JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

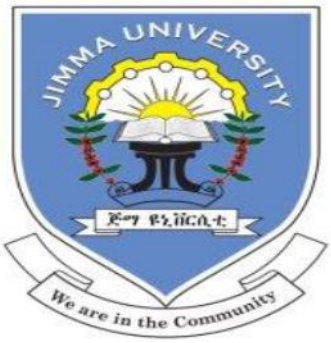
APPLYING THESAURUS BASED SEMANTIC COMPRESSION FOR AFAAN OROMO TEXT RETRIEVAL

By

TESFAYE TADELE

JIMMA, ETHIOPIA

Mar. 11, 19



JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

APPLYING THESAURUS BASED SEMANTIC COMPRESSION FOR AFAAN OROMO TEXT RETRIEVAL

By

TESFAYE TADELE

PRINCIPAL ADVISOR: DEBELA TESFAYE (Ph.D.)

CO-ADVISOR: SHIMELIS S. (MSC)

A THESIS SUBMITTED TO SCHOOL OF COMPUTING OF JIMMA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

APPLYING THESAURUS BASED SEMANTIC COMPRESSION FOR AFAAN OROMO TEXT RETRIEVAL

By

TESFAYE TADELE

Name and signature of members of the examining board

Title	**Name**	**Signature**	**Date**
1. Advisor	Dr. Debela Tesfaye		March 11, 2019
2. External Examiner	_____	_____	March 11, 2019
3. Internal Examiner	_____	_____	March 11, 2019
4. Chairperson	_____	_____	March 11, 2019

DECLARATION

This thesis is my original work and it is not been previously submitted by other candidate as a partial requirement for a degree in any University.

Signature: _____

Tesfaye Tadele Sorsa

March 11, 2019

ACKNOWLEDGEMENT

First, I give praise to the almighty God in all aspects of my life. Next, I would like to say thanks for those people who had helped me directly or indirectly during my thesis work.

Secondly, I would like to express my gratitude to my Advisor Dr Debela Tesfaye for his guidance, support and his continuous enthusiasm and encouragement throughout my thesis work. I am also very grateful to extend my sincere thanks to my co- advisor Shimalis S. (Msc) and other staff members of school of computing of Jimma University for their cooperation. I also extend many thanks to Dr. Elfneh Uddessa from London institute of linguistic and literature who have helped me and give me correction, support and suggestion on thesaurus approval as well as English grammar, syntax, and paragraphs coherence during my thesis.

I want to express my grateful thanks for my beloved family, for their strong cooperation by standing with me throughout every step of my academic work. Lastly but not the least, I want to say thanks to my friends who have helped me by suggestions, technical support and resource sharing during my thesis work.

ABSTRACT

Nowadays, the amount of information resource available on the Internet and World Wide Web (WWW) is increasing rapidly from day to day. Due to this reason, retrieving the appropriate information from these resources is becoming more and more complex task for information seekers. This situation makes it difficult for the users to identify the relevant information that match their information needs. Like in other languages, there are huge amount of electronic information resources that generate every day in Ethiopian languages. Specially, in Afaan Oromo. The growing amount of information resources are challenging for archival and searching specific information from them. Therefore, developing an effective information retrieval system for Afaan Oromo that allows searching and retrieving relevant documents as per user information needs is important. In an attempt to address this need, researchers have developed some information retrieval systems (IR). However, there is evidence that these IR systems have yet attained any promising system performance. This underperformance can be due to various reasons; the main cause of the IR system underperformance is due to semantic term mismatch or semantic nature of the natural language. Besides this, there has been no attempt made for Afaan Oromo IR (AOIR) system using semantic compression approach. Therefore, to solve these issues, integrating of thesaurus based semantic compression technique with generic IR system will improve performance of AOIR system.

In this study, we have developed AOIR using thesaurus based semantic compression approach. The developed prototype has both indexing and searching parts. For experiment, 106 Afaan Oromo text documents collected from news articles, the Afaan Oromo Bible, websites and books by the researchers. Then, different text operations such as; tokenization, normalization, stop word removal and stemming were used to identify content bearing terms. Following this, tf-idf term weighting scheme is used to compute the weight of each content bearing term. Finally, semantic compression is applied in which semantic representation of each term is identified and thesaurus based inverted indexing structure is used in document indexing. The experiment is conducted using two ways, the first experiment is conducted without considering the synonymous search term and the second experiment is conducted by using thesaurus based indexing. From conducted experiment, the average result of 58.451% precision, 78.438% recall, and 61.520% F-measure are registered without thesaurus based semantic compression approach and thesaurus based semantic compression technique showed 71.014%, 89.287% and 77.715% average precision, recall and F-measure respectively. However, the system performance is still affected by the problem of polysemy. Therefore, additional work has recommended in order improving the system performance so that it can advance the AOIR further by using different techniques.

Keywords: **information retrieval, indexing, semantic compression, thesaurus**

TABLE OF CONTENTS

DECLARATION i
ACKNOWLEDGEMENT ii
ABSTRACT iii
LIST OF TABLES vii
LIST OF FIGURES viii
LIST OF ACRONYMS ix
LIST OF APPENDICES x
CHAPTER ONE 1
1.1. INTRODUCTION 1
1.2. Background 1
1.3. Statement of the Problem 5
1.4. Objective of the study 7
1.4.1. General Objective 7
1.4.2. Specific Objectives 7
1.5. Methodology of the study 8
1.5.1. Literature Review 8
1.5.2. Data collection and Corpus Preparation 9
1.5.3. Research Design 9
1.5.4. Development Tools 9
1.5.5. Testing Procedures 9
1.6. Scope and Limitations 10
1.7. Application of the results 11
1.8. Organization of the study 12
CHAPTER TWO 13
LITERATURE REVIEWS 13
2.1. Introduction 13
2.2. Overview of IR 13
2.2.1. Information Retrieval Process 15
2.2.2. IR Models 19
2.2.3. IR System Performance Evaluation 25
2.3. Approaches to semantic compression 27
2.3.1. Ontology-Based Approach 28

2.3.2. Frequency-Based Approach 30
2.3.2.1. Term Frequency 30
2.3.2.2. Term frequency-inverse document frequency 30
2.3.2.3. N-Gram Based Approach 31
2.3.2.4. Thesaurus-Based Approach 33
2.4. Related Works 37
2.4.1. Semantic Compression Based IR for Foreign Language 37
2.4.2. Semantic compression based IR for local languages 39
CHAPTER THREE 44
MORPHOLOGY OF AFAAN OROMO 44
3.1. Introduction 44
3.2. Overview of Afaan Oromo 44
3.3. Afaan Oromo Alphabets 44
3.4. Afaan Oromo Morphology 45
3.5. Afaan Oromo Sentence Structure 47
3.6. Afaan Oromo Grammar 48
3.6.1. Gender 48
3.6.2. Personal Pronoun (bamaqaa/ gulummoo) 48
3.6.3. Adjective (Ibsa Maqaa) 49
3.6.4. Adverbs (Ibsa xumuraa) 50
3.6.5. Prepositions 51
3.6.6. Postpositions 51
3.6.7. Conjunctions 52
3.6.8. Punctuations 53
3.6.9. Definiteness 53
CHAPTER FOUR 55
METHODS AND TECHNIQUES 55
4.1. Introduction 55
4.2. Proposed System Architecture 55
4.3. Dataset preparation and Document pre-processing 56
4.4. Document Pre-processing 57
4.4.1. Tokenization 57
4.4.2. Normalization 58

4.4.3. Elimination of Stop Word 59
4.4.4. Words Stemming 60
4.5. Term Weighting 62
4.6. Semantic Compression Algorithm 64
4.7. Inverted Index 67
4.8. Evaluation Techniques 67
CHAPTER FIVE 68
IMPLEMENTATION AND EXPERIMENTAL RESULTS 68
5.1. Introduction 68
5.2. Thesaurus Construction 68
5.3. Index Construction 69
5.4. Thesaurus based semantic compression 70
5.5. Description of prototype system 70
5.6. Test query selection 74
5.7. Experiment and Performance Evaluation 75
5.7.1. Experiment Setting 75
5.7.2. Data Collections 75
5.7.3. Manual documents Classification 75
5.7.4. Evaluation Parameters 76
5.7.5. Experimental Results 78
5.7.6. Discussion of the Results 80
CHAPTER SIX 83
CONCLUSION AND RECOMMENDATIONS 83
6.1. Conclusion 83
6.2. Recommendations 84
REFERENCE 86
APPENDIXES 90

LIST OF TABLES

TABLE 2.2-1: THE VECTOR SPACE MODEL TERM BY DOCUMENT MATRIX 22

TABLE 3.3-1: LIST OF QUBEE AFAAN OROMO 45

TABLE 3.6-1: SAMPLE LIST OF AFAAN OROMO PREPOSITIONS 51

TABLE 3.6-2: SAMPLE LIST OF AFAAN OROMO POSTPOSITIONS 52

TABLE 3.6-3: SAMPLE LIST OF AFAAN OROMO CONJUNCTIONS 53

TABLE 4.4-1: SAMPLE LIST OF AFAAN OROMO STOP WORDS 60

TABLE 5.2-1: SAMPLE LIST OF SYNONYMOUS TERMS IN AFAAN OROMO 69

TABLE 5.7-1: MANUAL CLASSIFICATION OF DOCUMENT RELEVANCE WITH RESPECT TO 10 QUERIES 76

TABLE 5.7-2: THE PERFORMANCE OF THE PROTOTYPE SYSTEM WITHOUT CONSIDERING SYNONYMOUS WORDS 78

TABLE 5.7-3 THE PERFORMANCE OF THE PROTOTYPE SYSTEM USING SEMANTIC COMPRESSION BASED ON THESAURUS BASED INDEXING TECHNIQUE 80

LIST OF FIGURES

FIGURE 2.2-1: THE GENERAL ARCHITECTURE FOR INFORMATION RETRIEVAL 14

FIGURE 4.2-1: THE ARCHITECTURE OF SEMANTIC COMPRESSION MODEL FOR AFAAN OROMO TEXT RETRIEVAL 56

FIGURE 4.4-1: ALGORITHM FOR TOKENIZATION 58

FIGURE 4.4-2: ALGORITHMS FOR REMOVING STOP WORDS 60

FIGURE 4.4-3: ALGORITHM FOR STEMMING 62

FIGURE 4.6-1: ALGORITHM FOR SEMANTIC COMPRESSION 66

FIGURE 5.5-1: SHOWS THE SCREEN SHOOT OF THE PROTOTYPE SYSTEM 71

FIGURE 5.5-2: PYTHON CODE FOR TOKENIZATION 72

FIGURE 5.5-4: PYTHON CODE FOR REMOVING STOP WORDS 73

LIST OF ACRONYMS

ASNET- Amharic Semantic Network

AOIR – Afaan Oromo information retrieval

BT – Broader Terms

DF – Document Frequency

GB – Giga Byte

IDF – Inverse Document Frequency

IR – Information Retrieval

IRS – Information Retrieval System

LSI – Latent Semantic Indexing

NLP – Natural Language Processing

NT – Narrower Term

OBN – Oromia Broadcast Networks

OMN – Oromia Media Networks

POS – Part of Speech

POST – Part of Speech Tagging

RC – Reference Concept

RT – Related Term

SOV – Subject –Object-Verb

SVO – Subject –Verb -Object

SVD – Single Value Decomposition

UF – USE_FOR

VSM – Vector Space Model

WWW – World Wide Web

LIST OF APPENDICES

Appendix A: Relevance Judgement Table ---------- 88

Appendix B: List of Afaan Oromo stop words ----------91

CHAPTER ONE

1.1. INTRODUCTION

This chapter gives general information about this study. It gives the general background of the study area, the statement of the problem that motivated the research, the objective of the study, the methodologies employed to come up with the solution(s) of the problem, test procedure, scope and limitation of the study, significance and organization of the study is presented.

1.2. Background

Now a day, the amount of information resource available on the internet and World Wide Web (WWW) is increasing rapidly from day to days. Due to the availability of this huge amount of information resource on the internet and World Wide Web, retrieving the proper information from it is becoming more and more complex task for information seekers [1], [2]. This situation made difficult for the users to find the relevant information that matches their information needs. The interest of human being to find useful information resource from the collection of documents stored in the database is increased with computer technology that has the capability to store a large volume of information. Thus, in 1950, information retrieval (IR) has emerged to manage such a huge amount of electronic documents that is increasing rapidly from time to time. After a few decades, Information retrieval became the leading information access system that overtakes the traditional database searching style [3]. Thus, many business enterprises, educational institutions (i.e. university, colleges and etc..), cooperate and public libraries are using information retrieval systems to provide access to their resources such as books, journals, newspapers, and other electronic documents. These information resources can exist in various formats such as text, audio, video, picture, and multimedia.

Currently, information retrieval is a broad hot research area in computer science and Information Technology fields. Mainly it is focusing on how to fulfill the users need in finding information of their interest. It deals with the representation, organization, storage and access to the information items that exist as textual documents, and unstructured information and multimedia. Information retrieval also includes modeling, web search, and text classification, summarization, filtering, and presenting information items [1]. It is also defined as the act of finding an unstructured document that satisfies information requested from large collections of documents [4]. In other words, it is

the science of searching for a document contains the information required by the users as per information of the query term. As we understand from the definitions given above, the main goal of an information retrieval system is to retrieve the relevant documents as per user information needs.

In fact, information retrieval systems do not actually retrieve the information itself rather it retrieves the documents from which the information can be acquired and understood. So, searching for relevant documents from a huge amount of documents collection is interesting, if it is based on the information contents of the document rather than based on the key term (keyword) existing in or associated with the document[2]. To support content-based searching in advance, considering three basic processes of information retrieval is expected. These are; document representation (an indexing), query representation (query processing) and similarity measure between query and documents (searching and matching)[2]. Document representation or indexing is the process of describing information content in the documents. Query representation (query processing) is the process of formally describing user information need and the similarity measure is the process of applying a set of rules and procedures for matching the document contents and user information needs whether it is satisfying user need or not[3],[4]. All these three processes are performed in two IR subsystem. These are indexing and searching subsystem.

Typically, IR systems do not work on the documents themselves; rather it focuses on representation or abstraction, organization and storing of the documents. Therefore, determining whether a document is relevant to a user query is depends on the kinds of representation that is being used for the document and the query [5]. This is mainly depending on information retrieval models that are used for the representation of documents, representation of user information need and searching tasks. IR model is a general description of the IR system[6],[7]. Various information retrieval models have been proposed and developed by different scholars in IR. These models are the Boolean model, the statistical models (vector space and probabilistic models), and the linguistic and knowledge-based approach[7],[8].

Traditional information retrieval uses keywords (bag of words) to represent both documents contents and queries. To express the importance of the particular keyword in the given documents or queries, the weights are assigned to both index term and query term. The weighting scheme is based on a variety of commonly used tf*idf formula [10], [11]. When the users give their query

to the system, the system perform a task of matching those documents which have terms in the query, and the documents will be immediately retrieved if the match is exists.

A key characteristic of keywords based IR systems is that document query matching relies on the number of shared keywords that have no semantic relationships. This leads the technique to a lexical based relevance estimation that is less effective than a semantic-based one [10]. In this case, the relevant documents are not retrieved if they do not share words with the query and irrelevant documents that have common terms with the query are retrieved as a relevant list even if these terms are not describing the same concept of the documents and query terms.

Therefore, this approach misses many relevant documents because it does not capture the complete or deep meaning of the document contents and the user's query. When users use different words to express the same concept in several ways. User`s may uses different query terms to generate their information need that describing the same concept. In this case, the IR system retrieves irrelevant documents that containing the query terms as a relevant list. However, the retrieved documents may not have the expected information contents with user`s query. In another way, some relevant documents are not retrieved, as they are not match with query terms even if they are semantically describing the same concepts with query terms[6]. These are the major problems in classical IR systems that are not well addressed yet in different natural languages.

In order to come up with the solution for these common IR problems, many scholars have been developed various methodologies and techniques to enable IR system retrieves relevant documents effectively and efficiently from huge amount of document collections as per user information need. Some of the scientific approaches to solve the problems are latent semantic indexing (LSI), query expansion and reformulation, sematic compression approach, and the others.

Latent semantic indexing (LSI) is different types of standard vector space model that mainly based on the principle of words used in the same context be likely to have the same meaning. It is used to overcome the problems of word mismatching during searching. To identify the pattern in the relationship between the words and concept contained in the documents, it uses a statistical technique called Single Value Decomposition (SVD). SVD uses the term by document matrix that produced in any vector space model to generate the indexing space [6]. Term by document matrix is broken down into the reduced dimension of three separate matrices products. The model uses these three matrices to store all the information of text documents and perform retrieval task. To

determine the similarity between a query and documents, LSI uses SVD vectors of each Di document. LSI solves two most problems exist in keyword-based search: these are synonyms and polysemy. Different words that have the same meaning are termed as a synonym and the same words that have different meaning are termed as polysemy. In LSI, a query and documents are represented in reduced dimensional spaces that save more disk space. In addition to this, LSI has also another benefit. Since it is a purely statistical method, it does not depend on the knowledge of the text. Which means it is a language-independent technique that we can apply to any languages. Not only this but also it is very tolerant of noise text like misspelled words, and the like errors. The weakness of the LSI approach is that as the document size increased, the computational cost is also extremely increased [3], [6].

Query expansion and reformulation approach is the technique that helps to enhance the performance of an IR system[9]. In query expansion, the term used in the query can be expressed in a various way but it has the same meaning (synonyms) and then the system will recommend the term that more express the idea in the query. Query expansion is domain and language dependent[3], [9].

Semantic compression approach is a technique that has been introduced in 2010 as a method of improving text document matching techniques both in terms of effectiveness and in terms of efficiency[10]. Semantic Compression technique allows replacing of the term that is more general, phrase, and concepts with the specific term, phrases or concepts. The least frequent terms are treated as unnecessary and they are replaced with a term that is more general [10], [11]. As a result, a reduced number of candidate terms can be used to represent text collections. It uses the reduced number of word-space dimensions that make it more efficient (less disk space is used to store descriptors). Dimensions reduction is performed by using the possible term as a descriptor. The descriptor is a term that is used to represent a set of synonyms or hyponyms in the processed text collection. To select the descriptors, the decision is made by considering the semantic relationship between the terms and their domain frequency. To achieve this, the semantic compression approach uses ontology-based semantic compression and frequency based semantic compression techniques [8].

The ontology-based semantic compression technique is the description of a set of concepts such as items, events and their relations that are specified in give the domain to share common

knowledge. Ontology in IR plays a great role to improve the process of retrieving relevant documents from huge corpus by identifying specific properties of a domain as well as the semantic relationships between different concepts from that domain [8]. The benefit that Ontology provides for information retrieval are; it shares and reuses stored information, it avoids language ambiguity and standardizes users query expression, it allows semantic understanding of document contents, ontology-based search process has excellent reasoning capability and it provides a consistent representation of information [5], [12]. However, constructing ontology is not an easy task and there is no perfect ontology for any domain. The ontology-based representation allows the system to use fixed-size of document vector that consists of one concept per base concept. The main drawback of ontology-based representation is that, if some terms are not defined in the ontology but related to a specific domain is absent in machine-readable dictionary that used to define the concept-based version of the documents, the information loss will occur. These issues affect the performance of information retrieval. Thus, thesaurus based semantic compression is a technique that allows for correct generalization of terms in some given contexts.

Frequency-based semantic compression is mainly based on the cumulative frequency of the terms and semantic relationships. It uses semantic network and frequency dictionary to compress the meaning of the terms[8], [10].

Therefore, in order to balance the effectiveness and efficiency of Afaan Oromo text retrieval system, a new approach should be tried out. To deal with these issues, semantic compression approach using thesaurus based indexing technique is proposed to handle term mismatching problems in Afaan Oromo text retrieval during search times. Thus, the thesaurus helps the users to discover the appropriate term that matches with their information need.

1.3. Statement of the Problem

There are more than 80 languages in Ethiopia. Afaan Oromo is one of the languages with the largest number of speakers under the Cushitic language family[13],[14], [15], [16]. It is also called Oromiffa or Afaan Oromo and it is the third most widely spoken language in Africa, after Arabic and Hausa. The language is spoken by more than 40 million people as their mother tongue in Oromia region currently and it has been the official language of Oromia regional state, which is the largest region among eleven regional states in Ethiopia. In addition to these, in this state except

English and Amharic language courses, all subjects are delivered in Afaan Oromo in primary and secondary schools i.e. 1-8[4], [14].

Nowadays, a different resource such as journals, magazines, newspapers, online education, books, entertainment media, videos, pictures, and movies are released and available in an electronic format both on the internet and offline sources [4]. Not only this, but there is also a huge amount of information being released in Afaan Oromo since it is a language of education, research, administration, politics and welfare activities[4]. Therefore, it is a rapidly growing language in technological activities and social interactions. As a result, this language plays a great role in the social, political and business arena nowadays. This increased use of Afaan Oromo leads to the need for developing efficient and effective information storage and retrieval systems to represent, store and retrieve relevant documents from language document collections.

In Ethiopia, there are many attempt to develop information retrieval system for local languages especially for Amharic, Tigrigna, and Afaan Oromo. Several research works has been done for Amharic language to search and retrieve Amharic documents from large collections of Amharic documents which written in Ge`ez script characters. However, there are few research works done for Afaan Oromo in the area of information retrieval. Some of the works in information retrieval for Afaan Oromo are Afaan Oromo news text summarization[17], Afaan Oromo-English Cross-Lingual Information Retrieval (CLIR): A Corpus Based Approach[18], English – Afaan Oromoo Machine Translation: An Experiment Using Statistical Approach[16] and Afaan Oromo information retrieval system using vector space model[4].

However, among these work, the major problems of vector space model base information retrieval system is the term mismatching problems that concerns about the representativeness between terms used for represent documents (i.e. document indexing) and the terms used by the searcher to describe their information need. These happen due to different reasons like that of if the searcher uses different query terms to describe the same concept. This means that when user's uses different query terms that are not used in the index terms but expressing the same concept, term mismatching is occur and the IR model consider that the document that not share query terms is irrelevant for query term if the term is not in index terms. This problem also includes synonyms and polysemy words in Afaan Oromo. Thus, the recent work, Afaan Oromo text retrieval prototype developed by Gezahgn Egg using vector space model is unable to handle synonymous and polysemy words

of Afaan Oromo text documents during search time. So one-way of controlling terms mismatch or vocabulary problem is using thesaurus based indexing that shows the semantic relationship among the terms.

Therefore, in order to overcome the term mismatch problem, we need a retrieval model that captures semantic relations between terms to handle Afaan Oromo synonymous words. The integration of term relations contributes to reduce the gap during the matching between a query representation and a document index. Accordingly, this research work is proposed to design the semantic compression model for Afaan Oromo text retrieval using thesaurus-based indexing technique to enhance the effectiveness of Afaan Oromo text retrieval system.

