2021-10

INFORMATION RETRIEVAL FOR þÿSILT E TEXT USING LATENT SEMANTIC INDEXING

Yoseph, Mare Guwta

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTING

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

INFORMATION RETRIEVAL FOR SILT'E TEXT USING LATENT SEMANTIC INDEXING

By

Yoseph Mare Guwta

BAHIR DAR, ETHIOPIA

October 2021

INFORMATION RETRIEVAL FOR SILT'E TEXT USING LATENT SEMANTIC INDEXING

BY

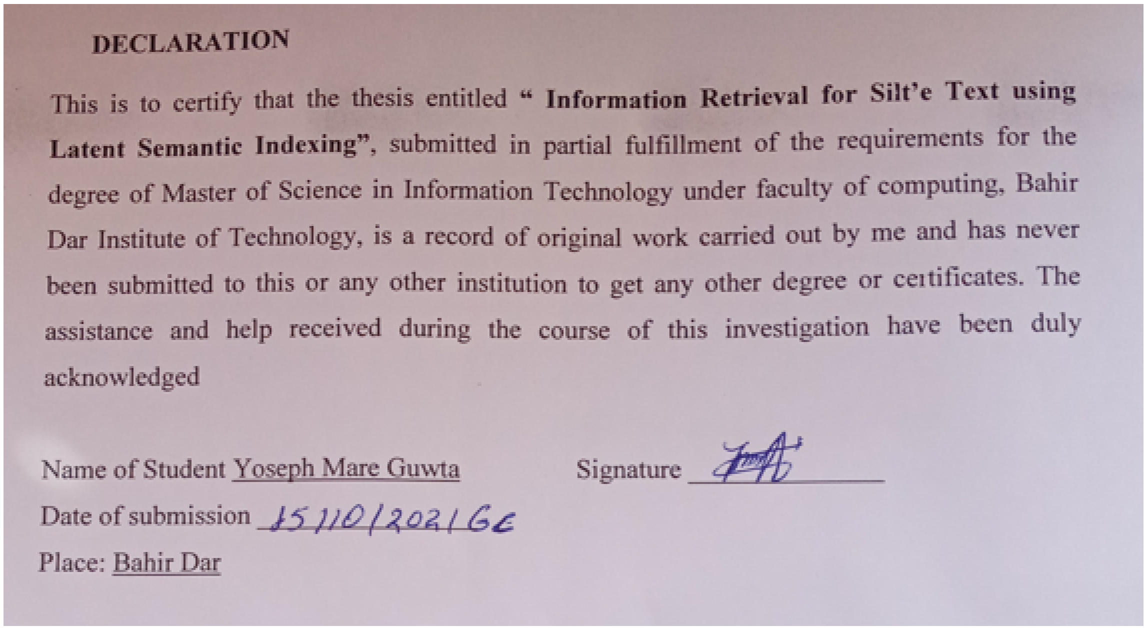
Yoseph Mare Guwta

A thesis report submitted to the school of Research and Graduate Studies of Bahir Dar Institute of Technology, BDU in partial fulfillment of the requirements for the degree of Masters of science in the Information Technology in the Faculty of Computing.

Advisor Name: Mekonnen Wagaw (PhD)

Bahir Dar, Ethiopia

October 2021



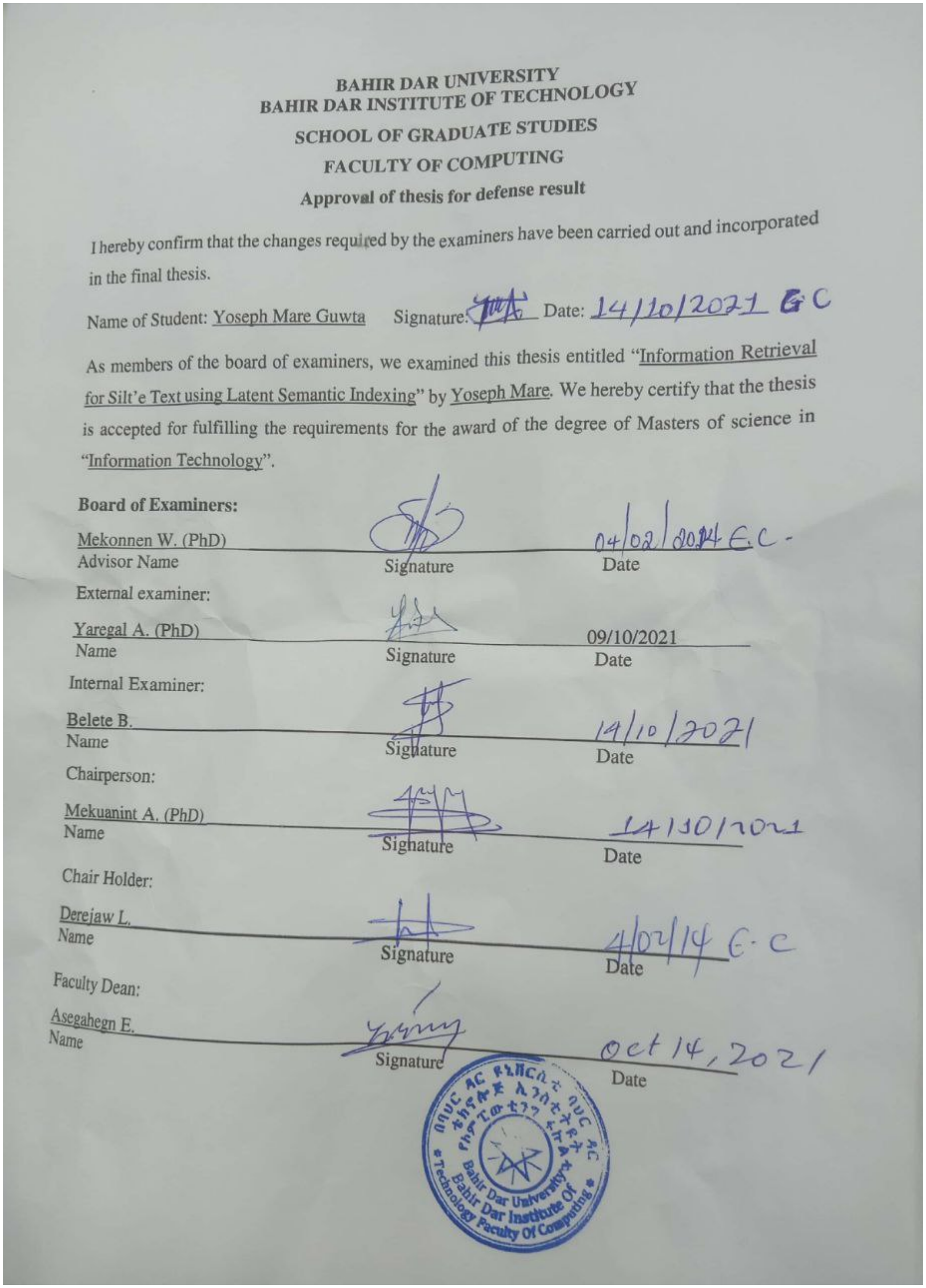
DECLARATION

This is to certify that the thesis entitled **“ Information Retrieval for Silt’e Text using Latent Semantic Indexing”**, submitted in partial fulfillment of the requirements for the degree of Master of Science in Information Technology under faculty of computing, Bahir Dar Institute of Technology, is a record of original work carried out by me and has never been submitted to this or any other institution to get any other degree or certificates. The assistance and help received during the course of this investigation have been duly acknowledged

Name of Student Yoseph Mare Guwta Signature ________

Date of submission 15/10/2021 GC

Place: Bahir Dar



BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
FACULTY OF COMPUTING
Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student: Yoseph Mare Guwta Signature: Date: 14/10/2021 G.C

As members of the board of examiners, we examined this thesis entitled "Information Retrieval for Silt'e Text using Latent Semantic Indexing" by Yoseph Mare. We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of science in "Information Technology".

Board of Examiners:

Mekonnen W. (PhD) 04/02/2014 E.C.
Advisor Name Signature Date

External examiner:

Yaregal A. (PhD) 09/10/2021
Name Signature Date

Internal Examiner:

Belete B. 14/10/2021
Name Signature Date

Chairperson:

Mekuanint A. (PhD) 14/10/2021
Name Signature Date

Chair Holder:

Derejaw L. 4/02/14 E.C
Name Signature Date

Faculty Dean:

Asegahegn E. Oct 14, 2021
Name Signature Date

በባህር ዳር ዩኒቨርሲቲ ባህር ዳር ቴክኖሎጂ ኢንስቲትዩት የኮምፒውቲንግ ፋካልቲ
Bahir Dar University Bahir Dar Institute Of Technology Faculty Of Computing

To Hailu Guwta

ACKNOWLEDGMENTS

First and foremost, I would like to thank Almighty God for giving me health, strength, ability, and opportunity to undertake this research study, to persevere and complete. Without his blessing, this achievement would not have been possible.

Next, I would like to thank my advisor Mekonnen Wagaw (PhD) for his continuous support of this research, for his patience, and motivation.

It is pleasure to thank those who helped me by different mechanism when I was working on this study specially my Mother Belay Mira, Amogne Andualem, Alemayeh Negash, Firaol Tesfaye and others people those their name is not mentioned here.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

IR	Information Retrieval
LSI	Latent Semantic Indexing
MAP	Mean Average Precision
SVD	Singular Value Decomposition
TF-IDF	Term Frequency Inverse Document Frequency
VSM	Vector Space Model

LISTS OF TABLES

LISTS OF FIGURES

ABSTRACT

Information retrieval is a mechanism that enables finding relevant information material of unstructured nature that satisfies information needs of user from large collection. Since there are usually many ways to express the same concepts, the terms in the user's query may not appear in a relevant document. Alternatively, many words can also have more than one meaning which may confuse the retrieval system. This research intended to apply latent semantic indexing to handle synonymous and polysemous words in the Silt'e text document and users' query. Silt'e text retrieval developed in this study has indexing and searching subsystems. While indexing organizes index terms, searching enables matching query terms with index terms in order to retrieve relevant documents. For the experimenting purpose, we have used 700 Silt'e text documents and 56 queries were used to test the prototype of the system. Silt'e text document corpus is prepared by the researcher encompassing different reports from Silt'e culture and tourism bureau and books. Also, various techniques of text preprocessing including tokenization, normalization, stop word removal and stemming were used to identify content-bearing words. Experimental result shows that the prototype registered on the average 68% recall, 79% precision and 72% F-measure. The major challenges that affect the performance of the IR prototype include lack of standard dataset for Silt'e language and the ineffectiveness of Silt'e stemmer to conflate Silt'e inflectional words into their stem. Therefore, in order to improve the performance of the prototype, there is a need to develop Silt'e dataset as well as Silt'e stemmer.

Keywords: Information Retrieval, Latent Semantic Indexing, Singular Value Decomposition

CHAPTER ONE: INTRODUCTION

1.1. Background

The advancement of computing technology in storage and processing capacity-controlled a lot of the experimental explosion and mass production of electronic information recorded from the day-to-day activity of societies (Babu & L., 2014). Data are being generated nowadays by different organization and institution. The stored data can be used for different purposes, but not all the stored on a computer are equally important at a certain point of time. Finding useful information from a huge collection of documents stored for a long period of time in different language remains as the problem still now and requires a series focus (Abu-Salih, 2018).

To retrieve relevant information that satisfies the interest of the user, some sort of systems should be developed. This enforces the researcher to devise systematic ways of maintaining and retrieving relevant and valuable information from the huge amount of stored document. The idea of organizing and systematically retrieving information has been raised and used by librarians before 2000 years ago (Johanna, 2006). They arrange books on catalog in terms of book's title or author and the retrieval process was a physical search from book catalog which is tedious and time-consuming, As the volume of the electronic data exponentially increases, this method becomes impractical, even if some libraries are using this method still now. Due to the exponential production of electronic materials, the librarians forced to employ more efficient methods of storing and retrieving information. The systematic organization of library material in the digital library came to use one century ago (Dramé et al., 2014). As the written information becomes large in size and digital documents easily available. it is difficult to retrieve relevant documents among the accumulated document collections.

Information retrieval (IR) is a mechanism that allows the user to retrieve the related unstructured information material from a large collection of information (Manning, 2009). The major goal of IR system is to make the necessary information available to the intended user at right time. Information retrieval has three main subsystems; indexing, weighting and searching with ranking (Bachchhav, 2016). Indexing involves identifying potentially important and content bearing terms from large document collection and depict the using indexing. This is the basic phase where key terms are extracted, preprocessed and be ready for term weighting function. Additional indexing subsystems save storage space and speed up searching by removing undiscriminating terms like stop word. The weighting subsystem involves computing the importance of each index term in document collection to the user query. This is because not all terms in documents are equally important. The third subsystem is searching which involves measuring the similarity of index terms to query terms. The searching subsystem will compute the relationship between the user's query terms and individual document terms.

There are many IR techniques. Most techniques to retrieve textual materials from database depend on the exact term match between terms in the user's query and terms by which documents are indexed. In another word, the items that have common entries in both the database and users query are returned to users. However, since there are usually many ways to express the same concept, the terms in the user's query may not appear in a relevant document. Alternatively, many words can have more than one meaning (Marco Suárez Barón, 2009). So the standard information retrieval models (e.g., Boolean, VSM, probabilistic) treat words as if they are independent (Baeza-Yates & Ribeiro-Neto, 1999). Due to these facts, term matching methods are likely to miss relevant documents and also retrieve irrelevant ones.

Latent Semantic Indexing(LSI) is one of the information retrievals techniques to develop IR based on the vector representation of documents and user's query (Deerwester et al., 2000). Because it uses concepts or topics instead of individual words to index and retrieve the documents, consequently allowing a relevant document to be retrieved even when it shares no common words within the query (Baeza-yates, 1999; Manning, 2009).

The motivation behind Latent Semantic Indexing is Silt'e to allow the users to make sufficient use of the available technologies because many terms can have more than one meaning present in any language provide great difficulty in the use of information retrieval as terms in human language that occur in a particular context can be interpreted in more than one way depending on the context. Hence, designing an efficient Silt'e text document retrieval using LSI technique is one basic solution to overcome the limitations of the previous Silt’e text document retrieval systems, we aim at designing Silt’e text document retrieval using LSI.

1.2. Statement of the problem

Silt’e language is one of the languages that are spoken in Ethiopia. It is a Semitic language spoken in Southern Nations, Nationalities and People’s Region in the Silt’e Zone (Johar, 2017). A lot of valuable information is being produced in government organizations and schools in Silte zone. The growth of digital text information makes the utilization and access of the right information difficult. Thus, developing an IR system that enables searching and retrieving relevant documents written in Silt’e is important.

So far two works have been done on information retrieval for Silt’e language. These are: text information retrieval system for Silt’e (Johar, 2017) and Silt’e text information retrieval system based on VSM (Jemal, 2016). Johar’s work uses probabilistic technique, Jamal’s work, on the other hand, is based on vector space model which is not capable to control Silt’e synonymous and polysemous terms. So his research has a great impact in retrieving relevant information for users query because of the reason that the system it doesn’t handle synonymous and polysemous terms.

In Silt’e language lexical variation is very common (Kedir, 2012). For instance, the word “ሰቾ” and “ተቃወ” are lexically two different words, but semantically they have the same meaning to mean “drink” in English. Thus, in a literal search, only documents containing the literal query terms will be matched, causing the search results to exclude relevant documents. And Polysemy (words are characterized by different meanings based on the context). For Instance, The Word “ዱግማግለ” in Silt’e can mean “slept”, or it can also mean “increase ”. So terms in a user's query will literally match terms in irrelevant documents The existence of such synonymy and polysemous words leads to decrease in the performance of the system (Abebaw, 2015).

So far of our knowledge there is no any research done in semantic based IR system for Silt’e language. Our research will be the first attempt of controlling the synonym and polysemy nature of Silt’e words, to enhance the performance of Silt’e IR systems. This research attempts to design semantic indexing based IR Silt'e text document , in order to control the effect of synonymous and polysemous behavior of Silt'e words.

Therefore, the aim of this study is to apply LSI approach to handle Silt'e synonymous and polysemous terms that affect the performance of information retrieval systems thereby retrieving relevant documents as per users query. Because it addresses on the basis of concepts, not keywords; searches are not constrained by the specific words that users choose when they formulate queries. So by using statistical techniques like Singular Value Decomposition (SVD) that LSI can retrieve relevant documents even when those documents do not share any words with a query. For this purpose we had developed semantic- indexing based information retrieval for Silt'e language. This study answered the following research questions:

- What are the suitable techniques to control synonymous and polysemous terms in order to enhance the performance of the Silt'e IR system?
- How semantic-indexing improve searching for relevant Silt'e documents in Silt'e text retrieval?
- To what extent Latent Semantic Indexing enables to design Silt'e text retrieval system?

1.3. Objective of the study

1.3.1. General objective

The general objective of the study is to develop an information retrieval prototype for Silt'e text using Latent Semantic Indexing for enhancing the performance of Silt'e text retrieval system.

1.3.2. Specific objective

In order to achieve the general objective, the study deals with the following specific objectives.

