Hawassa University
ሀዋሳ ዩኒቨርሲቲ

QUERY EXPANSION FOR AFAAN OROMO INFORMATION RETRIEVAL USING AUTOMATIC THESAURUS

MASTER OF SCIENCE THESIS

SAMUEL MESFIN BAYU

HAWASSA UNIVERSITY

HAWASSA, ETHIOPIA

NOVEMBER, 2021

QUERY EXPANSION FOR AFAAN OROMO INFORMATION RETRIEVAL USING AUTOMATIC THESAURUS

SAMUEL MESFIN BAYU

A THESIS SUBMITTED TO THE

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF INFORMATICS, SCHOOL OF

GRADUATE STUDIES, HAWASSA UNIVERSITY

HAWASSA, ETHIOPIA

IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE

DEGREE OF

MASTER OF SCIENCE IN COMPUTER SCIENCE

NOVEMBER, 2021

Declaration

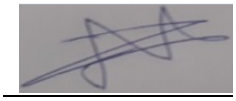
I hereby declare that this MSc Specialty or equivalent thesis is my original work and has not been presented for a degree in any other university, and all sources of material used for this thesis/dissertation have been duly acknowledged.

Name: - Samuel Mesfin Bayu.

Signature:-_____

This MSc Specialty or equivalent thesis has been submitted for examination with my approval as thesis advisor.

Name:-Melkamu Beyene (PhD)

Signature:- 

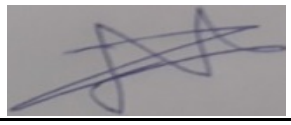
Place and Date of Submission:- _____

SCHOOL OF GRADUATE STUDIES

HAWASSA UNIVERSITY

ADVISORS APPROVAL SHEET

This is to certify that the thesis entitled **"Query Expansion for Afaan Oromo Information Retrieval Using Automatic Thesaurus"** submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science, the Graduate Program of the Department of Computer Science, and has been carried out by Samuel Mesfin Bayu ID. No PG/CoScR/0012/11 under my supervision. Therefore, I recommend that the student has fulfilled the requirements and hence hereby can submit the thesis to the department.



Melkamu Beyene (PhD) _____ _____

Name of major advisor Signature Date

SCHOOL OF GRADUATE STUDIES

HAWASSA UNIVERSITY

EXAMINERS APPROVAL SHEET

We, the undersigned, members of the Board of Examiners of the final open defense by Samuel Mesfin Bayu have read and evaluated his thesis entitled **"Query Expansion for Afaan Oromo Information Retrieval Using Automatic Thesaurus "** and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirements for the degree

_____	_____	_____
Name of the Chairperson	Signature	Date
Melkamu Beyene (PhD)	_____	_____
Name of Major Advisor	Signature	Date
_____	_____	_____
Name of Internal Examiner	Signature	Date
Michael Melese (PhD)	_____	_____
Name of External examiner	Signature	Date

Final approval and acceptance of the thesis is contingent upon the submission of the final copy of the thesis to the School of Graduate Studies (SGS) through the Department/School Graduate Committee (DGC/SGC) of the candidate's department.

_____	_____	_____
SGS Approval	Signature	Date

Acknowledgment

First and foremost, I have everlasting thank to Almighty God for keeping me alive and giving me the strength to accomplish this work. And I would like to thank my mother Saint Marry for all of the things she has done in my life. Next, I would like to express my sincere gratitude to my advisor Dr. Melkamu Beyene, for his encouragement and supervision in writing this thesis. Similarly, I am also thankful to all of the instructors who thought me at Hawassa University. I would like to thank my father Mesfin Bayu, my brothers Solomon and Anduwalem, my sister Yordanos Mesfin, for their endless love, support, and encouragement.

Lastly, my heartfelt thanks go to Moti G, Kebede N, and Belay T. for contributing to the achievement of this thesis by participating in the process of judging the quality of the automatic thesaurus. I am equally thankful to Wubayo Olani for her willingness to participate in relevance judgment.

May God bless you all!

Abbreviations

QE: Query Expansion

AFAAN OROMO: Afaan Oromo

IR: Information Retrieval

CBOW: Continuous Bag-Of-Words

TF-IDF: Term frequency-Inverse Document Frequency

PRF: Pseudo Relevance Feedback

RF: Relevance Feedback

LSI: Latent Semantic Indexing

Table of Contents

List of Tables

List of Figures

Abstract

Recently, the amount of textual information written in Afaan Oromo language is increasing dynamically. Likewise, the need to access the information also increases. But, it is difficult to retrieve and satisfy one`s own information need, because of the inability of the users to formulate a good query and the terminological variation or term mismatching among the world of readers and the world of authors. Hence, query expansion is an effective mechanism to reduce term mismatching problems and also to improve the retrieval performance of IR systems. The idea behind query expansion is to reformulate the user's original query by adding related terms. In this study, an automatic Afaan Oromo thesaurus is constructed from manually collected documents. After the text preprocessing tasks are performed on the document corpus, the preprocessed words are vectorized in multidimensional space by using Word2Vec`s skip-gram model. In which, words that share similar context have similar vector representation. Then cosine similarity measure was applied to construct the thesaurus. A one-to-many association approach was employed to select expansion terms. Hence top five terms that have the highest similarity score with the entire query were selected from the thesaurus and added to the original query of the user for query expansion. Then the reformulated query was used to retrieve more relevant documents.

Experimentations were performed to observe the quality of the constructed thesaurus and the effect of integrating query expansion into the Afaan Oromo IR system. The result shows that the constructed thesaurus generates related terms with average relatedness accuracy of 62.1%. *On the other hand, the integration of query expansion registered performance improvement by 14.3 % recall, 2.9 % F-measure, and performance decrement of 5.5% for precision.*

Keywords: Query expansion, information retrieval, thesaurus, Word2Vec, skip-gram.

Chapter One: Introduction

1.1 Background

Information has a crucial role in people's daily activities. Since the development of the World Wide Web, people have started sharing huge amounts of information over the internet. The information is written and published using different languages. Due to the explosion of information over the internet, retrieving and satisfying one's own information needs was very difficult. These problem motivated researchers around the world to explore a mechanism for searching and retrieving information over large collections. As a result, information retrieval systems were introduced. Information retrieval is the task of finding a material (usually documents) of unstructured nature that satisfies information needs (user query) from large collections [1]. In IR systems, peoples formulate queries to obtain whatever information they need. Then the system responds with a set of documents in response to the user`s query.

The goal of IR is to help users to obtain useful information that is relevant to their query with minimum effort regardless of the increasing complexity of information and dynamic nature of user query. Given a set of documents and user queries, IR systems find a ranked set of documents that are relevant to the query [1]. The result that the user obtains is highly dependent on the ability of the user to formulate a query. However, most users are not good at formulating queries [2].

In addition to the inability of the users to formulate a good query, most retrieval systems work by matching the keywords of the user query with the indexed terms of each document in the collection, which leads to a vocabulary mismatching problem. Due to this problem, retrieval systems face difficulty in retrieving the most relevant result. This is because concepts of the user query and/or the document can be expressed using different terminologies, this problem is called term mismatch or vocabulary problem. A term mismatch problem can be caused by a combination of synonyms and polysemy terms [3].

Synonyms are multiple words that have the same meaning and polysemy is a word that has more than one meaning whose meaning depends on the context it is being used. In term- matching based retrieval systems, the retrieval effectiveness of the system is highly reduced since most relevant results are not retrieved in response to the user query.

The recent focus in the area of information retrieval research is to improve the performance of retrieval systems. To deal with the problem of term mismatching and to increase the performance of retrieval systems, query expansion can be taken as one of the ideal solutions. Query expansion is an approach used to improve the term mismatch between short user queries and relevant documents [4]. In query expansion, the user's original query is reformulated by adding new terms that are significant for the performance improvement of the retrieval system [3]. Terms that are synonyms with or related to original query terms are added to the query, and then the reformulated query or expanded query is used to retrieve more relevant results. This process is called query expansion [1]. New terms that are added to the original query helps to retrieve more relevant results and greatly improves the effectiveness of retrieval systems.

Based on data sources used, query expansion approaches are broadly classified into two, namely Global analysis and Local analysis [3]. In a global analysis, the source of expansion term is a hand-built knowledge resource or a document corpus of a large collection. In this approach, expansion terms are obtained either by examining term correlations in the entire corpus or by building thesaurus manually or automatically [5]. On the other hand, local analysis adjusts the query by using the documents returned as relevant in response to the original query of the user. These documents are used as a source of expansion terms for expanding the query [6, 1].

Thesaurus can be defined as a data structure that defines semantic relatedness between words [7]. According to [1] [8] thesaurus can be constructed manually or automatically. Manual thesaurus is a set of synonymous names for concepts that are built manually by humans [1]. WordNet is one example of a manually constructed thesaurus. An automatically constructed thesaurus is a thesaurus that is automatically built by using word co-occurrence statistics over a collection of documents in a domain. The correlation between words shows the relationship between them. Manual construction of thesaurus is costly. It is time-consuming, requires deep domain knowledge and it is difficult to build unless experts are involved. Therefore, methods that automatically extract synonyms and other word relations from a large collection of documents are more preferable.