To this end, this research work attempts to explore and answer the following research questions.

1. How semantic compression approach is applied to control synonymous terms in Afaan Oromo IR system?
2. Does thesaurus based semantic compression technique improve the effectiveness of IR system in searching for relevant Afaan Oromo document?
3. What are the challenges of developing semantic compression model for Afaan Oromo text retrieval using thesaurus based indexing technique?

1.4. Objective of the study

The general and specific objectives of the study are discussed below in this subsection.

1.4.1. General Objective

The general objective of this study is to design thesaurus based semantic compression model for Afaan Oromo text retrieval using thesaurus based indexing approach.

1.4.2. Specific Objectives

In order to achieve the set general objective of the study, the following specific objectives are formulated to be attain:

- ✓ To conduct a review of related literature in the area of information retrieval that uses semantic compression technique in IR context and acquiring deep understanding about the application of the concept of semantic compression in the information retrieval system.

- ✓ To understand Afaan Oromo writing systems and its structures (morphology) of the language.
- ✓ To prepare corpus from Afaan Oromo documents collection and applying linguistic pre-processing
- ✓ To design thesaurus based semantic compression model for Afaan Oromo text retrieval.
- ✓ To develop a prototype for designed model for Afaan Oromo text retrieval.
- ✓ To test and evaluate the performance of the proposed model.
- ✓ To discuss and report the findings of the experimental evaluation results.
- ✓ To identify the challenges of conducting a research on a thesaurus based semantic compression for Afaan Oromo text retrieval.
- ✓ To draw a conclusion and forward recommendation for further study to improve the performance of Afaan Oromo IR system.

1.5. Methodology of the study

In order to achieve the objectives of the study stated above, we followed different methods and activities. These methods are; literature reviews, corpus preparation, research design, prototyping, and performance evaluation of the model.

1.5.1. Literature Review

In order to achieve the objectives of the study stated above, we followed different methods and activities. These methods are; literature reviews, corpus preparation, research design, prototyping, and performance evaluation of the model.

In this section, various related literature in the area of information retrieval and utilization of semantic compression approaches for IR are reviewed to understand the state of the art of the study area. Especially in the area of approaches to applying semantic compression for IR with different techniques and approaches are reviewed. The review involves different resources such as published research papers, journal articles, previous related research works, and electronic materials on the web. In addition to this, some printed materials such as Afaan Oromo textbooks, Afaan Oromo dictionaries, and other materials are reviewed in detail to obtain a deep understanding of the subject area. As the study is conducted on Afaan Oromo text retrieval that uses Afaan Oromo textual documents, the history, characteristics, and morphology of Afaan Oromo are also studied in parallel.

1.5.2. Data collection and Corpus Preparation

Afaan Oromo does not have publically available compiled text corpus used for any information retrieval tasks so far. As a result, the dataset used for carrying out the experiment in this study is collected from different Afaan Oromo official media such as Oromia Broadcast Network (OBN) News online, Afaan Oromo online BBC news, Voice of American news (VOA), Afaan Oromo bible; Afaan Oromo textbook, legal documents and any other resource available on the internet were used. The collected text documents were reviewed to discovery the synonymous words and later on, it is used for construct the thesaurus manually from the text corpus.

1.5.3. Research Design

In this study, the thesaurus based semantic compression approach is used to design the Afaan Oromo text retrieval model that index Afaan Oromo text documents based on the thesaurus. The whole process is divided into the following sub-phases: first, it starts with documents pre-processing phase that includes (tokenization, normalization, stop word removal, and stemming) and then followed by term weighting using term frequency inverse document frequency (tf*idf) technique. Finally, thesaurus based semantic compression is applied to indexing and searching.

1.5.4. Development Tools

To accomplish this study, development tools are needed for implementation of the proposed model. Python 3.4.0 programming language is used to develop the prototype system. We select Python 3.4.0 as programming language tool because it is easy to learn and it is a powerful programming language, especially for natural language text processing. It is a dynamic programming language used in a wide variety of natural language application domains. The prototype system was run on Windows 10 64 bits' environment.

1.5.5. Testing Procedures

The developed prototype system is tested based on formulated test queries and relevance judgments that were prepared by the researcher. To evaluate the effectiveness of a prototype system; the most common IR performance evaluation metrics such as precision, recall, and F-measure are used. During evaluation time, the user judgment is involved in order to ensure that whether the system correctly retrieves relevant documents to the user query or not. In this work, F-measure is used to measure the overall performance achieved by the prototype system, and from

this measure, the greater result of F-measure is interpreted as the best attempt to find the best possible compromise between precision and recall.

1.6. Scope and Limitations

Semantic compression based IR is a very complex task that needs understandings of natural language processing techniques in details. A complete semantic compression based IR system will involve a variety of natural language processing tools such as part of speech (POS) tagger, parser, stemmer, WordNet and ontology. However, it is difficult to integrate all this applications to the non-researched language like Afaan Oromo. Even though some of these NLP tools have been developed by a number of scholars for the dissertation, they are not freely available for integration with the system we proposed to develop. Therefore, the scope of this study is to develop a prototype IR system for Afaan Oromo by applying semantic compression approaches using thesaurus based indexing technique. The prototype system is designed to search Afaan Oromo text from large Afaan Oromo document corpus. For experiment, Afaan Oromo text documents are collected from different domain and text operation such as tokenization, normalization, stop word removal and stemming is performed to extract content-bearing terms that used for indexing. Then after, for each content-bearing term, term weight is computed using tf-idf term weighting technique. Finally, thesaurus based inverted file indexing is used to organize the index term. In this study, thesaurus based indexing is used to handle Afaan Oromo synonymous words in Afaan Oromo text. There are a number of word senses and semantic relations in thesaurus such as hyponyms, hypernyms, synonyms (synset) and antonym.

- ✓ Among all these words semantic relations listed above, only synonyms (synset) word semantic relations is used in this research work to handle different words that have the same meaning.
- ✓ The noun synonymous terms are extracted from corpus because other word categories are rarely exist as a concept.
- ✓ The study involves both an indexing and a searching part.

The main work is limited to a textual document of Afaan Oromo corpus only. However, there are other data types such as image, audio, video, and graphics, which are out of the scope of this study. Due to the lack of standardized large size document corpus and, lack of standardized thesaurus for

Afaan Oromo text, extracting all word sense and semantic similarity from small document corpus is difficult. Therefore, controlling polysemy words is another limitation of this research.

1.7. Application of the results

Nowadays, the increasing interest in the field of information retrieval is mostly related to its practical applications and several settings. Information retrieval can be applicable in commercial settings, political settings, academic settings, entertainment, and the like. IR system is mainly used to retrieve relevant information resource from a huge amount of information resource storage as per request present to it by the user. Currently, information seekers search for relevant information about politics, academic, commercial, events, religion etc. of a particular situation for their own basic need. Therefore, the output of this research work can be used in particular for the following purpose;

- The output of this research work can be applied in Afaan Oromo IR system to improve the efficiency and effectiveness of the system. When thesaurus based semantic compression is integrated with AOIR system, it supports users who are poor in Afaan Oromo and native language speakers to retrieve information that meets their information needs by retrieving those relevant documents, which have a similar meaning to the user query.
- It can be implemented in governmental, non-governmental, private institution and other organizations in order to enable users to search relevant text document from a huge amount of information storage.
- It will be used as a motivation for other researchers to work on this area for further improvement of retrieval system in Afaan Oromo and other local languages.
- In addition to that, it can be used in the following application area of natural language;
  - ✓ Text matching
  - ✓ Semantic search engine
  - ✓ Question answering
  - ✓ Query expansion and reformulation

1.8. Organization of the study

The rest part of this thesis report is organized into five chapters as follows.

Chapter 2 presents literature reviews and related works conducted on the area of Information retrieval. In the first subsection, overviews of information retrieval, information retrieval process, and Information retrieval models are reviewed. Secondly, Semantic compression approaches. Finally, Reviews of any related works that integrate semantic compression approach in IR for foreign and local languages is presented. In chapter three, understanding of writing system and morphology of Afaan Oromo is discussed. In chapter four, description of the system design and Architecture of the semantic compression model for AOIR; starting from documents pre-processing step up to document indexing and searching. The fifth chapter presents, implementation, evaluation, experimental results of the prototype and discussion on the findings are briefly explained. Finally, in chapter six, conclusion and recommendations are presented.

CHAPTER TWO

LITERATURE REVIEWS

2.1. Introduction

This chapter is concerned with review of the literature, which is divided into two broad sections. Section one discusses overviews of information retrieval, information retrieval processes, Information retrieval models and the series of activities that involve in the documents and query indexing processes, as well as the criteria by which information retrieval systems are evaluated. In the second section of this chapter, some research works related to semantic compression based IR for different natural languages are studied.

2.2. Overview of IR

Information retrieval (IR) deals with the representation, storage, organization and access to information items. The information items can exist in the form of documents, Web pages, online catalogs, structured and semi-structured records as well as in the form of multimedia objects[19]. The representation and organization of these information items should provide the user with easy access to the information, in which he/she is interested [19], [20]. Information Retrieval includes activities like; modeling, Web search, text classification, systems architecture, user interfaces, data visualization, filtering and languages in which the documents are written [1],[4], [19]. The main goals of information retrieval are to retrieve all the documents that are relevant to a user query while retrieving as few non-relevant documents as possible. The better performance achieved by particular information retrieval system is based on how much the system retrieves all the relevant documents (information items) stored in the corpus and how much it rejects all non-relevant information items or documents as per user request.

In order to judge this, the whole IR system is divided into two main subsystems: Indexing and searching [4]. Indexing is the process of preparing index terms, which are either content bearing or free text, extracted from documents corpus. Searching is a process of matching users information via query to documents in the collection via index terms. For effective retrieving of relevant documents from the collection of the certain corpus, the documents are typically transformed into a suitable representation using information retrieval strategies in which each information retrieval strategy incorporates a specific information retrieval model for its document

representation purposes. These indexing and the searching process are controlled by different information retrieval models. The information retrieval models that are used for the representation of document and query are; Boolean model, the statistical model, which includes the vector space and the probabilistic retrieval model, and the linguistic and knowledge-based approach [8], [7], [20].

Once the documents are represented by respective IR model, some performance evaluation techniques are used to assess the accuracy of an information retrieval system. The evaluation of an information retrieval system is the process of judging how well a system meets the information needs of its users. In addition, how it approves the accuracy of IR system. There are several IR effectiveness evaluation techniques. However, Precision and recall are the two commonly used techniques that are used to measure the performance of information retrieval system during the evaluation of an information retrieval system. Precision is the fraction of the retrieved result documents that are relevant to the user's information need whereas Recall is defined as the fraction of relevant documents in the entire collection that are returned as result documents. Figure 2.2-1: shows the general architecture of information retrieval components and their interrelation.

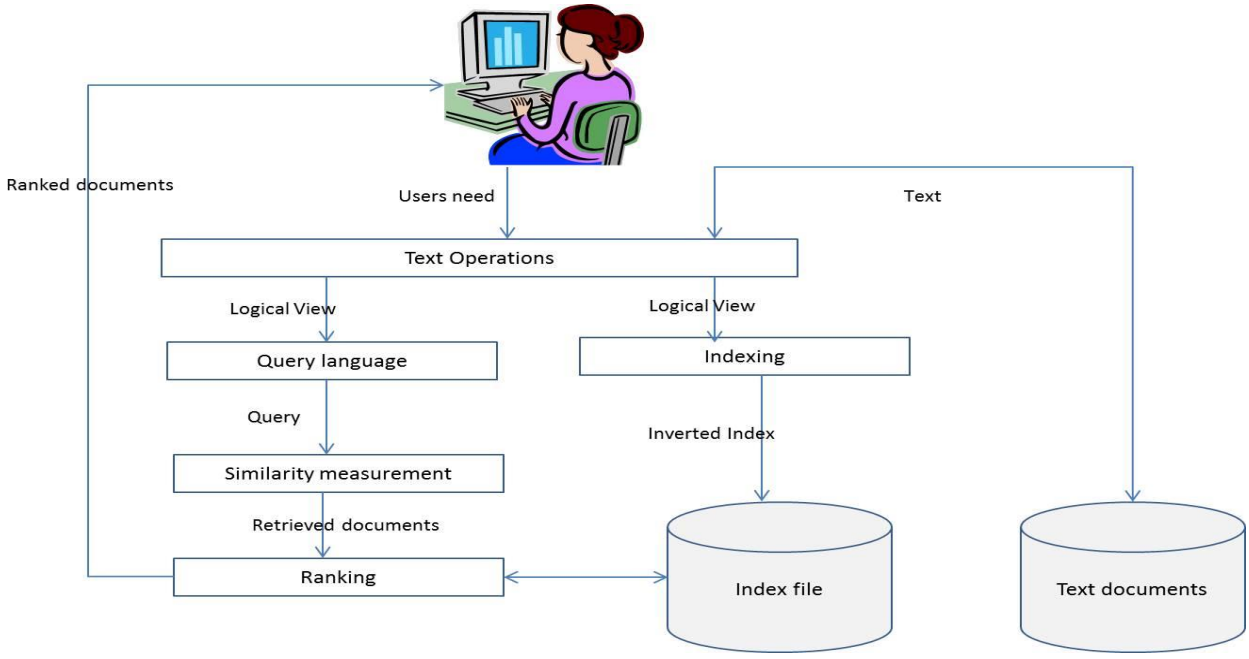


Figure 2.2-1: The General Architecture for information retrieval

2.2.1. Information Retrieval Process

Information retrieval is described as the process of representation, organization, storing of and accessing to information items from the collection of documents [21]. Generally, all these tasks are performed by passing through three basic information retrieval processes. The first one is extraction of terms from document collections and descriptors into a logical representation of term (indexing), the second one is handling user`s information needs or query and transforming it into an abstract representation (query processing) and the third and the final one is searching both representations (matching and ranking) [21].

Indexing

The indexing process is an offline process of organizing documents using a list of keywords that are extracted from documents collection [4]. The process of generating the data structure that represents term to document is called indexing. Therefore, the indexing task deal about identifying the proper word that specifies the document contents that it is extracted from document. This is because of; all words in the text are not equally important for representation of document contents, as well as not all good words, are equally describing or defining the contents of the document. Therefore, it is necessary to pre-process the documents collection to identify the appropriate word to be used as an indexing term.

There are several index structures used for organizing index terms; such as sequential files, inverted files, suffix tree, suffix array and signature file [4], [21]. The sequential file is an indexing structure, which accesses elements of record in a predetermined ordered sequence. In a sequential file, the records are arranged in serially fashion one after the other in lexicographic order based on the value of some key field. Inverted file is an indexing structure that stores a mapping of contents list such as word, numbers and the like to its location in the set of documents [4], [21], [1]. It is the most popular data structure used in text information retrieval systems. Suffix tree and suffix array are a data structure that processes the suffixes of a particular string to allow fast implementation of string operations. A signature file is a data structure that works based hash coded and it is a word-oriented index structure[1] . From various indexing structure discussed above, the most popular one in text searching is inverted file. Inverted file is a technique used for indexing of text collection to make the searching process fast. Inverted file is built from two files:

the vocabulary file and posting file. The vocabulary file is a collection of index term in the text document that is organized by terms. Vocabulary file stores all of the key words from all documents in lexicographical order and for each word a pointer to posting file.

Building an index using an inverted file from a documents collection involves several steps to improve the speed of text retrieval. To build the index in advance, the major steps are followed: The first step is collecting the document to be indexed. The second step is tokenizing the whole document into a list of tokens. The third step is performing linguistic pre-processing activities such as normalization, stop word elimination, and stemming. Finally, indexing the document by creating an inverted index.

Tokenization is the process of chopping off the whole documents into a piece of document unit so-called tokens and at the same time throwing away certain characters that have less contribution to document indexing. A token is an instance of a sequence of characters in some particular document that is grouped together as a useful semantic unit for processing.

Normalization deals with case folding. Every words, phrases, and sentence existing within the document collection should be converted into a uniform case format. Which can be upper case, lower case, or a mix of the two.

Stop word elimination refers to removing of every frequent term from the indexing process, as they are poor in document contents indication.

Stemming refers to the conflation of words to their base form or root. Where the morphological variants of words are stripped to one single suffix entry.

Query processing

Once the documents are indexed, the next step is parsing of user query into an internal form. This involves several steps. First, the user specifies her/his information need using natural language supported by underlying IR system and the IR system accepts it as an input from the user. Then, the system parses and transforms the user query into internal logical representation by applying query pre-processing activities (such as query tokenization, normalization, stop word removal and stemming) as applied in document indexing. After query pre-processing operations are completed,

the list of query terms is weighted based on term weighting scheme used in document indexing task. The mathematical formula to compute query term weighting is given as:

$$Wiq = \left(0.5 + \left(\frac{0.5tfiq}{maxtf}\right)\right) * \log\left(\frac{N}{ni}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots Equation\ 2.1$$

Where:

- **Wiq** is stands for the weight of term i in query **q**
- t**fiq** stands for the frequency of term i in query **q**
- **max tf** stands for the maximum frequency value of all query term
- **N** stands for total number of documents in the corpus (this is not including queries)
- **ni** stands for total number of documents in which the query term is found.

Matching

Following the end of the query-processing phase, user query term is matched against index term. Because of this operation, a set of documents are returned as per user information need. This process is called matching[22]. Before display the result to the user, the IR system will rank, sort a set of retrieved documents, according to the relevance function criteria, and return them by decreasing order of estimated relevance. The main problem of information retrieval system is how it identifies the relevant document and which are not properly. To differentiate the relevance among the document, information retrieval system using the ranking algorithm to make a decision. Ranking algorithm plays a great role in information retrieval system by providing the mechanism in which the relevant documents are displayed for the users in decreasing order based on their similarity score with the query. The document with high similarity score is considered as a relevant one to the query and it is ranked at the top of the relevant document list that is retrieved. Then, from a set of returned relevant documents, the user identifies the relevant document that contains useful information about their query[4], [21].

Therefore, ranking algorithms are considered as the core component of IR systems since they are important to determine the performance of the IR system. From different literature, there are several ways to measure how two or more documents are similar to each other or how the documents and query are matching each other[23]. Using similarity measure, a set of documents can be compared to a query term and the most similar document will be returned. Some common

similarity measurements in IR are Euclidean distance, dot product, Cosine similarity and the like [23]. Euclidian distance computes the similarity between document and query using the "root of square differences between coordinates of a pair document and query terms". The similarity between vectors for document *Dj* and query Q can be calculated as:

$$\mathrm{sim}(\mathrm{dj},\mathrm{q}) = |\mathrm{dj} - \mathrm{q}| = \sqrt{\sum_{\mathrm{i}=0}^{\mathrm{n}} = (\mathrm{wij} - \mathrm{wiq})2} \quad \text{---------- Equation 2.2}$$

Where:

Wij stands for the weight of a term *i* in a document *j* and *wiq* stands for the weight of term *i* in query *q*.

The second document and query similarity measuring technique is the dot product. The dot product is also called inner product and it computes the document and query similarity by taking the product of the magnitudes of query term and document vector. For a give document dj and query q vectors, their similarity is computed as:

$$\mathrm{sim}\,(\mathrm{dj},\mathrm{q}) = \sum_{\mathrm{i}=0}^{\mathrm{n}} = \mathrm{wij}\,.\,\mathrm{wiq} \quad \text{---------- Equation 2.3}$$

Where:

Wij stands for the weight of the term i in document j and wiq stands for the weight of term i in query q.

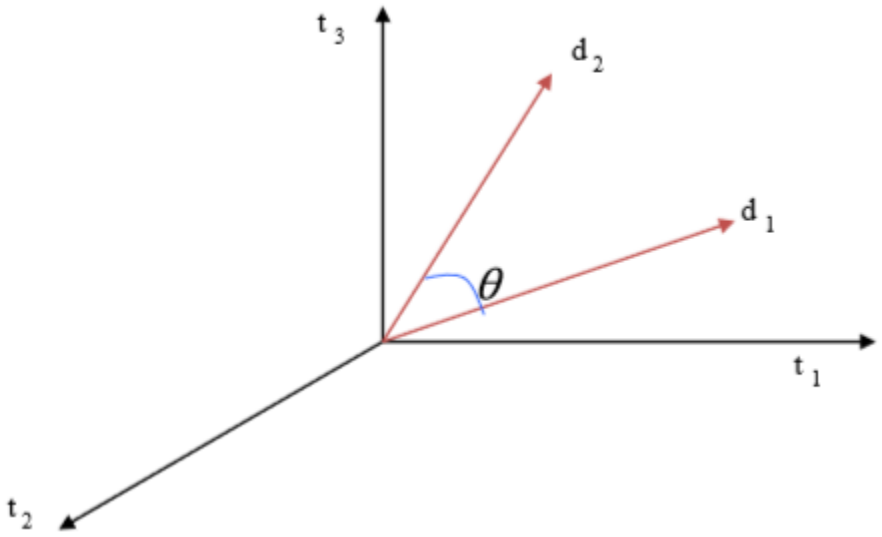
The third document and query similarity measuring technique is cosine similarity, which is a popular similarity measure in information retrieval. Cosine similarity measure use document vector, query term vector and the angle between the two, which is θ. If the query term is not appearing in the document, the similarity measure score will be zero. Else, if the query term frequently appearing in the document, its similarity score will be higher and it is considered, as the document is relevant. This is actually based on two factors such as the length of the vector and the direction of the vectors. If the query and document vector are pointing to the same direction in the space, they are likely similar. The cosine similarity measure is calculated as un-normalized similarity:

$$\text{sim}(Q, Di) = \sum_{j=1}^{t} wqj * wdij \quad \text{Equation 2.4}$$

Then, from the given equation above; the cosine similarity is a normalized inner product that can be calculated as:

$$\text{Cosine similarity: } \text{sim}(Q, Di) = \frac{\sum_{j=1}^{t} wqj * wdij}{\sqrt{\sum_{j=1}^{t} (wqj)2 * \sum_{j=1}^{t} (wdij)2}} \quad \text{Equation 2.5}$$

For a given query Q that contains query terms t1, t2 and t3, suppose there are two documents vectors d1 and d2 that represent the document in the space. Therefore, the cosine similarity for the given two-document vector will be:



Finally, the retrieved documents are ranked based on cosine similarity score result. The document which is with higher cosine similarity score is ranked at the top of the list and others follow in descending order from higher to lower[4].

2.2.2. IR Models

A Model is an idealization or abstraction of the actual process. An Information retrieval system uses different models for document and query representation, which have a great influence on retrieval performance. It is required for a better understanding of the document and query representation process as well as it provides a description to guide system implementation[6],[23].