- To review literatures and previous works related with IR systems.
- To collect and prepare Silt'e language text corpus.
- Designing Silt'e information retrieval with Semantic indexing.

- To design a prototype that can search a relevant document from Silt'e text corpus.
- To compare the result with the performance results obtained for the vector space model.
- To evaluate the performance of the prototype.
- Forward conclusion and recommendations.

1.4. Scope and the limitation of the study

The focus of this study is on designing Silt'e text information retrieval prototype based on Latent Semantic Indexing that searches relevant text documents from Silt'e text corpus. It involves indexing and searching mechanisms of IR for the text data type. Other data types, such as image, video, and graphics are out of focus of the research. Stemming of index terms is not done; because semantic-based indexing system by its nature partially handles the problems that arise due to morphological variation of index terms. Of course, had stemming been used, it would have at least reduce the number of index terms and in turn allow to add some additional documents into the test collection.

1.5. Significance of the study

Developing an information retrieval for a local language helps the users of the language to access information. Generally, the study will have the following significances:

- This study will be helpful in providing information for those who are interested in making a further study in this similar area.
- It has the potential to improve the development of the language.
- It will have a significant contribution in an attempt to use LSI for cross-language retrieval in which Silt'e one of the components.
- The result show that the advantage of LSI based IR for Silt'e text documents.

1.6. Organization of the thesis

The thesis is organized into five chapters. The first chapter introduces the background of the research, statement of problem, objective, the scope of the study, and significance of the study. The remainder of the thesis is organized as follows.

Chapter two is devoted to the literature review and related works. General concepts on IR and Silt'e writing system applicable to the research are discussed. Latent semantic indexing and related researches on latent semantic indexing are also reviewed. The third chapter presents the major techniques and methods used in this study and the proposed architecture of the prototype. Chapter four describes the experimentation and evaluation of the proposed prototype and discusses the results obtained from the experiment. Finally in chapter five most important findings including faced challenges are written as a conclusion and forwards recommendations for future work.

CHAPTER TWO: LITERATURE REVIEW

2.1. Introduction

This chapter is broadly divided into three sections. The first section discusses the features of Silt'e writing system related to information retrieval. The Silt'e alphabets, synonym and polysemy, numbers and grammar are also introduced. The second section discusses the concepts related to IR, indexing, the series of activities involved in the document and query indexing process, components of IR system, Approaches to solve the problem of synonym and polysemy and latent semantic indexing. Concepts such as term extraction, the semantics of the term and term weighting are briefly introduced. The third section discusses research works related to keyword-based IR and semantic-based IR.

2.2. Overview of Silt'e language

Silt'e language is one of the languages that are spoken in Ethiopia. It is a Semitic language spoken in Southern Nations, Nationalities and People's Region in the Silt'e Zone (Johar, 2017). Silt'e is a medium of instruction in primary schools from grade 1 to 4 and taken as a subject from grade 5 to 12. In addition, Silt'e is being delivered as a field of study in Hosanna College of Teacher Education. The Silt'e people adopted the Ethiopic script for their writing system (Kedir, 2012).

2.2.1. Silt'e Morphology

Morphology is the study of the way words are built from smaller units. A morpheme is a minimal meaning-bearing unit in a language. A morpheme in Silt'e can be free or bound, where a free morpheme can stand as a word on its own whereas a bound morpheme cannot occur on its own as a word. for example, these are free morphemes of Silt'e፡-ሰብ "person", ጋር "house", ቃዋ "coffee", they can stand by themselves. In contrast to these bound morphemes of Silt'e can't stand by themselves. For example, we can add the suffix "ቾ" to the word "ሰብ" to show paucal number; therefore is bound morpheme of Silt'e (Kedir, 2012).

2.2.2. Word formation in Silt’e language

Affixing (adding suffix, prefixes and infixes), compounding, duplicating (reduplicating) and changing vowel patterns are used to give different word forms in Silt’e language (Kedir, 2012). The addition of suffix, prefix and infix are the usual ways of word formation in Silt’e. Although it is not common, the addition of prefix and suffix at the same time is also used to form word variant. The possibility of prefixing and suffixing more than three prefix or/and suffix result in long word-formation. Hence, the complex morphological structure of a single Silt’e word can give very large number of variant.

The application of compounding is another means to contribute in the process of word variant formation in Silt’e language. A compound is made up of two or more simple words (roots) and as a meaning that is different from that of its parts: አባዴ (abbaadde/abaad), ‘parents’ from አቦ(abbo/abo) ’father’ and አዴ (adde/ade) ‘mother’. ወዘነቁበሌ (wazana-ḳuble)‘faint-hearted’,ወዘነ(wazana) ‘heart’ and ቁበለ(ḳubla)’deficent’. Duplication is also the other means which is used to form word variation: there are at least two distinct kinds of reduplicative formations in Silt’e language: type one involves the reduplication of the first letter (that is ላ (laa)) and ላላሀ(laalaaha) ’send many times’ formed from ላሀ (laaha)‘send’,type two by reduplicating the second letter by using its forth form that is ማ(ma)):ጀማመረ(jimaamara) ‘starts’ to little which is formed from ጀመረ(jamara)‘start’.

2.2.3. Dialect and verities

Every language that is spoken over any significant area is spoken in somewhat different forms in different places; these are its regional dialects. Moreover, even in a single community, the language may be spoken differently by members of different social groups; these different forms are social dialects. The Silt’e language has four dialects, these are Azernet-Berbere, Silti, Wuriro and Ulbareg. These four varieties depend on the geographical area and there are strong similarities among them (Jemal, 2016).

2.2.4. Silt’e writing system

Silt’e has its own writing system which is taken from Ge’ez character. The Silt’e writing system consists of twenty-six characters and seven vowels (Jemal, 2016) .

Consonants

According to Jemal (2016), Silt’e consonants are similar to other Ethiopian Semitic languages. The portion of the Ge’ez alphabets and their phonetic order are shown in table 2.1.

Table 2. 1 Silt’e Consonant

ሀ(ha)	ለ(le)	መ(me)	ረ(re)	ሰ(se)	ሸ(ša)	ቀ(qe)	በ(ba)	ተ(te)	ቸ(ce)
ነ(ne)	ኘ(ña)	አ(a)	ከ(ka)	ወ(we)	ዘ(za)	ዠ(ža)	የ(ya)	ደ(de)	ጀ(ja)
ገ(ga)	ጠ(ť e)	ጬ|ĉe|	ጰ(Ṗa)	ፌ(fe)	ፐ(pe)	ገ(ga)	ጠ(ť e)	ጬ|ĉe|	ጰ(Ṗa)
ጬ|ĉe|	ጰ(Ṗa)	ፌ(fe)	ፐ(pe)						

Vowels

Silt’e has seven vowels, these seven vowels are mapped onto ten, vowels, these are the five short vowels አ(a), ኤ(e), ኡ(u), ኦ(o), እ(i) and five long vowels ኣ(aa), ኦኦ(oo), ኢ(ii), ኡኡ(uu), ኤኤ(ee) (Muzeyn, 2012). The long vowels and consonants can be obtained by doubling the corresponding short vowels and consonants respectively. Table 2.2 shown Silt’e long and consonant.

Table 2. 2 Silt’e long and short vowel

Short vowels	Long vowels
አ(a), ኤ(e), ኡ(u), ኦ(o), እ(i)	ኣ(aa), ኦኦ(oo), ኢ(ii), ኡኡ(uu), ኤኤ(ee)

2.2.3. Inflectional and derivational morphology of Silt’e

Inflectional affixes describe word as stems are combined with grammatical markers for things like person, gender, number, tense, and case. There are five kinds of speech in Silt’e: adjectives, nouns, verbs, adverbs, and propositions. Prepositions and conjunctions are totally unproductive for IR purpose or other natural language processing. The same thing happens for adverbs and is few in number. Therefore, the discussion of derivational and inflectional morphology concentrates on the remaining part.

Gender

Gender distinction is simply a sex-based categorization of a given noun. Like most of the other languages, Silt’e also identifies two gender forms named as masculine and feminine. These distinct can have their own marker in Silt’e language. Masculine marked by vowel “ወሪ” and feminine marked by “አይሪቴ”. For example, gender for astronomical elements ወሪ(warii) “moon” define the masculine gender and አይሪቴ(ayiriite) “sun” define the feminine gender (Kedir, 2012).

Number

Numbers refer to the singularity and plurality of the given word. Like nouns, Silt’e adjectives also attach the suffix “ች” for plurality. Since adjectives and nouns form a common class, they behave identically at least morphologically as different linguistics agrees.

Silt'e language has formal means to express differences of quantity. It uses the three numbering techniques to express a number of nouns and pronouns; these are singular, plural and paucity. For an example, ሲን(siin) refer the single or large number of cups, but ሲንቻ(siinca) mention a small number of cups. Noun paucity in Silt’e formed by suffixation of “ቻ”. The other way of forming the noun paucal is by reduplicating the last consonant of the noun.

Personal pronoun

Silt’e pronouns include personal pronouns (refer to the persons speaking, the persons spoken to, or the persons or things spoken about), indefinite pronouns, relative pronouns (connect parts of sentences) and reciprocal or reflexive pronouns (in which the object of a verb is being acted on by verb's subject). Table 2.3 showed personal pronouns of the Silt’e language.

Table 2. 3 Silt’e personal pronoun

First Person				Second person		Third Person			
Singular		Plural		Singular		Singular		plural	
English	Silt’e	English	Silt’e	English	Silt’e	English	Silt’e	English	Silt’e

I	ኢኼ(ihee)	We	ኢኘ(inaa)	‘You’	ኣሽ(aash) f
					ኣተ(ataa)m
						He
						She
							ኡሀ(uuha)
							ኢሽ(iisha)
								They	ኣሁን(uuhun)

Adjective

An adjective is a word that describes, identify or quantify a word by preceding the noun or pronoun which it modifies. The adjective in Silt’e used to describe characteristics of nouns. By doing this it modifies the meaning of a noun, so it has an important function for daily communication. Sample Silt’e adjectives of the Silt’e language are indicated in table 2.4.

Table 2. 4 Sample Silt'e Adjective

Adjective	Meaning	Verb	Meaning
በቺሎ	Someone who cries	በቼ	Cry
ሸባልቶ	Singer	ሸባል	Sung
በሬዶ	So beautiful	በሬዶ በሬዷ	Beautiful

Adverb

An adverb is a word that tells us more about a verb. It qualifies or modifies a verb, adjective and other adverbs. In Silt’e modifiers of verb or verb phrase usually express time, location, manner, etc. Some Silt'e adverbs are listed in table 2.5.

Table 2. 5 Sample Silt’e adverbs

Adverb of time	Adverb of	Adverb of place	Adverb of frequency

		manner					
English	Silt’e	English	Silt’e	English	Silt’e	English	Silt’e
Today	ሀውጆ	Fast	ኮሞ	here	ሂኔ	Always	ሁለም ነግ
Later	በዘፍ	very	ፈየኮ	there	ሀኔ	sometimes	ደደ ነግ
tomorrow	ጌስ	together	ሀዴኘ	everywhere	ሁለምኤት	never	ረም
after tomorrow	ሴስተ					occasionally	አለፈ አለፋኔ

Verb

Silt’e has a subject, object, verb word order. The verb is a word that describes the subject’s action or state within a sentence. Sample verbs from Silt’e: ቃዋ (coffee), ገባት (night). Silt’e verbs are either perfection or imperfection. Perfection includes past or completed actions whereas imperfection indicates present, continuous and future action. 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. Table 2. 6 Shows sample Silt’e verbs.

Table 2. 6 Sample Silt'e Verbs

Two radical	Three radical	Four radical
ቃዋ(qaawaa) “coffee”	ገባት(gabaata) “night”	አዋልከ(aawalake) “speak”

2.2.1. Synonymy and Polysemy

Synonymy

Synonymy refers to the fact that the same underlying concept can be describe using different terms (Baeza-Yates & Ribeiro-Neto, 1999). The phenomenon of synonymy is sufficiently widespread to account for the popularity. The notion of synonymy has a deceptively simple definition of different lexemes with the same meaning. The synonyms can substitute for one another in a sentence without changing either the meaning or the acceptability of the sentence in the document (Baeza-Yates & Ribeiro-Neto, 1999). In LSI, the concept in query, as well as all documents that are related to it, is all likely to be represented by a similar weighted combination of indexing variables.

For an example: ጀማል ቃዋ ተቃወ . "Jemal drinks coffee ".

The underlined word ተቃወ can be substituted by the word "ሰቸ" which cannot be used in this sentence. These two words can replace each other without changing the meaning of the sentence.

Polysemy

Polysemy describes a single word that has more than one meaning, which is common property of language. Large numbers of polysemous words in the query can reduce the precision of a search significantly. By using a reduced representation in LSI, one hopes to remove some "noise" from the data, which could be explained as rare and less important usages of certain terms. The idea of polysemy allows to state that sense is related to, and probably derive from, without asserting that it is a distinct lexeme (Baeza-Yates & Ribeiro-Neto, 1999). For example: ጀማል ዱግማለ ባለ . Here the word underlined ዱግማለ which mean "slept" and "increased" are two related meanings according to the contexts of the sentence. As one suspect, the task of distinguishing homonymy from polysemy is not quite straightforward. There are two criteria that are typically invoked to determine whether or not the meanings of two lexemes are related or not: the history, or etymology, of the lexemes and how the word s are conceived of by native speakers (Baeza-Yates & Ribeiro-Neto, 1999). In the absence of detailed etymological evidence, a useful intuition to use in distinguishing homonymy from polysemy is the notion of coincidence. On the other hand, it is far more difficult to accept cases of polysemy as coincidences. useful intuition to use in distinguishing homonymy from polysemy is the notion of coincidence. On the other hand, it is far more difficult to accept cases of polysemy as coincidences.

2.3. Overview Information retrieval system

Information Retrieval (IR) is the discipline that deals with the representation, storage, organization and retrieval of unstructured data(document, web pages, records abstract, summaries, etc.), in response to a user query, which may itself be unstructured (Baeza-Yates & Ribeiro-Neto, 1999).

The main objective of the IR system is to provide easy access to unstructured data, and successful IR methods are required for an increasing number of documents on any deskto p.The information retrieval system is a computer program that stores and manages record information and allows users to retrieve information of interest.

Information retrieval system is a computer program that store and manage information in the document and assist users in retrieving information of their interest. It finds useful information from the document or the document from a huge repository of document that contains the required information and returns to the user with its location. Those document that satisfies the information needs of the user called relevant document.

In the information field of study, some related such as data retrieval, document retrieval, text retrieval, and information retrieval are interchangeable (George & Hameed, 2013). Data retrieval involves the finding of a document from the repository that contains the keyword in the query, whereas information retrieval focuses on looking for and returning the actual information about a certain issue or subject (Baeza-Yates & Ribeiro-Neto, 1999). But text retrieval and document retrieval are synonymous terms intended for user queries (David, 1996)**.** IR can also cover other kinds of data and information beyond textual document; it could be an image, multimedia or other data (Manning, 2009).

The records that IR takes on are often called document, IR involves the retrieval of document from an unstructured repository. The central problem in IR is the quest to find the set of relevant documents, amongst a large collection, containing the information sought thereby satisfying an information need usually expressed by a user with a query (Baeza-Yates & Ribeiro-Neto, 1999).

2.4. Fundamental of IR system process

An effective IR system involves three components such as document repository, query formulation and models and its evaluation. This component works in three basic processes (phase) (Hiemstra, 2000). The first process is a representation of the document and the user query. in an information retrieval system, the document and user search statement should be first represented in a proper encoding method called indexing. Both document and query pass through text processing operation during the indexing process. Indexing is an offline process used to facilitate access to preferred information from document corpus accurately and efficiently as per users query (Manning, 2009). This process applied before the retrieval process start and it is used to representing the document.

The second process is the compression of the user query and all available document in the repository. This is termed in another way as a matching (searching) process. The searching process computes the frequency of the term in the user query and document in the repository and retrieves the most related document to the user query. In this phase, the IR model comes to play a role in assigning weight to induced term and measure their similarity score with the query term. The third process in IR is the ranking process which involved ordering the retrieved document according to their level of importance to the user query. Document ranking is the essential process in modern IR systems which increase the quality of retrieval system.

2.4.1. Automatic indexing

Automatic indexing is the ability for a computer to scan large volumes of documents against a controlled vocabulary, taxonomy, thesaurus or ontology and use those controlled terms to quickly and effectively index large document depositories. As the number of documents exponentially increases with the proliferation of the Internet, automatic indexing will become essential to maintaining the ability to find relevant information in a sea of irrelevant information (Lahtinen, 2000). There are two major approaches for the automatic indexing of text documents: statistical approaches that rely on various word counting techniques, vector space model used this approach. Statistical indexing aims to capture content bearing words that have a good discriminating ability and a good characterizing ability for the content of a document. Discrimination ability means that the words can distinguish documents from one another. And those terms are going to be counted and will have some weight. The other approach is the linguistic approach that involves syntactical analysis (Névéol et al., 2007). It is based on the knowledge based of the language it is working on (Keifer & Effenberger, 2013). Therefore, it needs ontology of the language to determine the terms which describe a document. It can use the extraction of the semantics of the document using content bearing words extracted from the document that going to be indexed (Keifer & Effenberger, 2013).

