 2017].

[33] [Online]. Available: http://www.apache.org/licenses/LICENSE-2.0. [Accessed june 14 2017].

[34] W. B. C. D. M. T. Strohman, Search Engines Information Retrieval in Practice, Pearson Education, Inc., 2015.

[35] I. Y. Y. Yao, "nformation Retrieval Support Systems," *Boca Raton:Taylor & Francis Group,* pp. 1-778, 2012.

Appendix I. Lists of Ethiopic alphabets used for Silt'e

1st order	2nd order	3rd order	4th order	5th order	6th order	7th order
ሀ ha	ሁ hu	ሂ hii	ሃ haa	ሄ he	ህ hi	ሆ ho
ለ le	ሉ lu	ሊ lii	ላ laa	ሌ le	ል li	ሎ lo
መ me	ሙ mu	ሚ mi	ማ maa	ሜ me	ም mi	ሞ mo
ረ re	ሩ ru	ሪ ri	ራ raa	ሬ re	ር ri	ሮ ro
ሰ se	ሱ su	ሲ si	ሳ saa	ሴ se	ስ si	ሶ so
ሸ ša	ሹ šu	ሺ šii	ሻ šaa	ሼ še	ሽ ši	ሾ šo
ቀ qa	ቁ qu	ቂ qii	ቃ qaa	ቄ qe	ቅ qi	ቆ qo
በ ba	ቡ bu	ቢ bii	ባ baa	ቤ be	ብ bi	ቦ bo
ተ te	ቱ tu	ቲ tii	ታ taa	ቴ te	ት ti	ቶ to
ቸ ca	ቹ cu	ቺ cii	ቻ caa	ቼ ce	ች ci	ቾ co
ነ ne	ኑ nu	ኒ nii	ና naa	ኔ ne	ን ni	ኖ no
ኘ ña	ኙ ñu	ኚ ñii	ኛ ñaa	ኜ ñe	ኝ ñi	ኞ ño
አ a	ኡ u	ኢ ii	ኣ aa	ኤ ee	እ i	ኦ o
ከ ka	ኩ ku	ኪ kii	ካ kaa	ኬ ke	ክ ki	ኮ ko
ወ wa	ዉ wu	ዊ wii	ዋ waa	ዌ we	ው wu	ዎ wo
ዘ za	ዙ zu	ዚ zii	ዛ zaa	ዜ ze	ዝ zi	ዞ zo
ዠ ža	ዡ žu	ዢ žii	ዣ žaa	ዤ že	ዥ ži	ዦ žo
የ ya	ዩ yu	ይ yii	ያ yaa	ዬ ye	ዪ yu	ዮ yo
ደ da	ዱ du	ዲ dii	ዳ daa	ዴ de	ድ di	ዶ do
ጀ ja	ጁ ju	ጂ jii	ጃ jaa	ጄ je	ጅ ji	ጆ jo
ገ ga	ጉ gu	ጊ gii	ጋ gaa	ጌ ge	ግ gi	ጎ go

Appendix II. Document-Query matrix used for relevance judgment

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}
Doc-1.pdf	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Doc-2.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-3.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-4.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-5.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-6.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-7.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-8.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-9.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-10.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-11.pdf	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Doc-12.docx	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc-13.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-14.docx	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc-15.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Doc-16.docx	NR	NR	NR	NR	NR	R	NR	R	NR	NR
Doc-17.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-18.pdf	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Doc-19.pdf	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc-20.docx	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Doc-21.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-22.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Doc-23.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-24.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Doc-25.docx	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-26.docx	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-27.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Doc-28.docx	NR	NR	NR	R	NR	NR	NR	R	NR	NR
Doc-29.docx	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-30.docx	R	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-31.docx	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Doc-32.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-33.docx	NR	NR	R	NR	NR	NR	NR	NR	NR	NR

Doc-34.docx	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Doc-35.docx	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Doc-36.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-37.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-38.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-39.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-40.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-41.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-42.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-43.docx	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Doc-44.docx	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Doc-45.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-46.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-47.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-48.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-49.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-50.docx	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-51.docx	R	NR	R	NR	NR	NR	NR	R	NR	NR
Doc-52.docx	NR	R	R	NR	NR	NR	NR	NR	NR	R
Doc-53.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-54.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-55.docx	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Doc-56.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-57.docx	NR	R	NR	NR	NR	R	NR	NR	NR	NR
Doc-58.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-59.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-60.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-61.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-62.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-63.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-64.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-65.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-66.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Doc-67.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-68.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-69.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-70.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR

Doc-71.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-72.docx	R	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc-73.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-74.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-75.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-76.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-77.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-78.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-79.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-80.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-81.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-82.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-83.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-84.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-85.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-86.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-87.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-88.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-89.docx	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Doc-90.docx	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Doc-91.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-92.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-93.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-94.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-95.docx	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Doc-96.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-97.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-98.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-99.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-100.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-101.docx	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Doc-102.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-103.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-104.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-105.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-106.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-107.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR

Doc-108.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-109.docx	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-110.docx	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Doc-111.docx	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc-112.docx	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc-113.pdf	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc-114.docx	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc-115.pdf	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc-116.docx	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc-117.pdf	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc-118.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-119.docx	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-120.pdf	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-121.docx	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-122.pdf	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-123.pdf	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-124.docx	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-125.docx	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-126.pdf	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Doc-127.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-128.pdf	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-129.pdf	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Doc-130.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-131.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-132.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-133.docx	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Doc-134.docx	NR	NR	NR	NR	NR	NR	NR	R	NR	NR