Information retrieval models are classified into two broad approaches, which are semantic approach and statistical approach. The semantic approaches are works on syntactic and semantic analysis of the natural languages texts. Some information retrieval models based on semantic approaches are latent semantic indexing, linguistic and knowledge-based approach and neural network. The statistical approaches include vector space model and the probabilistic model. The statistical models use the statistical measure to retrieve documents that are highly ranked. The most commonly used information retrieval models that are based on a statistical approach are the Boolean model, vector space model and the probabilistic model [6], [7], [22].

Boolean Model

The Boolean model is the first information retrieval model that represents documents and queries as a set of an index term. It works based on set theory and boolean algebra that allow the users to specify their query using a combination of three different boolean operators such as the product operator (AND), the sum operator (OR) and the negation operator (NOT) [7], [8], [4]. The logical operator AND is applied to merge or group a set of terms into single query statement that will return the document if and only if all the query terms exist or match the index term of the document. Else, if one of the query terms does not match the index term from the query, there is no any document will be retrieved. The weight for index term is binary which is either 1 or 0. The similarity between the documents to the user query is based on criteria that say if the document satisfies the Boolean query; the similarity between documents to query is one (present) which is interpreted, as the document is relevant for user query [7]. Else, if the similarity between documents to the query is equals to 0 (absent), it is interpreted as the document is not relevant for the user query.

For example; Q. what is Data AND communication AND networking?

For the query statement data AND communication AND networking, there is three query term that is combined by AND logical operator. In this case, only a document that is containing index terms with all three terms will be retrieved. Else, if one of these terms is not indexed with any document in the collection, nothing will be retrieved. In the Boolean model, users can also employ the OR logical operator to represent their query. If a term in a user, query is linked by OR logical operator, the document with either of the terms or document with all terms will be retrieved.

For example; Q. what is data OR communication OR networking?

Here three query terms are linked by OR logical operator. In such a case, the document containing index term data OR communication, communication OR networking, data OR communication OR networking as well as data OR networking will be retrieved.

The Boolean model has two major drawbacks. The first one is its retrieval strategy is based on binary criteria, in which a document is predicted to be either relevant or non-relevant. Due to this reason, it does not provide an appropriate basis for ranking the retrieved results. This results in low precision when the retrieval space is too big. The second drawback is; it is difficult for users to translate their information need or query into a Boolean expression using logic operators [4].

Statistical model

The vector space and probabilistic models are the two types of the statistical retrieval approach in which both of them use statistical information in the form of term frequencies to determine the relevance of the documents with respect to a user query. Both of them produce their output as a list of documents ranked by their estimated relevance [4].

Vector Space Model

The vector space model is an algebraic model that represents documents and queries as a vector in N-dimensional space. Each dimension of the space corresponds to a separate term in the documents collection vector [23],[4]. For each separate term, the value is assigned to show how it is important to represent the semantic feature of the document. This assigned value is called weight. This indicates, if the term t occurs in given document dj, the weight of the term will be expressed as the coordinate of dj in document space. Whereas if the term t is not appearing in the document dj, the weight of the term will be zero. For a total collection of documents in the corpus, the document d that containing term t that described the documents itself is represented as a term by document matrix. Each column of the matrix is representing a document vector in the space and each row of the matrix is representing a term vector in the space [4], [6], [23].

Term list	*Doc1*	*Doc2*	*Doc3*	*Doc4*	*Doc5*	*Doc6*	*Doc n*
Term1	**W11**	**W12**	**W13**	**W14**	**W15**	**W16**	**W1n**
Term2	**W21**	**W22**	**W23**	**W24**	**W25**	**W26**	**W2n**
Term3	**W31**	**W32**	**W33**	**W34**	**W35**	**W36**	**W3n**
Term4	**W41**	**W42**	**W43**	**W44**	**W45**	**W46**	**W4n**
Term5	**…….**	**……**	**…….**	**…….**	**…….**	**…….**	**……**
Term m	**Wm1**	**Wm2**	**Wm3**	**Wm4**	**Wm5**		**Wm n**

Table 2.2-1: The Vector Space model term by document matrix

To measure the degree of importance of the term in the document or term in the query, vector space model assign weight to term in the document and term in the query using the term weighting scheme. In information retrieval, the most common term weighting scheme for vector space model is tf-idf technique [4], [24]. It includes term frequency (TF), inverse document frequency (IDF) and term frequency-inverse document frequency (tf-idf).

Term frequency (tf): it is the frequency of the term in the document. It determines how many times the word appears in the document. It is calculated as a number of times a word appears in a document divided by the total number of words in that document. The mathematical equation for term frequency computation is given as:

$$tfij = \frac{fij}{\max\{fij\}} \quad \text{Equation 2.6}$$

Where ***fij*** is a number of term **i** in document **j** and max {***fij***} is the total number of terms in document j.

Inverse document frequency: - deals with measuring how important a term is in the document. It is defined by using the fraction of ***N/dfj***, where N is the total number of documents in the collection and ***dfj*** is the number of documents in which the term j occurs. The fewer the documents in which a term occurs, the higher the weight it has. The lowest weight of 1 is assigned to terms that occur in all documents. Because the terms that are limited to a few documents are useful for discriminating those documents from the rest of the collection. The terms that are frequently occurring across the entire collection are not as helpful for representation of the keywords[25]. So that, inverse document frequency is used for assigning of higher weighting to rare words which are more discriminative. Mathematically it is calculated as;

$$\mathrm{idf} = \log 2\ (1 + \frac{\mathrm{N}}{\mathrm{dfj}})) ------------------\ -------- \text{Equation 2.7}$$

Where *N* is the number of all documents in the corpus and ***dfj*** represents the number of documents containing the term t.

However, if the term frequency is zero, the term doesn`t occur in the document. Then, the log of zero is negative infinity, which is problematic. So, to solve this problem, one is added to the term frequency if the term does not occur in the document. Then, log of one is equivalent to zero and it return an answer of zero if the term doesn`t occur in the document as well as in the query term.

Term Frequency-Inverse Document Frequency (tf-idf): - is a weighting scheme that is commonly used in information retrieval tasks. It is a well-known method to evaluate how important a word in a document is. The tf-idf is a very interesting way to convert the textual representation of information into a Vector Space Model (VSM). It is composed of two terms: the first one is normalized term frequency (TF) which is the number of times a word appears in a document divided by the total number of words in that document[4], [24]. Then the second term is the inverse document frequency (IDF) which is computed as the logarithm of the documents in the corpus or collection of documents divided by the number of documents where the term *ti* appears. Then, TF-IDF is the product of TF *IDF. Mathematically it is written as:

$$\mathrm{tf.idf(t, d, D)} = \mathrm{tf(t, d)} * \mathrm{idf(t, D)} --------\ ----------- \text{Equation 2.8}$$

From the above expression, *tf-idf* is computed as:

$$\mathrm{tf} * \mathrm{idf} = \mathrm{tf} * \log 2 \left(1 + \frac{\mathrm{N}}{\mathrm{dfi}}\right) ------------------------ \text{Equation 2.9}$$

In vector space model, the similarity between the document and query is determined by an association between the document vectors dj and query q. The association is measured by the associative coefficient based on the inner product of the document vector and query vector where word overlap is indicating the similarity of the words.

In information retrieval, there are several ways of document and query similarity measurements. The most common and popular document and query similarity measure used in the vector space model is a cosine similarity measure. Cosine similarity measure uses the angle between the document vector and query vector to measure the similarity between the document and query.

Cosine similarity measures how much the query vector and document vector are pointing in the same direction in which the angle between two vectors is used as a measure of divergence between the vectors. The cosine of the angle is used as the numeric similarity between the index term and query term [8]. The cosine similarity score is 1 (one) when both the index term and query term vectors are pointing in the same direction. However, if the cosine similarity score is between the document vector and query vector is zero or approaching to zero, this shows that the document vector and query vector are dissimilar [26], [9].

The document that matches the query term with a high cosine similarity score is judged as to be a relevant and ranked on the top of the list whereas the documents that match the query term with a low cosine similarity score is ranked at a lower level of the list from relevant documents that retrieved.

Vector space model has several strengths when compared to the Boolean model. The first advantage of VSM is it supports partial matching between index term and query term. It allows retrieving documents that approximately close to the query term based on their degree of cosine similarity. It improves information retrieval performance by using term weighting scheme, it sorts and ranks document based on their degree of relevance to query based on cosine similarity measurement. It is simple to implement and fast [4], [6].

However, vector space model has some limitations. The first limitation of the vector space model is; each term is treated as independent components in the semantic space and it is not possible to include term dependencies into the model. Which means it couldn`t handle the issues of synonyms and hypernym words in natural languages. The other drawback is that it needs more storage space to represent term by document matrix for large text documents [6], [7].

Probabilistic Model

The third IR model is a Probabilistic model. This model uses the probability theory to build the search function and its operation mode. The information used to compose the search function is obtained from the distribution of the index terms throughout the collection of documents or a subset of it. This information is used to set the values of some parameters of the search function, which is composed of a set of weights associated with the index terms [7]. The fundamental idea behind the probabilistic model is, there are a given user query q and a document di in the collection,

then the probabilistic model tries to estimate the probability that the user will find the set of document di as a relevant set. In this model, the probability of relevance is depending on the query and document representation only [7].

In addition to this, the probabilistic model assumes that, there is subsets of all documents that the user prefers as the answer set for the query q. such an ideal answer set is considered as it increases the overall probability of relevance to the user query. The document in the set is predicted as relevant to the user query. Whereas document not in this set is predicted as not- relevant to user query [6], [7].

Like that of other models, the probabilistic model has its own limitations. The first limitation of the probabilistic model is it does not take into account the frequency of the index terms inside the documents. The second one is, it considers the sets of documents initially as a relevant and no relevant separately.

Linguistic and Knowledge-Based Approach

Linguistic and knowledge-based approaches have been developed to solve the problem of presence and absence of the exact keyword strings as specified by the logical representation of the query. To solve this problem, it uses methods such as morphological, syntactic, and semantic analysis to retrieve documents more effectively. Finally, it has to resolve word ambiguities and generates relevant synonyms based on the semantic relationships between words [8]. The development of a sophisticated linguistic retrieval system is difficult and it requires complex knowledge bases of semantic information and retrieval heuristics [26]. Hence, the systems often require techniques that are commonly referred to as artificial intelligence or expert systems techniques.

2.2.3. IR System Performance Evaluation

As stated in[26], Information retrieval systems, evaluation is broadly classified into three types. The first one is a functional evaluation. An evaluation technique used to test the specified system functionalities one by one. The second one is the performance evaluation. It is the most common performance evaluation technique in IR, which measures the system performance in terms of time and space used by the system to perform a specific task. According to this performance evaluation technique, the system with shorter response time and smaller space used is considered as a system with better performance. The last one is retrieval performance evaluation. This type of evaluation

technique is deal with assessing how well the IR system satisfies the information need of its users. There are two types of retrieval performance evaluation such as user-based retrieval performance evaluation and system based-retrieval performance evaluation. The first one measures the user's satisfaction with the system. User-based retrieval performance evaluation is focused on the attitude of the users. It is much more informative and useful but it is extremely expensive and difficult to achieve. Whereas the second technique deals, with how well the system is able to rank the relevant documents. It is based on the design model, in which experiment is performed in the retrieval process by controlling some of the variables that affect retrieval performance. It is much less expensive compared to user-based retrieval performance evaluation. System based retrieval performance evaluation has several sub-techniques used to measure system performance. Among those techniques, the two common information retrieval performance evaluation metrics are precision and recall. Precision is the number of relevant documents retrieved by the system divided by the total number of documents retrieved by the system.

$$\text{Precision (P)} = \frac{\{\text{Relevant documents}\} \cap \{\text{retrieved documents}\}}{\{\text{Total no. of retrieved documents}\}} \quad ------2.10$$

Recall is the relevant documents retrieved by the system divided by the total number of relevant documents in the corpus.

$$\text{Recall (R)} = \frac{\{\text{Relevant documents}\} \cap \{\text{Retrieved documents}\}}{\{\text{Total no.of Relevant documents}\}} \quad ---------- 2.11$$

Even though the most popular measures to evaluate the IR retrieval performance is precision and recall, they have some drawbacks that limit their effectiveness in certain cases:

The first case is, to determine the maximum recall for a given query; it needs deep knowledge of all the documents in the collection. The second case is, when users use interactive systems, they can specify their information need using a series of interactive steps with systems. In this circumstance, precision and recall evaluation measures are insufficient. Thirdly, using precision and recall evaluation measures individually does not contribute enough information about IR system success.

Due to these limitations, another evaluation measure that is the combination of both precision and recall is proposed by the scholars. This technique is called F-measure. F-measure is used to negotiation the difference between precision and recall [26].

F- Measure

Precision and recall should be used together to obtain a more complete evaluation metric. Since F-measure is the harmonic mean of the precision and recall, it provides more strong evaluation criteria. Mathematically is computed as:

$$F - measure\ (F) = \frac{2 * Precision * recall}{Precision + recall} -------------------2.12$$

2.3. Approaches to semantic compression

Semantic compression is a technique that is intended for generalization of terms in a given context without losing the basic information contents of the terms [11]. The semantic compression technique is used in information retrieval as a method that improves text document matching by increasing both effectiveness and efficiency. It is addressed using semantic networks and data on term frequencies. A semantic network is used to group the term based on their semantic relationship to some concept from one or more controlled vocabularies according to the used organizational structure that is common to all involved controlled vocabularies. Data on Term frequency is used in the form of frequency dictionary in which the least frequent term is considered as less important and it is replaced by more general term (hypernyms) [10],[11]. This more general term is stored in the semantic network and later used for searching. As a result, a reduced number of terms are used to represent the text documents without missing the actual information content. Using few words or terms to represent text document result in words or terms dimension reduction. This dimension reduction is achieved by using the descriptors. Descriptors are any term or word that represent the document contents. So, they are selected to represent a set of synonyms or hyponyms in the given pre-processed documents. While representing a set of synonyms words with their respective descriptor, two basic things that should be considered are the relationships among the terms and their frequency in context domain. To do this, it is based on cumulative frequencies of the term or word and information about semantic relationships among the terms within the document. First, consider that there is N- number of concepts that belong to each document contents and the total frequency of each concept f (Ci) has to be calculated for all documents in the corpus. To achieve this, cumulated term frequency and integrating of information from synonyms relationship is needed. Cumulated term frequency is computed by adding the sum of hyponyms frequency to the hypernym. Following this, the synonym with the highest cumulative

frequency is selected to incorporate the synonym relationship information. As a result, the term with the largest cumulative frequency is chosen and it needs to represent the document. Therefore, cumulative concept frequency is computed by adding the sum of hyponyms` frequencies to the frequency of the hypernym moving upwards in the hierarchy [10],[8],[11]. This is mathematically expressed as;

$$Cum\ f(Ci) = f(ci) + \sum_j cum_f(c_j)\ where\ C_i\ is\ a\ hypernym\ of\ c_j$$

Then, N-concepts with highest cumulative frequencies is sorted and selected as a descriptor list (target key concepts). Finally, the compression mapping rules are defined and nested to handle every occurrence of Cj as its hypernym is Ci in further processing.

However, semantic compression allows natural language processing tasks such as text matching, to operate at concept level rather than focusing on a level of the single term. This concept level semantic compression can be performed by replacing longer phrases with their more compressed forms that allow capturing a common meaning expressed with a different set of terms [11],[10]. However, from the grammatical rule of the language point of views, investigating concept interdependency and generating the more compact phrase that has common meaning with the original phrases is challenging tasks.

Many studies have been conducted on the IR system using different approaches to enhance the performance of the IR system. In IR system, semantic compression approach can be utilized in IR using two approaches. Based on ontology approaches and frequency-based approaches[8]. Ontology-based approaches are concerned with describing the domain knowledge and frequency based approaches are more focuses on choosing the appropriate word for query term expansion.

2.3.1. Ontology-Based Approach

Ontology-based semantic compression is a technique that uses a clear description of a shared conceptualization and their relationships. Ontology is a formal and explicit description of domain concepts, which are usually considered as a set of entities, relations, instances, functions, and axioms. Based on the knowledge area it covers, ontology can be classified as general and domain-specific ontology[26],[12]. It can be created manually or automatically. The main aim of ontology is to provide knowledge about a specific domain in a way that understandable for the computers and developers. Ontology can be used in a different aspect of information retrievals, such as

semantic web, query expansion, text classification, and text clustering and text summarization [12], [20].

Ontology improves information retrieval efficiency and effectiveness by defining the common terms or concepts that used to describe and represent the area of knowledge. It provides the well-defined meaning of each term or concepts with their desired semantic relationships[27],[20].

To days, there is a lot of growing interest in ontologies for managing data in information retrieval systems. In fact, ontologies provide terms semantic relations to understand the meaning of data. They are broadly used in all information retrieval systems to overcome the problems caused by the term or keyword-based indexing[28],[29]. Ontologies are used to represent knowledge in a conceptual manner that can be distributed among various applications. Every information item represented by ontology is at a conceptual level rather than focusing on a keyword or individual keyword.

Conventional IR approaches are representing documents as vectors of term weight. Such representation uses a vector with one component for every significant term that occurs in the documents. This representation has several limitations, like that of; different vector position is allocated to the synonyms of the same term. In this case, there is an information loss because the importance of a determinate concept is distributed among different vectors components. The other problem is that the size of a document vector has to be at least equals to the total number of words used to write the document. Every time a new set of terms is introduced (which is a high-probability event), all document vectors must be reconstructed. In order to solve this problem, the ontology-based approach is proposed.

As described in [29],[8] ontology-based information retrieval system suffers some problems such as the absence of some terms in the ontology. This is to mean that if particular terms that are related to specific domains (like bio-medical, mechanical, business, etc.) Are not defined in the machine-readable dictionary which is used to define the concept based version of the documents, at this time, there is a loss of information that can affect the final retrieval results.

2.3.2. Frequency-Based Approach

Frequency-based semantic compression is based on word cumulative frequencies and information stored in semantic relationships. Different techniques are used to apply frequency-based semantic compression approach in IR system.

2.3.2.1. Term Frequency

Term frequency is simply defined as the number of times the term appears in the document. Hence, it measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term could appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (the total number of terms in the document) which is used as a way of normalization. This actual count is normalized to prevent a bias towards longer documents to give a measure of the importance of the term ti within the particular document dj[24], [30]. Thus, the normalized term frequency is used in term semantic compression by computing the cumulated term frequency for each term and this information is stored in a semantic relationship. From two or more terms that are synonyms to each other, the one with the highest frequency is selected as a more general term and the remaining synonyms terms are replaced by it[11],[10]. However, the main problem with the term frequency approach is that it gives more priority for the frequent term and scales down the rare terms, which are empirically more informative than the highly frequent terms. The basic idea in the state of art indicates that a term that occurs frequently in many documents is not good discriminator and in contrast to this, the term that occurs rarely in a document is good content discriminators.

2.3.2.2. Term frequency-inverse document frequency

Term frequency-inverse document frequency is a statistical measure that used to evaluate how important a term is to a document in a collection or corpus. While computing the tf of all terms, they are considered as equally important. However, there are certain terms, which appear many times but have little importance in the document, especially, regarding the content of the document. Therefore, the most frequently occurring term is not a good keyword to distinguish relevant and non-relevant documents[23],[24]. Rarely occurring terms is a good term to distinguish relevant documents from the non-relevant documents in the corpus. So, that an inverse document frequency factor is used to diminish the weights of terms that occur very frequently in document collections

and increases the weight of terms that occur rarely. Inverse document frequency (IDF) is used to reduce the weight of commonly used words and increases the weight of words that are not used very much in a collection of documents. If the term is occurring in all document collections, the resultant IDF weight of the term will be zero[24],[31].

Typically, the tf-idf term weighting scheme is composed of two factors: such as normalized term frequency (tf) and inverse document frequency (idf) factor. Term frequency is the number of times the word or term appears in the document divided by the total number of the words in that document. Inverse document frequency (idf) is a measure of how much information the term provides within the document. That is dealing with whether the term is common or rare across all documents. It is computed as the logarithm of the number of the documents in the corpus divided by the number of documents containing the term[1], [4].

The importance increases proportionally to the number of times a word appears in the document but it is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given to a user query[32].

2.3.2.3. N-Gram Based Approach

An N-gram is a character sequence of a length N that extracted from a document. Typically, N is fixed for particular documents and the queries made against that corpus. N-gram is a probabilistic model that used to predict which word is appropriate to come after a given N-1 words sequence in a given text [33],[34]. Based on the character sequence it extracts at a time, N-gram is classified as a unigram, bigram, tri-gram, tetragram, and pentagram where N ranges from 1-5 respectively[33], [34].

The concepts of N-gram have been used in many areas such as; spell checker, string searching, prediction, speech recognition, handwriting recognition, machine translation and information retrieval [33], [9]. From all these applications of N-gram, we are focusing on application of N-gram in information retrieval because of its relatedness to our work. N-gram is used in information retrieval for indexing. Based on the feature it extracts from a text document, N-gram approach is categorized as character-based N-gram and word-based N-gram [34],[9]. In character based N-gram, a set of n consecutive characters are extracted from words. In this method, the value of n is

2 or 3 that corresponds to bi-gram and tri-gram respectively[33], [34]. For instance, for a word "COMPUTER" the bigrams are given as; *C, CO, OM, MP, PU, UT, TE, ER, R* and trigrams **C, *CO, COM, OMP, MPU, PUT, UTE, TER, ER*, R**. Where * represents a padding space. There are n + 1-bigram pattern and n +2 trigrams pattern for extraction of n character in a word. Using character based N-gram in IR for indexing has both advantages and disadvantages. The advantage is; it is very resistant for countering problems regarding grammatical issues and printing errors of the documents. It does not need language preprocessing such as stemming and stop word elimination. Beside this, character based N-gram has its own limitations. The first limitation is creation of an incorrect index that is semantically meaningless, the indexing becomes massive as it is not controlled and managed. As a result, finding the user`s request among indexes is time-consuming, this makes problem in the speed of retrieving. The other limitation is; the indexed words can include overlapped substrings of conclusive words, which cannot be reflect the conceptual information of the text[9], [35], [8].

Word-based N-grams are a sequence of n consecutive words that extracted from the text. Word level N-gram models are quite robust for modeling language statically as well as for information retrieval without many dependencies on language [33], [34]. The generic equation for probability of finding wn word after a sequence of $w1n-1$ words is given by:

P ($wn/w1\ n-1$)

However, it is not easy to compute the probability finding a word after a long sequence of preceding words.