2.4.2. Semantic indexing

To improve the performance of a traditional keyword-based search, web documents should be represented with their concept rather than bag-of-words. However; most of the previous works on indexing and information retrieval depend on lexical analysis and statistical methods. Using these techniques, it is difficult to abstract the semantics of the documents (Kang, 2003). Typically, information is retrieved by literally matching terms in documents and query. Since there are usually many ways to express a given concept (synonymy), the literal terms in a user's query may not match those of a relevant document. In addition, most words have multiple meanings (polysemy), so terms in a user's query will match terms in irrelevant documents. A better approach would allow users to retrieve information based on a conceptual topic or meaning of a document (Kang, 2003). Latent Semantic Indexing tries to overcome the problems of lexical matching by using statistically derived conceptual indices instead of individual words for retrieval. LSI assumes that there is some underlying or latent structure in word usage that is partially obscured by variability in word choice. SVD is used to estimate the structure in word usage across documents (Rosario, 2000). The other method to solve the lexical matching problem is the lexical chains technique for the extraction of concepts from the document and represents by concept vectors. And then text vectors, semantic indexes and their semantic importance degree are computed. this indexing method has an advantage in being independent of document length because we regarded overall text information as a value 1 and represented each index weight by the semantic information ratio of overall text information (Kang, 2003).

2.4.3. Semantic Index term extraction

Terms that are extracted from textual document are known to be an important and fundamental linguistic descriptor of documents (Bin, 2006). Extracted terms are used for representing contents of specific. The terms extracted from the document can be used for indexing, it helps to distinguish a document from others (Hans, 2005). In many cases, the terms that best describe the contents of a document are at the same time terminological units of the text's domain (Hans, 2005). Extraction of key terms from the textual document for indexing purpose using a computer is said to be automatic index text extraction. It is a basic requirement for any text-related applications such as text clustering, indexing and others (Bin, 2006). There are two main kinds of approaches to automatic index text extraction, statistical method and linguistic method (Bin, 2006). The statistical method relies on word frequencies: words that are repeated frequently within a document are likely to be good descriptors of its content. On the other hand, terms that occur in several documents (like "the", "about" or "believe") cannot distinguish one document from another (Hans, 2005). It can be computed for a given term by multiplying its frequency in the current document (TF = term frequency) with its inverse document frequency (IDF) (Hans, 2005). The linguistic method relies on syntactic criteria and does not use any morphological processes. Most researchers seem to agree that terms are mainly noun phrases to represent the semantics of the text. Document Representation and Term Weighting.

The automatic indexing process can be considered to be composed of two tasks: first assigning terms or concepts capable of representing the content of each document in the databased, called term extraction and second, assigning to each term a weight or values that signify its importance for purpose of content description, called term weight (Hans, 2005).

2.4.4. Index Term Extraction

The task of extracting content descriptors itself is composed of a sequence of many activities like text preprocessing (Hans, 2005).

Stop word

Stop-words are the most frequent terms which are common to every document and have no discriminating power one document from the other. Sometimes, some extremely common words which would appear to be of little value in helping select documents matching a user need are excluded from the vocabulary entirely (Hans, 2005)

Stemming

Stemming is a process used in most search engines and information retrieval systems. It is a core natural language processing technique for an efficient and effective IR system. Generally stemming transforms inflated words into their most basic form. Stemming is a language-dependent process in a similar way to other natural language processing techniques. It is often removing inflectional and derivational morphology. E.g. automate, automatic, automation ⟶ automat. 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 the 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**.**

Term Weighting

In representing the content of a particular document, all the words included in the index list are not equally important. The main reason for computing term weight is to assigning weight depending on their level of import (Phadnis & Gadge, 2014). Weighting index terms increase the precision of retrieval. The functions use these distribution statistics to compute the weight of each term in each document and query (Rosario, 2000). The term frequency * inverse document frequency, also known as tf*idf, is a well-known method of determining the value of a word in Information Retrieval and Text Mining in a document. It is a statistical method that reflects how important a word/term is to a document in collection/corpus. The value of **tf*idf** increases with proportion to the number of times a word appears in the document and decreases as the term exist frequently in the document corpus.

Term Frequency

A document that mentions a query term more often has more to do with that query and therefore should receive a higher score. Therefore, assign to each term in a document weight for that term that depends on the number of occurrences of the term in the documen t (Manning, 2009). This concept can be applied through assigning the weight; it is equal to the number of occurrences of term t in document d. This weighting scheme is referred to as term frequency and is denoted **tft, d**, with the subscripts denoting the term and the document in its order (Manning, 2009). The exact ordering of the terms in a document is ignored but the number of occurrences of each term is material (in contrast to Boolean retrieval). We only tried to capture information concerning the number of occurrences of each term (Manning, 2009).

$$\mathrm{TFij} = \frac{\mathrm{fij}}{\max\{\mathrm{fij}\}} \qquad (2.1)$$

Inverse document frequency

It will create a problem when we make all terms are equally important; the impact would be explicitly viewed when it comes to assessing relevancy against a query (Manning, 2009). In fact, certain terms have little or no discriminating power in determining relevance. For instance, a collection of documents on the auto industry is likely to have the term auto in almost every document. To this end, we introduce a mechanism for attenuating the effect of terms that occur too often in the collection to be meaningful for relevance determination. An immediate idea is to scale down the term weights of terms with high collection frequency, defined to be the total number of occurrences of a term in the collection. The idea would be to reduce the tf weight of a term by a factor that grows with its collection frequency (Manning, 2009). document frequency of the term, **dft**, defined to be the number of documents in the collection that contain a term t. it aims to obtain document level statistics, which deals with the number of documents containing a term (Manning, 2009). Using dft is difficult to measure the discrimination power of the term among the documents. To come over this problem, it better to use inverse document frequency, which is logarithmic function of the total number of the document N and over dft (Manning, 2009). It is defined as

$$\mathrm{IDF} = \mathrm{logN/dft} \tag{2.2}$$

Tf-idf weighting

Combining the definitions of term frequency and inverse document frequency helps to produce a composite weight for each term in each document. The tfidf weighting scheme assigns to term t in document d given by

$$\mathrm{tfidft,d} = \ \mathrm{tf\,t,d} * \mathrm{idft,d} \tag{2.3}$$

tfidf$_{t,\ d}$ assigns to term t is a weight in document d that is

1. Highest when t occurs many times within a small number of documents (thus lending high discriminating power to those documents);
2. Lower when the term occurs fewer times in a document or occurs in many documents (thus offering a less pronounced relevance signal);
3. lowest when the term occurs in virtually all documents.

Ranking

In the modern IR system, document ranking is the crucial step that enhances the efficiency of the retrieval system. A ranking function receives a retrieved document set from the matching function and provides an ordered document set in descending order according to its level of relevance to the user query. The document ranking depends function is ranked lists retrieved according to the level of relevance

2.5. The Matching Process

After documents are logically organized and the user query is processed, the next step is comparing the query against the document representations, which is called, the matching process. The result of the matching is a ranked list of documents according to their likelihood of relevance (Baeza-Yates &Ribeiro- Neta, 2011).

Query matching model

Comparison of query and document representations is a very important activity of IR system; to achieve to retrieve documents that are more similar to the specific query. The three classic models of Information retrieval: the Boolean model, the Vector space model and the probabilistic model are often used to accomplish this particular task (Manning, 2009). The Boolean, vector space model and probabilistic models are briefly discussed below

Boolean Model

The Boolean model of Information retrieval is the oldest of the three classic retrieval models and it relies on the use of Boolean operators in combination with set theory (Baeza-yates, 1999). A query is usually a Boolean expression of words input directly by the user. The query can be converted into a Boolean expression taking from a user's query free sentence. The terms in a query are linked together with AND, OR and NOT (Nie, 2011). This method is often used in search engines on the Internet (e.g Google) because it is fast and can therefore be used online (Nie, 2011). Unfortunately, the Boolean model has got its own drawbacks. It requires the users to have some knowledge of the search topic for the search to be effective (Baeza-yates, 1999). A wrong word in a query could rank a relevant document non-relevant. In addition to that, all retrieved documents are considered to be equally important. Another problem with the Boolean model is that most users find it difficult to translate their information need into a Boolean expression (Baeza-Yates & Ribeiro-Neto, 1999). This means most users need an intermediary to work with the Boolean retrieval model, which in turn brings its own problems, such as misunderstanding and possible misrepresentation of the information needs of the user.

The similarity of document dj to query term q in Boolean model expressed as:

$$\text{Sim (dj,q)} = \begin{cases} \text{1 if document dj matches query q(relevant)} \\ \\ \underline{\text{0 if document dj does not match q(irrelevant)}} \end{cases}$$

Boolean model is simple to reconstruct query and very useful in retrieval system where the exact match or limited result are required.

Vector Space Model (VSM)

The vector space model an Algebraic model which depicts the document in n-dimensional vector space. It computes vector values and cosine values between document and query to measure their similarity to retrieval most relevant document for the user queries. The cosine measure how much the retrieved documents are close to the query term in vector space.

The VSM solves the problem of the binary exact matching Boolean model by enabling non-binary term to retrieve partial match ranking in decreasing order based on the degree of similarity to query term (Baeza-yates, 1999).

The VSM Model, represent the document and the query by term-document matrix. 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 index-term weight enables to calculate the level of similarity between query terms and each document and to rank the retrieved document according to the degree of relevance. The term weighting and similarity measuring property of VSM made it the popular IR model.

Boolean matching and binary weights are too limiting the vector model proposes a framework in which partial matching is possible this is accomplished by assigning non-binary weights to indexterms in queries and 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 (Baeza-Yates & Ribeiro-Neto, 1999).

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 indexterms)
- The representations of document d_j and query q are t-dimensional vectors given by

$$\overrightarrow{\mathrm{d}}\,\mathrm{j} = (\mathrm{w1j}, \mathrm{w2}\,, . . . , \mathrm{wtj}) \qquad (2.4)$$

$$\overrightarrow{\mathrm{q}} = (\mathrm{w1q}, \mathrm{w2}\,, . . . , \mathrm{wtq}) \qquad (2.5)$$

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

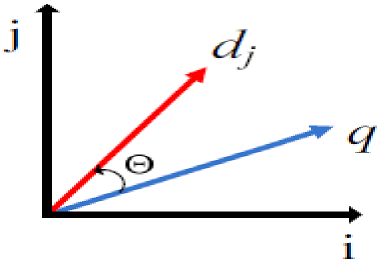


Figure 2. 1 Similarity between a document *d*j and a query q

$$\text{Cos (x)} = \frac{\overrightarrow{\text{dj}}\,.\,\vec{\text{q}}}{|\overrightarrow{\text{dj}}||\vec{\text{q}}|} \tag{2.6}$$

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.

VSM have several advantages. First, it improves retrieval performance using its term-weighting scheme. Second, the partial matching strategy of VSM allows retrieval of document approximate the query condition. Third, it sorts and ranks the documents according to their degree of similarity to the query using cosine similarity measurement. Finally, it is simple to implement and fast (Deerwester et al., 2000). However vector space model has got different drawback because it considers terms as unrelated objects in the semantic space. This means, if no common words are shared between the query and documents in text collection, the similarity value will be zero and no document will be retrieved. And also the model does not consider the uncertainty (Baeza-yates, 1999; Deerwester et al., 2000).

Probabilistic model

In the probabilistic model, the retrieval is based on the probability ranking principle. The central idea of probability is pre-estimation of the probability of relevance of the document to the user query (Baeza-Yates & Ribeiro-Neto, 1999). The characteristic of the probabilistic model to rank document to rank documents in decreasing order of their probability of relevance to the user query is taken as the most important feature of the model (Manning, 2009). Its advantages include; it provides users with a relevance ranking of the retrieved documents and it is easier to formulate queries because there is no need to learn any query language. In this model, the similarity measure is the ratio between the probabilities of finding relevant documents to the probability of finding non-relevant documents. The disadvantage of this model is, the need to guess the initial separation of documents and the adoption of the independence of index terms (Popescu, 2001).

2.6. Approaches to solve the problem of synonym and polysemy

A number of approaches have been employed to solve the problems of polysemy and synonymy in information retrieval systems (Baeza-Yates & Ribeiro-Neto, 1999). Some of these approaches are Ontology, thesaurus construction, relevance feedback and dimensionality reduction.

2.6.1. Ontology based approach

Ontology is a means to represent knowledge that is characterized by multiple concepts with the relationships between the concepts also represent, since the meaning of the text is not immediately obvious from the words or phrases (Staab & Hotho, 2003). Ontology based information retrieval recognizes the relations among terms by referring to the ontology.

The ontology definition of concepts can be used to describe the concepts and these concepts will be defined as document class. The real quality of ontology can be assessed only for its use in real application (Staab & Hotho, 2003). An ontology is a type of knowledge base that describes concepts through definitions that are sufficiently detailed to capture the semantics of a domain. An ontology captures a certain view of the world, supports intentional queries regarding the content of a database, and reflects the relevance of data by providing a declarative description of semantic information independent of the data representation (Sridevi. & Nagaveni., 2010). Creating ontology is not an easy task and obviously, there is no unique correct ontology for any domain (Sridevi. & Nagaveni., 2010). Calculating term importance is a significant and fundamental aspect for representing documents in conventional information retrieval approaches. It is usually determined through term frequency-inverse document frequency(TF-IDF).

Ontology-based representation allows the system to use fixed-size document vectors, consisting of one component per base concept. General approaches to ontology-based information retrieval are knowledge base and vector space model driven approach(Staab & Hotho, 2003). Knowledge base use reasoning mechanism and ontological query languages to retrieve instances. These approaches focus on retrieving instances rather than documents. Vector space model driven approach the idea is to represent each document in a collection as a point in a space (a vector in a vector space).

2.6.2. Thesaurus construction

Thesauri are valuable structures for information retrieval systems. A thesaurus provides a precise and controlled vocabulary which serves to coordinate document indexing and document retrieval. In both indexing and retrieval, a thesaurus may be used to select the most appropriate terms (Feldvari, 2000). When used during retrieval and searching, thesauri are useful in bridging the gap that exists between the metadata provided by the indexer and the concepts presented by searcher. The controlled vocabulary limits the terms available and increases the possibility that the query will use appropriate terms. This thesaurus is structured in a form of relationship which helps the searcher in navigation through the metadata and finding an appropriate query expression.

Manual Thesaurus Construction

The process of manually constructing a thesaurus is both an art and a science. We present here only a brief overview of this complex process. First, one has to define the boundaries of the subject area. (In automatic construction, this step is simple, since the boundaries are taken to be those defined by the area covered by the document database.) Boundary definition includes identifying central subject areas and peripheral ones since it is unlikely that all topics included are of equal importance. Once this is completed, the domain is generally partitioned into divisions or subareas. Once the domain, with its subareas, has been sufficiently defined, the desired characteristics of the thesaurus have to be identified.

Now, the collection of terms for each subarea may begin. A variety of sources may be used for this including indexes, encyclopedias, handbooks, textbooks, journal titles and abstracts, catalogues, as well as any existing and relevant thesauri or vocabulary systems. Subject experts and potential users of the thesaurus should also be included in this step. After the initial vocabulary has been identified, each term is analyzed for its related vocabulary including synonyms, broader and narrower terms, and sometimes also definitions and scope notes. These terms and their relationships are then organized into structures such as hierarchies, possibly within each subarea. The process of organizing the vocabulary may reveal gaps which can lead to the addition of terms; identify the need for new levels in the hierarchies; bring together synonyms that were not previously recognized; suggest new relationships between terms; and reduce the vocabulary size. Once the initial organization has been completed, the entire thesaurus will have to be reviewed (and refined) to check for consistency such as in phrase form and word form. Special problems arise in incorporating terms from existing thesauri which may for instance have different formats and construction rules. At this stage the hierarchically structured thesaurus has to be "inverted" to produce an alphabetical arrangement of entries--a more effective arrangement for use. Typically both the alphabetical and hierarchical arrangements are provided in a thesaurus. Following this, the manually generated thesaurus is ready to be tested by subject experts and edited to incorporate their suggestions.

Automatic Thesaurus Construction

In selecting automatic thesaurus construction approaches for discussion here, the criteria used are that they should be quite different from each other in addition to being interesting. Also, they should use purely statistical techniques. (The alternative is to use linguistic methods.) Consequently, the two major approaches selected here have not necessarily received equal attention in the literature. The first approach, on designing thesauri from document collections, is a standard one. The second, on merging existing thesauri, is better known using manual methods.

Here the idea is to use a collection of documents as the source for thesaurus construction. This assumes that a representative body of text is available. The idea is to apply statistical procedures to identify important terms as well as their significant relationships. It is reiterated here that the central thesis in applying statistical methods is to use computationally simpler methods to identify the more important semantic knowledge for thesauri. It is semantic knowledge that is used by both indexer and searcher. Until more direct methods are known, statistical methods will continue to be used.