Recently the concepts of distributional semantics are widely applied to many areas of natural language processing and information retrieval tasks. According to [9] distributional methods can also be applied for automatic thesaurus generation. These methods work based on the idea that the

meaning of a word is related to the distribution of words around it [10]. In distributional semantics, it is believed that words that are used in similar contexts will have similar meanings. Distributional semantic models are classified into two: co-occurrence-based models and predictive models [11]. The former is trained over the entire corpus to capture global dependencies and contexts while the latter captures local dependencies within a small context window [11].

Word embedding is an approach for word vector representations that can capture syntactic and semantic relationships between terms [12]. It is a representation of text where words that have the same meaning have a similar representation. Word2vec is a well-known word embedding model based on distributional semantics [13]. The model takes a large text corpus as input and produces a vector space in which each unique word is assigned a corresponding vector in the space [14]. It is a predictive model that trains word representations in such a way that words are embedded in space along with similar words based on their context [15]. The Word2Vec algorithm consists of two models namely: the continuous bag-of-words (CBOW) model and the skip-gram model [11]. The CBOW model is trained to predict a target word, given a context word c in both the left and right sides of the target word. Whereas the skip-gram model is the opposite of the CBOW model [14]. It is trained to predict the nearby context words given the target word [13]. Even though both models are efficient for learning high-quality vector representations from large amounts of unstructured text data, skip-gram architecture is better for capturing semantic relationships between words [11].

Afaan Oromo is a language from the Cushitic language family and it is spoken by the largest ethnic group in Ethiopia. It is one of the languages with the largest number of speakers in Africa after Arabic and Hausa [16]. It is spoken in the horn of Africa including in Ethiopia, Kenya, and Somalia. The language is also called Oromiffa [17]. It has been a working language of Oromia regional state and as a medium of instruction for primary schools across the regional state [18]. These days' huge amounts of documents are being published in Afaan Oromo and it is shared and available online over the internet. According to [19] Afaan Oromo text retrieval system is highly affected by term mismatching problem. This is because the retrieval systems works merely based on term matching. Term mismatching problem occurs because there are a lots of synonym and polysemy terms in the language due to its dialectic nature. Therefore, due to term mismatching between user query and relevant documents, the performance of Afaan Oromo text retrieval system

is highly reduced. To reduce the effect of term mismatching problem and enhance the performance of Afaan Oromo IR system, integrating query expansion into Afaan Oromo IR system can solve this problem.

The focus of this study is to construct an automatic Afaan Oromo thesaurus by applying the concept of distributional semantics and use them as a source of expansion terms for expanding user queries. This study integrates a query expansion system into the Afaan Oromo IR system to enhance its performance effectiveness.

1.2. Motivation

Afaan Oromo language is dialectic language. The language is said to have six major dialects. Because the language lacks standard form, all the dialects are used for writing documents and also in mass media. The dialectic nature of the language accounts for variation in vocabulary and pronunciation which in turn resulted in large amount of synonym words in the language. When user formulate their query by using synonym words, term mismatching based Afaan Oromo text retrieval system miss relevant documents that even contain words that are synonym to the query of the user. Therefore, the performance of Afaan Oromo text retrieval system is highly reduced.

The reduced performance of Afaan Oromo text retrieval systems because of term mismatching problem motivated this study to integrate query expansion into Afaan Oromo text retrieval system. Query expansion helps to reduce the effect of term mismatching problem and enhance the performance of Afaan Oromo text retrieval system.

1.3. Problem Statement

Since the Afaan Oromo language got an official script, a huge amount of documents are being written and published. To enable speakers of the language to get access to those documents with minimum effort some studies have been done in the area of Afaan Oromo text retrieval systems. The first attempt was done to develop the Afaan Oromo search engine [20]. In the study, the researcher has identified that although general-purpose search engines such as Google and yahoo allows users to search using some other languages other than English, they have limitation in considering some special characteristics (such as stemming and acronyms) that are specific to the

languages since they are originally designed for English. Therefore, the study attempted to fill this gap by building the Afaan Oromo search engine.

The other work was done by Gezehagn [19]. In the study, the researcher tried to fill the gap of the Cross-language retrieval system that makes use of Afaan Oromo queries to retrieve text written in the English language. The author stated that users who are searching for an Afaan Oromo document by using a query written in Afaan Oromo do not have a suitable environment to obtain the information they need. To fill this gap, the study attempted to develop Afaan Oromo text retrieval system, in this work the researcher developed a retrieval system that merely depends on term matching and stated that the performance of his study was affected by the term mismatching problem. To solve this problem, the study recommended that query expansion could greatly improve the performance of the retrieval system by controlling this effect.

In most collections, the same concepts can be expressed using different terminology. This happens because of a property of words called synonymy. Multiple words can have the same meaning, these words are said to be synonyms [21]. Because of the difference in terminology between queries of the user and terms that appear in the document, term mismatch or vocabulary problems will occur. Retrieval systems that merely depend on term matching are highly affected by this problem. Therefore, relevant documents cannot be retrieved in response to a user`s query. Hence, synonym words are responsible for the reduction of the performance of information retrieval systems. They reduce the percentage of recall and precision of the retrieval systems [3]. According to [19] Afaan Oromo information retrieval system is also affected by such kinds of problems.

Some studies have been attempted to control the effect of the term mismatching problem in Afaan Oromo IR systems. Berhanu [22] applied latent semantic indexing that needs to identify the relationship between a set of documents and words depending up on the assumption that words that are close or related in meaning will appear in the same document [11]. LSI analyzes word association within a set of documents by forming a term-by-document matrix. Since the matrix is very large, SVD (singular value decomposition) was used for dimensionality reduction. In his study, the researcher used only 70 text documents and nine queries for evaluating the system.

The other study conducted by Tesfaye [23] applied thesaurus-based semantic compression for Afaan Oromo text retrieval. In which Afaan Oromo thesaurus was constructed manually where

terms with the highest TF-IDF value are used as a descriptor or key terms. The key terms were used to represent synonym terms in the thesaurus, and then a semantic compression algorithm was applied for indexing the document. The algorithm checks whether a given term is represented as a key term or synonym term in the thesaurus, if it is used as a key term it is used for indexing otherwise its respective descriptor is searched from the thesaurus and used for indexing by replacing the synonym term. The study used a corpus size of 106 text documents for constructing the thesaurus and for evaluating the system.

Although previous works stated above attempted to increase the retrieval performance of Afaan Oromo IR systems, the size of the corpus they used is not enough for calculating semantic similarity, and also manually constructing thesaurus is very difficult for large collections. Moreover, it is very expensive to get a good general-purpose thesaurus [24]. On the other hand, although co-occurrence-based methods such as latent semantic indexing (LSI), are very good at capturing the word co-occurrence statistics of the corpus; they are poor at capturing syntactic and semantic regularities [25]. Additionally, LSI is very slow for large collections [11]. Hence, a fast and efficient technique that can capture semantic relationships between terms is required. To this end as far as our knowledge no attempt has been made to enhance the performance of Afaan Oromo IR systems by expanding the queries of the user.

To fill this gap, we proposed query expansion for the Afaan Oromo IR system by using an automatically generated thesaurus. Afaan Oromo thesaurus is constructed automatically from the entire document collection by applying the concept of distributional semantics. Which works based on the idea that the meaning of words can be derived from the context in which the word appears. That is, words that are used in similar contexts have similar meanings. The constructed thesaurus is used as a source of expansion terms for expanding the query and then the expanded query is used to retrieve more relevant documents and increase the performance effectiveness of Afaan Oromo IR systems.

To this end, this study tries to answer the following questions:

- To what extent the generated thesaurus terms are related to the given term?

- To what extent query expansion improves the performance of Afaan Oromo retrieval systems?

1.4. Objective

In this section, the general and specific objectives of the study are discussed.

1.4.1. General objective

The general objective of this study is to design and develop query expansion for Afaan Oromo information retrieval using automatic thesaurus.

1.4.2. Specific objective

In order to accomplish the general objective of this study the following sub-tasks will be done

- To review literatures for building both theoretical and computational foundations.
- To collect and pre-process a corpus for training and testing.
- To develop thesaurus automatically.
- To design the architecture of query expansion using automatic thesaurus.
- To develop the prototype of the system.
- To evaluate the performance of the system.
- To report the findings of the study.

1.5. Scope and limitation of the study

This study aims to resolves the effect of term mismatching or vocabulary problem and improve the performance of Afaan Oromo text retrieval system by integrating query expansion technique that make use of automatic thesaurus as a source of expansion term.

The number of expansion terms used to expand the query of the user is restricted to five, this is in order to reduce the complexity of the evaluation process and also to add top related terms to the query. On the other hand, the study used a limited corpus size for evaluating the performance of the system developed since relevance judgment preparation for a large corpus collection is time-consuming. Additionally, the quality of the constructed thesaurus was measured by language experts because of lack of reference lexicons with which the generated thesaurus terms are compared.