Term based or word based N-gram uses term weighting model to extract every terms or words in the documents and queries. It emphases on studying each of the words unigram existing in the documents and queries. then, it uses the term with higher frequency in the documents and user`s queries. This technique reduces the system precision rate because N-gram can be extract terms that have no semantic relation with user query term. As a result, non-relevant document list might be returned to the users. This is the main problematic issue in modern information retrieval. So, to solve this problem, N-gram based semantic compression technique is suggested. N-gram is applicable in semantic compression by using local context analysis on concepts in which the concepts are defined as a single noun, two adjacent nouns or three adjacent nouns instead of using

keywords or terms. The weight is calculated for each concept in the local passages and the top N concepts are used for indexing and later used for query expansion[36],[8],[37].

2.3.2.4. Thesaurus-Based Approach

There are a number of studies, which have been conducted on a thesaurus for different languages. From these studies, the definition of thesauri is vary based on the purpose of the thesaurus. However, many scholars define thesauri based on two perspectives. The first one is based on its function and the second one is based on its structure. Based on its function, the thesaurus is a terminological control tool used in translating from a natural language of a documents, indexers or users (searchers) into a more controlled system language and Based on its structure, the thesaurus is a controlled and dynamic vocabulary that consists of semantically related terms which cover a specific domain of knowledge[38],[31],[39]. The thesaurus is tools that permit both the indexer and researcher to use the same terms to describe the same subjects or concepts, used in information retrieval to enhance the performance of information retrieval. It also improves a user`s ability to find the information that they are looking for quickly and easily[40],[39],[41]. By doing so, it supports the indexing, retrieval, organization and navigation of information. The relationships, which exist in the thesaurus, guide users to more general or to concepts that are more specific by allowing them to navigate through the vocabulary to choose appropriate terms for their contents[42],[10]. This navigability of thesaurus makes it much more useful than a simple controlled list of terms as it permits a user to browse a subject domain. The semantic relationships which exist in the thesaurus are categorized as associative (related terms: RT), hierarchical (Broader, or Narrower: BT/NT) and equivalent (equivalent: USE/UF)[43],[44],[8].

Associative relationship (related terms: RT) is a relation existing between terms that belong to concepts that are much related to each other but not the same. In this semantic term relation type, a hierarchical representation is impossible and it is cannot contain the equivalence set. These types of semantic term relation are very complex and difficult to classify each term to their respective cluster semantically[44]. Thus, it is commonly exhibited by the manual construction of a thesaurus approach. In addition to this, the relative value of these relationships for information retrieval is still not clear. Because identifying such types of semantic relationships using automatic methods is difficult. Especially, exploiting the relationships between the terms using statistical relation among a term in the collection of documents is not correctly represent the semantic similarity

between the terms or concepts. As a result, associative term relationships are rarely used in automatic thesaurus constructions[44],[8].

The hierarchical semantic relation (Broader, or Narrower: BT/NT) is represented using the broader term (BT) and narrower term (NT) relationships[44]. This terms relation is used to indicate the relationships between the concepts in the thesaurus by showing which concept contains or contained by other concepts. In hierarchical relationships, BT corresponds to the terms that act as a super-ordinate whereas NT corresponds to subordinate in the classification [44], [8], . It is used to direct the user to a concept that is higher up in the thesaurus hierarchy and therefore, the broader or more general concept than the one that they are seeking will display for the user. Whereas the NT (narrower term) direct the user further down the hierarchy of the thesaurus to more specific terms[44].

The equivalence relationship (equivalent: USE/UF) is called synonymous relationships and it is used to indicate the semantic equivalence of terms. It is used to distinguish preferred terms and their synonyms. The synonym is a term that has the same meaning or that covers the same concepts as another term in the specific domain. These include also variations of spelling, combinations of terms and singular and plural versions[44]. One of these terms is used as the preferred term and the remaining terms become non-preferred terms. These non-preferred terms will assist users in navigating to the preferred terms. In an equivalent relationship, USE and UF are used to indicate the relationship between preferred and non-preferred term. USE relation is used as (A USE B) where as UF (USER_FOR) relation between A and B is described as (A USE_FOR B). It can only occur between preferred and non-preferred terms. It does not exist between non-preferred terms as well as it does not occur between a term and concepts because, the term that represents the concepts cannot represent the single term [44],[8]. There are two major approaches to thesaurus construction: the manual thesaurus construction and automatic thesaurus construction [44],[8],[41].

1. **Manual thesaurus construction**

Manual thesaurus construction is a highly conceptual and knowledge-intensive task and labor intensive[8],[42],[44]. In order to develop thesaurus manually, several steps have to be taken before the final product is ready to use. First, define the boundary of the subject area. In this step, define the central subject area and make a decision about which subject area should be covered

and in what subject area more emphasis should be given. Then the general subject areas are divided into sub-subject areas, which are limited to a specific domain. Once the subject area is definition is completed, the next step is gathering all literature related to each subject area, which needs to be indexed. The thesaurus is constructed based on the literature that collected from a different domain. These literature include a variety of resources such as textbooks, handbooks, articles, scientific journals, existing thesauri and relevant vocabulary system [8], [42],[44]. In order to have a deep understanding of each document contents, every expert and potential users of the thesaurus should be involved in this step. After identification of terms from each sub-area is completed, the following step is analyzing each terms relationship with its vocabulary including synonyms, broader terms, narrower terms and definition as well as a scope note[44] . After the relationship between terms is identified, the term and their relationships are organized into a hierarchical or associative structure possibly within their respective sub-areas. Once the organization of thesaurus structure is over, the entire thesaurus will be reviewed to examine the keywords or terms consistency. In this section, the developers identify which terms represent a single concept and which terms represent compound concepts. This is helpful to reduce the size and complexity of the subject area.

2. Automatic thesaurus construction

Automatic thesaurus construction is simple when compared to that of the manual construction thesaurus. There are three basic approaches to construct thesaurus automatically[8],[44],[30]. The first approach is designing the thesauri from document collections items, the second approach is merging the existing thesauri and the third one is user-generated thesaurus[44].

i. From a collection of document items

In designing thesaurus from a collection of documents, the collection of documents is used as the source to identify the most informative terms (words and phrases) for actual thesaurus construction. The important terms and the significant terms relationships are identified by applying statistical procedure. To construct thesaurus automatically from the document collections items, several steps are involved. The first step is to identify the appropriate documents collection. The only loosely stated criteria are that the collection should be sizable and representative of the subject area. The next step is to determine the required specificity for the thesaurus. If high specificity is needed, then the emphasis will be on identifying precise phrases; otherwise, more general terms

can be sought and terms can be selected from titles, abstracts, or even the full text of the documents if available[8],[44].

ii. Merging Existing Thesauri

This approach is used when two or more thesauri for a given subject exist that need to be merged into a single unit[8]. In this case, there are predefined thesauri in some given subject area then one can benefit by merging that thesauri rather than constructing the thesaurus from the scratch. This method is an actual manual method because there is no any activity that undertaken to construct the thesauri automatically rather than linking to existing thesauri. By doing so, it is possible to be more efficient than producing the thesaurus from scratch. The only problems that should be under consideration while merging the two or more thesauri are, the merger should not violet the integrity of any components of the thesaurus[44]. Because the thesaurus can be constructed for a specific domain. So that, the integrity between two or more thesauri should remain.

iii. User-Generated Thesaurus

This approach is based on tools from expert systems in which the expert system is used to build automatic thesauri using information gained from users search. In this approach, the users of IR systems are aware of and use many term relationships in search strategies long before they find their way into thesauri[8], [44]. In such case, the intention is to capture the thesaurus knowledge from user`s search. This approach utilizes mainly expert system. Generally, thesaurus provides the precise and controlled vocabulary that supports efficient and effective indexing, and retrieval process in IR[44]. In both indexing and retrieval, a thesaurus helps the users to select the appropriate term that matches their information need.

Therefore, in this study, we use the equivalence relationships (set of synonyms) to handle words that have the same meaning and represent them by one word that is common for all synonym words. The reason why we select this approach is that, there is no previously prepared standard thesaurus for Afaan Oromo and it is difficult to develop the complete thesaurus for the language.

2.4. Related Works

A various research work has been done in the area of information retrieval for Afaan Oromo using different types of techniques and approach by different scholars. However, there is no work has been done in the past to integrate semantic compression in information retrieval system in order to improve the performance of Afaan Oromo text retrieval. However, there are some works that have conducted by different researchers in a different language by applying a number of approaches that integrates semantic compression in information retrieval system in order to increase the performance of information retrieval system of respective languages. So that in this subsection we have summarized the review of some related works for different languages in which different approaches of semantic compression were applied to enhance the performance of information retrieval system.

2.4.1. Semantic Compression Based IR for Foreign Language

Dariusz Ceglarek et al [11] have conducted the research on semantic compression for specialized information retrieval system. The study was conducted in the Polish language. The main objective of the study is to examine the utilization of semantic compression technique in IR system. In addition to that, to check that if it reduces the number of vector space dimensions significantly without the deterioration of the results. To do this, the experiment was conducted based on Seneca net, which is a semantic net that uses term frequency. To extract term frequency, they used the brown university standard corpus of present-day American English, which is readily available for anyone. For document clustering procedure, a two-sample set of documents was used. Document pre-processing task was applied.

To evaluate the successfulness of utilizing semantic compression in information retrieval, the experiment has been done on two sample sets of documents that contain 780 and 900 terms each. All documents were in Polish and collected from six different knowledge area such as astronomy, ecology, IT, culture, law, and sport. All collected documents were manually labeled with its respective categories by researchers. For comparison purpose, all documents were clustered 8 times and the prototype system is evaluated without using semantic compression technique and using semantic compression technique.

The results of the experiment show that semantic compression method reduced the number of concepts and vector dimension in a significant way. Thus, using semantic compression techniques in information retrieval can enhance the performance of IR system.

Darius [10] conducted a study on semantic compression for text document to compare the global semantic compression and domain-based semantic compression technique. The study was conducted using term frequencies that combine data from two sources, which is a frequency dictionary and concept hierarchy from the semantic network. Global semantic compression uses term frequencies from frequency dictionary that belongs to specific domain and concept hierarchy from the semantic network to compress term meaning in a text. The domain-based semantic compression technique uses domain-specific term frequencies from frequency dictionary to compress the term meaning in the documents. Thus, the author developed an experiment to check in which technique semantic compression yield better results when applied to a specific document. For the experiment, document corpus was collected from various domains; such as business, crime, culture, health, politics, sport, biology, and astronomy. All of the documents were written in English. The evaluation experiment was performed by grouping the whole documents into two cluster, one cluster is semantically not compressed and the other is semantically compressed. To verify the results, all documents were initially labeled manually by the researcher. The first run was done without semantic compression mechanism and from 25000 identified concepts the system only returns 50 of them. Secondly, the semantic compression was applied and it returns 12000 first, at next iteration 10000, 8000, 6000, 4000 respectively in the different execution phase. From the experiment results, semantic compression is more effective when a text domain is identified and an appropriate domain frequency dictionary is used to perform the process. Based on the previous experiment result further experiments were conducted to evaluate global semantic compression and domain-based semantic compression. The experiment results show that domain based semantic compression yield better in natural language tasks. Because the domain frequency dictionaries more reflect the characteristics of natural language. As a conclusion, domain-based semantic compression use concept that is less ambiguous that allows for better understanding of the text. Whereas global semantic compression leads to a more generalized concept that may have very broader meaning.

E. Mohammed et al [21] were developed semantic Arabic IR model based on the use of ontology. The proposed model has two main parts: semantic inverted index construction and semantic query processing and retrieval. In inverted index construction, the first collection of documents is built and index creation algorithm is used to manipulate each document of the collection by extracting and preprocessing its contents one after the other. The preprocessing operations on this phase include stop word removal and stemming. From the resulted set of words in the previous phase, the reference concept is constructed from ontology link between these set of words. Finally, each word is stored in the form of [words, reference concepts, DOCId] with their reference concepts. DOCId uniquely identifies to which document the reference concept and words are belonging. For semantic query processing, the same algorithm is used for document collection preprocessing was used for query processing and a set of the query term is extracted from the user query. Then, matching the query term against the semantic index is done. Finally, the ontology filters the RCs that have a relation with RCs of the query term and return to the user. By doing so, they have represented the relation and the meaning of each word in the index based on its context.

The results show that the new approach enhanced the precision and made it 100% accurate in all cases. However, it is time-consuming. In a semantic model, automatically detecting reference concepts (RC) for a query from the ontology is achieved. In addition to that, the semantic vector space model implemented and compared its performance with traditional IR system. Both systems are evaluated by the same query prepared by the researchers. Based on the query given to it, the traditional model retrieves 987 documents from 1000 document in the collection for all queries. Whereas the semantic model retrieved 782 document for all queries. This indicates, in the traditional model, many documents differ in the same queries domain. Finally, the achieved precision for the top 10 retrieved documents by the traditional model is 66% and 87% by semantic model respectively. This shows that the semantic vector space model enhances the ranking process as well as the precision of the returned result.

2.4.2. Semantic compression based IR for local languages

There are about 80 local languages in Ethiopia. Therefore, we need to conduct reviews to find out the works that have been done previously by different scholars for local languages. From the reviews that we have been done, few works have been done for the local language.

Abey Bruck [45] conducted a study on Enhancing Amharic IR system based on statistical co-occurrence techniques. The main aim of the study was to design a good system that can enhance the recall of the Amharic text retrieval without affecting the system precision. To achieve this, they integrate query expansion model with statistical co-occurrence technique in Amharic IR system in order to find the words that have similar meaning with user queries and retrieves relevant documents that satisfy user information need. This method generates synonymous by expanding the terms for a query term based on the index terms co-occurrence information. It explores the presence of terms with another term to decide whether they have the same or meaning. This is used to handle if the query term is synonymous and polysemy terms.

The researcher has evaluated the proposed approach in order to compare the performance of the new model with based IR system for the Amharic language. To evaluate the proposed model, precision, recall, and F-measure were used. From evaluation of the implemented prototype, they have got a better result using statistical co-occurrence method over bi-gram based information retrieval system by achieving the precision of the relevant document decrease by 4%, whereas the recall of retrieving the relevant document was improved by 6% and the F-measure was increased by 2%. Therefore, as a conclusion the statistical co-occurrence method shows better improvement than the bi-gram method with the result of recall 6% and F-measure 2%. One of the recommendations given in this research work is designing the hybrid approach of the bi-gram and statistical co-occurrence based query expansion for Amharic text retrieval system can more the performance of Amharic IR system.

Tewodros Hailemeskel [6] developed a text retrieval prototype using latent semantic indexing approach with single value decomposition (SVD). The research is conducted to solve the problem of synonyms and polysemy found in Amharic text, which decrease the performance of Amharic text retrieval system. The experiment was conducted to evaluate the performance of the system. The experiment is conducted on 206 Amharic documents and 25 queries were prepared to test the system. From the experiment result, method recorded average precision of 71% and the curve of method existed above the standard vector space approach.

Alelign Tefera [46] conducted a study on the Automatic construction of Amharic semantic network (ASNET). WordSpace model was used for of semantically related concepts from Amharic corpus. The process of relation identification among these concepts utilizes the extracted text pattern. They

have designed the system that handles the 'part-of' and 'type-of' relation between the concepts. Two different datasets are used for experimentation. The first is one, which collected from the information center that composed of 1064 news items and all the news items are tagged with part of speech. This dataset is used for the extraction of concepts in the corpus used for the study. The second dataset was collected from Ethiopian news agency and it is composed of 3261 untagged free text news items. This dataset is used for the extraction of the possible frequency of concepts that are extracted from the tagged dataset. They test system prototype in three different phases with different datasets collected from Ethiopia news agency and information center. The accuracy of the system to extract pairs of having 'type-of' and the 'part-of' relation is 68.5% and 71.7% respectively.

Tewdros Abebaw [8] conducted the study on the Amharic language in order to enhance the effectiveness and efficiency of Amharic text retrieval system using a thesaurus based semantic compression approach. The author was initiated by the problem of vector space model that its inability to handle synonyms and hypernyms terms in Amharic text retrieval and high dimension representation of terms in the space. This decreases the effectiveness of Amharic IR system. The main objective of the study was to handle the problems related to synonyms and hypernyms words in Amharic text retrieval that happen during search time using thesaurus based indexing. For the study, 300 different Amharic text documents were collected from news media. Once the corpus is collected, linguistic pre-processing techniques were used for identifying the content-bearing terms that used for indexing and searching process. Linguistic pre-processing includes; tokenization, normalization, stops word removal and stemming subtasks. Since there is no upper case and lower case in Amharic writing system, instead of case conversion, the researcher use normalization subtask to replace the character that represents in different symbols but conveys identical meanings. List of stop words were removed from Amharic documents collection. The stemming part of the pre-processing is done to convert Amharic variant words to its root or stem. Then, the identified terms were weighted. The tf*idf term weighting technique is used for measure the importance of the terms in the document. Then after, for each weighted terms, they construct thesaurus (synonyms of each term) manually from Amharic dictionary (Amharic mezgeba kaliat). Finally, thesaurus based inverted file is created for document indexing and searching purpose.

To evaluate the prototype system, recall and precision were used. The system is tested based on 300 documents collected from different disciplines of Amharic documents and 15 queries are formulated by the researchers. From the experiment, thesaurus based semantic compression technique shows the best IR effectiveness with 70.77% precision and 77.13% recall. Using semantic compression technique, index terms are reduced by 61% when compared with the vector space model. Even if a good result is achieved from the experiment, there are some challenges that greatly affect retrieval performance. The first one is that the stemmer is unable to handle the words variants, doesn`t remove infixes as well as unable to handle the ambiguity of word in the Amharic language. The other challenges are there is no standard thesaurus, standard Amharic corpus that can be used for experimentation. This greatly affects the retrieval performance. Therefore, the researcher believes that the performance of Amharic IR system can be enhanced if ontology-based semantic compression is designed, standard Amharic thesaurus is constructed and the stemming algorithm is improved. This research work is conducted on text document only. However, other types of information that exist in video, audio, graphics, and pictures format are open for further study.

Gezahegn Gutema [4] was developed "Afaan Oromo text retrieval system" at Addis Ababa University in 2012. The main objective of the study is to develop an information retrieval system that can enable to search for relevant Afaan Oromo text document from the corpus. For this research work, the corpus is prepared by collecting 100 different Afaan Oromo text documents from different news Media. This collected document includes different subject areas like politics, education, culture, religion, history, social, health economy, and other events. The prototype system is implemented as it has two main components, which is indexing and searching. To accomplish the indexing part, various text pre-processing tasks were performed on the documents to make them suitable for indexing. The text pre-processing tasks involve activities like tokenization, normalization, stop word removal and stemming. In the study, tokenization is used to break down the whole documents into a list of the sequence of tokens based on white space. Following tokenization, normalization is used to convert all the characters in the list of tokens into lower case and used to remove different punctuations marks and symbols except "`" which have a different meaning in Afaan Oromo. Stop words were manually prepared by the researcher and those lists of stop word are checked against a normalized list of normalized tokens, if the token is not a stop word, it will be stemmed by stemming algorithm. Else it will be dropped. The final

activity of the pre-processing task, which is stemming is used to convert different words variants into their common root or stem. To do this, rule-based stemmer developed by Debela Tesfaye and Ermias Abebe (2010) which was based on porter stemmer algorithm is used. Once every term in the corpus is stemmed, the term weighting technique is applied to determine how the term is important to describe the document content. For this, tf*idf term weighting scheme is used. The index file structure used in this research work is the inverted index file structure. Cosine similarity is used to measure the similarity between document and query in which the search strategy is based on the vector space model. The prototype system is implemented using python 2.7.1 script.

The designed model is tested using 100 Afaan Oromo text documents and 9 queries to make relevance evaluation. To measure the performance of the designed model, precision and recall are used. From experiment results, Afaan Oromo text retrieval using vector space model registered an average performance of 57.5% precision and 62.64% recall. Finally, the researcher believes that the performance of the system can be increased if the stemming algorithm is improved, standard test corpus is used, and ontology and thesaurus are used to handle polysemy and synonymy words in Afaan Oromo language.

In reviewed related work, several IR systems have been developed for both foreign language and local languages using different approach and techniques to enhance the performance of IR system for the respective language. For local languages like the Amharic language, many scholars attempt to improve the effectiveness of Amharic text retrieval system. Whereas for Afaan Oromo, there is limited work to improve the effectiveness of Afaan Oromo IR system. The work done by Gezahegn Gutema is based on the vector space model. However, the use of vector space model couldn`t handle term mismatch and unable to retrieve a relevant document from Afaan Oromo text corpus when different words with the same meaning are used for searching. Thus, our work is proposed to solve this gap of the reviewed related works. Accordingly, we used the semantic compression approach using thesaurus based indexing technique to handle Afaan Oromo synonymous words.

CHAPTER THREE

MORPHOLOGY OF AFAAN OROMO

3.1. Introduction

In the second section of the chapter, the feature of Afaan Oromo language is discussed in details. The following sub-section; overview of Afaan Oromo, Afaan Oromo alphabets, Afaan Oromo sentence structure, grammar and morphology is discussed.

3.2. Overview of Afaan Oromo

Afaan Oromo is a member of the Cushitic branch of the Afro-Asiatic language family and spoken predominantly in Ethiopia [4],[47],[48],[49],[14]. The language is spoken as a first language by more than 40 million Oromo people in the region and neighboring peoples in Ethiopia (2015 census).

It is the official working language of the Oromia regional state, which is the largest state among the eleven regional states of the country [17],[13]. In addition to this, the language is used as a medium of instruction in elementary schools (grade 1 to 8), the regional colleges and as a school subject in elementary (from grades 1-10) and secondary schools (grades 11 to 12). It is also taught as a major field at a BA and an MA level in six universities of the country [4],[50].

Apart from this, the language is also used in other African countries like Kenya, Somalia, Sudan, Tanzania, South Africa, Egypt, Libya, Eritrea and other parts of the world like the US, specifically Minnesota, Europe, Australia, and Saudi Arabia. This makes it African's the fourth most widely spoken language after Hausa, Arabic, and Swahili [15].

3.3. Afaan Oromo Alphabets

Afaan Oromo is written using a Latin alphabet called "**Qubee"** which formally adopted from Latin script and became the official script of Afaan Oromo since 1991[4],[17],[51]**.** The language has 33 characters; from these 26 of them are consonants. Among 26 Oromiffa consonants, three consonant letters are borrowed from English language. These are P, V and Z whereas seven of them are made of two consecutive consonants (CH, DH, SH, NY, PH, TS, ZH) that are used together as one consonant or a combination of two characters that serves as a one character to give

a new sound in the language. Like that of English language, Afaan Oromo also has five vowels which are: a, e, i, o and u[13],[14],[50].

In Afaan Oromo writing system, the geminated consonants and long vowels are represented by double letters[14]. In addition to seven compound symbols, not all the 26 letters are corresponding with their English sound representation as shown in the table below.