By Merging Existing Thesauri

This second approach is appropriate when two or more thesauri for a given subject exist that need to be merged into a single unit. If a new database can indeed be served by merging two or more existing thesauri, then a merger perhaps is likely to be more efficient than producing the thesaurus from scratch. The challenge is that the merger should not violate the integrity of any component thesaurus.

2.6.3. Relevance feedback

Relevance feedback approach uses some ranked retrieved documents to reformulate the original user's query (Baeza-Yates & Ribeiro-Neto, 1999). Feedback phase's work in documents that are known to be relevant to the original query q are used to reformulate to qm. The expectation is that the query qm will return a good result of documents relevant to q. However, obtaining documents relevant to the original user's query is not easy and requires the direct interference of the user (Baeza-Yates & Ribeiro-Neto, 1999). This approach pass through some steps with involvement of end users. First users formulate a query to the system and retrieve some ranked documents as a result second, from these ranked documents, users select some relevant documents to their query, the system uses the ranked documents to reformulate the original query of users, finally the system retrieve some relevant document based the reformulated query (Xu & Croft, 2006). While the IR system expect users on deciding whether the top 10 results for a given query are relevant or not, unfortunately most users are unwilling to provide this information, mainly on the web(Baeza-Yates & Ribeiro-Neto, 1999). Because of this problem, the idea of relevance feedback has been difficult to use for years. But if the information collected from users is related to the original query, it's expected that relevance feedback will produce good results feedback (Baeza-Yates & Ribeiro-Neto, 1999). Feedback approach is composed of two steps explicit feedback and implicit feedback. In the explicit relevance feedback approach, the relevance judgment is provided directly by the users or by a group of human assistants. Whereas In implicit relevance feedback, there is no involvement of user in the relevance judgment. Instead, the feedback information is derived implicitly by the system.

2.6.4. Dimensionality Reduction

Dimensionality reduction technique seeks to resolve the problems of synonymy and polysemy by examining the "latent" structure of a document and the terms within it (Deshmukh & Hegde, 2012). These techniques decompose words and documents into vectors in a low dimensional space. These techniques decompose words and documents into vectors in a low dimensional space.

The variability in word choice between the system users and authors of documents are addressed because any word can be matched with another word to some degree in the low dimensional space (Deerwester et al., 2000). Latent semantic indexing is an example of these techniques. In this research Dimensionality reduction approach is proposed to handle synonymous and polysemous term and retrieving relevant information.

2.7. Latent semantic indexing (LSI)

According to Deerwester et al. (2000), The classic information retrieval models (Boolean, standard vector space and probabilistic) are used to finding relevant textual material depends on matching individual words in users requests with individual words in databased texts. There are usually many ways to express a given concept, so the literal terms in a user‘s query may not match those of a relevant document (Marco Suárez Barón, 2009). In addition, most words have multiple meanings, so terms in a user‘s query will match terms in documents that are not of interest to the user (Deerwester et al., 2000). LSI model looks for concept rather than looking at relevant term in the document corpus. There may not be direct query term occurrences in the document based on the semantic of language. LSI tries to overcome the problems of lexical matching by using statistically derived conceptual indices instead of individual words for retrieval (Baeza-Yates & Ribeiro-Neto, 1999). As

Baeza-yates (1999) stated LSI model is implemented in IR System in order to search and to find information based on the overall meaning of a document not only the meaning of the individual words. In the LSI model, a document collection is a built-in form of vector space by using Singular Value Decompositions (SVD) technique, a technique from Linear Algebra (Deerwester et al., 2000). A truncated singular value decomposition (SVD) is used to estimate the structure in word usage across documents. Retrieval is then performed using the databased of singular values and vectors obtained from the truncated SVD.

The benefit of the LSI model is that it solves two of the most problematic constraints in literal keyword searches: different words have the same meaning (synonymy) and the same word having different meanings (polysemy) (Fleur, 2015).

2.7.1. LSI Indexing approach

LSI uses common linear algebra techniques to learn the conceptual correlations in a collection of text. In general, the process involves constructing a weighted term by document matrix and then use rank reduced singular value decomposition to construct a low-rank approximation of this matrix (Fleur, 2015).

The term by document matrix is decomposed into a set of "r", a rank of the matrix, orthogonal factors from which the original matrix can be approximated by linear combination. Typically, any rectangular matrix, X, for instance, a tXd matrix of terms and documents can be decomposed into the product of three other matrices (Wei et al., 2008)

$$\mathrm{X} = \mathrm{USV}^{\mathrm{T}} \text{ or}$$

$$X = U0 * S0 * V0T \tag{2.7}$$

such that columns of Uo[u1,u2,u3,…ur] is a tXr matrix that define the left singular vectors of matrix X and Vo[v1,v2,v3,…vr] is a dXr matrix that defines the right singular vectors of matrix X, which are the orthogonal eigenvectors associated with XX^T and X^TX respectively. So is a diagonal matrix representing the singular values of X arranged in decreasing order, which are non-negative square roots of the Eigenvalues of X^TX (Baeza-Yates & Ribeiro-Neto, 1999 ; Popescu, 2001). This is called Singular Value Decomposition (SVD) of X. If only the "k" largest singular values of So (where K<= r (rank of A)) are kept along with their corresponding columns in the matrices Uo (u1, u2.. uk) and Vo(v1, v2 …vk), and the rest avoided, yielding matrices Uk, Sk and Vk, the resulting matrix Xk is the unique matrix of rank "k" and is closest in the least-squares sense to the original matrix X (Popescu, 2001; Baeza-Yates & Ribeiro-Neto, 1999).

$$\underset{tXd}{X \sim Xk} = \underset{tXk}{Uk} * \underset{kXk}{Sk} * \underset{kXd}{VkT} \qquad (2.8)$$

The point here is that the reduced matrix Xk, by keeping only the first "k" independent linear components of X, captures the major associational structure in the matrix and also much of the noise caused by, such as, polysemy and synonymy, that leads to poor information retrieval is removed with the dimensionality reduction (Rosario, 2000).

Hence, the selection of the right dimensionality or the value of "k" is a crucial issue. Of course, the value of "k" should be large enough to integrate all the real structure in the document collection, but also it should be small enough so that noises that are caused by the variability in word usage are not included (Deerwester et al., 2000). A document which has no words in common with a user's query may be near to the query if that is consistent with the major pattern of word usage. Geometrically, the location of terms and documents in the approximated k dimensional space is given by the row vectors from the Uk and Vk metrics respectively (Deerwester et al., 2000). The cosine or dot product between vectors in this space corresponds to their estimated similarity. The goal is not to describe the concepts verbally, but to be able to represent the documents and terms in a unified way for exposing document-document, document-term, and term-term similarities. The representation of both term and document vectors in the same space makes the computation of the similarity between any combination of terms and documents very easy. From IR point of view, three basic similarity comparisons in the reduced matrix Xk are important (Deerwester et al., 2000 ; Rosario, 2000).

Term by term comparison

In the term or rows of the vectors, the comparison is used to determine the extent to which two terms(term to term) have a similar pattern of occurrence across the document. In the term or rows of the vectors, the comparison is used to determine the extent to which two terms(term to term) have a similar pattern of occurrence across the document.

$Xk.\ Xk^{T} = (Uk.Sk.Vk^{T}).\ (Uk.Sk.Vk^{T})^{T} = UK.\ Sk.\ Vk^{T}.\ Vk.\ Sk.\ UK$

$= Uk.\ Sk.\ (Vk^{T}.\ Vk).\ Sk.\ Uk^{T}$

$$= (Uk.\ Sk).\ (Uk.\ Sk)^{T} \qquad (2.9)$$

Here, we can see that the i, j cell of the square matrix Xk. Xk^{T} can be obtained by taking the dot product between the ith and jth rows of the matrix Uk. Sk. We can consider Uk. Sk as coordinates for terms, because since Sk is a diagonal matrix, it is merely used to stretch or shrink the axis of the reduced vector space without affecting the position of Uk in the space.

Document By Document Comparison

The analysis is used to determine the similarity between the documents by identifying the extent to which the two documents have a similar term occurrence pattern(Deerwester et al., 2000). This can be achieved by computing the dot-product of column vectors of the reduced matrix Xk i.e.

$Xk^{T}.\ Xk = (Uk.Sk.Vk^{T})^{T}.\ (Uk.Sk.Vk^{T}) = UK.\ Sk.\ Vk^{T}.\ Vk.\ Sk.\ UK$

$= Vk.\ Sk.\ (Uk^{T}.\ Uk).\ Sk.\ Uk^{T}$

$$= (Vk.\ Sk).\ (Uk.\ Sk)^{T} \qquad (2.10)$$

The i, j cell of Xk^{T}. Xk, which is a document-by-document square matrix, is obtained by taking the dot product between the ith and jth row of the matrix as coordinates for documents, Vk. Sk. Though, the space of Vk. Sk is just a stretched or shirked version of the space of Vk, in proportion to the corresponding diagonal elements of Sk.

Term by Document Comparison

The third comparison is between a term and a document. However, the comparison between a term and a document is just the value of an individual cell of Xk. Hence,

$$\mathrm{Xk} = \mathrm{Uk.Sk.Vk}^{T} \quad (2.11)$$

$\mathrm{Uk.\ Sk}^{\frac{1}{2}}.\ \mathrm{Sk}^{\frac{1}{2}}\mathrm{Vk}^{T} = (\mathrm{Uk.\ Sk}^{\frac{1}{2}})\ (\mathrm{Vk.Sk}^{\frac{1}{2}})^{T}$

This implies that the i, j cell of Xk is obtained by taking the dot-product between the i^{th} row of the matrix Uk .Sk½ and the jth row of the matrix Vk .Sk½ (or the jth column of (Vk .Sk½)^{T}). Here again, the effect of the factor Sk½ is stretching or shrinking of the axes by a factor of its value.

Query representation

One of the major challenges in information retrieval is the description and representation of information needs in terms of a query. Among the central and essential feature of information. Retrieval is matching the text of the query to the text of the document in the corpus. In LSI based IR systems, a user's query is often considered as a pseudo document and is represented as a vector in the reduced term-by-document space (Deerwester et al., 2000 ; Rosario, 2000)

First, the terms used by the searcher are represented by an (m X 1) vector "q" whose elements are either zero or correspond to the frequency of terms that exists in the database of reduced vector space. The local and global term weights used for the document collection are applied to each non-zero elements of the query vector q. Then the query vector is represented in the reduced LSI space by the vector

$$\hat{p} = qT.\,Uk.\,SkT \quad (2.12)$$

Where q^{T}. Uk is the sum of term vectors precise by vector q scaled by Sk^{T} (Rosario, 2000 ; Deerwester et al., 2000). Finally, the query vector can then be compared to all existing document vectors, and the documents ranked by their similarity (nearness) to the query. One common measure of similarity is the cosine between the query vector and document vector. Typically, the first "n" closest documents or all documents exceeding some cosine threshold are returned to the user (Deerwester et al., 2000).

2.8. Related works

2.8.1. Information retrieval for Non- Ethiopian language

Claveau & Kijak (2016) proposed distributional thesauri for information retrieval based on probabilistic methods using latent semantic indexing (LSI). The corpus was collected from AQUAINT-2 composed of articles in English containing a total of 380 million of words. For a given input word these thesauri identify semantically similar words based on the assumption that they share a similar distribution than the input word's one. The random projection was selected for dimensionality reduction and they have tested the performance of the thesaurus for query expansion. For 50 queries and over more than 170,000 English documents were considered. By selecting the top 10,50,100 thesaurus terms resulted to have the best gain and the precision was above 14% by considering the intrinsic measure. Intrinsic means taking sample words from WordNet and determines how 30 much it is similarly based on the words co-occurrences. Using the intrinsic measure, a total of 38 neighbors were found from 12243 common nouns using and they used the cosine and mutual information similarity and weighting measures respectively. But using intrinsic measure is not advisable because the neighboring words are limited to a certain synonym set or totally may be absent from the WordNet or Moby reference lists, but extrinsic comparisons are better for accurate query retrieval and expansion. The evaluation of distributional thesaurus through information retrieval tasks has been explored and the performance was tested using different cut-offs using information retrieval and high gain and precision were achieved.

AbuZeina et al. (2018) conducted experiments to examine the effectiveness of LSI in the Arabic language. The authors proposed a novel approach to enhance the quality of the retrieved documents in search engines for the Arabic language. They used a new extension of the LSI technique instead of just using the standard LSI technique. The proposed method is based on employing the word co-occurrences to form a term-by-document matrix. they evaluated the performance of the proposed method using an Arabic data collection that contains more than 500 documents. The authors concluded that the proposed method is more effective than the standard LSI to enhance the quality of the retrieved documents in search engines.

2.8.2. Related works of Ethiopian language

Multiple works on Ethiopian languages have been done in information retrieval aimed to improve the performance of the retrieval process. But the work founded by the reviewer involves only Amharic, Afaan Oromo and Silt'e related studies in the area of IR.

Amharic IR systems

Bruck (2015), conducted an experiment to articulate the problem that was seen in the Amharic IR system, specially the impact of polysemy and synonymy of terms in the documents. As a result of that different users may describe their information needs differently, for that matter, their queries may differ lexically. Therefore to overcome this problem the researcher proposed semantic-based query expansion. Bruck (2015), has used text preprocessing to make avail the terms in the document to indexed in efficient manner. vector space model was devised to the IR system developed by him. The expansion of the query given to the system based on the knowledge based of Amharic language or thesaurus. In his study, the performance was evaluated using the usual IR system measurement criteria, recall and precision. The performance of the system measured compared with the standard vector space model. The performance of the system registered recall of 0.92 and precision of 0.68. These shown us query expansion in Amharic IR system registered better recall but the precision gone down.

Promising work was done on Amharic text retrieval system for enhancing Amharic IR by (Tewodros, 2003). He was aimed to solve problem exact term matching techniques by using latent semantic index (LSI) with singular value decomposition. The researcher pursued three successive phases to develop a latent semantic indexing model to make the mode suited with Amharic documents. In the first phase preprocessing (extracting terms, term-document matrix generation and calculating term weighting) and indexing were included. The next phase could be performing K-dimensional Singular Value Decomposition (SVD). Finally, Query Projection, Matching and Ranking of Relevant documents The system evaluated by the two commonly known parameters, recall and precision. The average precision for the LSI approach (0.7157) and the average precision of the vector space approach (0.6913) with 206 news articles and 9256 indexing terms. This implies a 2.4% improvement over the standard vector space method.

Semantic indexing and document clustering for Amharic information retrieval has done by (Wordofa, 2013). His work mainly focuses on the indexing and clustering process, to solve the problem found in the semantic natures of the language. The system retrieves the information needs of the user by retrieving documents which have similar meaning to the query formulated by the user rather than exact terms match. It comprises three basic components indexing, clustering and searching. The system comprises all processes exist in generic IR plus to that C-value technique multi-word term extraction, k-means algorithms document clustering, cluster based searching strategy used. In his study, the performance was evaluated using the most common and basic statistical IR system measurement criteria, precision, recall and F-measure. In his study, the performance was evaluated using the most common and basic statistical IR system measurement criteria, precision, recall and F-measure. The performance of the system registered an average of 88% recall, 51% precision and 72% f-measures for frequent queries string. And average 60% recall, 42% precision and 60% F-measures for less frequent string queries. When we calculate the average performance over frequent and less frequent query strings, we found an average of 74% recall, 46% precision, and 66% F-measures. These shown us semantic indexing and document clustering based Amharic IR system registered 6% of F-measure improvement. This means the performance of the Amharic IR system increases from F-measure of 60% (Amanuel) to F-measure of 66% accuracy. The researcher believes that the performance of the system can be increased if a context-aware IR system using a well-developed thesaurus to aware the context to which the terms in the text appears. In addition, using well-tagged corpus of bigger size could help to discriminate the context in which terms are presented.

Afaan Oromo retrieval system

Promising work was done on semantic indexing based Afaan Oromo text retrieval system for enhancing Afaan Oromo IR by (Anbase, 2019). In his study, the potential of applications of Latent Semantic indexing approach in Afaan Oromo text retrieval is investigated. For experimentation, the researcher prepared 70 Afaan Oromo text documents corpus and 9 user queries. All text documents were obtained from the website of Oromia National Regional State, Voice of America Radio Afaan Oromo service and the International Bible Society official website. To develop the prototype he used the NetBeans. The designed prototype has two main components: indexing and searching. Once the corpus has been collected different pre-processing (stop word removal, tokenization, normalization and stemming) activities were employed on the documents to make them suitable for indexing. In his study, the System evaluation was based on precision and recall and his experiment showed that the performance of the prototype is on average 67% precision and 63% recall registered.