1.6. Significance

One of the major importance of query expansion is that it increases the chance to retrieve the relevant information on the internet, which is not retrieved otherwise using the original query. Hence, integrating query expansion into Afaan Oromo IR system helps to enhance the performance effectiveness of the retrieval system. In addition to this, the query expansion mechanism can be applied to a wide variety of Afaan Oromo language processing and retrieval systems such as Question Answering (QA) system to overcome the mismatch problem and to improve document retrieval. It can also applied into Cross-Language Information Retrieval systems, where query expansion can be used during query translation to overcome translation errors. It can also be used in other systems such as multimedia information retrieval systems, information filtering, plagiarism detection, etc. [3].

1.7. Organization of the thesis

This thesis work is organized into six chapters. The first chapter discusses the background of the study, motivation, statement of the problem, the general and specific objective, scope and limitation, and finally the significance of the study. The second chapter gives a review of literature that is relevant for this study covering a brief overview of IR systems, query expansion, query expansion approaches, distributional semantics, word embedding approaches, similarity measures, and evaluation methods. Additionally, an overview of Afaan Oromo language that includes about Afaan Oromo writing system, morphology, and challenges of Afaan Oromo language for IR system are also discussed. Finally, a review of some prior works on query expansion for foreign languages, and previous works that are done to tackle term mismatching for Afaan Oromo language and prior works on query expansion for Amharic and Tigrigna languages are presented. The third chapter presents research methodologies that are applied for this study. Chapter Four discusses a detailed description of the design and implementation of the proposed system. The fifth chapter discusses the experimentations and discussion of the result obtained. Finally, the sixth chapter presents the conclusion, recommendations and gives directions for future research.

Chapter Two: Literature Review

In this chapter, literature works that are useful for this study are reviewed. This is in order to show the overview of the basic concepts of information retrieval and to understand query expansion, the resource used, and the existing approaches of QE. Additionally, an overview of thesaurus construction methods, approaches for automatic thesaurus construction, basic concepts of distributional semantics, word embedding methods, and widely used similarity measures are also discussed. In addition to this literature and books written about the Afaan Oromo language are reviewed to provide an overview of the language and its challenge for information retrieval systems. Lastly, literature works that are related to this study specifically those that are done to control term mismatch or vocabulary problems for both Afaan Oromo and other languages are reviewed.

2.1. Overview of Information Retrieval Systems

People started archiving and finding information thousands of years ago. It became more easily achievable with the invention of computers, hence people started to store large amounts of information. Therefore, a way of finding useful information from such a collection is required. This leads the field of information retrieval to be born in the 1950s [24]. Since then, the amount of published information is increasing rapidly. As the amount of published information is growing, managing the information becomes more difficult. This situation is called information overloading. Information retrieval systems are used to reduce the effect of what has been called information overloading.

Information retrieval is a broad area of computer science that is aimed at providing information to the user according to their interest [25]. “Information retrieval can be defined as the process of finding material, usually documents of unstructured nature (usually text) that satisfies an information need from within a large collection (usually stored on computers)” [1]Information retrieval is about the representation, organization, storage and access to information items such as documents, web files, online catalogues, multimedia objects and so on [25]. The goal of information representation and organization is to facilitate easy access.

IR systems serve as a bridge between the world of authors and the world of readers. The authors express their idea by writing documents and while the reader seeks information by formulating a query, IR systems are used to retrieve the most relevant document that matches the user query. But an IR system that merely depends on term matching is inefficient due to term mismatching or vocabulary problems. This problem happens because the same concept can be expressed using different terminologies. Hence, there can be term variations in user queries and the documents in the collection. Synonym and polysemy words are responsible for this variation [1]. When a user formulates a query by using a synonym word that does not appear in the document, term matching-based retrieval systems miss the relevant document even when its synonym word, expressing the same concept appears in the document.

2.2. Query Expansion

In most collections, the same concepts can be expressed using different terminology. This happens because of a property of words called synonymy. Synonym words are words that have the same meaning [21]. Words that exhibit such property greatly reduce the effectiveness of retrieval systems by reducing the number of relevant documents retrieved in response to the user query. This kind of problem is known as term mismatching or vocabulary problem [3]. Query expansion is an approach used to tackle this kind of problem.

Query expansion is a technique for improving term mismatching between queries formulated by the user and relevant documents [26]. In query expansion, after stop words are removed from the query, the user's original query is expanded by adding new terms that are semantically related or synonym with terms in the query. For each term t in a query, synonyms or related terms are added to expand the query automatically [1]. Then the reformulated query or expanded query is used to retrieve more relevant results [3]. This enhances the performance effectiveness of the retrieval systems [6]. The choice of data source for obtaining terms that are added to expand the query is the key part of query expansion.

2.2.1. The process of query expansion

The process of expanding a query involves four steps: preprocessing of data source, term weighting and ranking, term selection, and query reformulation [3]. The preprocessing activities applied depends on the type of data source used for obtaining expansion terms. The most common

preprocessing activity includes tokenization, stop word removal, and stemming [4]. Once the data source is preprocessed, the next step is term weight and ranking. The relevance of the candidate term to a query is determined by its weight. According to [4] the approaches for weighting and ranking candidate expansion terms are classified into four:

- **One-to-one association**: - in which each candidate expansion term is related at least to one query term. In this approach, weight and rank are given to candidate terms based on the one-to-one association between query terms and candidate expansion terms. The most common approach to establish a one-to-one association between query terms and candidate expansion terms is to employ term similarity measures. Then terms that have the highest score are taken as expansion terms.
- **One-to-many association**: - in which each candidate expansion term is correlated to multiple query terms. The candidate expansion term is added to the user query if it is related to multiple terms of the original query. If the relationship of candidate terms with the entire query or with multiple query terms needs to be considered, the one-to-many association should be used [3].
- **Feature distribution of top-ranked documents:** - which considers top-weighted terms that are extracted from top retrieved documents in response to the user`s original query.
- **Query language modeling**: constructs statistical language model over the term collection. The widely used language model is the relevance model. It works based on top retrieved documents in response to the original query. The language model determines the probability of a term in relevant documents collection based on its co-occurrence with the query terms. Then the terms with the highest probability are chosen as expansion terms [27].

After term weighting is completed the terms with the highest score are selected as expansion terms to expand the query. Then the query is reformulated by adding expansion terms to the original user query.

2.3. Query Expansion Approaches

There are several query expansion approaches but, based on the data source used all methods can be categorized under two broad classes. Hence, query expansion techniques are classified into two namely, local analysis and global analysis [24] [3] [6].

2.3.1. Local Analysis

In local analysis methods, documents that are obtained in response to the user's original query are used as a source for the selection of expansion terms [3]. In this method, users' unmodified (initial) query is used to retrieve relevant documents, and then from the set of documents retrieved expansion terms are selected. Once the expansion terms are selected, the query will be expanded by combining them with initial query terms. In the local analysis approach, there are two ways for expanding a query: relevance feedback (RF) and pseudo relevance feedback (PRF) [24].

2.3.1.1. Relevance Feedback (RF)

Relevance feedback is a query expansion approach that involves the user in the retrieval process [1]. Since relevance feedback requires direct interference of the user, obtaining information about the set of documents that are relevant to a query is very expensive. This is due to the unwillingness of users to provide feedback information. To overcome this problem a group of specialists used to make the relevance assessment [24]. In this approach, first, the user issues a query, a set of documents are retrieved in response to the query, then the user marks the result of the system as relevant or irrelevant, the system represents the information need or the query based on user feedback and then retrieve the revised set of the retrieval result. This kind of one or more iteration might be carried out. [1] [24] Here, the intention is to modify the query in such a way that the modified query will be moved towards relevant documents and away from non-relevant ones.

Relevance feedback for vector model: Rocchio's method

The application of relevance feedback to a vector model works based on an assumption that documents that are identified as relevant for a given query, have similarities among themselves. The core idea is to reformulate the query in such a way that it gets closer to the neighbourhood of the relevant documents and away from the neighbourhood of non-relevant documents in the vector space [25]. The best query vector for distinguishing relevant documents from non-relevant

documents where the complete set of relevant documents for a given query is given by the following equation.

$$\vec{q}_{opt} = \frac{1}{|C_r|}\sum_{\forall \vec{d}_j \in Cr} \vec{d}_j - \frac{1}{N-|C_r|}\sum_{\forall \vec{d}_j \notin Cr} \vec{d}_j \quad \text{(1.1)}$$

Where C_r: is the set of relevant documents to a given query, $\vec{d}_j$ is a weighted term vector associated with document d_j, N is the number of document collection and $\vec{q}_{opt}$ is the optimal weighted term vector for a query q. The problem of equation (1) is that obtaining the set of all relevant documents (C_r) for a given query q, is something that is not unachievable [1]. To overcome this problem, formulating the initial query and incrementally changing the initial query vector is taken as a possible solution [25]. The incremental change is done by limiting the computation to the documents known to be relevant. There are three classic and similar ways to calculate the modified query vector $\vec{q}_m$ as follows

Standard_Rocchio: $\vec{q}_m = \alpha\,\overrightarrow{q} + \frac{\beta}{N_r}\sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \frac{\gamma}{N_n}\sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$ ……. (1.2)

Ide_regular: $\vec{q}_m = \alpha\,\overrightarrow{q} + \beta\sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma\sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$ ……. (1.3)

Ide_Dec_Hi: $\vec{q}_m = \alpha\,\overrightarrow{q} + \beta\sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma\, Max_{Rank}(D_n)$ ……. (1.4)

Where $Max_{Rank}(D_n)$ is a reference to the highly ranked non-relevant document and α, β, γ are tuning constant. Rocchio fixed α = 1 and Ide fixed α = β = γ = 1. According to [25] relevance feedback for a vector model is simple and returns a good result. This is because modified term weights are directly calculated from the set of retrieved results and good results are obtained.