Qubee		IPA	Qubee		IPA	Qubee		IPA
A	a	/a/	L	l	/l/	W	w	/w/
B	b	/b/	M	m	/m/	X	x	/t'/
C	c	/č'/	N	n	/n/	W	w	/w/
D	d	/d/	O	o	/o/	Z	z	/z/
E	e	/e/	P	p	/p/	CH	ch	/č/
F	f	/f/	Q	q	/k'/	DH	dh	/ɗ/
G	g	/g/	R	r	/r/	NY	ny	/ñ/
H	h	/h/	S	s	/s/	PH	ph	/p'/
I	i	/i/	T	t	/t/	SH	sh	/š/
J	j	/ǧ/	U	u	/u/	TS	ts	/s'/
K	k	/k/	V	v	/v/	ZH	zh	/ž/

Table 3.3-1: List of Qubee Afaan Oromo

List of Qubee Afaan Oromo[14],[4].

3.4. Afaan Oromo Morphology

Morphology is a branch of linguistics that studies and describes the internal structure of the words and how it formed in a language[47],[50],. It is a study of the way in which words are built up from smaller meaning-bearing units, that is called morphemes. A morpheme is often defined as the minimal meaning-bearing unit in any language. Morphology is classified into two type, which is derivational and inflectional morphology. Derivational morphology deals with a combination of the words stem with a grammatical morpheme that produces different word class. Most probably, it is difficult to predict the actual meaning of the word. By affixing the existing derivational morphemes, forming the new lexemes is possible. If the attached suffix is detached from the word, the meaning of the word can be altered. Inflectional morphology deals with the combination of a word stem with a grammatical morpheme in the same word class. In inflectional morphological,

inflectional morphemes are morphemes that serve as a pure grammatical function that never creates a new word[50].

Morphemes are the smallest and building blocks of linguistic unit, which has meaning or grammatical function. Accordingly, words are composed of these smallest and building block units of morphemes[50].

Morphemes in Afaan Oromo

Morpheme is a smallest unit which are individually meaningful elements in the utterances of the language[14]. Morphemes in Afaan Oromo are divided into two broad categories: free morphemes (dhamjecha walabaa) and bound morpheme (dham jecha hirkata). Free morphemes are morphemes that can convey full meaning without attaching itself to other morphemes. It provides the lexical meaning of the word and it can occur as a word independently. Whereas bound morpheme is attached to other morphemes to convey the meaning. In case, it adds additional meanings of varies kinds to the word. It does not occurs independent as a word by itself. For instance, in ***malamaltummaatiin*** 'by corruption', ***/-tiin/*** is a bound morpheme and ***malamaltummaa*** 'corruption' is a free morpheme[51], [50].

Bound morphemes

In Afaan Oromo, based on their contents there are two types of bound morphemes[50], [51].These are: Bound root (hundee hirkataa) and affix (fufii). Bound roots are morphemes that provides the most real and factual role to the word meaning. For example, the word like **burqaa, burqe, burqitte, burqisiisse burqisiitte, burqisiissan, burqisiissuu, burqisiisicha, burqisiifne and burqisiifna** has a bound root **burq-.** Most of the root words in Afaan Oromo are bound morphemes[51]. Affix is a bound morpheme that attached to the root. Affix is categorized into four division based on their location in the words. ***fufii dura*** "prefix**",** ***fufii jidduu*** "infix***", fufii naannee*** "circumfix", and ***fufii duubee*** "suffix". These can be divided into prefixes, suffixes, infixes, and circumfixes. The prefixes precede the stem or root, suffixes follow the stem or come after stem, infixes inserted in the stem and circumfixes do both that means, it can come before the stem and after the stem. The other classification of Afaan Oromo affixes is based grammatical functionalities and the types of words class they change, affixes is classified into derivational affixes (fufii yaasaa) and inflectional affixes (fufii hortee)[50]. Derivational affixes are an affixes

that attached to the root word and change the original meaning as well as the category of the word. Some of the derivational affixes in Afaan Oromo are hin- eenya, -ummaa etc. But, inflectional affixes in Afaan Oromo indicates grammatical formation such as: Tense(yenna/henna), verb(xumurtoota) numbers (dannuu), persons (rammadii kooniyaa) and possessions (abbummaa)[50]. In all these case, inflectional affixes do not change the meaning and category of the word that they attached. Some of inflectional affixes in Afaan Oromo are: -oota, -dhaan, -lee, -waan, -awwaan, -dhaaf, -fi, -tuu, -te, -e, -an, -f, and etc.

Free morphemes

Afaan Oromo free morphemes are classified as: lexical morphemes (dhamjecha hiika) and functional morphemes (dhamjecha tajaajilaa)[50],[51]. Lexical morphemes are free morphemes that have their own one content meaning. For instance, the word **mana** (house) can be used as subject or object in a given sentence but its meaning is remaining unchanged. However, functional morphemes recognize some kinds of grammatical role as a result it conveying little meaning of their own. Functional morphemes generally realize some kind of grammatical role, conveying little meaning of their own. Most of the time it is used for specify relationship between others morphemes class. Example, kana, sana, tana, tun, kun, sun, and etc. are functional morphemes in Afaan Oromo.

3.5. Afaan Oromo Sentence Structure

Both Afaan Oromo and English language use Latin script in which the majority of the letters used in both languages are similar. However, they are different in sentence structure. English language use subject–verb–object (SVO) agreement in sentence forming[14]. Afaan Oromo uses subject-object-verb (SOV) agreement in sentence forming. Subject-object-verb is a sentence structure, in which the subject comes first, the verb comes next and ten object in Afaan Oromo sentence. For instance, in Afaan Oromo sentence “Tolaan saree ajjeese/ijjeese” to mean, “Tola killed the dog”. In this sentence, “Tolaa” is a subject, “saree” is an object and “ajjeese/ijjeese” is a verb. Therefore, Afaan Oromo uses SOV sentence structure. In addition to this, in Afaan Oromo the adjectives follow a noun or pronoun that they modify while in English language adjectives typically precede the noun or pronoun. For instance, “Tolaan bareeda dha” to mean, “tola is a handsam”. In this sentence, “Tolaa” is a noun and “bareeda” is an adjective follows tolaa (noun).

3.6. Afaan Oromo Grammar

In linguistics, grammar is the set of structural rules governing the composition of clauses, phrases, and words in any given natural language. The term grammar refers to the study of rules, and this field includes morphology, syntax, and phonology, frequently complemented by phonetics, semantics, and pragmatics. Like that of another language in the world, Afaan Oromo has also its own syntax and rules that used to structure the language feature[4],[48],[17].

3.6.1. Gender

Like most Afro-Asiatic languages family, Afaan Oromo has two grammatical genders, which is feminine and masculine.

Number

Afaan Oromo has a singular and plural numbers. However, a noun that refers to multiple entities is not obligatorily plural. That is if the context is clear, a formal singular noun may refer to multiple entities like that of nama “man”, or “people” nama torba “seven-man” or “seven people”. Another way of looking at this is to treat the "singular" form as indefinite for the number. When it is important to make a plurality of a referent clear, the plural form of a noun is used[50], [48]. Plurals forms of a noun are formed through the addition of suffixes to the nouns. The most common plural marker suffixes in Afaan Oromo are: {–oota}, {-wwaan}, {-een}, {-o(o)ta}, {-lee} and {-yyii}. The final vowel is dropped before the suffix and the western dialects, the suffix becomes –ota following a syllable with a long vowel: mana “house”, manoota “houses”, hiriyaa “friend”, hiriyoota, “friends”, barsiisaa “teacher”, and barsiisoota “teachers”. Other common plural suffixes are including: (-wwaan), -een and –(a)-an and the latter two may cause preceding consonants to be doubled. For example waggaa “year”, waggaawwan “years”, laga “river” laggeen “rivers”, ilma “son”, ilmaan “sons”[4], [14], [48].

3.6.2. Personal Pronoun (bamaqaa/ gulummoo)

In Afan Oromo, like in other languages, pronoun is a word that is used instead of a noun or noun phrase. They are characterized based on number and gender. For instance, ishee ‘her’, isa ‘him’, isaan ‘they’ are some[15].

3.6.3. Adjective (Ibsa Maqaa)

a. Gender form of adjectives

In Afaan Oromo adjectives can exists as masculine, feminine and neutral [14],[48]. Masculine adjectives used to modify masculine nouns, whereas feminine adjectives are used to modify feminine nouns and neutral adjectives can be used with any noun. All non-neutral adjectives can be made masculine and feminine by attaching an appropriate suffix to the nouns. Masculine suffixes for adjectives are -aa, -aawaa, -achaa, icha and –eessaa. Feminine suffixes include -oo, -tuu, ooftuu, itii, ttii and eettii. Standard morphology rules are applied when attaching these suffixes.

For example:

English Meaning	Masculine	Feminine
Adorable	Jaallatamaa	Jaallatamtuu
Beautiful	bareedaa	Bareedduu
Fast	Si`awaa	Si`ooftuu
Sweet	Mi`awaa	Mi`ooftuu/mi`ootuu
Small	xiqqaa, xinnaa	Xinnoo, Xiqqoo
Small	diqqaa/diqqicha	diqqoo/diqqitii
Black	Gurraacha	Gurraattii
Poor	Hiyyeessa	Hiyyeettii

A neutral adjective uses the same form for both masculine and feminine nouns. For example, the adjective “adii” which means “white” is used the same form for both genders.

b. Plural form of adjectives

When adjectives are used to modify a noun, typically the noun remains in singular form and number is shown by adjectives only. Plural adjectives are formed by repeating the first syllable[48]. For example:

English meaning	Singular	Plural
White	Adii	Adaadii
Beautiful	bareedduu	babareedduu

Dry	Gogaa	Gogogaa

Some masculine adjectives will change their ending to suffix "-oo" when pluralized. However, some of them do not repeat the first syllable as a plural marker.

For example:

English Meaning	Singular	Plural
Knowledgeable	Beekaa	Beekoo
Strong	Cimaa	Ciccimmoo
Large	Guddaa	Guguddoo/gurguddoo
High	Olaanaa	Olaanoo

c. Adjectives with pronoun

In Afaan Oromo, to express an adjective with a pronoun one can simply use the correct pronoun in front of the adjective. The pronoun used depends on its role in the sentence[48], [14]. For example, "gurraachi kun dansaa fakkaata" which means "the black one looks nice" and "isa gurraacha nan barbaada" to mean "I want the black one" or "gurraacha an ni fedha" to mean "I need the black one".

3.6.4. Adverbs (Ibsa xumuraa)

In the English language, an adverb comes after the verb that it modifies but in Afaan Oromo, the adverb comes before the verbs they modify. There are four major types of adverbs in Afaan Oromo. Those are adverbs of time (ibsa xumura/dhumaa yeroo/yenna), adverbs of manner (ibsa xumura/dhumaa haala), adverbs of place (ibsa xumura/dhumaa bakka/ idoo/addee) and adverbs of frequency (ibsa xumura irra deddeebii). For example: "***ani dafee deema*****"** which means (" I will go quickly"), here the word "***dafee***" indicate in which manner of action he will go", inni **jabeesse hojjate**" which means ("he works hard"). Ishiin/isiin **suuta/laana** deemti/deenti (she walks slowly), Isaan **boru** dhufu (they will come tomorrow), Mana **kooti/kiyyatti** si affeerun/yaamuu barbaade/fedhe (I would like to invite you to my house), Oromiyaa **keessa** laga baayeettu/heddu`utti jira (there are many rivers in Oromia).

3.6.5. Prepositions

The preposition in Afaan Oromo language links a noun, pronouns, and phrases to an action, to another noun, or to other words in the sentence. The prepositions in Afaan Oromo are divided into two categories, which are named as prepositions and postpositions[47]. In Afaan Oromo language, the prepositions come before the noun while the postpositions coming after the noun they relate to. Table 3.5.1: Shows sample list of Afaan Oromo prepositions.

Afaan Oromo Prepositions	***Equivalence in English***	***Afaan Oromo prepositions***	***Equivalence In English***
Gara	*Towards*	*Tti*	*To*
Eega, erga	*Since, from, after*	*Garas, garasan*	*towards*
Haga, hanga	*Until*	*Jala, gajjallaa*	*Under*
Hamma, haga	*Up to, as much as*	*faallaa*	*Unlike*
Akka	*Like, as*	*Karaa*	*Via*
Waa`ee, waan	*About, in regard to*	*Wajjiin, woliin*	*With*
Ergii, kaasee	*Since*	*Keessatti*	*Within*
Manna	*Than*	*Malee*	*Without*
Gidduu, jidduu	*Through*	*Irraa/siqee/hiiqee/fa gatee*	*Far from*
Kanaaf/tanaaf	*Because of, due to*	*Bira dhihoo/dhiyoo*	*Close to*
Alati, malee	*Except*		*Unlike*

Table 3.6-1: Sample list of Afaan Oromo prepositions

3.6.6. Postpositions

Personal pronouns are not used with prepositions. Instead, possessive pronouns are used as personal pronouns. Examples: “Towards me” gara koo/kiyya (tti) [not gara na], “Like us” akka keenya/keenna correct, “about you” waa`ee kee/waan keetti, “According to him” akka isaatti, “About you” waa'ee kee, and etc. Table 3.5.1 shows Sample list of Afaan Oromo postpositions.

Oromiffaa postpostions	***Equivalence in English***	***Oromiffaa postpositions***	***Equivalence in English***
Ala	*Out, outside*	*Itti*	*To, at, in*
Bira	*Beside, with, around*	*Jala*	*Under, beneath*
Booda, duuba	*After*	*Jidduu*	*Middle, between*
Cinaa	*Beside, near, next to*	*Keessa*	*In, inside*
Dur, dura	*Before*	*Malee*	*Without*
Duuba	*Behind, back of*	*Wajjiin, waliin/woliin*	*With, together*
Irra	*On*	*Gubbaa*	*On, above*
Irraa	*From*	*Fuldura/fuladura*	*In front of*
Itti	*To, at, in*	*Gadi*	*Down, below*
Jala	*Under, beneath*	*Ol(i)*	*Upward, above*

Table 3.6-2: Sample list of Afaan Oromo postpositions

Mana keessa “in the house”, irra deebi'i “repeat it”, Yuunivarsitii Haroomayaatti barsiissaa “Teacher at Haromaya University”, mana nyaataa kanatti “at this restaurant”, hanga/haga torban dhufu “until next week”, gammachuu wajjin/woliin “with pleasure”, Keeniyaan Itoophiyaarraa (gara) kibbatti argamti/dhagganti “Kenya is located (to the) south of Ethiopia” etc.

From the examples given above, we notice that postpositions **tti**, **irra, wajjiin/woliin** and **irraa** most often occur as suffixes –tti, -rra and –rraa on the nouns they related to. On other hand, Postpositions take the accusative form of personal pronouns. Examples: “At you” sitti, “From me” narraa, “Under him” isa jala and the like. When an adjective modifies a noun, the postposition follows the adjective as in “nama guddarraa” which means “from the elder man”.

3.6.7. Conjunctions

As that of prepositions are used to links nouns to other parts of the sentence, conjunctions are also connecting the complete thoughts together[47]. Conjunctions come between the two clauses they connect, though “garuu” and “immoo/ammoo” can also come after the first noun or noun phrase

in the second clause. Some commonly used conjunctions in Afaan Oromo are listed in the given table 3.5.3 below.

Afaan Oromo conjunctions	Equivalence in English
Fi [also –f suffix]	And
Garuu, immoo	But
Yookiin[for declarative] moo [for question]	Or
Haa ta`u malee/te`uu malee	However
Ta`us/te`ulle	Though
Kanaaf, kanaafu	So, therefore
Sababin isaa, sababiinsa	Because [its reason]
Akka	So that, in order to

Table 3.6-3: Sample list of Afaan Oromo conjunctions

3.6.8. Punctuations

The punctuation marks are used in both Afaan Oromo and English languages are the same and used for the same purpose except apostrophe. Apostrophe mark (“**`**”) in English shows possession but apostrophe mark in Afaan Oromo is used in writing to represent a glitch (called ***hudhaa***) sound[14]. The “***hudhaa***” punctuation mark in Afaan Oromo has another role in reading and writing system. This is different from the English language. For example, it is used to write the word in which most of the time two vowels are appeared together like that, “***taa`umsa***” to mean (“seat”), “***re`ee***” to mean (“goat”), “***du`a***” to mean (“death”) with the exception of some words like “***har’a***” to mean “today” which is identified by the sound created[4].

3.6.9. Definiteness

In Afaan Oromo, definiteness is a grammatical category that is used to distinguish noun phrases according to whether their reference in a given context is supposed to be uniquely identifiable[15],[47]. Afaan Oromo has no indefinite article like a used in English language. But definite article used in English language like the is indicated in Afaan Oromo by adding ***–(t)icha*** suffixes to noun for masculine nouns and ***–(t)ittii*** for feminine nouns. Before add this suffixes, the final vowel is dropped and the suffix ***–(t)icha*** is attached to masculine nouns and ***–(t)ittii*** suffix attached to feminine nouns. For instance, ***saree*** “dog” ***saricha*** “the dog”, ***nama*** “man”

namicha/namticha “the man”, ***re`ee*** “goat” ***re`ittii*** “the goat”, ***durba*** “girl” ***durbittii/dubartittii*** “the girl” and etc. Making a noun definite is less common in Afaan Oromo than in English language and it is only used for objects that known to both the speaker and listener[48]. A noun can exist in either definite or pluralized form but not both.

In English language, Indefiniteness is marked by “an” or “some”, but in Afaan Oromo, indefiniteness uses the noun alone without modification. For example, the word “***tokko***” which means “one” is used to indicate “a certain” something as the same time the word “***tokko-tokko***” can be used to mean “some”. For instance, ***Kitaaban barabaada***, which means, “I want a book” (any book), ***Kitaaba tokkon barbaada***, which means, “I want a (certain) book***”, Kitaaba tokko-tokkon barbaada*** which means, “I want some books”.

CHAPTER FOUR

METHODS AND TECHNIQUES

4.1. Introduction

In this chapter, the architectural design of thesaurus based semantic compression for Afaan Oromo text retrieval, the components of the architecture and the algorithm to implement the components are presented and discussed in detail.

4.2. Proposed System Architecture

The proposed system architecture for thesaurus based semantic compression model for Afaan Oromo text retrieval consists of two main components, which are indexing and searching. On the indexing side, Afaan Oromo text documents are given to IR system and it organizes the contents of the documents using index structure to improve the searching process. To achieve this, document pre-processing task is performed. The first step is tokenization in which text documents are split into a stream of tokens. The system accepts Afaan Oromo text documents as an input and tokenizes it into a stream of tokens. This is followed by normalization in which all words written in a different case (UPPER, lower and mix of the two) are converted to a similar case format. In our case, all words are converted to lower case format. Because most of the time readers and writers use lower case format. Once this is completed, the normalized list of tokens or words is checked against a list of Afaan Oromo stop words, as it does not stop word. Then, non-stop word tokens are stemmed. For all stemmed words, its respected weight is calculated automatically. Finally, the synonym of each weighted word is constructed and an index is constructed based on thesaurus using an inverted index file structure.

On searching side similar text pre-processing (text tokenization, normalization, elimination of stop word, stemming and term weighting) technique is applied to user query. Then, a similarity measurement is performed by applying a semantic compression algorithm to retrieve and rank relevant documents. Here is the architecture of thesaurus-based semantic compression for Afaan Oromo text retrieval system.

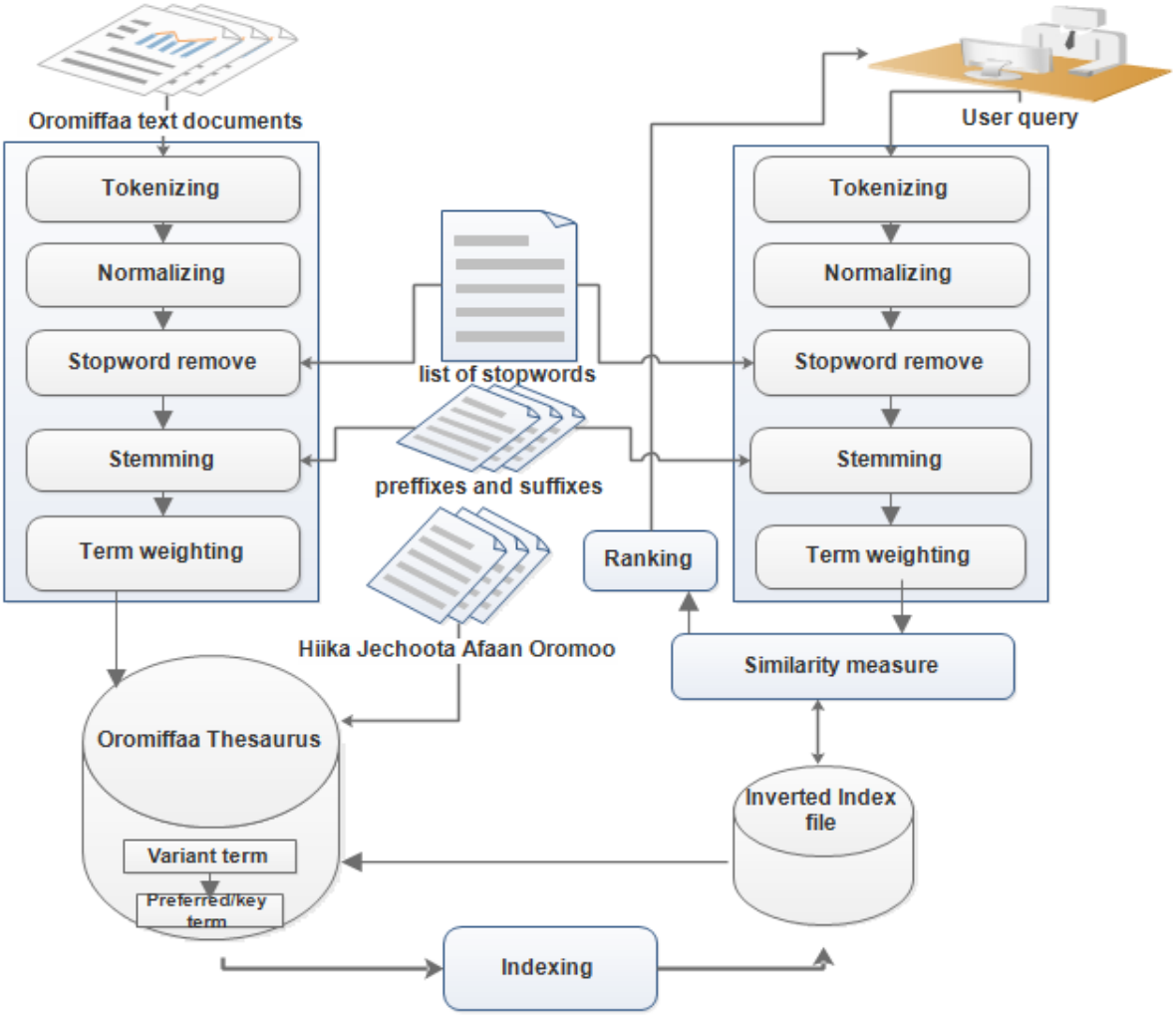


Figure 4.2-1: The architecture of semantic compression model for Afaan Oromo text retrieval

4.3. Dataset preparation and Document pre-processing

Dataset/corpus is a collection of documents that are collected from a various source that used to evaluate information retrieval system after passing through further pre-processing computations. In this study, a corpus was developed by gathering Afaan Oromo text documents from a different source that involves information about politics, social affairs, sports, entertainments, religion, culture, and business. To get these documents and significant information regarding the resource, some domain experts were consulted and some linguistics literature in Afaan Oromo has been reviewed.