Afaan Oromo-Amharic bilingual dictionary using parallel documents were developed by Eyob. All the collected documents were used for the construction. In the collected data different pre-processing tasks, like tokenization, normalization, stemming, is used for the word alignment tool and information retrieval task. The most mandatory and minimum requirement input files for word alignment are the vocabulary file and the bitext file. These files were generated by the packages available in GIZA++ toolkit. Then, as input for the GIZA++ to create word alignment statistical information of vocabulary and bitext file generated was used. In this way, the researcher developed the bilingual dictionary and this dictionary served as a translation knowledge source. The system is tested with 50 queries and 50 randomly selected documents. Two experiments were conducted, the first experiment is done by using only one possible translation to each Afaan Oromo query term and the second experiments is done by using all possible translations. The retrieval effectiveness of the system is evaluated by the two commonly known parameters, recall and precision for both monolingual and bilingual runs. Accordingly, the result of the first experiment showed a maximum average recall value of 0.58 and maximum average precision of 0.81 for the monolingual run; and a maximum average recall of 0.38 and maximum average precision of 0.45 for the bilingual run. The result after conducting the second experiment returned a maximum average recall of 0.7 and a maximum average precision value of 0.6. The result of the second experiment showed a better result of recall and precision than the first experiment. The result obtained in the second experiment is a maximum average precision of 0.60 for the bilingual run and the result for the monolingual run remained the same. From the second experiment, it can be concluded that using all possible translation can be used to improve the overall retrieval effectiveness of the system.

Promising work was done by (Gezehagn, 2012). He designed and developed information retrieval for Afaan Oromo Text. He aimed to apply and evaluate the VSM approach to Afaan Oromo text Retrieval to make effective retrieval of the document in the language with Afaan Oromo query. For the study, 100 different textual documents were collected from different news media. The collected corpora involve different subjects like politics, education, culture, religion, history, social, health, economy and other events. The similarity measure is done by using the popular cosine similarity measure. The searching component is based on the Vector Space Model and this was implemented using python script.

For testing the designed system all the collected documents and 9 queries were prepared; these queries are marked across each document as either relevant or irrelevant to make relevance evaluation. The study used precision and recall as a measure of effectiveness. Results from the experiment returned an average performance of 57.5% precision and 62.64% recall.

Silt'e information retrieval system

Vector Space Model-based Silt'e text retrieval has done by (Jemal, 2016). The researcher aimed to apply and evaluate the Vector Space Model approach to Silt'e text retrieval to make effective retrieval of documents in the language with Silt'e query. For the study, 100 different textual documents were collected from schools books. The collected corpora involve different subjects like politics, education, art, social, health, and other events. The designed system has two main components: indexing and searching. Once the corpus has been collected different pre-processing activities were employed on the documents to make them suitable for indexing. The index file structure used in the study is inverted index. Inverted file index has two files Vocabulary file and Post file which were used in building vectors of a document versus terms. Index terms should be content bearing words and for this task, stop word list has been prepared manually. The term weighting technique used in the study is term frequency-inverse document frequency (tf-idf). The similarity measure is done by using the popular cosine similarity measure. For testing the designed system all the collected documents and 10 queries were prepared; these queries are marked across each document as either relevant or irrelevant to make relevance evaluation. To develop the prototype the researcher used the Apache Solr. Results from the experiment returned 65% F-measure .

Johar (2017), experimented the efficiency of Probabilistic IR Model for Silt'e text retrieval employing relevance feedback for query expansion. For the study, the researcher prepared 134 different textual documents from Silt'e textbook and Silt'e cultural and tourism bureau. The collected corpora involve different subjects like politics, education, culture, religion and other events. The similarity measure is done by using the popular cosine similarity measure. Apache Solr was used to develop the prototype. The system registered 68% F-measure. The researcher believes the system's efficiency can be improved by improving the stemming algorithm, and using standard test corpus.

Latent Semantic Indexing is an effective technique to improve the performance of information retrieval systems. In different studies showed that latent semantic indexing technique leads to performance improvement. Research have been done for Silt'e retrieval system without latent semantic indexing technique. Even if different researchers attempt to develop semantic indexing based information retrieval system for different languages, latent semantic indexing techniques depend on characteristic of the language. As far as our knowledge, there is no attempt of semantic indexing system for Silt'e language. Therefore, Silt'e language needs development of semantic indexing based information retrieval which considers the characteristic of the language.

CHAPTER THREE: METHODOLOGY

3.1. Introduction

This chapter discusses the methods and procedures in order to development of semantic indexing based IR for Silt'e text. Two researchers had conducted researches to tackle problems exhibited in Silt'e text retrieval and to enhance the performance of the retrieval system as discussed in section 2.9.2. The researchers tried to overcome the problems using VSM and using probabilistic methods of Silt'e IR. Though these research were conducted, still there are several problems with regard to Silt'e IR . This is because the problem related with the synonymous and polysemous word. In this research, the problems stated above are handled and tried to solve.

3.2. Data set preparation

Data sets are required to train and test the effectiveness of IR system. buts there is no standardized corpus prepared before in Silt'e language. For this research work, we prepared the data set by collecting the text document written in Silt'e language from different sources that can represent the language in the general aspect education, culture, arts, health and politics. Such sources are school textbooks prepared in Silt'e language and different reports from Silt'e culture and tourism bureau and Hosanna teacher training college. The collected documents are filtered manually and we applied some text preprocessing operations, which are language-dependent processes such as tokenization, normalization, stop words removal and stemming to make it ready for use in the IR system. The size of the corpus is an important factor in the performance of the IR system; with the general point that the amount of data is big, it has a better quality (Manning, 2009). However, it is very difficult to obtain standardized data set for under-resourced language like Silt'e that why a limited size of corpus we used.

For prototype evaluation, a total of 700 documents were collected and stored as utf-8 encoding format which is suitable to support the languages. As shown in table 3.1, the document corpus contains five clusters, which are health, education, social, political and art.

Table 3. 1 Corpus used for developed IR system

No	Types of Documents	Number of Documents
1	Health related	145
2	Education related	200
3	Social related	155
4	Politics related	115
5	Art related	85
Total		700

3.3. Test procedure

To test the system test queries were formulated and relevancy judgments were prepared. For this research, we prepared 56 text queries. The queries used in the experiment are attached as **Annex 5**. These queries are marked across each document as either relevant or irrelevant. These queries are selected subjectively by the research after reviewing the content of each document manually. To perform relevance judgement we selected 16 people with different backgrounds.

3.4. System Architecture

Designing an information retrieval system passes through a series of steps. The standard architecture of an IR system consists of two major parts, i.e., indexing and searching. Indexing is an offline process that is dedicated to organizing documents using a list of words, which are extracted from the documents themselves. The second part of the IR system is an online process that is searching for relevant documents. It consists of query processing, matching, and document ranking. The architecture of the proposed prototype of the system is depicted in figure 3.1 The components are discussed in detail afterwards.

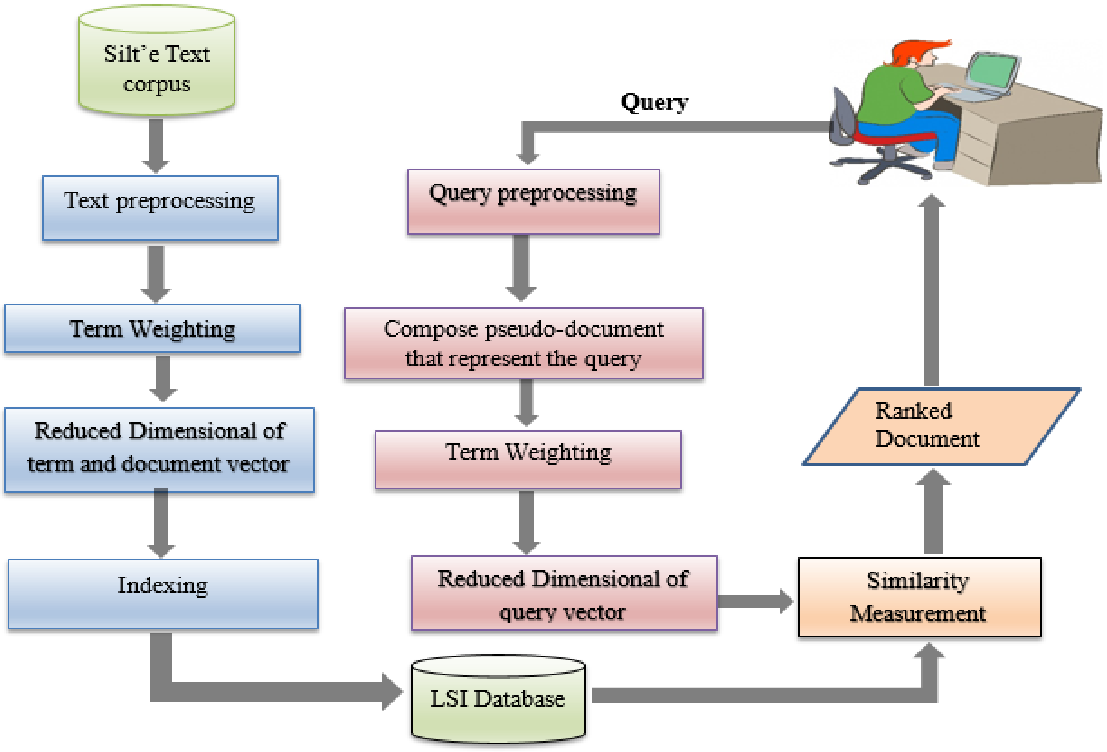


Figure 3. 1 System Architecture for Silt'e Information Retrieval

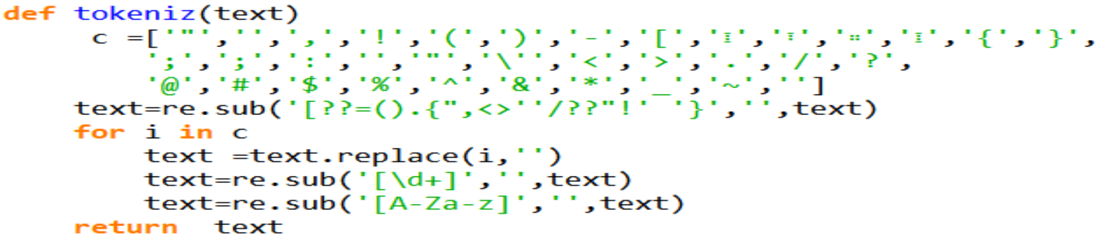
The first thing we have to do in information retrieval system research is collecting documents that are used for indexing. After the documents were collected text preprocessing task is followed. After prepossessing is done, the next step is term weighting , Both the term-document matrix and the query vector are weighted using tfidf weight to increase or decrease the significance of each term based on their ability to represent the content of a document and also discriminate it from other document collection. The weighted term-document matrix is then given to the SVD algorithm as an input and a reduced dimensional representation of the matrix is generated from it. The reduced-dimensional representation of the query vectors is also obtained and reduced into the reduced space. After the terms, documents and queries are represented in the same reduced dimensional space, the cosine similarity measure is used to identify the ranked list of relevant documents for each query of the system.

3.4.1. Text Preprocessing

The major text preprocessing tasks are discussed below.

Tokenization

The Silt'e writing system uses white space to separate one word from other in the document the termination of a sentence is always end up with punctuation marks. The stream of characters in natural language text must be split into distinct meaningful units before any further natural language processing task. The data collected from different documents were in sentence form with a punctuation mark, numbers, special characters and control characters. Consider the statement "ሉላሉሌ ዪንጄ ሉባምቸነ ‹‹ፋገ››፣ ‹‹፣አረሺ››፣ ‹‹ቡደ››፣››" the tokens after tokenizing were done splitting the sentence are "ሉላሉሌ", "ዪንጄ", "ሉባምቸነ", "ፋገ", "አረሺ" and "ቡደ". This tackle treating individual words from the entire. So, removing these irrelevant things should be done before data processing began; and then, separating a sentence or a paragraph into tokens called tokenization. figure 3.2. is python code that tokenize and remove punctuation mark.

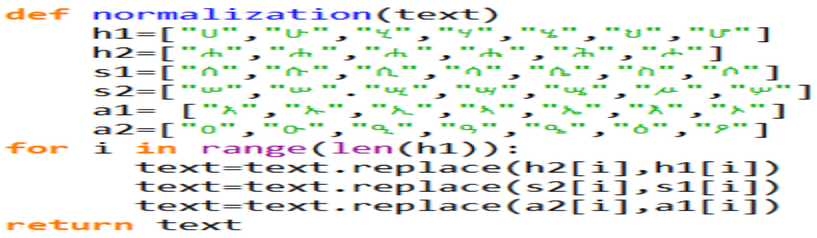


```
def tokeniz(text)
     c =['"',' ',',','!','(',')','-','[','፤','፣','።','፤','{','}',
         ';',';',':','.','"','\'','<','>','.','/','?',
         '@','#','$','%','^','&','*','_','~','']
    text=re.sub('[??=().{",<>''/??"!'  '}','',text)
    for i in c
        text =text.replace(i,'')
        text=re.sub('[\d+]','',text)
        text=re.sub('[A-Za-z]','',text)
    return  text
```

Figure 3. 2 python code for tokenization and punctuation mark removal

Normalization

In contrast to Amharic, Silt'e uses distinct 26 Ethiopic alphabets (in Silt'e language writing system different letters that has similar sound, like , ሐ, ሠ, ዐ are not used, instead only ሀ,ሰ አ, are used). But, in some document of the real life, people experienced writing of these similar sound characters which are used in Silt'e writing systems. Even though these symbols sound the same, in Silt'e language they are not part of the alphabet. But in case people use them they must be changed to alphabets that are used in Silt'e, that have similar sound form. Therefore, for the sake of flexibility the prototype will handle this kind of situations as they appear in the documents. figure 3.4 has been used to normalize text.



```
def normalization(text)
    h1=["ሀ","ሁ","ሂ","ሃ","ሄ","ህ","ሆ"]
    h2=["ሐ","ሑ ","ሒ ","ሓ","ሕ","ሖ"]
    s1=["ሰ","ሱ","ሲ","ሳ","ሴ","ስ","ሶ"]
    s2=["ሠ","ሡ "."ሢ","ሣ","ሤ","ሥ","ሦ"]
    a1= ["አ","ኡ","ኢ","ኣ","ኤ","እ","ኦ"]
    a2=["ዐ","ዑ","ዒ","ዓ","ዔ","ዕ","ዖ"]
for i in range(len(h1)):
      text=text.replace(h2[i],h1[i])
      text=text.replace(s2[i],s1[i])
      text=text.replace(a2[i],a1[i])
return text
```

Figure 3. 3 Python code for normalization

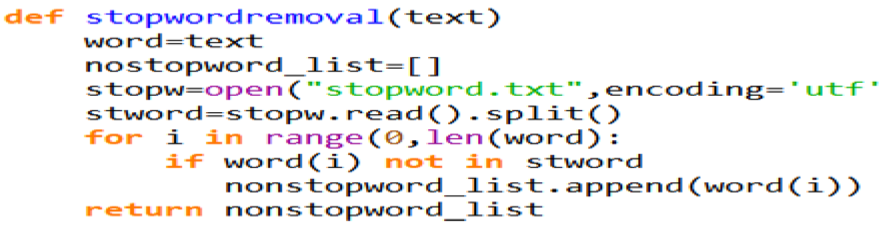
Stop words

Stop-words are the most frequent terms which are common to every document and have no discriminating one document from the power other. Sometimes, some extremely common words which would appear to be of little value in helping select documents matching a user need are excluded from the vocabulary entirely (Hans, 2005).

Stop word elimination helps to reduce the size of the index structure and to save storage space by removing no-content bearing terms, thereby increasing the speed of searching (Manning, 2009). Different stop word removal techniques are developed. The two commonly used techniques are IDF and dictionary lookup. IDF (Inverse Document Frequency) is a frequency analyzing methods that count the frequency of occurrence of words in one or more document and assumes the more frequent words as stop word. The problem in IDF techniques is that it will consider and remove all more frequent words as stop word even if they are not.

Since stop words do not have significant discriminating powers in the document collection; the researcher filtered stop words list to ensure only content bearing terms are included.

Silt’e stop word list is saved in the text file “stopword.txt”. First, the algorithm reads the files and saves them on variable. Then the normalized token is checked to be different from the terms in the stop word. Terms not in stop word are forwarded to the stemmer function. The figure 3.3 has been used for detecting and removing stop word from the given text.

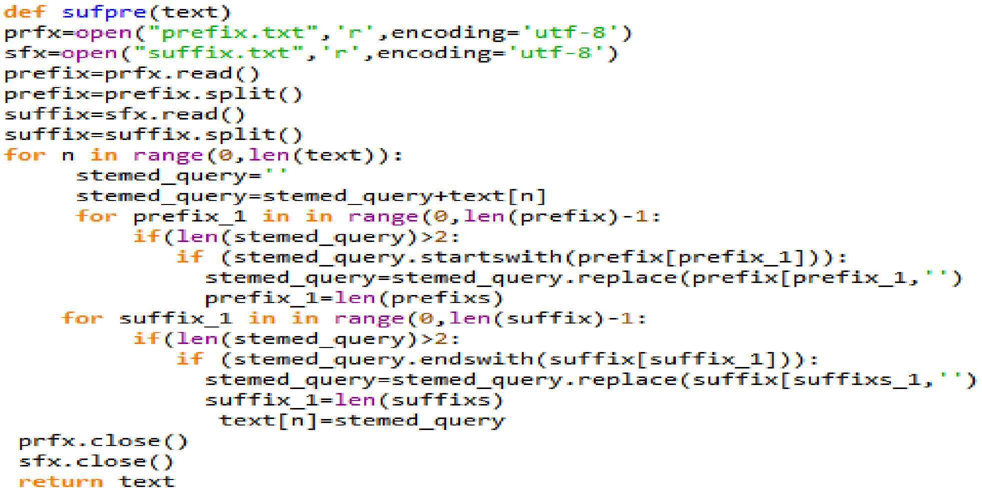


```
def stopwordremoval(text)
     word=text
     nostopword_list=[]
     stopw=open("stopword.txt",encoding='utf'
     stword=stopw.read().split()
     for i in range(0,len(word):
          if word(i) not in stword
             nonstopword_list.append(word(i))
     return nonstopword_list
```

Figure 3. 4 Python code for remove stop word.