Relevance feedback for probabilistic model

In the probabilistic model, documents are dynamically ranked according to their similarity to query q by using probabilistic ranking principles [25]. In the probabilistic model, the similarity of document j to a query q is given by:

$$sim(d_j, q) = \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{P(k_i|R)}{1 - P(k_i|R)}\right) + \log\left(\frac{1 - P(k_i|\bar{R})}{P(k_i|R)}\right) \quad \text{(1.5)}$$

Where $P(k_i|R)$ the probability of observing term k_i in the set R of relevant documents and $P(k_i|\bar{R})$ is the probability of observing term k_i in the set $\bar{R}$ of non-relevant documents. Unlike the procedure in the vector space model, there is no query expansion in the probabilistic model instead; the original query terms are reweighted using feedback information given by the user [25].

2.3.1.2. Pseudo Relevance Feedback (PRF)

The other approach for query expansion under local analysis is pseudo relevance feedback. In contrast to relevance feedback, in pseudo relevance feedback, the user is not directly involved in the query refinement process [1]. As it automates manual parts of relevance feedback, it is also called blind relevance feedback [25]. In this approach the user`s original query is used to retrieve the initial set of relevant documents, then top k ranked documents are assumed as relevant and used as a source of expansion terms [3]. There are two query expansion strategies based on implicit user feedback: local clustering and local context analysis.

Implicit feedback through local clustering

In this approach by using the user`s original query q, a set of documents are retrieved as relevant D_l By using unique words or vocabulary words (v_l) from the set of documents retrieved and documents retrieved, the frequency of a term $k_i \in v_l$ in the document d_j, $d_j \in D_l$ is called $f_{i,j}$ is calculated. Then term-document matrix $M_l = [m_{i,j}]$ is constructed using vocabulary words and documents in the relevant set. Given that M_l^T is the transpose of a matrix M_l, and then the matrix $C_l = M_l M_l^T$ is a local term-term correlation matrix. Each element $c_{u,v}$ express the correlation between terms K_u and K_v. Based on the co-occurrence of terms inside the document set that are retrieved as relevant in response to the user`s original query, a term relationship is established. This correlation is used to calculate the local cluster of neighboring terms. Then terms that belong to clusters associated with query terms are used to expand the user`s original query [25].

Implicit feedback through local context analysis

Local context analysis is a technique that combines local feedback and global analysis for query expansion [6]. It is based on noun groups, i.e., single nouns, two adjacent nouns, and three adjacent

nouns in the text. Noun groups that are obtained from top-ranked documents are treated as document concepts. Concepts are chosen based on their co-occurrence with query terms for expanding user queries. However, the best passages are used for determining the term co-occurrence instead of the whole document [25]. The local context analysis technique works in three steps:

- First, top k ranked passages are retrieved using the original query. This is done by breaking up the documents retrieved into a text window of fixed size and ranking them as if they were documents.
- Second, the similarity ***sim (q, c)*** between each concept in the top-ranked passages and the whole query q (not individual terms) is calculated using the following formula.

$$sim(q,c) = \prod_{ki \in q} \left(\delta + \frac{log(f(c,ki) \times IDFc)}{log\, n} \right)^{IDFi} \quad \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (1.6)$$

$f(c,ki)$ Is the function that quantifies the correlation between concept c and query term k_i and it is given by:

$$f(c,k_i) = \sum_{j=1}^{n} pf_{i,j} \times pf_{c,j} \quad \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots .. (1.7)$$

$$IDF_i = \max\left(1, \frac{\log_{10}(N/np_i)}{5}\right) \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (1.8)$$

$$IDF_c = \max\left(1, \frac{\log_{10}(N/np_c)}{5}\right) \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots .. (1.9)$$

Where n is the number of top-ranked passages, $pf_{i,j}$ and $pf_{c,j}$ are the frequency of a term k_i in the jth passage and frequency of concept c in jth passage respectively, IDFc and IDFi are inverse concept frequency and inverse term frequency respectively. npi and npc represents the number of passages containing the term k_i and the number of passages containing concept c.

- The final step is to expand a query using top k ranked concepts retrieved according to $sim(q, c)$ calculation. The concept added is assigned a weight given by $1 - 0.9 \times \frac{i}{k}$

  Where i is the position of the concept in the concept ranking. Whereas the terms in the original query are assigned a weight of 2.

2.3.2. Global Analysis

In local analysis methods first, the user formulates a query and in response to that query certain documents are retrieved then these documents are used as a source for obtaining expansion terms whereas, in global methods, the user query is expanded independent of the user`s original query and results returned from it [3]. In a global analysis, expansion terms can be obtained from some form of thesaurus [1]. Thesaurus can be defined as a book of words together with their synonyms[1]. It can also be defined as a data structure in which semantic relationships between terms are defined [7]. According to [1] [8], there are two methods for constructing thesaurus namely: manual thesaurus construction and automatic thesaurus construction.

Manual thesaurus is a set of synonymous names for concepts that are built manually by humans [1]. In a manually constructed thesaurus, first, the subject area has to be clearly defined. Once the subject area is known, terms are collected from various sources to form a vocabulary, and then, for each term, its related vocabulary including synonyms, broader and narrower terms are identified. Finally, terms together with their relationships are structured into some form of structure such as hierarchy [8]. WordNet is a good example of a manually constructed thesaurus.

On the other hand, an automatically constructed thesaurus is a thesaurus that is automatically built from a large collection of documents. There are several methods to automatically construct a thesaurus. But all of the methods work based on co-occurrence between terms. The co-relation between words shows the relationship between them. Therefore, it is used for selecting expansion terms for expanding the query [3]. The selected terms have some semantic relationship with terms in the query.

[1] Merriam-Webster dictionary

When thesaurus is used as a source of expansion terms, for each query term t, terms that are synonyms and/or related to term t, from the thesaurus are used for automatically expanding the query. Thesaurus-based query expansion does not need any user input [1].

2.4. Approaches to Automatic Thesaurus Construction

Automatic thesaurus construction is preferable over manual thesaurus construction. This is because of the fact that manually constructed thesaurus are costly, labour-intensive, and difficult to update. Several approaches were followed to construct thesaurus automatically. In this subsection, some approaches that are successfully applied for automatic thesaurus construction are discussed as follows.

2.4.1. Document clustering

Document clustering can be used for automatic thesaurus construction. In this approach, a thesaurus is constructed by considering the occurrence of terms in a document. First documents are clustered into small clusters based on similarity measures by using clustering algorithms. Two documents are said to be similar if they share a sufficient number of terms. Once the documents are clustered, then terms are grouped by their occurrence in the document clusters [28].

2.4.2. Co-occurrence

A co-occurrence-based thesaurus is constructed based on the notion that words that co-occur in a document are likely to be similar in some sense [11]. The most common way to compute co-occurrence-based thesaurus is by using term-term similarities. In this approach, the first step is to construct term-document matrix M, where each cell $M_{t,d}$ is weighted count $w_{t,d}$ for term t and document d. Then if we compute $C = MM^T$, then $C_{u,v}$ is a similarity score between a term u and v [7]. The larger the score, the more the two terms are similar [1]. Based on the similarity score, terms that are closer to a given term are generated as thesaurus terms.

2.4.3. Lexical co-occurrence

Lexical co-occurrence-based thesaurus construction is a corpus-based approach for automatically generating thesaurus from a text corpus. Two terms are said lexically co-occur if they appear within some distance of each other (typically, if they co-occur in a window of k words). The assumption here is that words that have similar meanings will occur with similar neighbours. For generating

thesauri automatically, several researchers have used lexical co-occurrence. For instance, Jing and Croft [29] have used lexical co-occurrence information to build a thesaurus. Schutze et.al [7] also applied lexical co-occurrence for constructing thesaurus automatically from a text corpus. In these studies, the lexical co-occurrence of each term was represented as a vector in multidimensional space, and co-occurrence statistics information is captured. Then, similarity measures were applied on terms by comparing their corresponding vectors. Finally, for a given term synonyms or related terms that are nearest neighbours to the term based on their similarity score are generated as thesaurus terms.