4.4. Document Pre-processing

Document pre-processing is the process of applying any type of computation on unstructured raw data and transforms it into a format that more easily and effectively processed in another procedure. After the documents were selected and collected, document pre-processing tasks is applied to convert them into a suitable way to be being indexed and searchable. This document pre-processing procedure includes tasks such as tokenization, normalization, stop word removal, and stemming. All these are discussed as follows.

4.4.1. Tokenization

Tokenization is the process of breaking down of input documents into stream of tokens. These tokens could be individual words like (noun, verb, adverb, pronoun, article, conjunction, preposition, punctuation marks, numbers, and alphanumeric). The chopping of text document into these sequences of tokens is performed without consideration of their meaning and relationship of the tokens. In text document, words or phrases are separated from each other by white space, semicolon, commas, quotes, and periods [30],[4].

Separating words from one another in text is varying from one language to another language. In some language, characters and white space are not used to separate words in the documents. For examples; languages like Chinese, Japanese, Korean and Arabic uses a special character set which does not use white space to separate words in the sentence or in the documents. However, those languages whose scripts are in Latin use white space for separation of words from one another within the sentence.

Afaan Oromo is also one of the Cushitic family that uses Latin script for textual purpose and it uses a whitespace character to separate words from each other in text documents[4]. For example, “Tolaan saree gurraacha ajjeese/ijjeese” which to mean, “tola killed the black dog”. In this sentence, the words “tolaan”, “saree”, “gurraacha” and “ajjeese/ijjeese” are separated from each other by a whitespace character.

In this study, tokenization is used for splitting of input text document into list of tokens at its words boarder (based on white space character) without consideration of their meaning and relationship. These list of tokens are used for further processing purpose. In addition to this, we also use

tokenization to remove some special characters such as **@, #, $, %, ^, &, *, ~, (), [], _, =, <>, {}, £, -,** and all punctuation marks **(?, :, ;, ., …, /, /;, !, :-“ ”)** and numbers (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) that are attached to text in the corpus. The same technique is applied at query processing side. After documents tokenization process is successfully completed, linguistic pre-processing tasks is started.

```
" Corpus Tokenization"
For file in corpus
    Define word delimiter to space
    Read files
        For file in Read
        If there is word delimiter
            Put each terms as a separate token
         End If
        End For
End For
```

Figure 4.4-1: Algorithm for tokenization

4.4.2. Normalization

Peoples use different character form to write the same word. They can use uppercase, lower case or mix of the two. In Afaan Oromo writing system, using the same word with spell inconsistency is common issues. This use of inconsistent spelling might be occurring in corpus that needs to be used for system evaluation purpose. Therefore, if those words are not replaced with uniform word format throughout the corpus, words that describe the same concept are treated in a different way by the system. Due to this reason, more than one index term will be occurring for single word within the document. As result, the system computational time is increased, more space is occupied, and system effectiveness (accuracy) will be decrease.

Therefore, to reduce such type of problems, all words having different forms but the same in their meaning are converted to a uniform format. The process is so-called normalization. The technique

is used to handle different writing style. For instance; the Afaan Oromo word “Dhukkuba” to mean disease is written as “Dhukkuba” at the beginning of the sentence and it is written as “dhukkuba” at the middle of a sentence while it has the same meaning at both locations in the sentence. So, case normalization is used for handling such variations throughout the corpus. In our case, all the text in the corpus is converted into most commonly used case format, which is lower case to make uniformly understandable by the system. This task is performed using python 3.4 scripts.

4.4.3. Elimination of Stop Word

Words that are common among the documents in the collection are not good discriminators. Such words are called stop words. Stop words are a set of commonly used words in any language. Such words should be removed from text document since it is not having contribution for document content descriptions or representation. Stop words removal is critical to information retrieval applications because, while we remove the words that are very common in a given text, we are focusing on the important words instead. For example, the user query is “how to develop information retrieval applications”. If the search engine tries to find documents that contained the terms “how”, “to” “develop”, “information”, ”retrieval”, “applications”, the search engine is going to find more documents that contain the terms “how”, “to” than documents that containing information about developing information retrieval applications because the terms “how” and “to” are so commonly used in the English language. Therefore, if we neglect these two terms, the search engine can actually focus on retrieving documents that contain the keywords: “develop” “information” “retrieval” “applications” which would more closely return documents containing related information.

The other reason for the removal of stop words is it reduces the index size or space. Some examples of these insignificant words are articles, conjunctions, pronouns, prepositions (such as from, to, in, and about), demonstratives, (such as this, these and there), and interrogatives (such as where, when, and whom). The same to English language, there are also words that are used as a stop words in Afaan Oromo, which has no any contributions to document meaning or semantics. Therefore, such types of words are removed from the document collection to prepare the term that exactly represents the document. Hence, stop word elimination improves the size of the indexing structures. In this study, we have use the stop word list prepared by Gezahegn Gutema (2012) and

add some of them from different literature and Afaan Oromo books. Table 4.4.1: shows sample list of Afaan Oromo stop words.

Akkam	Sun	Allatti	Haa ta`u malee	Yookiin
Akkasumas	Inni	Alla	Yoo	Garuu
Akkuma	Ishee	Amma	Gubbaa	Booddee
Ala	Ana	Ammo	Irraa	Akka
An	Ani	Bira	Fi	Irraatti
Kana	Ati	Booda	Ammallee	Ta`ullee

Table 4.4-1: Sample list of Afaan Oromo stop words

```
" Elimination of Stop word"
Read stop_word_list file
Open the file for processing
Do
    Read the content of the file line by line
     Assign the file content to String variable
       For word in String Variable Split by space
            If word in stop_word_list
             Remove from index term
            Else
                Continue
           End If
     End For
While End file
```

Figure 4.4-2: Algorithms for removing stop words

4.4.4. Words Stemming

In Afaan Oromo writing system, many words are morphologically variant. These morphological variant words have similar semantic understanding in the language. These words inflected by prefixes, infixes and suffixes. In IR system, those words considered as an equivalent word.

Therefore, to achieve computational efficiency and better retrieval performance, it is important to conflate morphological variations of the words. Accordingly, Stemming is a technique used for removing all affixes forms such as prefixes, infixes, and suffixes from words[48],[4]. Stemming reduces a different word variation that conveys the same meaning into a single form or to its root form. The application of stemming technique in information retrieval makes the processes less dependent on particular forms of words and reduces the potential size of the index terms[7]. As a result, the performance of IR system will enhanced. Mainly, it improves the recall. Therefore, applying stemming to Afaan Oromo text corpus and queries has a crucial contribution for our system efficiency and performance.

Stemming techniques are language dependent tasks. Since it is developed based on the morphological structure of the language, every language required to have language specific stemming techniques[7]. For instance, the stemming technique used for Afaan Oromo, Amharic language and English language is too different. Because, each language has their own meaningful pattern formation rules/syntax and structure that has to be followed.

The stemming algorithms in IR are used to improve the efficiency of the information retrieval and improve the recall. Conflation is the term that frequently used to refer mapping multiple morphological variants to a single root form representation. The main role of stemming is to remove various suffixes as result in the reduction of a number of words, to have exactly matching stems, to minimize storage requirement and maximize the efficiency of IR Model. In this study, we have used stemming algorithm developed by Debele Tesfaye (2010) using rule base approach. The stemming algorithm used in this study is given below.

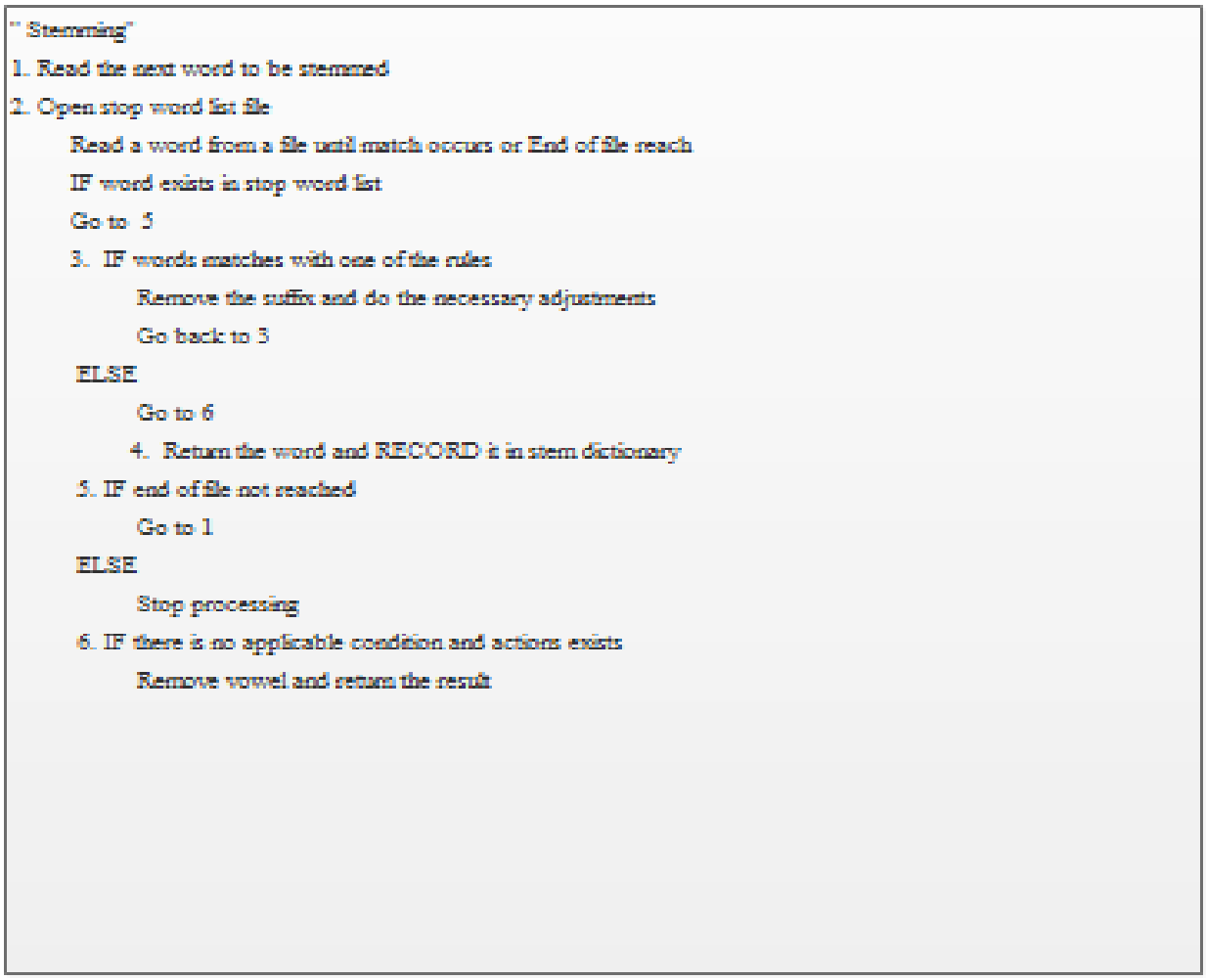


Figure 4.4-3: Algorithm for Stemming

4.5. Term Weighting

Term weighting is the process of calculating and assigning the respective weight to the terms within the documents. It is used to define the significance of a terms or word in specific document and in the overall documents collection. In another words, it is the identification of best terms that represent document content and it distinguish certain specific document from the rest of the collection. This implies that, the best terms should have high term frequencies in the document but have low term frequencies in overall collection[24].

Term weighting scheme consists two basic factors to avoid favor of long documents over short documents. these are: the local factor and the global factor[24]. The local factor examines and indicates how much the term is important in specific document. The global factor examines and

indicates how much the term is important in the documents collection. In this case, normalization factor is used to avoid the favoring of long documents over the short documents.

Local term weighting

As the name indicates, the computation of term weight is depending only on the frequencies of the term within the specific document and it is not depending on inter-documents term frequencies. It uses several techniques to measure term importance in the documents. these techniques are: Binary weighting, term frequency, logarithmic term frequency, and augmented normalized term frequency technique[24][52]. Among these techniques, term frequency and logarithmic term frequency are popular.

i. *Binary weighting* technique is not compute the actual frequency the term has in the document. Instead of computing the frequency of the term, it simply assigns equal relevance for all terms or words appear in the documents. This is totally based on the presence or absence of the term or words in the document. Therefore, it lacks partial matching.

$$\boldsymbol{li,j = 1, \quad if\ the\ term\ i\ exists\ in\ document\ j, \quad otherwise\ 0}$$

ii. *Term frequency* focus on how many times the term occurs in the document and it consider the more the term occurred frequently within the document, the more it is predicted as important one than the term that occur less frequently[52],[3].

$$\boldsymbol{li,j = fij, \quad fij\ indicates\ the\ numbers\ of\ occurrences\ of\ term\ i\ in\ document\ j}$$

iii. *Logarithmic term frequency* it works in the same to the term frequency but it uses log function to normalize the effect. Because, it is impossible to say that, the term that appears hundred times in a document is hundred times more important than the term or word appears only once[3],[52].

$$\boldsymbol{li,j = \log(fij + 1),\ fij\ indicates\ the\ numbers\ of\ occurrences\ of\ term\ i\ in\ document\ j}$$

Global term weighting

Global weighting gives more emphases to terms that are discriminating. Several schemes are based on the theory of the less frequently a term appears in the whole collection, the more discriminating it is[52].

The inverse document frequency (idf) is a most popular measure of words` importance throughout the whole collection. It`s defined as the logarithm of the ratio of total number of documents in a collection to the number of documents containing the given word or term[52]. This implies, rare terms or words have high idf and common terms or words have low idf value.

$$idf = log2(\frac{N}{dfi}) \quad ----------4.1$$

Therefore, tf*idf computed as:

$$tf * idf = tfij * log2\left(\frac{N}{dfi}\right) \quad ----------4.2$$

In this study, we use tf*idf term weighting schemes to compute weight of each terms. Because it is the normalized term weighting technique and it is the most popular term weighting technique used in information retrieval and search engine now a day. Therefore, it is more helpful in our works.

4.6. Semantic Compression Algorithm

Thesaurus is one of many tools used in information retrieval that helps the users to find the appropriate term during search time. It provides the controlled vocabulary that organized in a known order and structure to ensure the consistency of indexing and promotes retrieval that is more satisfactory. It solves semantic mismatch problem, in a condition where the users query term do not match with index term. For instance, if the user uses Afaan Oromo query “Qoricha HIV AIDS” (HIV AIDS medicine) and “Dawaan dhukkuba HIV AIDS” (HIV AIDS medicine). The query terms “qoricha” and “Dawaa” are synonym to each other but have different lettering in the query. Thus, a user`s query that contains the word “dawaa” will not match any documents that contains the word “qoricha”. All documents that contain the word “qoricha” would not be retrieved for query term “dawaa”, because there is no semantic relationship between word in the user`s query and those documents that contain the word “qoricha”. It is clear that the words mismatch results in poor performance of Afaan Oromo text retrieval system, which in turns, reduces the accuracy of information retrieval significantly. Thesaurus solves this problem by defining the semantic relationship among the terms during document indexing. Thesaurus use three basic term relationships to define semantic relationship among the terms[11],[8],[44]. These are equivalence relationships (synonyms terms: USE, UF), hierarchical relationships (broader or narrower terms:

BT/NT), associative relationships (related terms: RT). In this study, we develop a thesaurus that employs equivalence relationships among terms to handle synonymous words problems throughout Afaan Oromo text retrieval.

The process of manually constructing a thesaurus has a number of broad steps. These are:

1. The first step is selecting and recording terms from text documents collection.
2. Identify the wanted characteristics of the terms. In which type of term relationship the term can classified. Such as only synonyms, hierarchical or related term (Broader Term and Narrower Term) and scope notes.
3. Analyze each terms for its related vocabulary and group them based on their type synonyms, Broader Term (BT) and Narrower Term (NT), definitions and scope notes.
4. Organize these terms and their relationships into structures such as hierarchies (BT/NT), equivalence relationship (synset), Associative relationship (related).
5. Review the entire thesauruses for consistency check of word forms.
6. Finally, "invert" the hierarchy structure of the thesauruses into alphabetical arrangement of entries. This step is depending on relationship structure that exists between terms such as equivalence relationship, hierarchical, and associative.

These six steps are used to identify the appropriate term from the corpus and construct the thesaurus manually. To construct the thesaurus from corpus manually, first, the whole corpus is pre-processed automatically and content bearing terms are identified. Once content bearing terms are filtered, the weight of each term is computed using term weighting scheme. As described in section 4.5, due to its advantage, we used term frequency inverse document frequency (tf-idf) technique to compute the weight of the term. Therefore, thesaurus is constructed and term with high tf-idf is used as descriptor or key term in thesaurus. The descriptor (key) term used to represent synonymous terms in the thesaurus.

Then after, semantic compression algorithm is applied for indexing the documents based on information in the thesaurus. For index the documents, the algorithm analysis whether the term is used as a descriptor term or used as a synonymous term in the thesaurus. If the term is used as a descriptor term in thesaurus, the term is used for indexing the document. However, if the term is a synonymous term in the thesaurus, its respective descriptor in the thesaurus is searched and the

term will be replaced with descriptor term from the thesaurus. Finally, the completely indexed documents are organized using inverted file structure.

To illustrate this procedure, assume that there are three different user query that containing the term “hacuuccaa”, garboomsaa” and “cunqursaa”. In this case, the user can generate their query using one of the term as a query term and retrieve all document containing all three-query terms. Because, all are synonym to each other. From these three synonymous terms one of them are used as an index term and the remaining are expanded during search time. The candidate index term is considered as a preferred/descriptor term and the remaining are considered as variant terms. Preferred/descriptor term is a term used for document indexing whereas variant terms are represented by preferred term in the thesaurus and query term is expanded to include all list of synonyms terms in the thesaurus. During searching, the query term is checked whether it is an index term or not. If query term is an index term in inverted file, the system will return relevant documents including the document that containing the synonymous terms to query term. Else, if the query term is not an index term in the inverted file, the query term is checked against thesaurus to which descriptor term it belongs in the thesaurus and replaced with its descriptor. However, if the query term is not belonging to any descriptor terms and synonymous terms from thesaurus, the system will return no match found for the query prompt. Figure 4.6-1 shows algorithm for semantic compression.

```
Open thesaurus
Open term_dict
For term in term_dict
    Read each terms
    For each term in the term_dict
        If term is a key term (a descriptor) in thesaurus then the term is used as index term
         Else
         Search it`s descriptor from thesaurus and replace the term with it`s descriptor in inverted index file.
     End if
   End for
End for
```

Figure 4.6-1: Algorithm for semantic compression

4.7. Inverted Index

The basic concept in information retrieval is concerned about how documents collection is going to be represented in information retrieval system. The documents are logically represented by content bearing words or terms that occurs in it. In information retrieval, inverted file some time called inverted index and it is the standard data structure for efficient indexing of texts by their own terms[22], [8]. Inverted index is a dictionary of terms in which each word or term is followed by the identifier of every document that contains the word or term[22]. In addition to this, the number of occurrences of each word in a document is stored this structure. To create inverted file, the major steps that has to be followed are; the first task is collecting the documents that need to be indexed (preparing corpus). The second step is performing text pre-processing task such as tokenization, normalization, remove stop words, and stemming. For a list of root or stem words returned from stemming process, the respective term weighting is automatically computed. The term weighting scheme used in this work is tf*idf as discussed in section 3.4.5. As a result, list of terms that are used for indexing is identified. The data structure used for build document index and way of organizing it has significant impact on the performance of the entire retrieval system. An inverted file is a technique that permits an IR system to search quickly what documents contain a given set of words or terms, and determine how frequently the word appears in the document[22]. This can enhance the performance of the IR system to some extent. Thus, in this study, we used inverted index for document indexing purpose. Finally, the inverted file is created.

4.8. Evaluation Techniques

As discussed in section 2.1.3, in this study, we used three techniques intended for measuring of IR achievement, which is precision, recall and F-measure. Precision is expressed as the ratio of the number of relevant documents retrieved to the number of retrieved documents by the system. Recall is defined as the ratio of the number of relevant documents retrieved by the system to the total number of relevant documents in the collection. F-measure is an evaluation measurement that computed by combining precision and recall together. It is a harmonic mean of recall and precision used to measure the performance of the IR system as whole. Mathematical equation is already given in section 2.1.3.

CHAPTER FIVE

IMPLEMENTATION AND EXPERIMENTAL RESULTS

5.1. Introduction

In this chapter, the developed prototype for the designed thesaurus based semantic compression model and its evaluation is presented. The subsequent sections present thesaurus construction, index construction; thesaurus based semantic compression, Test query selection, experiment and performance evaluation parameters and challenges of the study.

5.2. Thesaurus Construction

Searching for a document based on their content meaning is depends on the word that used to index the document. The Retrieval system that based on word matching suffers from the vocabulary mismatch problem, which is a common phenomenon in the usage of natural languages. This is happening due to the existence of several ways of expressing of the same concept. For instance, for Afaan Oromo words “qorichi” and “dawaa”, the two words are write in different ways but they have the same meaning and convey the same message in a given Afaan Oromo text. This term mismatch degrades the performance of Afaan Oromo IR system. In order to come up with solution for this problem, we have proposed thesaurus based semantic compression for Afaan Oromo text retrieval. In our work, the thesaurus is manually construct from text corpus. As there is no previously available Afaan Oromo thesaurus, we have used different resource to build Afaan Oromo thesaurus manually. First, the term that needs to be included in the thesaurus are extracted from corpus. This task is performed by using text pre-processing (such as tokenization, normalization, stop word removal and stemming) the whole document corpus. Then, for list of stemmed terms, term frequency is automatically calculated using tf*idf term weighting scheme. Then, we manually construct the thesaurus from Afaan Oromo dictionary, which is entitled as “Galmee Jechoota Afaan Oromo”, as well as from Elellee (English-Oromo-Amharic Dictionary) (Galmee Jechoota Afaan Inglilizii-Oromo-Amaaraa) by Hinsene Mekuria Books. Then after, from list of synonyms of terms, the term with high frequency is used as a preferred term or key term in the thesaurus whereas the remaining synonym terms are considered as a no-preferred terms.