Stemming

Stemming is the process of converting words to their based form in order to match different tenses, moods, and other inflictions of a word. The based 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 (Manning, 2009). Stemming is a language-dependent process in a similar way to other natural language processing techniques. The stemming operation most of the time is applied to the textual data. In developing stemming algorithms, the knowledge of a particular language or language expert supports are required. It is often removing inflectional and derivational morphology. E.g. በስልጤ, የስልጢኘ, የስልጤ → ስልጤ. Thus, stemming of the Silt'e corpus and query is done by using a rule-based stemmer developed by (Kedir, 2012). Stemming techniques are language-dependent. Therefore, every language is using its own specific stemming technique for increasing efficiency. Figure 3.4 has been used for remove prefix and suffix in the given text.



```
def sufpre(text)
prfx=open("prefix.txt",'r',encoding='utf-8')
sfx=open("suffix.txt",'r',encoding='utf-8')
prefix=prfx.read()
prefix=prefix.split()
suffix=sfx.read()
suffix=suffix.split()
for n in range(0,len(text)):
     stemed_query=''
     stemed_query=stemed_query+text[n]
     for prefix_1 in in range(0,len(prefix)-1:
         if(len(stemed_query)>2:
            if (stemed_query.startswith(prefix[prefix_1])):
              stemed_query=stemed_query.replace(prefix[prefix_1,'')
              prefix_1=len(prefixs)
    for suffix_1 in in range(0,len(suffix)-1:
         if(len(stemed_query)>2:
            if (stemed_query.endswith(suffix[suffix_1])):
              stemed_query=stemed_query.replace(suffix[suffixs_1,'')
              suffix_1=len(suffixs)
               text[n]=stemed_query
 prfx.close()
 sfx.close()
 return text
```

Figure 3. 5 python code for stemming

3.4.2. Term weighting

The term weighting was done by using tfidf weighting scheme for measuring how much information is given by a term according to its document frequency. The term by document matrix is generated in the term by document matrix. Then, the weighted matrix is given into the SVD function. Then the weighting to each of its elements and returns a weighted term documents matrix as the value that shows the significance of the term in the document collection used for index term. The basic rational behind weighting is that a term has high weight if it is frequent in the relevant documents but infrequent in the document collection (Tewodros, 2003).

$$\mathrm{WT} = \mathrm{TF} * \log\left(\frac{\mathrm{N}}{\mathrm{DF}}\right) \tag{3.1}$$

Where TF is the frequency of each term in the respective document and DF is the number of documents that contain the given term.

3.4.3. K-dimensional Singular Value Decomposition (SVD)

The weighted term-document matrix “W” is computed in the previous stage through (tfidf) is used as an input for SVD and then, the SVD also computed for a reduced dimension K, where K is less than both the number of terms and the number of documents in the matrix. The K-dimensional SVD of the weighted matrix is performed. The SVD function accepts the weighted matrix and the number of dimensions as an input and returns three matrices, because it reduces the dimension of the data set and the three matrices are: the K-largest left singular vectors, Uk, of W, the K-largest right singular vectors, Vk, of W, and a diagonal matrix Sk whose non-zero entries in the principal diagonal are the singular values of W arranged in decreasing order and then the system checks if the user entered some value denoting the number of singular values. In a simple way, [UK, Sk, Vk] =svd (W, k), where W is the term-document weighted matrix and k is the reduced dimensionality of the weighted term-document matrix. In the LSI there is no way to determining the optimal number of dimensions to use for performing the SVD. As we have mentioned in section 2.8, there is no general rule that can be used to select the optimal value of dimension K. So, in this thesis, the value for the dimension is chosen by the principle of “What works best” and suitable depending on the nature of the document collection. That means, the retrieval performance of the LSI model was examined for several different dimensions, and the dimensionality that maximizes the performance of the system was chosen.

3.5. Query Projection, Matching, and Ranking of Relevant Documents

3.5.1. Query Projection

The system takes user queries and then the queries are treated as pseudo-documents for the user. Therefore, the NLP performed on the collected test document is, also performed on the queries and the vectors representing the frequency of each term in the queries. In addition to that, the query could represent in the same space with the collection test document, after that, they are weighted by using the tfidf weighted used in weighting the documents and then projected into space. A SVD function, designed to compute the weighting and project the queries into the reduced space using the formula "q=qT *Uk * Sk-1".

3.5.2. Matching and Ranking of Relevant documents

When you find an LSI-indexed databased, the queries and the documents are located in the same reduced LSI space, it is possible to compute the Euclidean distance between the queries and each document and take the nearest documents as the best ones for the specific query because ideally, concept space is much less than the word space. Consequently, having the reduced dimensional representation of the query vector "q" and the scaled document matrix. "d". After the cosine measure is computed, the documents are sorted according to the cosine coefficients, the larger the cosine coefficient, the more relevant the document and LSI returns the relevant documents that do not contain the keyword at all. 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 (Baeza-Yates &Ribeiro-Neta, 2011).

3.6. Evaluation of information system

The goal of IR system is retrieving relevant documents from the collection that satisfies user's information need to evaluate the performance of Silt'e IR, In this study, the three widely used retrieval effectiveness measure such as precision, recall, and F- measure are used.

Recall

Recall is a measure of the ability of a system to present most relevant items that are available in the corpus.

$$\text{Recall} = \frac{\text{Number of relevant items retrieved}}{\text{Number of relevant in collection}} \qquad (3.2)$$

It is important to measure recall for circumstances where the searcher wants as much information on the topic as possible and therefore is interested in retrieving as many relevant results as possible. Recall on its own is not very useful, we need to compare it with the number of no relevant documents by calculating precision (Baeza-Yates & Ribeiro-Neto, 1999).

Precision

Precision is a measure of the ability of a system to present only relevant items taking into account all retrieved documents

$$\text{Precision} = \frac{\text{Number of relevant items retrieved}}{\text{Total number of items retrieved}} \qquad (3.3)$$

IR systems aim to have high precision because this means that the majority of documents retrieved are relevant to the user needs. It should be noted recall and precision clearly trade off against one another.

F-Measure

The F-measure is used to measure the performance of the system since it balances the precision and recall values. F-measure is a single measure that trades off precision versus recall. It is the weighted harmonic mean of precision and recall.

$$\text{F - measure} = \frac{2 * (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \qquad (3.4)$$

CHAPTER FOUR: RESULT AND DISCUSSION

4.1. Introduction

This chapter reports on the experiments conducted using the architecture designed in chapter three, and the findings from the experiment. In this research, an attempt has been made to design an information retrieval system for Silt'e language using latent semantic indexing. For this research study, we prepared a dataset collected from different sources.

4.2. Description of the Prototype System

In developing the prototype of LSI based IR system for Silt'e language, several components are integrated together. This system has indexing and searching. The main objective of this prototype system is to represent documents, index terms, and queries in a similar reduced-dimensional space so that comparison of the documents, index terms and queries in this reduced-dimensional space is possible. A standard VSM, obtained in the middle of the prototype model. This makes the concept space is ideally much lesser than the word space because the LSI model is straightforward. The document and the sample query files stored in their respective folders are read and each term are extracted excluding the stop word. The term by document matrix and the query vector are generated from it. To enhance or reduce the significance of each term based on their capability to represent the content of a document and also distinguish it from other documents in the collection of the documents; both the term-document matrix and the query vectors are weighted. The weight of the term-document can be calculated using tfidf term weighting methods. The term weighting was done by finding the frequency of terms in each document and its synonyms in the documents and product of its inverse document frequency of it. The weighted term-document matrix is then given to the SVD algorithm as an input and a reduced dimensional representation of the matrix.

After the documents, terms and queries are represented in the similar reduced dimensional space, the cosine similarity measure is used to categorize the ranked list of relevant documents for each of the required query from a collection of documents.

Text preprocessing computations were applied to the collected documents to make them readily available to be being indexed and searched. After the document identified and collected the text preprocessing tasks are followed. The first preprocessing task is tokenization. Tokenization in this work also used for splitting the document into tokens and detaching certain characters such as punctuation marks or non-content bearing words. Finally, the tokenized document will be returned for the next process. After the documents tokenized, the next step is normalizing. After text is normalized the prototype extracts the content bearing terms like nouns and verbs of the document from the token set to keep the semantics of the document. On the semantics set of the token set some major activities carried out; stop word removal, and stemming.

Silt'e stop word list is saved in the text file "stopword.txt". First, the algorithm read the files and saves them on the variable. Then the normalized token is checked to be different from the terms in the stop word. Terms not in stop word are forwarded to the stemmer function. Stemming can be done for computational efficiency and better retrieval performance it is crucial to conflate morphological variations. The next part is creating an inverted file form by using term by document matrix generation latent semantic indexing being a variant of vector space method, the documents, as well as the queries, are represented mathematically as vectors in some vector space (Baeza-Yates &Ribeiro- Neta, 2011).

We used a function called term_document_matrix to find the number of times a particular word appears in each document. It consists of words as columns and document numbers as rows. Then, a text file containing the list of all individual terms is created. The frequency of each term in each document is read from the text file generated. Then the numerical data-pre-processing/weighting was done by using tfidf term weighting methods. The term weighting was done by finding the frequency of terms and their synonyms in the documents and the product of its inverse document frequency in the documents. The body of the term by document matrix generated in the term by document matrix generation, and saved into a text file in a folder. Term weighting has been employed in the form of a double dimension array to allow building of the term-document matrix. The weighted term-document matrix "W" of the preceding step is used as an input and its SVD is computed for a reduced dimension K, where K is less than both the number of terms and the number of documents in the matrix. The function accepts the weighted matrix and the number of dimensions as an input and returns three matrices: the K-largest left singular vectors, K-largest right singular vectors, and a diagonal matrix Sk whose non-zero entries in the principal diagonal are the singular values of Weight arranged in decreasing order and then the system checks if the user entered some value denoting the number of singular values. The system takes user queries and then the queries are treated as pseudo-documents. So, the natural language preprocessing achieved on the collection of the test document is also performed on the queries and the vectors specifying the frequency of each term in the queries are obtained. In addition to that, in order to represent the queries in the same space with the test document collection, they are weighted using the same scheme used in weighting the documents, tfidf, and then projected into the same space. A SVD function, designed to perform the weighting and project the queries into the reduced space using the formula q=qT *Uk * Sk-1. Now, that the queries and the documents are located in the same reduced LSI space. Because the concept space is ideally much lesser than the word space, we can limit the number of singular values such that important semantic information is not lost and noise is removed from the system. Fifty Six queries were identified for the testing purpose. To handle the query entered by the user and then compute the similarity of the query against documents in the corpus using Latent Semantic Indexing.

Consequently, having the reduced dimensional representation of the query vector "q" and the scaled document matrix "d", the angle between them can be computed by cosine similarity formula.

$$sim(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{\left|\vec{d}_j\right|\left|\vec{q}\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,q}^2}}$$

Where, ||q|| and ||d|| are the norm of the query and document vector respectively.

After the cosine measure is computed, the documents are sorted according to the cosine value, the larger the cosine value, the more relevant the document to the user query using the concept of the LSI approaches.

4.3. Experimentation and evaluation

The classic recall, precision and f-measure are used to evaluate the retrieval performance of the two indexing approaches. An attempt has been made to compare the results of the Latent Semantic Indexing (LSI) method against the standard Vector Space Model. For the vector space model, the same term by document matrix, that was the starting point for the LSI method, is used. Using VSM, the prototype retrieved most of the relevant documents in the collection out of the total relevant documents in the corpus. However, the result of the recall is lowered, This is because documents containing one of query terms but not-relevant are retrieved. Documents are irrelevant because the query term found in those documents not express the meaning of the query with respect to other terms found in the query. On the other hand, some documents are relevant for the query but the document may not contain the query term on the document. Because of this the relevant documents are also not retrieved. Therefore, in order to enhance the performance of the IR system, LSI is applied . The results obtained for the VSM is attached as **Annex 6**.

The average result of precision, recall and F-measure of the prototype using LSI are 68%,79% and 72% respectively. This shows results registered on the improvement of IR prototype system. Performance increase in percentage of precision 9 %, recall 5 %, and F-measure 6% compared to the result obtained from VSM. The results obtained for the LSI model is attached as **Annex 7.**

User relevance judgment

The aim is to make sure how well system is performing on the eyes of users to assure that the system is acceptable and usable by them. To perform relevance judgment, we selected 16 people with different backgrounds. Due to time factor we limit the number of involved people to test the prototype. We discussed the details of the evaluation in table 4.1.

Table 4. 1 User relevance judgment

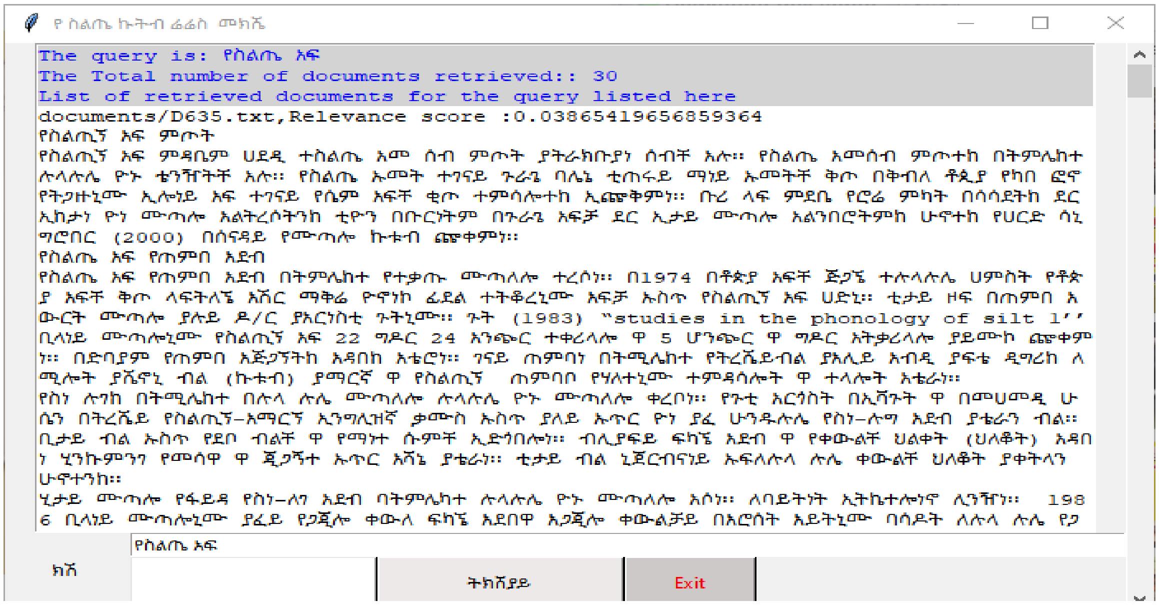
Number of peoples	Background of the peoples	Inspected elements	Response result
9	Have computing back Ground in the field of Computer Science and Information Technology	Terminologies, coloring input and output formats used in the prototype interface and retrieved result	59.7% of the involved peoples provide positive response on the inspected elements. 41.3% of the involved peoples accept the terminology, and input/ output formats and retrieved result, but they were not satisfied with the interface of the prototype.
7	Understand the language used in the prototype but not have computing background	Easiness to use the system and satisfaction with the retrieved result	57.5% satisfied by both easiness and retrieved result. 42.5% 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.

4.4. Discussion

The better result is registered by applying LSI for Silt'e information retrieval with average precision, recall and f-measure of 68%, 79% and 72% respectively. This increases by 9% in precision, 5% in recall, and 6 % in F-measure. The overall performance of the prototype system increased when it is compared with VSM. The reason is that the synonymous terms and polysemous are partially handled in LSI using SVD.

There were irrelevant documents retrieved from the corpus and some of the relevant documents from the corpus couldn‘t be retrieved. For instance, when we tried to search relevant document from the corpus using “ያበሮስ ቅጬ ሃለት” query. There were 23 relevant documents in the corpus, the prototype retrieved 20 documents out of that 18 were relevant documents. This is because the term “ቅጬ” has many meanings and is found in different documents.