Appendix III. Lists of Silt’e stop words

ሀነይ	ሬራቀደን	አቢሌ	እነይ
ሀነግነ	ቀለ	አብሌ	እነይ
ሀዳድኑም	ቀደ	አብተቴ	ኦናኔ
ሁልምከ	ቀደን	አብታይ	ወክት
ሁኖ	ቂጦ	አተም	ወይ
ሁኖተኒመዋ	ቅጨ	አተቴ	ዋ
ሂነሚ	በሁነትንሙ	አቲ	ዋ
ሂንኩምንገ	በሁኖት	አታይ	የገግ
ሂንኩምንገ	በለ	አታይ	ዩተቴ
ሆነምታሌ	በሉሌ	አነይ	ዩታይ
ለሄ	በልዳሌ	አናግና	ያሸ
ለሄው	በልዳሌ	አዩ	ያተቴ
ለሰባድሸ	በሰቹ	አይታይ	ያታይ
ለሰባድኑም	በውኖትምከ	አይነኮ	ዬዬ
ለሰገጋሙ	በዬ	አይነኮ	ዮለ
ለሰገግሸ	ቢትላይም	አይኔ	ዮልስ
ለሰገግኑም	ብቾ	አድ	ዮናነይ
ለሰገግከ	ብቾ	አድአድ	ደር
ለሳድባድከ	ተሬር	ኡሀ	ደር
ለደር	ተዮን	ኡስጥ	ድባየ
ለገነ	ተደር	ኡንኮ	ገነ
ለገነገናም	ቱፍትሉፍት	ኢንኩምንገ	ገና
ሉሀ	ቲታሚ	ኢንኩምንገ	ገናሚ

ሉሉሌ	ታሌ	ኤት	ገናገናይ
ሉላሉሌ	ታቼን	ኤጋህ	ገገ
ሉላሉሌ	ትዮኑ	ኤጋመ	ገገኑ
ሉሌ	ናሩ	ኤጋሽ	ገጌ
ላይሽ	ናር	እሊ	ጉት
ማ	ናርት	እቢ	ግን
ምን	አለቢ	እብተቴ	ግዘ
ሱር	አሊ	እተቴ	ግዘቸ
ሱር	አልቀሬ	እቲ	ፈሬ
ሱርዋ	አሎነ	እታይ	ፎኖ
ስር	አበይ	እነኮ	
ሬራ	አቢ	እነይ	

Appendix IV. Lists of Silt’e prefixes

ኢለው	አል	የ
እለው	አት	ይ
በል	ተይ	ት
ኢለ	በ	ከ
በሰ	ላ	ቲ
አይ	ቃ	ተ
እተ	ጫ	ሻ
እት	አ	ና
እል	ሰ	ያ
እለ	ለ	እ

Appendix V. Lists of Silt’e suffixes

ኩሞመ	ንቾ	ሸ	እ
ቢያኔ	እሎ	ቸ	አ
አታም	ዬን	ኮ	ው
አተኘ	ኔት	ኩ	ም
አመ	ታም	ሁ	ቴ
ሼታ	ተኛ	ነ	ካ
ኩመ	ኢ	ን	መ
ኒመ	ይ	ሺ	መ
ሼሸ	ኤ	ኤ	ሚ
ታት	ሸ	አ	ዋ
ኤን	ኔ	ተ	ሶ
አኘ	ከ	ት	ኒ
ተኘ	ሎ	ኡ	ኝ
			ክ

Appendix VI. Schema structure for Silt'e text information retrieval system

```
<?xml version="1.0" encoding="UTF-8"?>
<schema name="example" version="1.6">
  <uniqueKey>id</uniqueKey>
  <fieldType name="_bbox_coord" class="solr.TrieDoubleField" stored="false" useDocValuesAsStored="false" docValues="true" precisionStep="8"/>
<fieldType name="alphaOnlySort" class="solr.TextField" omitNorms="true" sortMissingLast="true">
    <fieldType name="text_general" class="solr.TextField" positionIncrementGap="100">
    <analyzer type="index">
      <tokenizer class="solr.StandardTokenizerFactory"/>
      <filter class="solr.StopFilterFactory" words="stopwords.txt" ignoreCase="true"/>
                        <filter class="solr.WordDelimiterFilterFactory" catenateNumbers="1" generateNumberParts="0" generateWordParts="0" catenateAll="0" catenateWords="1"/>
                        <filter class="solr.KeywordMarkerFilterFactory" protected="protwords.txt" />
      <filter class="solr.StemmerOverrideFilterFactory" dictionary="stemdict.txt"/>
                        <filter class="solr.RemoveDuplicatesTokenFilterFactory"/>
    </analyzer>
    <analyzer type="query">
      <tokenizer class="solr.StandardTokenizerFactory"/>
      <filter class="solr.StopFilterFactory" words="stopwords.txt" ignoreCase="true"/>
                        <filter class="solr.WordDelimiterFilterFactory" catenateNumbers="1" generateNumberParts="0" generateWordParts="0" catenateAll="0" catenateWords="1"/>
                        <filter class="solr.SynonymFilterFactory" expand="true" ignoreCase="false" synonyms="synonyms.txt"/>
                        <filter class="solr.RemoveDuplicatesTokenFilterFactory"/>
                        <filter class="solr.KeywordMarkerFilterFactory" protected="protwords.txt" />
                        <filter class="solr.RemoveDuplicatesTokenFilterFactory"/>
    </analyzer>
 <field name="silte" type="text_general" indexed="true" stored="true"/>
  </schema>
```