Preferred term used as index term. Table 5.2.1 shows some of the synonyms terms in Afaan Oromo.

Key term/preferred term	Synonym terms	Key term/preferred term	Synonym terms
goolaba	xumura, fixinsa, guduunfa, cufinsa	gowwummaa	daallummaa, doofuma, wallaalummaa
gurmaa`ina	qindoomina, tokkumma, walta`insa, gamta	shaggaa	gaarii, mishaa, dansaa, hosee, baroo, baayeesa
Guddina	misooma laalisu, badhaadhu, dagaaggina, horuu	sodaa	Gantu, yaraa, gadhee, dabeessa
gamnummaa	qarummaa, qaruxee	gumaata	arjoomu, kenna, sadaqaa, dabaree
Xiyyeeffannaa	hubannaa, xinxala, yaadaan, qayyabannaan, herregu	yaala	wal`aansa, cidhuu

Table 5.2-1: Sample list of synonymous terms in Afaan Oromo

5.3. Index Construction

As discussed in section 4.6, index construction is a natural language dependent process that need document pre-processing tasks. This pre-processing task involves various steps; such as tokenization, normalization, stop word removal and stemming. The next step is term weighting and indexing. In this work, tokenization is used to break down the whole collection of documents into the stream of tokens based on the given delimiter and at the same time some special characters, numbers, punctuation marks and other symbols are eliminated during document tokenization task. Normalization is used to handle different writing style in order to make uniform case throughout the document corpus. Thus, the system understands easily. In this research work, we used normalization to convert every words in the text corpus to lower case format. Stop word removal

is the process of eliminating non- content bearing words from the stream of tokens that are normalized. In this study, 206 Afaan Oromo stop word list prepared by Gezahegn Gutema (2012) and some others are identified by the researcher are used. This list of stop words are saved as a "stopwords.txt" file and checked against normalized tokens of the corpus. If the token is not in stop word list, it will be stemmed by stemming algorithm. Once the normalized tokens are checked, as it is not stop word, then all the remaining tokens are considered as content bearing words in the documents. Thus, they are stemmed and their respective term frequency is automatically computed. Based on term frequency, the term with higher term frequency is selected and its synonym is constructed. Then, from the synonymous terms, the one with high tf-idf is selected as an index term and finally document index is constructed based on inverted index file structure.

5.4. Thesaurus based semantic compression

We have a list of terms with their respective term frequency that returned from term weighting activity. Therefore, thesaurus based semantic compression technique uses term frequency as a one principle to index the document. The term with high term frequency is chosen and checked against the thesaurus whether the term used as a descriptor/key term in the thesaurus or it used as synonymous term. If the term used as a descriptor term/ key term in the thesaurus, it is also used as an index term. Else, if the term has higher term frequency and it is a synonymous term or if it is a synonymous term for predefined key term in the thesaurus, it replaced by key term/ descriptor and it used as an index term. Once the thesaurus based semantic compression is successfully completed, the text corpus is indexed using inverted file.

5.5. Description of prototype system

Implementing the prototype system that integrates thesaurus based semantic compression technique with basic AO IR system contains several components as depicted in figure 4.2-1 in chapter three. Those components describe the major activities done in development of applying thesaurus base semantic compression approach for Afaan Oromo text retrieval. The prototype system has implemented using python 3.4.0 version. Figure 5.5-1 shows the screen shoot of the prototype system. The users can write their query in Afaan Oromo texts towards any target in the input box. The system displays the list of relevant documents that relevant to the user query.

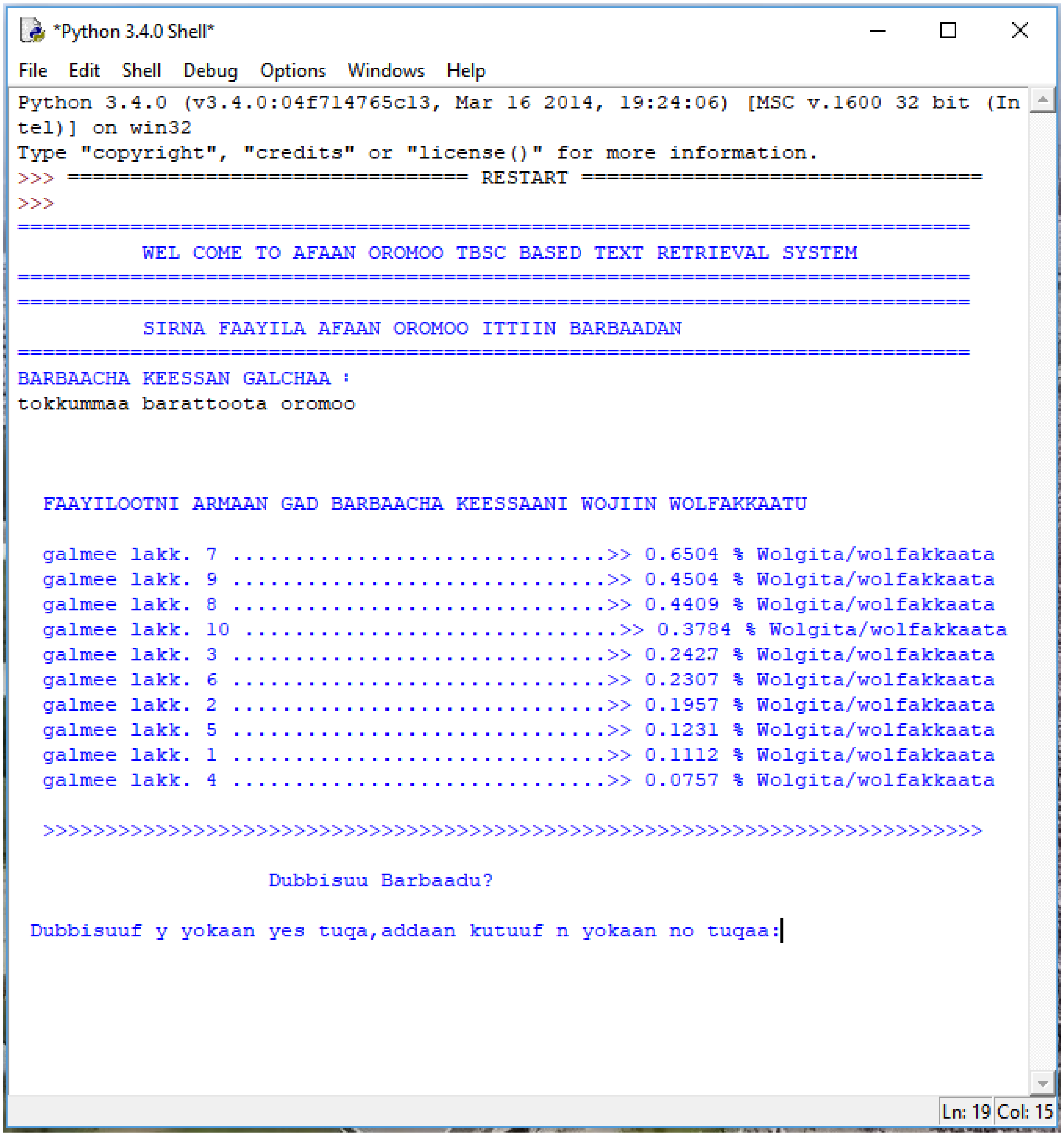


Figure 5.5-1: shows the screen shoot of the prototype system

The main objective of the prototype system is to retrieve relevant documents from oromiffaa text corpus as per user information needs. To achieve this objective, the following procedures are followed; first oromiffaa text documents are pre-processed to extract index terms, this involves activities such as tokenization, eliminating stop words, nomalization, and stemming. Once each term is converted to its stem or root, the weight is calculated for each term based on tf-idf term

weighting scheme. Finally, documents are indexed based on thesaurus using inverted index file structure. The same procedure is followed for user query processing as done in document pre-processing and indexing. To match the query term with index terms, the searching process follows thesaurus based query term and index term matching. Therefore, the indexing process is implemented to construct inverted index based on thesaurus. In constructing inverted index, the first step is extracting the terms from all documents in the corpus used for indexing. This step is started by tokenizing terms for all document collection. To do this, the python code read all files in the corpus line by line and generate each words, special characters, punctuations, and numbers. These words, special characters, punctuations and numbers are splitted based on white space and any special characters, punctuations and numbers are removed. Then, list of words or tokens are stored in "wordList" variable for further processing. Figure 5.5-2 Shows python code implemented to tokenize terms exists in afaan Oromo text corpus.

```
Punctuation = re.compile(r'[& %(.?!, $0123456789":)-/;\*]"')
filepaths = glob.glob("C:/Python34/masters/corpus/*.txt")
 word_list = [ ] # store each terms in the documents
     for files in filepaths:# for each files in the directory read its contents
         with open(files, 'r') as f:
             Words = f.readlines()
             Words = re.split('\s+', open(files).read().lower())
         word_list.append(words)
```

Figure 5.5-2: Python code for tokenization

After list of terms are extracted and tokenized, the next step is removing stop words from tokenized list of terms. At this time, the python code read stop word list from file saved as "*stopwords.txt*" and compare the tokenized terms with stop word list. If tokenized term is similar with stop word in stop word list, it removes from wordlist. Figure 5.5-4 Shows python code implemented to remove stop word list from tokenized terms.

```
stopword = open("C:/Python34/masters/stopwords.txt", 'r')
Stop_word = stopword.readline()
Stop_list = stop_word.split().lower()
For I in range (0, len(stop_list)):
    If word in stop_list[i]
        Pass
    Else
        Word_list.append(word)
```

Figure 5.5-3: Python code for removing stop words

The next step is text normalization. As discussed in section 4.4.2, In Afaan Oromo writing system, the letters can be exists as uppercase, lower case or mix of the two in text. To convert these letters alphabets to common form in the text, normalization code is implemented. In this study, all the alphabets in the text are converted to lower case to make uniform letter case throughout the corpus. This is already done in tokenization as it is simply performed by single python function which is "*wordList.lower()".*

Since Afaan Oromo is morphologically rich language, words can exists in variant format. This variant of words can have the same or similar semantic interpretations in the given context. Human being can simply identify the variant word that belongs to similar semantic class. However, computer system cannot identify these word variants. Therefore, there is a need to reduce words to their stem or root. In this research work, the stemming implementation has two modules. The module used for prefix removal and the module used for suffix removal. The prefix module used for removing of prefix attached to the word. To do this, the python code takes list of index terms and checks if the index term exists in stop word list or not. If the index term exists in stop word list, the index term omitted from further processing and process next word if any. If the index term is not exists in stop word list, the python code iterate on word and check characters that match with one of the prefix exists in prefix list. If the character matched, it is a prefix and removed from the word. Once the prefix removed from Afaan Oromo index term, the next step is removing the suffix from index term. To remove suffix from index term the same steps are followed like prefix. Figure 5.5-4 Shows python code for implementation of stemming.

```
def stem(w):
      if len(w) < 3:
return w
      Else
          if _s_v.match(stem):
              w = stem + "y"
               m = _step2.match(w)
              if m:
                    stem = m.group(1)
                     suffix = m.group(2)
              If _mgr0.match(stem):
                   w = stem + _step2list[suffix]
                   m = _step3.match(w)
             if m:
               Stem = m.group(1)
                suffix = m.group(2)
            if _mgr0.match(stem):
               w = stem + _step3list[suffix]
                m = _step4_1.match(w)
             if m:
                  stem = m.group(1)
              if _mgr1.match(stem):
                  w = stem
              else:
                   m = _step4_2.match(w)
               if m:
                  stem = m.group(1) + m.group(2)
                  if _mgr1.match(stem):
                w = stem
                 m = _step5.match (w)
              if m:
                   Stem = m.group(1)
                  if _mgr1.match(stem) or (_meq1.match(stem) and not _c_v.match(stem)):
                   w = stem
           if w.endswith("ll") and _mgr1.match(w):
              w = w[:-1]
          if first_is_y:
             w = "y" + w[1:1]
   return w
```

Figure 5.5-4: Shows python code for implementation of stemming.

5.6. Test query selection

To test the developed prototype system, 10 queries that are able to describe the document is prepared by speakers of Afaan Oromo language. These queries are prepared by considering the synonymous words in Afaan Oromo based on the collected text corpus. Then, these queries are used to retrieve relevant documents from text corpus. It is used to measure the performance of the prototype system in which the user enters the query to the system and the system retrieves the relevant document from Afaan Oromo text corpus. This task is repeated for all 10 queries and the system output will be recorded. To make relevance judgement whether the retrieved relevant document by the prototype system is relevant or not, the relevant document retrieved by the system and the manually classified document is crosschecked to now the document is either relevant or irrelevant to the query.

5.7. Experiment and Performance Evaluation

The prototype system is developed to meet the proposed functionality of thesaurus based semantic compression for Afaan Oromo text retrieval. In this section, experiment setting, data collection, evaluation metrics (precision, recall and F-measure), discussion of experimental results and the challenges and finding are the subtopics that discussed as follows.

5.7.1. Experiment Setting

The experimentation has done on a laptop PC with windows 10 operating system, 2.40 GHZ Intel CPU, 4 GB RAM and 1tera hard disk. Python 3.4.0 programming language for the development and testing of the developed model.

5.7.2. Data Collections

The developed prototype system is evaluated using 106 Afaan Oromo text documents collected from news articles, Afaan Oromo bible, websites, books by the researchers. The collected documents involve various subject areas like education, politics, and culture, religion and history events. This used for indexing purpose. Thesaurus manually constructed from document corpus based on two Afaan Oromo dictionaries. Which are hiika galmee jechoota Afaan Oromo and Elleele dictionary.

5.7.3. Manual documents Classification

This activity is concerned with labeling of Afaan Oromo text documents for experiment purpose. All the 106 text documents are manually categorized by the researcher to judge the relevance based on the set of queries that have prepared for system testing. For each query, the set of relevant document are manually classified. Manual classification of documents is interesting to evaluate the system retrieval performance. Table 5.7-1: shows the manual classification of the number of relevant documents for each queries.

No	Queries	No of relevant document	No of non-relevant document
1	Qoricha dhukkuba busaa	13	93
2	Gamtaa barattoota yuuniversiiti	14	92
3	Dhibee saree maraatee	12	94
4	Seena uummata Oromo	21	85
5	Tapha kubbaa miilaa guutu oromiyaa	9	97
6	Biyyoota gaanfa afrikaa	11	95
7	Mootummaa cunqursaa	6	100
8	Bulchiinsa sirna gadaa	4	102
9	Qorannoo ammayyaa	8	98
10	Woldaa Misooma oromiyaa	2	104

Table 5.7-1: Manual classification of document relevance with respect to 10 queries

5.7.4. Evaluation Parameters

As discussed in section 2.2.3 the basic statistical measures parameters that most frequently used to measure the effectiveness of IR system are precision, recall, and F-measure. Therefore, in this study, we used precision, recall and F-measure parameters to evaluate the effectiveness of the developed prototype system. Precision measure the accuracy or quality of the developed prototype. It measures to what extent the system correctly retrieves the relevant document in the text corpus for a given query. It is the ratio of the number of relevant documents a search retrieves to the total number of documents retrieved either it is relevant or irrelevant for specific user query term by the prototype. As the higher the precision result is, the better the system is performing. The higher precision result indicates that, the probability in which the system retrieve non- relevant document for a given query is reduced whereas the low precision result interpreted as more documents that are non-relevant are considered as a relevant and returned by the system. Therefore, precision is concerned about how many retrieved document are relevant for the given information need. The precision is computed by:

$$Precision\ (p) = \left(\frac{TP}{TP + FP}\right) - - - - Equation\ 5.1$$

Where;

- ✓ **TP** stand for True positives, which means the number of relevant documents that are correctly retrieved by prototype.
- ✓ **FP** stands for false positives, which means the number of irrelevant documents retrieved by the prototype as relevant documents.

Recall is a measure of completeness or quantity. It is the number of relevant documents retrieved by prototype system divided by the total number of existing relevant documents in the corpus. The recall is mainly concerned about how many relevant document is selected. It is computed by:

$$Recall\ (R) = \left(\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}\right) - - - - - Equation 5.2$$

Where,

- ✓ **TP** stand for True positives, which means the number of relevant documents that are correctly retrieved by prototype.
- ✓ **FN** stand for False Negatives, which means the number of relevant documents that are incorrectly rejected by the prototype.

There is an inverse relationship between precision and recall. Where it is possible to increase one at the cost of reducing the other. The greater precision result decreases the recall and the higher recall result will decrease the precision. F-measure is the harmonic mean of the precision and recall that takes the interpretation of both the precision and recall measures. It is computed as:

$$F - Measure\ (F) = \left(\frac{2 * Precision * Recall}{Precision + Recall}\right) - - - - Equation\ 5.3$$

In this study, these three parameters are used to measure the effectiveness of the proposed thesaurus based semantic compression model for Afaan Oromo IR system. The evaluation made to examine to what extent correctly the developed prototype system retrieves the relevant document from Afaan Oromo text corpus as per user query. Before testing is done, Ten (10) Oromiffa test queries are formulated to perform relevance judgement and document query matrix that show all relevant documents for each test queries is constructed. For each test queries, the system effectiveness is measured by precision, recall, and F-measure. Therefore, the average registered results of precision, recall and F-measure represent the total performance of the

prototype system. Finally, we have compare the total performance registered by semantic compression approach using thesaurus based indexing for Afaan Oromo text retrieval with previously done Afaan Oromo IR system based on Vector space model.

5.7.5. Experimental Results

The experiment is conducted using 106 Afaan Oromo text documents and 10 queries that used for searching of document from Afaan Oromo text corpus. For evaluation purpose, each query is given to prototype system and all returned results were recorded. Then, the results were crosschecked from manually document classification table. This is made to check whether the prototype system is retrieving the relevant document list or not as per user request. Therefore, the results of this experiment are presented in the following table using IR performance evaluation metrics such as precision, recall and F-measure. Table 5.7.2. Shows the effectiveness of the AO IR system without using thesaurus based semantic compression based on 10 queries selected for the experiment.

Query	Relevant	Retrieved by system	Relevant retrieved	Precision	Recall	F-measure
Qoricha dhukkuba busaa	13	8	5	0.6250	0.3842	0.4753
Gamtaa barattoota yuuniversiiti	14	10	8	0.8010	0.5712	0.6666
Dhibee saree maraatee	12	10	10	1.000	0.8333	0.9090
Seena uummata Oromo	21	29	18	0.6207	0.8570	0.7192
Tapha kubbaa miilaa guutu oromiyaatti	9	13	7	0.5385	0.7777	0.6363
Biyyoota gaanfa afrika	11	8	6	0.7501	0.5454	0.6315
Mootummaa cunqursaa	6	21	6	0.2857	1.000	0.4444
Bulchiinsa sirna gadaa	4	13	4	0.3075	1.000	0.4697
Qorannoo ammayyaa	8	12	7	0.5833	0.8750	0.6999
Woldaa Misooma oromiyaa	2	6	2	0.3333	1.000	0.5001
Total				5.8451	7.8438	6.152
Average				**0.58451**	**0.78438**	**0.61520**

Table 5.7-2: The performance of the prototype system without considering synonymous words

As it is shown in table 5.7.2, from the experiment average result of 58.451% precision, 78.438% recall, and 61.520% F-measure are registered using IR system without thesaurus based semantic compression approach. From this experiment, we observe that the percentage of recall dominates the percentage of precision by 19.987%. To balance the precision and recall the F-measure records is 61.520%, which indicates the performance of the system is not satisfactory.

From the above evaluation, the obtained result shows that the system retrieved most of the relevant documents in the corpus. However, the result of the precision indicates that non-relevant documents are retrieved as relevant documents with relevant documents in higher number than the relevant documents. This is because the system retrieves any documents that containing one of the query terms that are not relevant contextually. In this case, the retrieved documents are irrelevant because, the query term appear in those documents are not express the meaning of the query with respect to other terms exists in the query. For instance, for query "mootummaa cunqursaa" which express the idea of ***colonial government***, the system retrieved irrelevant documents such a ***file05.txt***, ***file08.txt*** and ***file88.txt*** because they contains query term "*mootummaa*" to mean "***government***". However, in these retrieved documents the term "*mootummaa*" is used to express different issues related to government that is not related with user query concept that is colonial government. In other hand, for user query "*qoricha dhukkuba busaa*" the total relevant documents for the query is 13 as indicated in the table 5.7 2, but the prototype returns 8 documents and out of these, five of them are relevant documents whereas three of them are irrelevant documents for the query. In this case, the IR system is unable to retrieve all relevant documents in the document corpus. This is because the query terms "*qoricha*" and "*dhukkuba*" has a synonym term such as "*dawaa*" and "*dhibee*" respectively. Due to this reason, the term "*dawaa*" and "*dhibee*" is not exists in query term but they have the same meaning with query terms. However, those document containing term "*dawaa"* and *"dhibee"* are ignored by prototype, as it is an irrelevant one. The same problems are discovered for other query term that has more than one meaning. The prototype is not perform well because it is affected by synonymous and polysemy nature of Afaan Oromo words. Considering documents having synonym terms of the user query term, the prototype system performance is evaluated. Table 5.7.3. Shows the result of the prototype system after using thesaurus based semantic compression for AO IR system based on 10 queries selected for prototype evaluation. Table 5.7.3 Show the evaluation results as follows.

No.	Query/gaafilee	Relevant	Retrieved by system	Relevant retrieved	Precision	Recall	F-measure
1	Qoricha dhukkuba busaa	13	14	11	0.7857	0.8461	0.8181
2	Gamtaa barattoota yuuniversiiti	14	16	14	0.8750	1.0	0.9333
3	Dhibee saree maraatee	12	14	11	0.7857	0.9166	0.8460
4	Seena uummata Oromo	21	29	19	0.6551	0.9047	0.7599
5	Tapha kubbaa miila guutu romiyaatti	9	10	9	0.90	1.0	0.9473
6	Biyyoota gaanfa afrika	11	12	7	0.5833	0.6363	0.6086
7	Mootummaa cunqursaa	6	10	6	0.60	1.0	0.750
8	Bulchiinsa sirna gadaa	4	12	3	0.25	0.75	0.375
9	Qorannoo ammayya	8	7	7	1.0	0.875	0.9333
10	Woldaa Misooma oromiyaa	2	3	2	0.6666	1.0	0.80
	Total				**7.1014**	**8.9287**	**7.7715**
	Average				**0.71014**	**0.89287**	**0.77715**

Table 5.7-3 The performance of the prototype system using semantic compression based on thesaurus based indexing technique

5.7.6. Discussion of the Results

Findings of the experiment

As shown in table 5.7.3, Applying Thesaurus based semantic compression for Afaan Oromo text retrieval has been registered average precision, recall and F-measure of 71.014%, 89.287% and 77.715% respectively. Relatively the prototype system performs well with term that has synonymous terms in the thesaurus than terms that has no synonyms word in the corpus as the result in the table show. There are several relevant documents containing different words that have the same meaning in document corpus for a given particular user query. For instance, for the query '*qoricha dhukkuba busaa', the word 'qoricha' and 'dhukkuba'* both have synonymous word in which *'qoricha'* is synonym for *'dawaa'* and *'dhukkuba'* is synonym for *'dhibee'.* For this query, the prototype system retrieve the same result for both document that is described by query term *qoricha* and *dhukkuba* as well as documents that described in synonymous term of *qoricha* and

dhukkuba, which is *dawaa* and *dhibee* respectively. At this time, the prototype performs 0.7857 precision, 0.8461 recall and 0.8181 F-measure. This indicates that the prototype system retrieves almost all relevant documents for query terms that have synonym words.