2.4.4 Bayesian Network

Automatic construction of thesaurus is usually accomplished by extracting term relationships automatically. Statistical analysis is the popular method for extracting term relations. However, for low-frequency terms, statistical information cannot be reliably used for deciding the relationship of terms. This problem is known as the data sparseness problem. Data sparseness is the major problem of statistical methods in automatic thesaurus construction. Since low-frequency terms are crucial for thesaurus construction, Young et.al. [30] Proposed a method that makes use of a Bayesian network to deal with a data sparseness problem. A Bayesian network built from local term dependencies can give a probabilistic similarity distribution among the terms. The distribution is different from that of the most frequent probabilities but will be used to classify terms.

2.5. Query expansion using a manually constructed thesaurus

The most widely used form of query expansion in the global analysis involves the use of some kind of handcrafted thesauri such as dictionaries, WordNet, and Ontology [3]. These thesauri are used as a source for obtaining expansion terms [1]. Terms that are synonyms and/or related to each query term are selected from the thesaurus, then added to the query in order to increase the percentage of relevant documents retrieved in response to a given query.

WordNet is a lexical database originally developed for the English language by Princeton University. It organizes nouns, verbs, adjectives, and adverbs into groups of synonyms called synsets and describes the relationship between them. It contains more than 118,000 different word forms and more than 90,000 different word senses [21]. WordNet has versions in many languages

and it can be built for any language [31]. It is widely used in word sense disambiguation and query expansion. Reference [32], [33], [34] have used WordNet as a source of expansion terms for query expansion. In the expansion process, key phrases are identified and extracted from the user query and mapped into the WordNet synsets. The synsets with the highest similarity score will be selected for expanding the query term [35]. According to [21] if the definition of two terms in a WordNet shares many common words, the two terms are said to have a strong relationship.

The other manually constructed resource that can be used as a source for query expansion is Ontology. An ontology is a type of structured vocabulary in which the terms and the logical relationships that hold between them are well-defined [36]. Ontologies can be general or domain-specific [37]. Domain ontology models a specific domain such as tourism, agriculture, etc. It represents the particular meanings of terms as they apply to that domain. Ontologies have been widely used as a source of candidate expansion terms for query expansion. Reference [35], [36], [38] have used domain-specific ontologies constructed for agriculture, vegetable E-commerce, and tourism domains respectively for enhancing their respective information retrieval systems.

Although using manually constructed thesaurus such as WordNet and ontologies as a source of expansion terms can enhance retrieval effectiveness, they are usually domain-specific and requires to be updated frequently for terminological developments [3]. Moreover, its construction is very difficult, labor-intensive, and time-consuming [37].

2.6. Query expansion using automatic thesaurus generation

Manual construction of thesaurus is costly as it is time-consuming, requires deep domain knowledge and it is difficult to build unless experts are involved. Therefore, methods that automatically extract synonyms and other word relations from a large collection of documents are more preferable. This is because they have the advantage of not involving humans in their construction and they cover all the terms in the entire collection. The most widely used approach for automatic thesaurus construction is word co-occurrence. The co-occurrence can be document-level co-occurrence that is based on the idea that words that co-occur in a document are likely to be similar in some sense [11] or lexical co-occurrence (co-occurrence in a window of k words) which works based on assumption that words with similar meaning will occur with similar neighbors [7].

According to [9] distributional methods can also be applied for automatic thesaurus generation. These methods work based on the idea that the meaning of a word is related to the distribution of words around it [10]. That is, the meaning of a word can be obtained from a set of words with which the word occurs. The details of distributional methods will be discussed in the following section.

2.7. Distributional Semantics

The terms distributional, context-theoretic, corpus-based, and/or statistical can all be used to characterize a rich family of methods to semantics that shares a "usage-based" view on meaning. The perception of distributional methods is that the meaning of a word is related to the distribution of words around it [9]. The assumption here is that the statistical distribution of words in context plays a key role in characterizing their semantic behavior [10]. The distributional hypothesis states, "The degree of semantic similarity between two linguistic expressions A and B is a function of the similarity of the linguistic contexts in which A and B can appear." Meaning, words used in similar contexts have similar meanings. That is, if two words often occur with the same set of words, then they are said to be semantically similar [11].

2.7.1 Distributed word representations

Textual data must be mapped to real-valued vectors in order to be ready to be used by algorithms. The process of mapping text into real-valued vectors is called *feature extraction* or *vectorization* [15]. There are several methods to do so, such as bag-of-words, one-hot, frequency, and TF-IDF encodings are some of them. However, they all ignore word order that is, the contexts in which the word appears. Therefore, they have no mechanism for capturing the meaning of words in the document. The ideal solution to this problem is word embedding.

A method for estimating continuous representations of words learned by neural network models is known as a neural embedding or simply embedding. It works based on the idea of distributional semantics [10] and it serves as a core representation of texts in the deep learning approach [11]. It is a representation of text, in which terms that have the same meaning have similar representation. Then the resulting representation is a word vector.

Word vectors, also known as distributed word representations can be defined as a numerical vector representation of word semantics or meaning, including the actual and implied meaning [39] [40]. When texts are encoded along a continuous scale with a distributed representation, the resulting word vector is not a simple mapping from token position to token score like that of bag-of-words, one-hot, or TF-IDF encodings. Rather, it is represented in a space that has been embedded to represent word similarity [15].

Uday Kamath et.al. [11] Divides distributional semantic-based models for learning word vectors into two co-occurrence-based and predictive models. Co-occurrence-based models are global matrix factorization methods that are trained over the entire corpus in order to capture global dependencies and context, whereas predictive models are local context window methods that are trained to capture local dependencies within a (small) context window. Techniques that make use of the concept of distributional semantics will be discussed in the following subsections.

2.7.1.1. Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a technique that uses word co-occurrence to identify topics within a set of documents [25]. It works based on the notion that words that are close in meaning appear in a similar piece of text [11]. In the attempt to identify the relationship between a set of documents and words, LSI analyzes the association between words by building a term-document matrix where each cell can be the frequency of occurrence or TF-IDF of a term within a document [11]. The matrix created can be very large, as a result, dimensionality reduction techniques such as Singular-value decomposition (SVD) are applied to find low-rank approximation. This low-rank space can be used to identify key terms and cluster documents for information retrieval [1].

Even though co-occurrence-based methods such as LSI are good at capturing term co-occurrence statistics over the entire corpora, they are highly dimensional, hence requires large storage spaces and computational time (very slow for a large collection of documents). To handle this problem dimensionality reduction techniques such as SVD or Random projection are usually applied. That is why co-occurrence-based methods perform poorly in capturing semantic word regularities [11].

2.7.1.2. Word2Vec

In 2013, Google researchers led by Tomas Mikolov et al. [13] proposed a technique that makes use of neural network architecture for computing a continuous representation of words called

word2Vec. The algorithm trains word representations in such a way that words are embedded in space along with similar words based on their context [15]. Hence, it is an efficient method for learning high-quality vector representations of words from large amounts of unstructured text data [13]. Word2Vec is computationally efficient since it does not involve dense matrix multiplications [13]. In word2vec embedding, a distributed representation of a word is used [11] [15]. Word2Vec consists of two models namely: the continuous bag-of-words (CBOW) model and the skip-gram model [11]. Both models are efficient for learning high-quality distributed vector representations that capture meaningful syntactic and semantic word relationships [41].

A. Continuous bag-of-words model

CBOW model is trained to predict a target word, given a context word c in both the left and right sides of the target word [39]. Imagine a sliding window over the text that includes the central word or target word, together with the c words to the left and right side of the target word. The context words form the input layer. Each word is encoded in a one-hot representation. The input layer maps each context word through an embedding matrix W to a dense vector representation of dimension *k*, and the resulting vectors of the context words are averaged across each dimension to yield a single vector of *k* dimension. The embedding matrix **W** is shared for all context words. The training objective of CBOW is to maximize the conditional probability of observing the actual output word (the target word) given the input context words, together with the weights. Thus, the CBOW model seeks to maximize the average log probability [11]:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c<j<c, j\neq 0}\log\left(p\left(w_t\,|w_{t+j}\right)\right) \quad (2.11)$$

Where *c* is the number of context words to each side of the target word.

B. The Skip-gram model

The skip-gram model is the opposite of the CBOW model [14]. Unlike CBOW, which predicts the target word given context words, the skip-gram model is trained to predict the nearby context words given the target word [13]. Skip-gram is constructed with the central word as a single input vector, and the target context words are now at the output layer.

The training objective of the Skip-gram model is to find word representations that are useful for predicting the surrounding words in a sentence or a document. Given a sequence of training words, $w_1, w_2, w_3, \ldots w_t$ the objective of the Skip-gram model is to maximize the average log probability [13] [11]

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c<j<c,j\neq 0}\log\left(p\left(w_{t+j}\,|w_t\right)\right) \ldots \ldots \ldots \quad (2.12)$$

Where c is the size of the training context (which can be a function of the center word w_t)

Both skip-gram and CBOW models can learn a vector representation with very high quality. However, skip-gram architecture is better for capturing semantic regularities than CBOW architecture, which is more effective for capturing syntactic regularities [11].