When the size of the input query increases the corresponding expanded query terms and the relevant documents have a higher probability of being retrieved. This enables the information retrieval system to retrieve the most relevant documents according to its rank in descending orders. This saves the searching time for the users while they are seeking the information they want. The sample screenshot when the program runs for searching the query የስልጤ አፍ used in evaluation is depicted below in figure 4.1.



```
The query is: የስልጤ አፍ
The Total number of documents retrieved:: 30
List of retrieved documents for the query listed here
documents/D635.txt,Relevance score :0.03865419656859364
የስልጢኘ አፍ ምጦት
የስልጢኘ አፍ ምዳቤም ሀደዲ ተስልጤ አመ ሰብ ምጦት ያትራክቡያነ ሰብቾ አሉ። የስልጤ አመሰብ ምጦተከ በትምሌክተ
ሎላሉሌ ዮኑ ቴንኸትቾ አሉ። የስልጤ ኡመት ተገናይ ጉራጌ ባሌኔ ቲጠሩይ ማነይ ኡመትቾ ቅጦ በቅብለ ቶጺያ የካበ ፎኖ
የትጋዙኒሙ ኢሉነይ አፍ ተገናይ የሴም አፍቾ ቂጦ ተምሳሎተከ ኢጬቅምነ። ቡረ ላፍ ምደቤ የሮሬ ምካት በሳሳደትከ ደር
ኢከታነ ዮነ ሙጣሎ አልትረሶትንከ ቲዮን በቡርነትም በጉራጌ አፍቻ ደር ኢታይ ሙጣሎ አልነበሮትምከ ሁኖተከ የሀርድ ሳኒ
ግሮበር (2000) በሰናዳይ የሙጣሎ ኩቱብ ጬቀምነ።
የስልጤ አፍ የጠምበ አደብ
የስልጤ አፍ የጠምበ አደብ በትምሌከተ የተቃጡ ሙጣለሎ ተረሶነ። በ1974 በቶጵያ አፍቾ ጀጋኜ ተሉላሉሌ ሀምስት የቶጵ
ያ አፍቾ ቅጦ ላፍትለኜ አሽር ማቅሬ ዮኖከ ፊደል ተትቆረኒሙ አፍቻ ኡስጥ የስልጢኘ አፍ ሀድኒ። ቲታይ ዞፍ በጠምበ አ
ውርት ሙጣሎ ያሉይ ዶ/ር ያአርነስቲ ጉትኒሙ። ጉት (1983) “studies in the phonology of silt 1’’
ቢላነይ ሙጣሎኒሙ የስልጢኘ አፍ 22 ግዶር 24 አንጭር ተቀረላሎ ዋ 5 ሆንጭር ዋ ግዶር አትቃሪላሎ ያይሙከ ጬቀም
ነ። በድባያም የጠምበ አጅጋኘትከ አዳበከ አቴሮነ። ገናይ ጠምባነ በትሚሌከተ የትረሼይብል ያአሊይ አብዲ ያፍቴ ዲግሪከ ለ
ሚሎት ያሼኖኒ ብል (ኩቱብ) ያማርኛ ዋ የስልጢኘ  ጠምባበ የሃለተኒሙ ተምዳሳሎት ዋ ተላሎት አቴራነ።
የስነ ሉገከ በትሚሌከተ በሉላ ሉሌ ሙጣለሎ ሉላሉሌ ዮኑ ሙጣለሎ ቀረቦነ። የጉቲ አርጎስት በኢሻጉት ዋ በመሀመዲ ሁ
ሴን በትረሼይ የስልጢኘ-አማርኘ ኢንግሊዝኛ ቃሙስ ኡስጥ ያለይ ኡጥር ዮነ ያፈ ሁንዱሉሌ የስነ-ሉግ አደብ ያቴራን ብል።
ቢታይ ብል ኡስጥ የደበ ብልቾ ዋ የማነተ ሱምቾ ኢድጎበሎነ። ብሊያፍይ ፍካኜ አደብ ዋ የቀውልቾ ህልቀት (ህለቆት) አዳበ
ነ ሂንኩምንገ የመሰዋ ዋ ጃጋኜተ ኡጥር አሽኔ ያቴራነ። ቲታይ ብል ኒጀርብናነይ ኡፍለሉላ ሉሌ ቀውልቾ ህለቆት ያቀትላን
ሁኖተንከ።
ሂታይ ሙጣሎ የፋይዳ የስነ-ለገ አደብ ባትምሌከተ ሉላሉሌ ዮኑ ሙጣለሎ አሰነ። ለባይትነት ኢትኬተሎነኖ ሊንኸነ።  198
6 ቢላነይ ሙጣሎኒሙ ያፈይ የጋጀሎ ቀውለ ፍካኜ አደበዋ አጋጀሎ ቀውልቻይ በአሮስት አይትኒሙ ባሳዶት ለሉላ ሉሌ የጋ
```

የስልጤ አፍ

Figure 4. 1 A Screenshot of retrieved document for a given query

CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS

5.1. Conclusion

A lot of valuable information is being produced in government organizations and schools in Silte zone. This leads to the need for developing efficient and effective information storage and retrieval systems to represent, store, search and retrieve relevant Silt'e documents from large document collections as per users' information need.

The goal of this research is to examine the benefits of LSI text retrieval approach for Silt' text retrieval. An experiment on the retrieval performance of two text retrieval approaches has been made: vector space model and Latent Semantic Indexing, and the experimental results have been presented.

To this end, 700 text document corpus are indexed and 56 queries are used for evaluation. Various techniques of text preprocessing were employed; preprocessing techniques such as tokenization, normalization; stop word removal and stemming were used for identifying content bearing terms for indexing and searching.

The analysis begins with a weighted term by document matrix in which each number in the cell of the matrix represent the importance of the term within as well as in the entire document collection. This weighted matrix is used for indexing and retrieval by the standard vector space method.

The weighted matrix was analyzed further by the Singular Value Decomposition to derive the latent structure model which was used for indexing and retrieval by the LSI retrieval model. Queries were placed in the resulting space at the results of their constituent terms.

Cosine similarity measure between the query vector and document vectors were used to identify documents which are relevant to each query and the documents are ranked according to their cosine measures.

According to experimentation, the prototype is registered 79% precision, 68% recall and 72% F-measure by using LSI. This increases by 9% in precision, 5% in recall, and 6 % in F-measure when it is compared with VSM. The overall performance of the prototype system increased when it is compared with VSM. This is a promising result to design an applicable IR system for Silt'e text document. However, there are several challenges faced which limit to register the optimum performance expected from the model in order to outperform the entire IR prototype developed for Silt’e language. The main challenge in the study is the inability of the stemmer. The Silt’e stemmer used doesn’t remove infixes and also ambiguity of words in the language. And also there is no standard Silt’e corpus that can be used for experimentation.

5.2. Recommendations

Silt’e information retrieval system has a wide-open place for future study. The area is on the beginning level. It needs the collaboration of the researchers and funding organizations. The following recommendations are forwarded based on the finding which includes the developments of resources and future research directions for Silt’e text.

- Query expansion technique improves performance of retrieval systems by including synonymous terms in search. Therefore, it is necessary to see the effect of query expansion on the retrieval performance of the system.
- Nowadays Ontology and deep learning is the main area of study in IR. Therefore, further research can investigate ontology and deep learning to enhance the performance of Silt’e IR system.
- 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, future researchers need to emphasize the development of corpus for Silt’e to reach an applicable information retrieval.
- Stemming algorithm used in this study doesn‘t work well for every inflated word in the language. Additionally, there are conditions which never addressed by the stemming algorithm. So there should be further study to come up with a better stemming algorithm that works for Silt’e documents.
- Applying LSI for cross-language retrieval on Silt’e text documents is highly advisable. This work is a text retrieval system for under-resourced languages like Silt’e text documents. Other types of documents like video, audio, graphics, and

pictures are not studied. It is very important to make more study to come up integrated and fully functional system.

REFERENCES

Abebaw, T. (2015). Applying thesaurus based semantic compression for improving the performance of Amharic text retrieval. Addis Ababa University.

Abu-Salih, B. A. (2018). Ontology-based approach for identifying the credibility domain in social Big Data. Journal of Organizational Computing and Electronic Commerce. 3(1), 354-377.

Anbase, B. (2019). Applications of Information Retrieval for Afaan Oromo text based on Semantic-based Indexing. Jimma University.

Babu, A., & L., S. (2014). A Survey of Information Retrieval Models for Malayalam Language Processing. International Journal of Computer Applications, 107(14), 19–23.

Bachchhav, K. P. (2016). Information Retrieval : search process , techniques and strategies. International Journal of Next Generation Library and Technologies, 2(1), 1–10.

Baeza-Yates, R. A., & Ribeiro-Neto, B. (1999). Modern Information Retrieval. ACM press New York.

Bin, H. (2006). Automatic Term Extraction in Large Text Corporae. Faculty of Computer Science, Dalhousie University, Canada B3H 1W5.

Bruck, A., & Tilahun, T. (2015). Bi-gram based Query Expansion Technique for Amharic Information Retrieval System. International Journal of Information Engineering and Electronic Business, 7(6), 1–7.

Claveau, V., & Kijak, E. (2016). Distributional thesauri for information retrieval and vice versa. Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016, 3709–3716.

Deerwester, S., Furnas, G. W., Landauer, T. K., & Harshman, R. (2000). Indexing by Latent Semantic Analysis. Kehidupan, 3(12), 34.

Deshmukh, A., & Hegde, G. (2012). A Literature Survey on Latent Semantic Indexing. International Journal of Engineering Inventions, 1(4), 2278–7461.

Dramé, K., Mougin, F., & Diallo, G. (2014). Query expansion using external resources for improving Information Retrieval in the biomedical domain. CEUR Workshop Proceedings, 1180, 189–194.

Feldvari, k.(2002). thesauri Usage in Information Retrieval Systems: Example of LISTA and ERIC Database Thesaurus. Osijek university.

Fleur, M. La. (2015). Conceptual Indexing using Latent Semantic Indexing. Uppsala University.

George, L. E., & Hameed, I. A. (2013). Text-Based Information Retrieval System using Non-Linear Matching Criteria. 2(6), 172–176.

Gezehagn, G. (2012). Afaan Oromo Text Retrieval System . Addis Ababa University.

Hans, F. (2005). Terminology extraction and automatic indexing - comparison and qualitative evaluation of methods. Oxford University Press.

Hiemstra, D. (2000). Using Language Models for Information Retrieval. Djoerd Hiemstra, Enschede, The Netherlands.

Jemal, z. (2016). Silt ' e text information retrieval system based on vector space model . Adama Science and Technology University.

Johanna. (2006). Latent Semantic Indexing and Information Retrieval. Heidelberg University.

Johar, A. (2017). Text information retrieval system for silt'e language. Adama Science and Technology University.

Kang, B.-Y. (2003). A novel approach to semantic indexing based on concept. 7(9),44–49.

Kedir, M. (2012). Designing a Stemming Algorithm for Silt 'e. Addis Ababa University.

Keifer, G., & Effenberger, F. (2013). Retrieval Effectiveness of an Ontology-Based Model for Information Selection. Angewandte Chemie International Edition, 6(11), 951–952.

Lahtinen, T. (2000). Automatic indexing : an approach using an index term corpus and combining linguistic and statistical methods. Helsingfors university.

Manning, C. D. (2009). Introduction to Modern Information Retrieval, Cambridge UP, New York s, 41(4), 305–306.

Marco Suárez Barón, K. S. V. (2009). An approach to semantic indexing and information retrieval. Journal of the American Society for Information Science and Technology 9(4), 78–98.

Névéol, A., & Aronson, A. R. (2007). Automatic indexing of specialized documents: Using generic vs. Biological, Translational, and Clinical Language Processing, 183–190.

Nie, J. (2011). A General Logical Approach to Inferential Information Retrieval. Artificial Intelligence Review. 10, 1–44.

Phadnis, N., & Gadge, J. (2014). Framework for document retrieval using latent semantic indexing. International Journal of Computers and Applications, 94(14), 37–41.

Rosario, B. (2000). Latent Semantic Indexing : An overview. Infosys 240, 1–16.

Sridevi., U. ., & Nagaveni., N. (2010). Ontology based Similarity Measure in Document Ranking. International Journal of Computer Applications, 1(26), 135–139.

Staab, S., & Hotho, A. (2003). Ontology-based Text Document Clustering. Intelligent Information Processing and Web Mining, Proceedings of the International. 451–452.

Tewodros, H. G. (2003). Amharic Text Retrieval : An Experiment Using Latent Semantic Indexing (LSI) With Singular Value Decomposition (SVD). Addis Ababa University

Wei, C. P., Yang, C. C., & Lin, C. M. (2008). A Latent Semantic Indexing-based approach to multilingual document clustering. Decision Support Systems, 45(3), 606–620.

Wordofa, M. (2013). Semantic indexing and document clustering for Amharic information retrieval. Addis Ababa university.

Xu, J., & Croft, W. B. (2006). Query Expansion Using Local Analysis and Global Document Jinxi Xu and W . Bruce Croft Center for Intelligent Information Retrieval. ACM, 4–11.

ANNEXS:

Annex 1: 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	ዠ žo
የ ya	ዩ yu	ዪ yii	ያ yaa	ዬ ye	ዩ 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
ጠ ťa	ጡ TU	ጢ ťii	ጣ ťaa	ጤ ťe	ጥ ťi	ጦ ťo
ጨ ča	ጩ ču	ጪ či	ጫ čaa	ጬ če	ጭ či	ጮ čo
ጸp̌a	ጹ p̌u	ጺ p̌i	ጻ p̌aa	ጼ p̌e	ጽ p̌i	ጾ p̌o
ፈ fa	ፉ	ፊ fii	ፋfaa	ፌ fe	ፍ fi	ፎ fo

Annex 2: Lists of Silt'e Stop words

ሀነይ	አበይ	ሬራቀደን	አቢሌ	ሉሉሌ	እነይ
ሀነግነ	አቢ	ቀለ	አብሌ	ሉላሉሌ	እነይ
ሀዻድኑም	ኤት	ቀደ	አብተቴ	ሉላሉሌ	ኦናኔ
ሁልምከ	ኤጋሀ	ቀደን	አብታይ	ሉሌ	ወክት
ሁኖ	ኤጋሙ	ቂጦ	አተም	ላይሽ	ወይ
ሁኖተኒመዋ	ኤጋሽ	ቅጨ	አተቴ	ማ	ዋ
ሂነሚ	እሊ	በሁነትንሙ	አቲ	ምን	ዋ
ሂንኩምንገ	እቢ	በሁኖት	አታይ	ሱር	የገግ
ሂንኵምንገ	እብተቴ	በለ	አታይ	ሱር	ዩተቴ
ሆነምታሌ	እተቴ	በሉሌ	አነይ	ሱርዋ	ዩታይ
ለሄ	እቲ	በልዻሌ	አናግና	ስር	ያሽ
ለሄው	እታይ	በልዻሌ	አዩ	ሬራ	ያተቴ
ለሰባድሽ	እነኮ	በሰቼ	አይታይ	ታሌ	ያታይ
ለሰባድኑም	እነይ	በውኖትምከ	አይነኮ	ታቼን	ዬዬ
ለሰገጋሙ	ገናገናይ	በዬ	አይነኮ	ትዮኑ	ዮለ
ለሰገግሽ	ገገ	ቢትላይም	አይኔ	ናሩ	ዮልስ
ለሰገግኑም	ገገኑ	ብቸ	አድ	ናር	ዮናነይ
ለሰገግከ	ገጌ	ብቾ	አድአድ	ናርት	ደር
ለሳድባድከ	ጉት	ተሬር	ኡሀ	አለቢ	ደር
ለደር	ግን	ተዮን	ኡስጥ	አሊ	ድባየ
ለገነ	ግዝ	ተደር	ኡንኮ	አልቀሬ	ገነ
ለገነገናም	ግዝቸ	ቱፍትሉፍት	ኢንኩምንገ	አሎነ	ገና
ሉሀ	ፈሬ	ቲታሚ	ኢንኵምንገ	ፎኖ	ገናሚ

Annex 3: Lists of Silt’e Suffixes

ኩሞሙ	ንቾ	ሸ
ቢያኔ	እሎ	ቸ
አታም	ዬን	ኮ
አተኘ	ኔት	ኩ
አሙ	ታም	ሁ
ሼታ	ተኛ	ነ
ኩሙ	ኢ	ን
ኒሙ	ይ	ሺ
ሼሸ	ኤ	ኤ
ታት	ሸ	አ
ኤን	ኔ	ተ
አኘ	ከ	ት
ተኘ	ሎ	ኡ