However, when we see for the query *'seena uummata Oromo'* the prototype system recorded better precision whereas the recall is decreased. This is because, the words in query, which are 'seena', and 'Oromo', have no synonym words. Thus, it is simply indexed with respect to the documents that containing it without consideration of their semantic meaning. Here, we observe that the document containing such a query term may convey different information in different contexts. For instance, the query '*seena uummata Oromo'* which means *history of Oromo people* can be include different domain such as political history, religion history, culture history, and the like. In this case, any document containing the term that match with the query term is considered as a relevant list. Therefore, the prototype system retrieves any document containing query term even if it is not relevant. This is interpreted as the prototype is retrieving non-relevant document while the relevant documents are left. This decline the recall of the prototype system. As generally, the developed prototype performs well in handling synonymous terms or word mismatch for different terms that have or convey the same meaning during search time. In addition to this, the result obtained from proposed model is promising when comparing with previously done Afaan Oromo IR system developed by Gezahegn [4] using vector space model.

Challenges of the study

Even though the promising result is registered in this study, there were several challenges that the researchers come across during the process of developing the prototype. These challenges hindered the prototype not to register the better result as expected. In this study`s experiment, we have learned some reasons and challenges for the distorted results of the experimental results. This is happened due to different reasons that are related to the nature of natural language complexities.

The first reason is that Afaan Oromo has no well-organized resource used for research experiment and the language still exists at infant stage in computational level. So that, getting synonyms semantic similarity or meaning of each term in Afaan Oromo text corpus is challenging. In our work, we have tried to retain the semantics of the document by recognizing and extracting terms that have more than one meaning in the document. However, this technique was not address the problem of polysemy. For instance, when we query *"Addunyaan baddee"* in this query, the term

"addunyaan" can translated as a *world* or *wealth* or it can be a girl whose name is *"addunya"*. It semantic meaning is depends on the context of the given document. In this case, the system couldn`t predict the context of the word in the document. Therefore, this problem has great influence on IR system performance.

The second reason is there is some term that has only single meaning. In this case, the term with high tf-idf used to index the document. This does not consider the meaning of the term according to the context of the document. The prototype system simply retrieves if query term is appear in the document. At this time, many document that are semantically not relevant to the user query term are retrieved as relevant one as well as some semantically relevant documents are missed. This situation affects performance of thesaurus based semantic compression model for Afaan Oromo text retrieval system.

On the other hand, some experimentation result shows that relevant document may not retrieved. This is happen due to the inability of stemming algorithm to stem the words variant to their correct root. In such case, the system interprets the query term as it has different meaning and returns irrelevant list.

CHAPTER SIX

CONCLUSION AND RECOMMENDATIONS

6.1. Conclusion

In this research work, we proposed and developed a thesaurus based semantic compression model for Afaan Oromo text retrieval. The main objective of the study is to integrating semantic compression approach using thesaurus based indexing technique. Before going to develop the proposed model, different information retrieval approaches developed for different languages, and the nature, structure and pattern of Afaan Oromo writing system have been studied. Thesaurus based semantic compression approach has been used as a development method to identify semantically related terms from Afaan Oromo text corpus for indexing. In this study, Afaan Oromo thesaurus is manually prepared from text corpus and algorithms used to implement the components of the model have been developed. Accordingly, the collected Afaan Oromo text documents are passes through each components of the model such as text preprocessing (i.e. tokenization, normalization, stop word removal and stemming), term weighting computation and semantic compression using thesaurus based indexing.

The text-preprocessing components is accountable for accepting the input text documents and returns a set of tokenized, normalized, and stop word free terms from text documents. The stemming part accepts the normalized and stop word free terms and decomposes them into their root words and affixes. All these are used for identification of content bearing terms for indexing and searching purpose. For each stemmed terms, their respective weight is computed using term weighting components using tf-idf term weighting technique. Based on weight of the term obtained from term weighting components, semantic compression is applied to compare term that has higher term weight and the key term used in the thesaurus and perform the sematic compression tasks. Finally, the text corpus is indexed using thesaurus based inverted file structure.

In order to evaluate the developed prototype, 106 Afaan Oromo text documents collected and 10 queries are used. In this study, system evaluation is performed in two phase. In the first phase, the prototype system is tested without considering thesaurus based indexing and in second phase, it is tested using thesaurus based indexing. Without thesaurus based semantic compression approach, the prototype system registered 58.451% precision, 78.438% recall, and 61.520% F-measure

respectively. Using thesaurus based semantic compression approach; the prototype system registered the average result of 71.014% precision, 89.287% recall, and 77.715% F-measure. The results obtained from this study are promising as it is the first experiment-using thesaurus based indexing for the language.

6.2. Recommendations

In this study, an attempt is made to develop and integrate semantic compression approach to Afaan Oromo text retrieval system. Developing complete semantic compression based IR system for Afaan Oromo text retrieval needs a long period of time, experts from different disciplines such as: linguistics, natural language processing, IT, large size of Afaan Oromo text corpus from different domains and full-fledged Afaan Oromo thesaurus (WordNet). The collaboration of all these different experts and the wealth availabilities of these resources can be make easy the complexity of developing a full-fledged semantic compression based Afaan Oromo text retrieval system. Therefore, the system we have developed has many rooms for further performance enhancements. Based on the knowledge obtained from literatures and the finding of our research work, the following recommendations are forwarded for the future research directions.

- ✓ Developing standard corpus for the language: In information retrieval study, consistent corpus is required for conduct experimentation and testing purpose. However, there is no compiled and developed standard corpus for Afaan Oromo so far. This needs immediate research works for the future.
- ✓ Constructing a standard thesaurus system for the language: the standard thesaurus annotated to Afaan Oromo text retrieval task is not available so far. For the purpose of this study, only small amount of the synonymous words of the language are annotated manually but since there is another term having more semantic relations and similarity, this is not sufficient to build robust semantic compression based IR model for the language. Therefore, developing a standard thesaurus or WordNet for Afaan Oromo can enhance the performance of the semantic compression based Afaan Oromo text retrieval system.
- ✓ Developing of ontology based semantic compression approach: nowadays, ontology plays a great role in semantic IR system. Therefore, any attempt to investigate ontology based semantic compression can be improving the performance of Afaan Oromo IR system.

- ✓ This research work is conducted on text documents, but other types of document like video audio, graphics and pictures are not studied yet. Therefore, conducting more study is important to integrate semantic compression for video, audio, graphics and pictures documents to come up with fully functional system.
- ✓ Developing of a full-fledged system: the model built in this work is for academic study that is at thesis level. Further works should be conducted to develop a full-fledged semantic compression based IR project that performs regular task of Afaan Oromo text retrieval to help speakers of the language.

REFERENCE

[1] M. Wordofa, “Semantic Indexing and Document Clustering for Amharic Information Retrieval Addis Ababa University School of Graduate Studies,” 2013.

[2] R. Etrieval, “Concept-Based Indexing in text Information Retrieval,” vol. 5, no. 1, 2013.

[3] M. La Fleur, “Conceptual Indexing using Latent Semantic Indexing,” 2015.

[4] G. G. Eggi, “Afaan Oromo Text Retrieval System,” no. June, 2012.

[5] M. F. Sánchez, “Semantically enhanced Information Retrieval : an ontology-based approach,” no. January, 2009.

[6] T. H. Gebermariam, “Amharic Text Retrieval: An Experiment Using Latent Semantic Indexing (LSI) With Singular Value Decomposition (SVD). Addis Ababa University School of Graduate Studies,” no. July, 2003.

[7] A. H. Madessa, “Probabilistic Information Retrieval System for Amharic Language,” no. June, 2012.

[8] T. A. CHEKOL, “Applying Thesaurus Based Semantic Compression for Improving of the Performance of Amharic Text Retrieval,” no. OCTOBER 2014, 2015.

[9] A. Bruck and T. Tilahun, “Bi-gram based Query Expansion Technique for Amharic Information Retrieval System,” no. November, pp. 1–7, 2015.

[10] D. C. B, “Semantic Compression for Text Document,” pp. 20–48, 2014.

[11] D. Ceglarek, K. Haniewicz, and W. Rutkowski, “Semantic compression for specialised Information Retrieval systems.”

[12] W. U. Dan and W. Hui-lin, “Role of Ontology in Information Retrieval,” vol. 4, no. 2, 2006.

[13] A. S. Genemo, “Afaan Oromo Named Entity Recognition using Hybrid Approach,” no. March, 2015.

[14] I. Bedane, “The Origin of Afaan Oromo : Mother Language,” vol. 15, no. 12, 2015.

[15] F. O. F. Computer and M. Sciences, “Named Entity Recognition for Afaan Oromoo School

of Graduate Studies Faculty of Computer and Mathematical Sciences," no. October, 2010.

[16] Sisay Adugna Chala, "English – Afaan Oromoo Machine Translation: An Experiment Using Statistical Approach a thesis submitted to the school of graduate studies of Addis Ababa University.," 2009.

[17] K. M. Jimalo, R. B. P, and Y. Assabie, "Afaan Oromo News Text Categorization using Decision Tree Classifier and Support Vector Machine : A Machine Learning Approach," vol. 47, no. 1, pp. 29–41, 2017.

[18] Daniel Bekele Ayana, "Afaan Oromo-English Cross- Lingual Information Retrieval (CLIR): A Corpus Based Approach," no. June, 2011.

[19] H. M. Harb, "Semantic Retrieval Approach for Web Documents," vol. 2, no. 9, pp. 67–76, 2011.

[20] S. Kumar and R. K. Rana, "Ontology based Semantic Indexing Approach for Information Retrieval System," vol. 49, no. 12, pp. 14–18, 2012.

[21] E. Mohammed and M. Alshari, "Semantic Arabic Information Retrieval Framework Semantic Arabic Information Retrieval Framework," no. 2014.

[22] M. Wordofa, "Semantic indexing and document clustering for Amharic Information Retrieval, A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Scien," 2013.

[23] V. K. Singh, V. K. Singh, G. Vishwavidyalaya, S. A. Programmer, and G. G. Vishwavidyalaya, "Vector Space Model: An Information Retrieval System," pp. 141–143, 2015.

[24] S. C. O. Riordan, "The Evolution and Analysis of Term-Weighting Schemes in Information Retrieval," 2008.

[25] P. Janarthanan, "Information Retrieval Using Second Order Co-occurrence PMI," vol. 9, pp. 79–88, 2013.

[26] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, and E. Motta, "Semantically

enhanced Information Retrieval : an ontology-based approach," 2009.

[27] D. H. Deshmukh, "Information Retrieval System Based On Ontology," 2016.

[28] A. Bouramoul, M. Kholladi, and M. C. B. P, "An ontology-based approach for semantics ranking of the web search engines results."

[29] D. Tecnologie, D. Tecnologie, C. Pereira, and D. Tecnologie, "An Ontological Representation of Documents and Queries for Information Retrieval Systems."

[30] A. Ayedh, G. Tan, K. Alwesabi, and H. Rajeh, "The Effect of Preprocessing on Arabic Document Categorization," 2016.

[31] M. Wedyan, B. Alhadidi, and A. Alrabea, "The effect of using a thesaurus in Arabic information retrieval system," vol. 9, no. 6, pp. 431–435, 2012.

[32] A. Aggarwal, G. Sharma, and S. Sharma, "An Enhanced Semantic Retrieval System of Trademarks using Machine Learning," vol. 9, no. September, 2016.

[33] W. B. Cavnar, J. M. Trenkle, and A. A. Mi, "N-Gram-Based Text Categorization."

[34] M. A. Al-ramahi and S. H. Mustafa, "N-Gram-Based Techniques for Arabic Text Document Matching ; Case Study : Courses Accreditation," vol. 21, no. 1, pp. 85–105, 2012.

[35] P. Majumder, M. Mitra, and B. B. Chaudhuri, "N-gram : a language independent approach to IR and NLP," vol. 3.

[36] S. H. Mustafa and Q. A. Al-radaideh, "Using N-Grams for Arabic Text Searching," vol. 55, no. March, pp. 1002–1007, 2004.

[37] M. Danesh, B. Minaei, and O. Kashefi, "A Distributed N-Gram Indexing System to Optimizing Persian Information Retrieval," vol. 5, no. 2, 2013.

[38] K. Feldvari, "Thesauri Usage in Information Retrieval Systems : Example of LISTA and ERIC Database Thesaurus," pp. 279–288.

[39] A. Kennedy, "Automatic Supervised Thesauri Construction with Roget ' s Thesaurus by," 2012.

[40] P. Q. Rashid, "Semantic Network and Frame Knowledge Representation Formalisms in

Artificial Intelligence," no. February, 2015.

[41] A. Tefera, "Automatic Construction of Amharic Semantic Networks From Unstructured Text Using Amharic WordNet," 2010.

[42] Andargachew Mekonnen Gezmu, "Automatic Thesaurus Construction for Amharic Text Retrieval Amharic Text Retrieval," no. July, 2009.

[43] N. Lagutina, I. Paramonov, I. Vorontsova, and N. Kasatkina, "An Approach to Automated Thesaurus Construction Using Clusterization-Based Dictionary Analysis."

[44] C. Ryan, "Ryan, C. (2014), 'Thesaurus construction guidelines: an introduction to thesauri and guidelines on their construction'. Dublin: Royal Irish Academy and National Library of Ireland.," *Guideline*, vol. no 5, p. 113, 2014.

[45] A. Bruck and T. Tilahun, "Enhancing Amharic Information Retrieval System Based on Statistical Co-Occurrence Technique," no. December, pp. 67–76, 2015.

[46] A. Tefera, "Automatic Construction of Amharic Semantic Networks (ASNet),School of Graduate Studies College of Natural Sciences Department of Computer Science," no. March, 2013.

[47] G. M. Wegari, "Parts of Speech Tagging for Afaan Oromo," pp. 1–5.

[48] D. Tesfaye, "Designing a Stemmer for Afaan Oromo Text : A Hybrid Approach Addis Ababa School of graduate studies Faculity of informatics," 2010.

[49] G. Tesema and A. Y. Abate, "Design and Implementation of Predictive Text Entry Method for Afaan Oromo on Mobile School of Graduate Studies Addis Ababa Institute of Technology (AAIT) Electrical And Computer Engineering School Of Graduate Studies INSTITUTE OF TECHNOLOGY (AAiT) ELEC," 2013.

[50] E. G. Desisa, "Sentiment Analysis for Opinionated Afaan Oromoo Texts E, Addis Ababa, Ethiopia October, 2017," 2017.

[51] Gaddisa Olaani Ganfure, "Design and Implementation of Afaan Oromo Spell Checker," no. June, 2013.

[52] N. Polettini, "The Vector Space Model in Information Retrieval - Term Weighting Problem

Local Term-Weighting," pp. 1–9, 2004.

APPENDIXES

Appendix A: Relevance Judgement Table

Text file	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*	*Q6*	*Q7*	*Q8*	*Q9*	*Q10*
File01.txt	NR	R	NR	R	NR	R	NR	NR	NR	NR
File02.txt	NR	NR	NR	R	NR	NR	R	NR	NR	NR
File03.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File04.txt	NR	NR	R	NR	R	NR	R	NR	NR	NR
File05.txt	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
File06.txt	NR	R	NR	R	NR	NR	R	NR	NR	NR
File07.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File08.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
File09.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File10.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File11.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
File12.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File13.txt	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
File14.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File15.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File16.txt	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
File17.txt	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
File18.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File19.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
File20.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File21.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
File22.txt	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
File23.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File24.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File25.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
File26.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
File27.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File28.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File29.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
File30.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File31.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File32.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File33.txt	NR	NR	NR	NR	NR	R	NR	NR	NR	R
File34.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File35.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File36.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File37.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File38.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR

File39.txt	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
File40.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
File41.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
File42.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File43.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
File44.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File45.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File46.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
File47.txt	NR	N R	NR	NR	R	NR	NR	NR	NR	NR
File48.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File49.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File50.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File51.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File52.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File53.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File54.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File55.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
File56.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
File57.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File58.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
File59.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File60.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
File61.txt	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
File62.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File63.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File64.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
File65.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
File66.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File67.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File68.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File69.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File70.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File71.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
File72.txt	NR	NR	R	R	NR	NR	NR	NR	NR	NR
File73.txt	NR	NR	R	NR	R	NR	NR	NR	NR	NR
File74.txt	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
File75.txt	R	NR	R	R	NR	NR	NR	NR	R	NR
File76.txt	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
File77.txt	R	NR	R	NR	NR	NR	NR	NR	R	NR
File78.txt	R	NR	R	NR	NR	NR	NR	NR	NR	NR
File79.txt	R	NR	R	NR	NR	NR	NR	NR	NR	NR
File80.txt	R	NR	R	NR	NR	NR	NR	NR	NR	NR
File81.txt	R	NR	NR	NR	NR	NR	NR	NR	R	NR
File82.txt	R	NR	NR	NR	NR	R	NR	NR	R	NR
File83.txt	NR	NR	R	NR	NR	NR	NR	NR	NR	NR

File84.txt	R	NR	R	NR	NR	R	NR	NR	NR	NR
File85.txt	R	NR	R	R	NR	NR	NR	NR	NR	NR
File86.txt	R	NR	R	NR	NR	NR	NR	NR	NR	NR
File87.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File88.txt	NR	NR	NR	NR	R	NR	NR	R	NR	NR
File89.txt	NR	NR	NR	R	NR	NR	NR	R	NR	NR
File90.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File91.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File92.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File93.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File94.txt	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
File95.txt	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
File96.txt	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
File97.txt	NR	NR	NR	NR	NR	R	NR	R	R	NR
File98.txt	NR	NR	NR	NR	NR	R	NR	R	NR	NR
File99.txt	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
File100.txt	R	NR	NR	NR	NR	R	NR	NR	R	NR
File101.txt	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
File312.txt	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
File201.txt	R	NR	R	NR	NR	NR	NR	NR	R	NR
file1003.txt	NR	NR	NR	R	NR	R	NR	NR	NR	NR
File910.txt	NR	R	NR	R	NR	NR	R	NR	NR	NR
File009.txt	NR	R	NR	R	NR	NR	R	NR	NR	NR

Keys: ➔R- Relevant

➔NR- Non-Relevant

Appendix B: List of Afaan Oromo Stop Words

aanee	**eennu**	**isa**	**jechuu**	**nurra**	**tahullee**	
agarsiisoo	**eennuu**	**isaa**	**jechuun**	**nurran**	**tana**	**yookaan**
akka	**enyu**	**isaaf**	**kan**	**nurraa**	**tanaan**	**yookiin**
akkam	**eenyuu**	**isaan**	**kana**	**nurraan**	**tanaaf**	**yoolinimoo**
akkasumas	**f**	**isaani**	**kanaa**	**nuti**	**tanaafi**	**yoom**
akkum	**faallaa**	**isaanii**	**kanaaf**	**nutti**	**tanaafuu**	
akkuma	**fagaatee**	**isaaniitiin**	**kanaafi**	**nuu**	**tahee**	
ala	**fi**	**isaanirraa**	**kanaafi**	**nuf**	**taahee**	
alatti	**fullee**	**isaanitti**	**kanaafuu**	**nufi**	**ta'ullee**	
alla	**fuullee**	**isaatiin**	**kanaan**	**nuuf**	**ta'uyyu**	
amma	**gajjallaa**	**isarraa**	**kanaatti**		**ta'uyyuu**	
ammaa	**gama**	**isatti**	**karaa**	**nulleen**	**tawullee**	
ammati	**gararraa**	**isee**	**kee**	**nuun**	**teenya**	
ammatti	**gara**	**iseen**	**keenna**	**nuy**	**teessan**	
ammaatti	**garas**	**ishee**	**keenya**	**odoo**	**tiyya**	
ammaattii	**garuu**	**ishii**	**keessa**	**ofii**	**turan**	
ammo	**giddu**	**ishiif**	**keessan**	**oggaa**	**turani**	
ammoo	**gidduu**	**ishiin**	**keessatti**	**oo**	**turaan**	
an	**gubbaa**	**ishiirraa**	**kiyya**	**osoo**	**turaani**	
ana	**ha**	**ishiitti**	**koo**	**otoo**	**turun**	
ani	**haa**	**ishiitti**	**kun**	**otumallee**	**turuni**	
ati	**hamma**	**isii**	**lafa**	**otuu**	**turuun**	
bira	**hanga**	**isiin**	**lama**	**otuullee**	**turuuni**	
booda	**haala**	**isin**	**malee**	**saaniif**	**too**	
booddee	**haaluma**	**isini**	**male**	**sadii**	**tti**	
dabalatees	**henna**	**isinii**	**manna**	**sana**	**utuu**	
dhaan	**hin**	**isiniif**	**maqaa**	**saniif**	**waa'ee**	
dudduuba	**hoggaa**	**isiniin**	**miti**	**si**	**waan**	
dugda	**hogguu**	**isinirraa**	**moti**	**sii**	**waggaa**	
dura	**hoo**	**isinitti**	**moo**	**siif**	**wajjin**	
duuba	**hoo**	**ittaanee**	**na**	**siin**	**warra**	
eega	**illee**	**itti**	**naa**	**silaa**	**woo**	
eegana	**immoo**	**itumallee**	**naaf**	**silaa**	**waawoo**	
eegasii	**ini**	**ituu**	**naan**	**simmoo**	**yammuu**	
ennaa	**innaa**	**ituullee**	**naannoo**	**sinitti**	**yemmuu**	
erga	**inni**	**jala**	**narraa**	**siqee**	**yeroo**	
ergii	**irra**	**jara**	**natti**	**sirraa**	**yommii**	
enu	**irraa**	**jechaan**	**nu**	**sitti**	**yommuu**	
ennu	**irraan**	**jechoota**	**nu'i**	**sun**	**yoo**	