2.8. Automatic thesaurus construction using Word2vec

In order to construct thesaurus automatically from the entire document collection by using word2Vec word embedding algorithm, the first step is to compute the word2Vec representation, after the representation is completed, the second step is to employee semantic similarity measures such as cosine similarity to calculate the similarity between two vocabulary terms by using their respective vector representation. That is, the similarity score between two vectors of the given terms, shows how much the two terms are related. If cosine similarity score between vectors of the given two terms is closer to 1 ($\cos 0^0$), then the two terms are said to be synonyms or related in meaning. In other words, the two vectors are close to each other [42]. To construct the thesaurus, terms that have the highest similarity score are selected for a given term.

In Word2Vec, word representation is learned by recurrent neural network language model [43]. Which is usually used for training sequential data. The architecture consists of an input layer, a single hidden layer, two weight matrices, and an output layer. The hidden layer consists of *N* neurons, which are equal to the number of dimensions used to represent a word. On the other hand, both input and output layers have *M* neurons, which is equal to the dimensionality of the vocabulary [43]. The input layer represents input words encoded using one-hot representation and the output layer`s activation function is softmax, which produces a probability distribution over

the words [39]. As we have discussed in section (2.7.1.2) above, there are two methods for training Word2Vec embedding, namely CBOW and Skip-gram. The Skip-gram algorithm tries to predict the neighboring n context words in both the right and left hand sides of the given target word t. whereas CBOW is the reciprocal of skip-gram [11].

In the Word2vec algorithm instead of training the network to learn the meaning of the target word given labels for that meaning, the network predicts words that are nearby the target word in a sentence (in the case of skip-gram). Here we have labels that are the nearby words we are trying to predict. However, in this case, the source of the label is the dataset itself and it does not require hand labelling, therefore the word2vec training algorithm is purely unsupervised [39]. To train a Word2vec model using a skip-gram model, the training set consists of the input word (target word) and the context (output) words that are (input, output) pairs wherein all the words are encoded in one-hot representation. One-hot is a way to encode data with 0`s and 1`s. if the vocabulary size is V these will be V-dimensional vectors with just one of the elements set to one, and the rest all zeros.

The softmax function in the output layer computes the probability of an output word being found as the context word of the input word. Then the output vector of word probabilities is converted to a one-hot vector, where the word with the highest probability is converted to 1 and all the rest terms will be set to 0 [39]. Once one round of the forward propagation is done, then obviously, there will be an error in the prediction compared to the actual value therefore, the error is calculated to update the weights so that the network corrects the error.

Once the training is completed the weight from input to hidden is used as embedding meaning, the weight matrix is the word embedding. The dot product between the one-hot vector representing the input and the weight matrix then represents word vector embedding. Therefore, each term from the vocabulary is represented by each row in the weight matrix [39]. This is because the result of the dot product is simply selecting the corresponding row from the matrix. In this way, semantically similar terms will have similar vector representations. This is because they were trained to predict similar neighbouring words. Hence, by employing similarity measures such as cosine similarity the semantically similar terms are selected for a given term. This way automatic thesaurus can be constructed using word embedding.

2.9. Similarity measurements

Text similarity measures are methods that are used to measure the degree to which two text entities are close to each other or far from each other [42]. The entities can be documents, sentences, or terms when similarity measures are employed for text similarity tasks. Text similarity methods measure how close two pieces of texts are both in terms of lexical similarity and semantic similarity. Words are said to be lexically similar if their character sequence is similar. On the other hand, words are semantically similar if they have the same theme [44]. Measuring the similarity between terms, sentences, and documents is one of the most important components in many areas such as information retrieval, text clustering, text summarization, etc. [45].

According to [42] there are three types of text similarity measures namely string-based, corpus-based and knowledge-based. The string-based metrics operate on string matching which is used for measuring lexical similarity. The corpus-based similarity is the one that measures semantic similarity between terms based on information obtained from a large corpus. On the other hand, knowledge-based similarity is a semantic similarity measure that makes use of information obtained from a semantic network for measuring the similarity between terms [42]. Knowledge-based and corpus-based measures are used for measuring semantic similarity. Some of the similarity measures that are widely used for measuring text similarity are discussed as follows.

I. Euclidean Distance

Euclidean distance is also known as the Euclidean norm, L2 norm, or L2 distance and is defined as the shortest straight-line distance between two points. It is the square root of the sum of squared differences between corresponding elements of the two vectors [42]. Mathematically this can be given as:

$$\boldsymbol{Ed(u,v) = \|u - v\|_2 = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}} \quad \text{..... (2.15)}$$

Where the two points u and v are vectorized text terms of length n

II. Jaccard Similarity

Jaccard similarity is a similarity measure that measures similarity as the intersection divided by the union of the objects. For text documents, the Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two documents but not the shared terms [46]. In other words, it is computed as the number of shared terms over the number of all unique terms in both strings. This can be mathematically given as:

$$\boldsymbol{SIM_J}(\overrightarrow{\boldsymbol{t_a}}, \overrightarrow{\boldsymbol{t_b}}) = \frac{\overrightarrow{\boldsymbol{t_a}} \cdot \overrightarrow{\boldsymbol{t_b}}}{\left|\overrightarrow{\boldsymbol{t_a}}\right|^2 + \left|\overrightarrow{\boldsymbol{t_b}}\right|^2 - \overrightarrow{\boldsymbol{t_a}} \cdot \overrightarrow{\boldsymbol{t_b}}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.17)$$

Similarity measure in the Jaccard coefficient ranges between 0 and 1. When $\overrightarrow{t_a} = \overrightarrow{t_b}$ the score is 1, meaning the two objects are the same. On the other hand, the score is 0, when $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$ are disjoint or completely different.

III. Cosine Similarity

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them [42]. Given two terms that are represented in their vector representation, then term vectors that have similar orientation will have nearly similar cosine score that is closer to 1 ($\cos 0^0$), which shows the vectors are very close to each other. On the other hand, term vectors with a cosine score that is close to 0 ($\cos 90^0$), indicates that the terms are unrelated. This is mathematically given by:

$$\boldsymbol{CS(u,v)} = \boldsymbol{\cos(\theta)} = \frac{\boldsymbol{u.v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|} = \frac{\sum_{i=1}^{n} \boldsymbol{u_i v_i}}{\sqrt{\sum_{i=1}^{n} \boldsymbol{u_i^2}} \quad \sqrt{\sum_{i=1}^{n} \boldsymbol{v_i^2}}} \ldots\ldots\ldots\ldots (2.16)$$

Where $CS(u, v)$is the cosine similarity score between term u and v. whereas u_i and v_i are n features of the two vectors.

2.10. Performance evaluation

Evaluations are required to show the performance improvements obtained as a result of applying new techniques or algorithms [1]. That is the change in the result obtained. The evaluations performed in this study are discussed in the following subsections as follows.

I. Evaluation of Automatic Thesaurus

According to [47] the quality of an automatically constructed thesaurus is evaluated in two ways. The first approach is to evaluate the thesaurus against reference lexicons such as WordNet. This technique is called intrinsic evaluation. The second approach is called extrinsic evaluation. Extrinsic evaluation involves the evaluation of the thesaurus by applying the generated thesaurus on downstream applications such as information retrieval, question answering, and so on.

II. Evaluation in information retrieval

Retrieval evaluation is a process of systematically associating quantitative metrics to the results produced by IR systems in response to a set of user queries. These metrics are directly associated with the relevance of the result to the user [25]. Hence, the standard approach for measuring the performance of information retrieval systems is based on the idea of whether a given document is relevant or irrelevant for given user information need expressed as a query [1]. The most widely used approach for evaluation is based on a comparison of the result produced by the system against the results suggested by humans for the same set of queries.

According to [1] three test collections are needed to measure information retrieval effectiveness. These are:

- A set of document collection
- A set of information needs that are expressed as a query, and
- A set of relevance judgments in which, for each query-document pair, binary assessment of relevance and non-relevance is annotated.

Relevance judgments are given by humans to the test document collection. Which is binary classification as either relevant or non-relevant for each query-document pair. There are various metrics for evaluating the retrieval performance of IR systems, but the most widely used metrics are recall and precision [1]. Some of the most widely used metrics for evaluating the performance effectiveness of IR systems are discussed as follows.