Annex 4: Lists of Silt’e Prefixes

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

Annex 5: Selected queries for prototype system evaluation

ለባድ ወልድ	ሱረ ሀድ	ዮባጀ ሱልቸ	የቀውል ማእነ
የልባስ ጡሃርነት ቂሮት	ክሼ ለሚሎት	ተሳማት ገግ ተገግ	ጎትተኔ ኤትቸ
የሰብ ወልድ ተረዘቆት	አመሰበኜ ተሳማት	አድጋቦት እንመንዳ	የመሀር ወክት
በደም ኢትላለፋነን ነቶ	ሰበኜ አህላቅ ላኪሞት	ተጋቦት ያአበሩስ	ቅሬታቶ
ስልጤዋ ያየር ንባረትከ	ያፍ አህላቅቸ	ተላፈቶት ባሸ	ቀሊሎ ቀውልቸዋ
የስልጤ አፍ	የስነ-ጠምበ	እንክት ባለ	ነቶ መንቃቆ
ያበሮስ ቅጬ ሃለት	ቲፌት ሄደ	ሸም ባለ በሮት	ያይዶይ ወራትቸ
የኢደ ርዝቅቸ	ቄነ በደ	ያጥናቦት ቁወ	ሰጬ ቃዋ
የባድ መሊቅ	ቃዋ ተቃወ	ዮባጀ ሱልቸ	በጌ ንበሮትዋ በከተማ ንበሮት
ያፍ አበሮስ	ኡንዱሩቄ አሼ	ያትኬሶን ዩግዣር ግዝቸ	ወከተ ቂሮትዋ ተድገለሎት
የቶጵያ አፍቾ ሩክቦኒሙ	መህማኞት	ዩርባዋ የዛንጂሮ ኤሶት	ያፍ አሸር ውጥን

Annex 6 : Experiment result for VSM

Query	Corpus	Retrieved	Relevant	Recall	Precision	F-measure
ለባድ ወልድ	22	18	17	0.77	0.94	0.85
የልባስ ጡሃርነት ቂሮት	15	16	11	0.73	0.68	0.7
የሰብ ወልድ ተረዘቆት	5	7	4	0.8	0.57	0.66
በደም ኢትላለፋነን ነቶ	13	11	9	0.69	0.81	0.75
ስልጤዋ ያየር ንባረትከ	15	17	13	0.86	0.76	0.81
የስልጤ አፍ	31	29	26	0.83	0.89	0.86
ያበሮስ ቅጬ ሃለት	23	20	18	0.78	0.9	0.83
የአደ ርዝቅቸ	6	4	4	0.66	1	0.8
የባድ መሊቅ	14	16	11	0.78	0.68	0.73
ያፍ አበሮስ	31	36	28	0.9	0.77	0.83
የቶጵያ አፍች ሩክቦኒሙ	11	10	8	0.72	0.8	0.76
በሸርቅ አዘርነት በርበሬ	8	7	6	0.75	0.85	0.8
የስልጤ አድ	42	37	35	0.83	0.9	0.88
ጎትተኔ ኤትቸ	7	4	4	0.57	1	0.72
ቅሬታቶ	17	16	14	0.8	0.875	0.84
ነቶ መንቃቆ	6	5	4	0.66	0.8	0.72
ያፍ አሸር ውጥን	3	4	3	1	0.75	0.85
የሱረ ሼሽት	4	3	2	0.5	0.66	0.57
የስልጥኘ ቀውልቸ ተቻበራት	17	16	14	0.82	0.87	0.84
የቀውል ማእነ	30	33	28	0.93	0.84	0.88
ሱረ ሀድ	8	6	5	0.62	0.83	0.71
ክሼ ለሚሎት	11	9	8	0.72	0.88	0.8
አመሰበኜ ተሳማት	11	10	8	0.72	0.8	0.76
ሰበኜ አህላቅ ላኪሞት	38	36	31	0.81	0.86	0.83
ያፍ አህላቅቸ	4	3	2	0.5	0.66	0.57
የስነ-ጠምበ	4	3	2	0.5	0.66	0.57
ቲፌት ሄደ	6	4	3	0.5	0.75	0.6
ቄነ በደ	7	5	3	0.42	0.6	0.5
ቃዋ ተቃወ	6	5	4	0.66	0.8	0.72
ኡንዱሩቂ አሼ	16	13	12	0.75	0.92	0.82
መህማኞት	7	6	5	0.71	0.83	0.71
በማእነ እላያነይ	4	3	2	0.5	0.66	0.57
ቦሽቲ ቆርቸ ጉት	9	7	6	0.66	0.85	0.75
ቀሊሎ ቀውልቸዋ	34	38	29	0.85	0.76	0.8
ያይዶይ ወራትቸ	8	4	3	0.37	0.75	0.5
ሰጬ ቃዋ	6	5	4	0.66	0.8	0.72
ለባይትከ ተቃሮት	8	6	5	0.62	0.83	0.71

ጌ ንበሮትዋ በከተማ ንበሮት	17	16	13	0.76	0.81	0.78
ወከተ ቂሮትዋ ተድገለሎት	8	7	6	0.75	0.85	0.8
ዩርባዋ የዛንጇሮ ኤሶት	5	4	3	0.6	0.75	0.66
ዮባጇ ሱልቸ	7	6	4	0.57	0.66	0.61
ዮባጇ ሱልቸ	12	10	7	0.58	0.7	0.63
ተሳማት ገግ ተገግ	6	5	4	0.66	0.8	0.72
አድጋቦት እንመንዳ	8	7	5	0.62	0.71	0.66
ተጋቦት ያአበሩስ	10	9	7	0.7	0.77	0.73
ተላፈቶት ባሽ	12	11	9	0.75	0.81	0.78
እንከት ባለ	14	13	11	0.78	0.84	0.81
ሽም ባለ በሮት	13	10	8	0.61	0.8	0.69
ያጥናቦት ቁወ	17	14	11	0.64	0.78	0.7
ዮባጇ ሱልቸ	4	3	2	0.5	0.66	0.57
ያትኬሶን ዩግዣር ግዝቸ	12	10	6	0.5	0.6	0.54
ዩርባዋ የዛንጇሮ ኤሶት	7	6	5	0.71	0.83	0.76
የነጠረ መዬ አፍሎኔዋ	5	3	2	0.4	0.66	0.5
የመሀር ወከት	11	6	5	0.45	0.83	0.58
የአደ ርዝቅቸ	6	5	4	0.66	0.8	0.72
Total				68%	79%	72%

Annex 7: Experiment result for LSI

Query	Corpus	Retrieved	Relevant	Recall	Precision	F-measure
ለባድ ወልድ	22	18	17	0.77	0.94	0.85
የልባስ ጡሃርነት ቂሮት	15	16	11	0.73	0.68	0.7
የሰብ ወልድ ተረዘቆት	5	7	4	0.8	0.57	0.66
በደም ኢትላለፋነን ነቶ	13	11	9	0.69	0.81	0.75
ስልጤዋ ያየር ንባረትከ	15	17	13	0.86	0.76	0.81
የስልጤ አፍ	31	29	26	0.83	0.89	0.86
ያበሮስ ቅጩ ሃለት	23	20	18	0.78	0.9	0.83
የአደ ርዝቅቻ	6	4	4	0.66	1	0.8
የባድ መሊቅ	14	16	11	0.78	0.68	0.73
ያፍ አበሮስ	31	36	28	0.9	0.77	0.83
የቶጵያ አፍች ሩክቦኒሙ	11	10	8	0.72	0.8	0.76
በሸርቅ አዘርነት በርበሬ	8	7	6	0.75	0.85	0.8
የስልጤ አድ	42	37	35	0.83	0.9	0.88
ጎትተኔ ኤትቻ	7	4	4	0.57	1	0.72
ቅሬታቶ	17	16	14	0.8	0.875	0.84
ነቶ መንቃቆ	6	5	4	0.66	0.8	0.72
ያፍ አሽር ውጥን	3	4	3	1	0.75	0.85
የሱረ ሼሽት	4	3	2	0.5	0.66	0.57
የስልጥኘ ቀውልቻ ተቻበራት	17	16	14	0.82	0.87	0.84
የቀውል ማእነ	30	33	28	0.93	0.84	0.88
ሱረ ሀድ	8	6	5	0.62	0.83	0.71
ክሼ ለሚሎት	11	9	8	0.72	0.88	0.8
አመሰበኜ ተሳማት	11	10	8	0.72	0.8	0.76
ሰበኜ አህላቅ ላኪሞት	38	36	31	0.81	0.86	0.83
ያፍ አህላቅቻ	4	3	2	0.5	0.66	0.57
የስነ-ጠምበ	4	3	2	0.5	0.66	0.57
ቲፌት ሄደ	6	4	3	0.5	0.75	0.6
ቄነ በደ	7	5	3	0.42	0.6	0.5
ቃዋ ተቃወ	6	5	4	0.66	0.8	0.72
ኡንዱሩቄ አሼ	16	13	12	0.75	0.92	0.82
መህማኞት	7	6	5	0.71	0.83	0.71
በማእነ እላያነይ	4	3	2	0.5	0.66	0.57
ቦሽቲ ቆርቻ ጉት	9	7	6	0.66	0.85	0.75
ቀሊሎ ቀውልቻዋ	34	38	29	0.85	0.76	0.8
ያይዶይ ወራትቻ	8	4	3	0.37	0.75	0.5
ሰጩ ቃዋ	6	5	4	0.66	0.8	0.72
ለባይትከ ተቃሮት	8	6	5	0.62	0.83	0.71

ጌ ንበሮትዋ በከተማ ንበሮት	17	16	13	0.76	0.81	0.78
ወከተ ቂሮትዋ ተድገለሎት	8	7	6	0.75	0.85	0.8
ዩርባዋ የዛንጄሮ ኤሶት	5	4	3	0.6	0.75	0.66
ዮባጀ ሱልቸ	7	6	4	0.57	0.66	0.61
ዮባጀ ሱልቸ	12	10	7	0.58	0.7	0.63
ተሳማት ገግ ተገግ	6	5	4	0.66	0.8	0.72
አድጋቦት እንመንዳ	8	7	5	0.62	0.71	0.66
ተጋቦት ያአበሩስ	10	9	7	0.7	0.77	0.73
ተላፈቶት ባሸ	12	11	9	0.75	0.81	0.78
እንክት ባለ	14	13	11	0.78	0.84	0.81
ሸም ባለ በሮት	13	10	8	0.61	0.8	0.69
ያጥናቦት ቁወ	17	14	11	0.64	0.78	0.7
ዮባጀ ሱልቸ	4	3	2	0.5	0.66	0.57
ያትኬሶን ዩግዣር ግዝቸ	12	10	6	0.5	0.6	0.54
ዩርባዋ የዛንጄሮ ኤሶት	7	6	5	0.71	0.83	0.76
የነጠረ መዬ አፍሎኔዋ	5	3	2	0.4	0.66	0.5
የመሀር ወክት	11	6	5	0.45	0.83	0.58
የአደ ርዝቅቸ	6	5	4	0.66	0.8	0.72
			Total	68%	79%	72%

Annex 8: Test Query with Relevant Document List

Queries	Relevant document
ለባድ ወልድ	15, 16, 19, 23, 22, 24, 45, 62, 67 , 128, 134, 148,161,162 165,278,392, 669, 673, 686, 689,700
የልባስ ጡሃርነት ቂሮት	32, 34 , 35 , 43 , 52 , 52 , 54 , 57 , 57 , 308 ,309, 385 , 396 , 434 , 437, 438 ,439, 440
የሰብ ወልድ ተረዘቆት	39, 42 , 42 , 42 , 377 , 387
በደም ኢትላለፉነን ነቶ	31, 32 , 33 , 44 , 44 , 51 , 52 , 52 , 55 , 386, 397 , 437
ስልጤዋ ያየር ንባረትከ	11, 13, 20 , 28 , 29 , 84 , 151, 250 , 272 , 285 , 306 , 394 , 536 , 692 ,693
የስልጤ አፍ	19 , 72 , 84 , 111 , 112 , 113 , 114 , 115 , 117 , 146 , 148 , 163 , 163 , 164 , 164 , 176 , 178 , 200 , 201, 305 , 332 ,333,335, 630 , 630 , 631 , 635 , 670 , 689
ያበሮስ ቅጬ ሃለት	18 , 20 , 21 , 25 , 22 , 54 , 57 , 57 , 84 , 106 , 132 , 149 , 150 , 302 , 304 , 309 , 309 , 434 , 438 , 621 , 672 , 691 , 694
የአደ ርዝቅቸ	45 , 46 ,47 53 , 389,390,
የባድ መሊቅ	30 , 31,36, , 51 , 84 , 100 , 142 , 148 , 149 , 395 , 615 , 683 , 689 , 691
ያፍ አበሮስ	24 , 25 ,26 , 27 , 28 , 29 , 116, 119 , 120 , 121 , 123 , 124 , 125 , 126 , 201 , 201 , 201 , 201 , 218, 224 , 253 , 393 , 394 , 406 , 422 , 425 , 426 , 427 , 428 , 429
የቶጵያ አፍች ሩክቦኒሙ	90 ,93, 94 , 97 , 101, 102, 110 , 157 , 614 , 627 , 627
በሸርቅ አዘርነት በርበሬ	89 , 156 , 208 , 298 , 298 , 655 698
የስልጤ አድ	2 , 12, 14 , 40 , 42 , 45 , 46 , 47 , 49 , 53 , 55 , 81, 82, 83, 84 , 89 , 90 , 111 , 112 , 113 , 114 , 115 , 117 , 156 , 214 , 239 , 277 , 279 , 280 , 283 , 332 , 383 , 389 , 437 , 630 , 630 , 631 , 635 , 666 , 670 , 698
ጎትተኔ ኤትቸ	8 ,10,16,17, 689, 690 , 691
ቅሬታቶ	81 , 84 , 87 , 88 , 89 , 142 , 143 , 144 , 145 , 156 , 162 , 683 , 684 , 685 , 698
ነቶ መንቃቆ	602, 603 , 604 , 608 , 689
ያፍ አሸር ውጥን	84 , 148 , 166
የሱረ ሼሽት	130 , 172 173 , 147
የስልጥኘ ቀውልቸ ተቻበራት	175 , 178 , 199 ,381,115 , 117 , 156 , 214 ,422 , 425 , 426 , 427 , 428,181 , 182 , 185
የቀውል ማእነ	84 , 130, 130 , 147 , 155 , 172 , 173 , 177 , 178 , 190 , 197 , 198 , 300 , 311 , 331 , 381 , 671 , 690 , 697

ሱረ ሀድ	76 , 84 , 148 , 152 , 155 , 165, 168 , 169 , 172, 173 , 176 , 177 , 178 , 181 , 182 , 185 , 190 , 197 , 198 , 200 , 204 , 208 , 215, 341 , 481 , 541 , 551 , 689 , 697
ክሼ ለሚሎት	84 , 150 , 151 , 178 , 179,183,184, 692 , 694
አመሰበኙ ተሳማት	205 , 567 , 568 , 574 , 578 , 579 , 582 , 586 , 587
ያፍ አህላቅቻ	190, 205 , 567 , 568 , 574 , 578 , 579 , 582 , 586 , 587
ሰበኙ አህላቅ ላኪሞት	84 , 172, 178 , 190 , 197 , 203 , 205 , 218, 224 , 253 , 271 , 406 , 422 , 425 , 426 , 427 , 428 , 429 , 432 , 446 , 544 , 556 , 558 , 560 , 567 , 568 , 574 , 578 , 579 , 582 , 586 , 587 , 610 , 661
የስነ-ጠምበ	84, 178 ,180,183
ቲፌት ሄደ	84,178 , 271, 273,275
ቄነ በደ	26,27,29,30,65,67,68
ቃዋ ተቃወ	24,41 44, 384 , 392 ,53
ኡንዱሩቄ አሼ	84,85,86,90,53,76,57,89,67,69,72,86,78,72
መህማቾት	59 , 61,62,63,64,65,66,67,670,671,673,675,680,681
በማእነ እላያነይ	610, 612 , 612 , 613,616 , 617
ቦሽቲ ቆርቻ ጉት	185,186,187,188,189,552,554
ቀሊሎ ቀውልቻዋ	84, 148 , 149 , 162 , 178 , 185 ,600 , 689 , 691
ያይዶይ ወራትቻ	332 , 383 , 389 , 437 , 630 , 635 ,600
ቃዋ ሰጨ	41,42,484
ለባይትከ ተቃሮት	191,192,193,201,203,204
በጌ ንበሮትዋ በከተማ ንበሮት	645,646,647,648,650
ወከተ ቂሮትዋ ተድገለሎት	205 , 218, 224 , 253,591,592,593
ዩርባዋ የዛንጄሮ ኤሶት	571,574,578,591
ዮባጆ ሱልቻ	84, 85, 149, 180, 185, 696
ተሳማት ገግ ተገግ	117, 118, 119, 121, 123, 126, 127
አድጋቦት እንመንዳ	167, 168, 169,344, 346,347,348,349
ተጋቦት ያአበሩስ	66,7,68,69,71,73,75
ተላፈቶት ባሽ	2,3,5,6,8,9 227,228,222
እንከት ባለ	76,78,79,84,88,306,307, 692,693,695
ሽም ባለ በሮት	377,355,367,358,359,600
ያጥናቦት ቁወ	312,314,324,326,317,319
ያትኬሶን ዩግዣር ግዝቻ	431,433,435,439,477,479,480
ዩርባዋ የዛንጄሮ ኤሶት	574,571,578,591
የነጠረ መዬ አፍሎኔዋ	20 , 84, 151 , 306 , 692
የመሀር ወከት	20,22,24,84,151,306,307, 692,693,695