Precision: is a fraction of retrieved documents that are relevant.

$$Precision\ (P) = \frac{\#(\text{relevant } documents \text{ retrieved})}{\#\ (retrieved\ items)} = P(relevant|retrieved) \dots\dots\dots\dots\dots\dots (2.13)$$

Recall: is a fraction of relevant documents that are retrieved

$$Recall\ (R) = \frac{\#(relevant\ documents\ retrieved)}{\#\ (relevant\ documents)} = R(retrieved|relevant) \dots\dots\dots\dots (2.14)$$

F measure: precision always decreases as the number of the retrieved document or recall increases and vice versa. A measure that trades off precision versus recall is the F measure, which is given by the following equation.

$$F\ measure = \frac{2PR}{P + R} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (2.15)$$

2.11. Overview of Afaan Oromo Language

Afaan Oromo is a language from the Cushitic language family and is spoken by the largest ethnic group in Ethiopia. It is one of the languages with the largest number of speakers in Africa next to Arabic and Hausa [16]. It is spoken in three countries in the horn of Africa including Ethiopia, Kenya, and Somalia [17]. The language is also called Oromiffa. The charter of the transitional government of Ethiopia granted the language to become regional official language by the year 1993 [18]. Since then, the language has been used as a working language of the Oromia regional state and as a medium of instruction for primary schools in the region [18]. Now a day's huge amount of documents such as textbooks, fiction, and newspapers are being written and published in Afaan Oromo.

2.11.1 Afaan Oromo Writing system

Afaan Oromo adopted Latin script for writing and it is called *Qubee*, it has 33 characters representing unique sounds with 26 of them are the same as English language and additional 7 compound characters called "Qubee Dachaa" (*ch, dh, ny, sh, ph, ts, and zy*) [17]. The language has a set of five short (a, e, i, o, u,) and five long (*aa, ee, ii, oo, uu*) vowels. The variation in the length of the vowels resulted in the change of meaning. The following table shows some examples.

Table 2.1. Length of vowels resulted in a change in meaning

Afaan Oromo	**English**
boru	tomorrow
booruu	dirty

Doubling consonant letters is also another characteristic of the writing system of the language. This brings variation in meaning just like that of long vowels. Table 2.4 gives some examples.

Table 2.2. Length of consonant letters resulted in a change in meaning

Afaan Oromo	**English**
hatuu	to steal
Hattuu	thief

2.11.2. Punctuation Marks in Afaan Oromo

Punctuation marks are placed in a text to make the reading easier and to make the meaning clear. In the Afaan Oromo language, the same punctuation pattern is used just like that of other languages that use the Latin writing system [48]. The most commonly used punctuation marks in Afaan Oromo are:

a) Tuqaa *Full stop* (.): is a statement terminator, it is used at the end of a sentence. It is also used in abbreviations.
b) Mallattoo Gaafii *Question mark* (?): is used at the end of questions or interrogative.
c) Rajeffannoo *Exclamation mark* (!): is used at the end of command and exclamatory sentences.
d) Qooduu *Comma* (,): it is used to separate listing in a sentence or to separate the elements in a series.
e) Tuqlamee *colon* (:): is used to separate and introduce lists, clauses, and quotations, along with several conventional uses, and etc.
f) Hudha apostrophe (‘): Even though apostrophe is as a punctuation mark in the English language, it is not a punctuation mark in Afaan Oromo it is part of words. For instance apostrophe in the word ***re`ee*** (which means goat in English) is part of the word.

2.11.3. Afaan Oromo Sentence Structure

Unlike English language that uses subject-verb-object (SVO) sentence structure, Afaan Oromo language uses subject-object-verb (SOV) structure for sentence construction [23]. In subject-object-verb sentence structure, the subject comes first, followed by the verb and followed by object. For instance, in Afaan Oromo sentence “caalaan ciree nyaate” to mean, “chala ate

breakfast". In this sentence, "caalaa" is a subject, "ciree" is an object and "nyaate" is a verb. 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 handsome". In this sentence, "Tolaa" is a noun and "bareeda" is an adjective follows noun.

2.11.4. Afaan Oromo Morphology

Morphology is a branch of linguistics that studies the structure of words or components of a word. There are two types of morphology: inflection and derivational. Inflectional morphology is concerned with inflectional changes in words where stems are combined with grammatically markers that does not resulted in a change in the parts of speech. On the other hand, Derivational morphology deals with those changes that result in changing classes of words (parts of speech). The smallest unit to which words can be broken down into is called morphemes. A morpheme either has meaning or conveys grammatical function [9]. There are two types of morphemes: free and bound morphemes. Free morpheme can stand alone with a specific meaning whereas, bound morpheme cannot stand alone with meaning. According to [9] morphemes can be divided into root (stem) and affixes. While the root (stem) is the main morpheme of the word which carries the main meaning, affixes add additional meanings of various kinds.

In Afaan Oromo stem are bound morpheme they cannot stand alone with meaning for instance the root "*bar-",* does not have meaning when it stands alone. Roots are pronounceable only when affixes are added to them. Likewise affix is also bound morpheme it cannot occur independently. These affixes are of three types: prefix, suffix and infix. Prefix and suffix occurs at the beginning and at the end of a root respectively. For instance, in a word, *barumsa* (education), *-umsa* is a suffix and *bar-* is a stem. On the other hand, infix is a morpheme that is inserted within another morpheme. Afaan Oromo does not have infixes [48]. Afaan Oromo is a language that has very complex and rich morphology. It is a language that involves very extensive inflectional and derivational morphological processes [20]. Most of the grammatical information in the language is conveyed through affixation. In Afaan Oromo, words can be formed through inflection, derivation, compounding, and reduplication [48].

2.11.5. Afaan Oromo Language Part of Speeches

The Afaan Oromo language words can be categorized into nouns, verb, adverb, adjective, pronoun and prepositions [49] [48].

i. Nouns

A noun is a word that names something, such as a person, place, thing, or idea. Nouns in Afaan Oromo are inflected for gender, definiteness and number [50].

a. Gender

Afaan Oromo has a two gender system (feminine and masculine). Frequent gender markers in Afaan Oromo include ***-eessa/-eettii***, ***-a/-ttii*** or ***–aa/tuu***.
Example:

Afaan Oromo	Construction	Gender	English
obboleessa	obol + eessa	male	brother
obboleettii	obol + eettii	female	sister
jabaa	jab + aa	male	strong
jabduu	jab + duu	female	strong

b. Number

Afaan Oromo has different suffixes to form the plural of a noun. The use of different suffixes differs from dialects to dialects. There are more than ten major and very common plural markers in Afaan Oromo including: *-oota, -oolii, -wwan, -lee, -an, een, -eeyyii, -oo,* etc.). For example:

Singular	Plural	English translation
hiriyaa	hiriyoota	Friend/ Friends
gaangee	gaangolii	Mule/Mules
laga	Lageen	River/rivers

c. Definiteness

In Afaan Oromo demonstrative pronouns like *kun* (this), *sun* (that) are used to express definiteness. In some Afan Oromo dialects the suffix *-icha* for male and *-ittii(n)* for female and for undermining usually has a singularize function is used where other languages would use a definite article.

For example:

namicha this/ the man (Subject)	kitaabni sun *that/ the book*
nititttii *this/ the women (Object)*	namtichi kun *this / the man*

ii. Verbs

A word that is usually one of the main parts of a sentence and that expresses an action, an occurrence, or a state of being. Afaan Oromo has base (stem) verbs and four derived verbs from the stem. Moreover, verbs in Afaan Oromo are inflected for gender, person, number and tenses [48]. There are four derived stems, the formation of which is still productive, Autobenefactive, Passive, Causative and Intensive.

a. Autobenefactive

The Afan Oromo autobenefactive (or "middle" or "reflexive-middle") is formed by adding -(a)adh, -(a)ach or -(a)at or sometimes -edh, -ech or –et to the verb root. This stem has the function to express an action done for the benefit of the agent himself.

Example:

Word (the verb)	Root/stem	meaning
bitachuu	bit	to buy for oneself

b. Passive

The Oromo passive corresponds closely to the English passive in function. It is formed by adding -am to the verb root. The resulting stem is conjugated regularly.

Example:

beek- *know*, beekam- *be known*

c. Causative

The Afaan Oromo causative of a verb corresponds to English expressions such as cause','make', 'let'.With intransitive verbs, it has a transitiving function. It is formed by adding -s, -sis, or -siis to the verb root.

Example:

deemuu - to go, deemsisuu - to cause to go

d. Intensive

It is formed by duplication of the initial consonant and the following vowel, geminating the consonant.

Example:

Waamuu - to call, invite wawwaamuu - to call intensively

iii. Adjectives

An adjective is a word that describes a noun or a pronoun. Adjectives in a sentence are used to modify nouns to show the quality of things. Consider the following examples:

‘boontun *bareeddu* dha’ (bontu is beautiful.)

‘caalaan *dheeraa* dha’ (chala is tall.)

In the above examples the word ‘barreeddu’ (beautiful) and ‘dheeraa’ (tall) are adjectives. Afaan Oromo adjectives can be formed from compound words [50]. For instance, ‘humna dhabeesssa’ (weak), ‘simbo qabeessa’ (handsome) are some of adjectives constructed from compound words. Adjectives inflect for number and gender in Afaan Oromo language.

singular	Plural	masculine	feminine
guddaa	gudguddaa	guddaa	guddoo
jabaa	jaboota	jabaa	Jabduu
Ko’eessa	Ko’eeyyii	Ko’eessa	Ko’eettii
cimaa	ciccimoota	cimaa	Cimtuu

iv. Adverbs

Adverbs are words that are used to modify verbs. In Afaan Oromo adverbs come before the verb they modify. Afaan Oromo adverbs are categorized as adverbs of time, place, and manner [48]. Adverbs of time show the time when the action takes place. Mostly adverbs of time answer the question of when the action takes place. Some of the words that can be used as adverbs of time in Afaan Oromo language includes: ‘amma’ (now), ‘boru’ (tomorrow),’kaleessa’ (yesterday),’yoom’ (when), “har’a” (today),‘galgala’ (tonight) etc. Consider the following example. ‘salamoon *kalessa* dhufe.’ (solomon came yesterday) ‘Adaama boru ni deemna.’(We will go to Adama tomorrow.) In these examples the word ‘kaleesa’ (yesterday) and ‘boru’ (tomorrow) are adverbs of time. Adverbs of place show the place where the action takes place.words that can be used as adverb of

place in Afaan Oromo includes: **‘**as’(here),‘achi’(there),‘gadi’(below),‘gubbaa’(above), ‘jidduu’(middle), ‘irra’(on),etc. consider the following example

‘Tolaan *mana* jira.’ (Tola is at home.)
‘Inni *konkolaataa irra* jira’ (he is on the car.)

Adverb of manner show how the action of the sentence is done. The following are Afaan Oromo words that can be used as adverb of manner ‘ariitin’(quickly), ‘suuta’ (slowly), ‘akka gaarii’ (well) etc. consider the following example:

‘Isheen ariitin figdi.’ (She is running quickly).
‘Calaan baay’ee cimaa dha.’ (Chala is very clever).
In the above two sentences the word ‘ariitin’ (means quickly) and ‘baay’ee’ (mean very) are adverbs of manner.

v. Pre, para, and post positions

Afaan Oromo languages use prepositions, postpositions and Para positions [23].

i. postpositions
   a. suffixed post positions

suffixed post positions	English equivalent
-ttii	in, at, to
-rra, irra	on
-rraa, irraa	out of, from

Example: hojitti yoom deebina? Which means (When shall we get back to work?)

   b. Postpositions as independent words

Some of the independent word postpositions are listed in the following table.

Postpositions as independent words	English Equivalent
ala	outside
bira	beside
booda	after, behind
duuba	behind
wajjin	with, together with

Example: Qarshii nu bira jiru fudhadha. Which means (Take the money with us.)

ii. Prepositions

Prepositions	English Equivalent
Akka	like, according to
Gara	to, in the direction of
Hanga, hamma	until, up to
Karaa	along, the way of, through

Example: Hanga deebi`e dhufutii na eegi. Which mean (Wait me until I come back.)

iii. Para positions

Para positions	English Equivalent
Gara…tti	to
Gara…tiin	from, from the direction of
Hanga…tti	up to,until

Example: booru gara mana keenyatti deebina. Which means (we will get back to our home tomorrow.)

2.11.5. Challenges of Afaan Oromo Language for IR systems

In this section, some characteristics of the Afaan Oromo language that makes Afaan Oromo Information retrieval systems challenging are discussed with examples.

I. Dialectic variation

According to [51] the Afaan Oromo language is said to have six major dialects, 1) Northern Afaan Oromo (Baate and Raayyaa), 2) Western Afaan Oromo (Macca), 3) Highland Shawan Afaan Oromo (Tuulama), 4) Eastern Afaan Oromo (Hararge), 5) Central Afaan Oromo (Gujii and Arsii), and 6) Southern Afaan Oromo (Boorana). Almost all of the varieties are being used in written materials and the mass media because the language does not have a standard form. Amanuel et.al [52] pointed out that because of the dialectic variation; there is a variation in terms of vocabulary and pronunciation. Such variations are responsible for misunderstandings in written communication among speakers of various dialects of the language [52]. Since the language is dialectic and there is no standard form, documents can be written and published by using all forms. As a result, there are many synonym words, which greatly reduce the performance effectiveness of Afaan Oromo retrieval systems [52].

II. Lexical ambiguity

The property possessed by a single word, to have two or more totally unrelated meanings is termed as homonymy [21]. The homonymous word can occur in the same position in utterances, this results in what is called lexical ambiguity [52], as in, for example, "Ani mana ballaa keessa hinjiraadhu." The above sentence has two distinct meanings. It means "I do not want to live in a wide/big house" on one hand or, it also means "I do not want to live in a blind person's house", this is because of the word "ballaa" that possess two distinct meaning in the sentence. There are several homonymous words in the Afaan Oromo language. Hence, this property of words will have a huge impact on the retrieval effectiveness of Afaan Oromo IR systems.

III. Short forms of compound words

Compound words in Afaan Oromo have short-form representation. In such representation, forward-slash (/) and period (**.**) is used to separate the letters in the short form representations. However, the forward slash is the most common. The short form may exist in place of long-form compound words in Afaan Oromo documents or user queries. In this case, both forms must be handled by information retrieval systems so that documents containing both forms will be retrieved.

Example:

Table 2.3. Abbreviation for compound words

Afaan Oromo	**English**
I.G (Itti Gaafatama)	Head
W.A (Waldaa Aksiyoonaa)	Share Company.
A.L.I (Akka lakkofsaa Itiyoophiya)	Ethiopian calendar

2.12. Related Work

In this section, research works that have been done to handle term mismatching problems and therefore attempted to enhance the performance of the retrieval systems are reviewed. Thus, global and local research works that make use of either manually or automatically constructed thesaurus as a source of expansion terms are reviewed.

2.12.1. Afaan Oromo Information retrieval system

Since the Afaan Oromo language got an official script, a huge amount of documents are being written and published. Therefore, to facilitate the speakers of the language to take advantage of that information, some studies have been conducted for instance in 2010, Tesfaye [20] designed and implemented the Afaan Oromo search engine. In addition to this Gezehagn [19] designed and implemented Afaan Oromo text retrieval system in 2012. In this subsection, prior research works that focus on the attempts to handle term mismatch problems for the Afaan Oromo language are reviewed.

In 2018, Firomsa [53] constructed an automatic thesaurus from Afaan Oromo text by using the term-term co-occurrence approach. The study aimed to provide an automatic thesaurus that can be used as a good resource for the development of enhanced Afaan Oromo retrieval systems. After performing pre-processing tasks such as tokenization, normalization, stop word removal, and stemming, the author computes term weighting by using TF-IDF and performs term co-occurrence analysis by building a term-term co-occurrence matrix. Then the study computed the similarity between terms by using cosine similarity and cluster them according to their similarity score. The author has used a corpus size of 100 documents from which 63434 words were collected from VOA, OBN, and Gospel Go. After pre-processing, the remaining 36869 words were used for thesaurus development. For evaluating the quality of the generated thesaurus, the study randomly selected 20 terms and generated 7 terms for each of them from the thesaurus. And the study found that on average 73.11 % of the generated thesaurus terms are related to their respective term.

Additionally, Berhanu [22] attempted to handle the effect of vocabulary problems in the Afaan Oromo IR system by applying latent semantic indexing (LSI) in order to identify topics or concepts within a set of documents. The study used LSI to analyse word association within a set of documents by forming a term-document matrix. To reduce the dimension of the matrix singular value decomposition (SVD) was applied. The queries were treated as pseudo documents and pre-processing tasks that were applied to the document collection were applied to the query as well. Then, query terms were weighted in the same way as that of documents and projected into the same reduced LSI space. Cosine similarity measure was applied to calculate the similarity among query terms and the documents to retrieve a set of relevant documents in response to the query.

The performance of the system was evaluated by using 70 text documents and nine queries. The system obtained on average 0.67(67%) and 0.63(63%) precision and recall respectively.

Tesfaye [23] applied thesaurus-based semantic compression for Afaan Oromo text retrieval to overcome the problem of term mismatching. In the study, the thesaurus was constructed manually by identifying content-bearing terms from the entire corpus where terms with high TF-IDF values are used as a descriptor or key terms. The key terms were used to represent synonym terms in the thesaurus, and then a semantic compression algorithm was applied for indexing the document. The algorithm checks whether a given term is represented as a key term or synonym term in the thesaurus, if it is used as a key term it is used for indexing otherwise its respective descriptor is searched from the thesaurus and used for indexing by replacing the synonym term. Hence, the documents that contain synonym terms were retrieved. The study used ten queries and a corpus size of 106 text documents for constructing the thesaurus and for evaluating the system. The system was evaluated with and without integration of semantic compression into the IR system and registered an average result of 58.451% precision, 78.438% recall, and 61.520% F-measure without thesaurus based semantic compression and 71.014%, 89.287%, and 77.715% average precision, recall and F-measure with thesaurus based semantic compression were registered. Although the study attempted to increase the retrieval performance of Afaan Oromo IR systems, the size of the corpus used is not enough for manually constructing thesaurus, and constructing thesaurus manually for large collections is very difficult. Moreover, it is very expensive to get a good general-purpose thesaurus [24].

2.12.2. Query expansion for Amharic information retrieval

In this section, more recent studies that have been made, in order to enhance the performance of Amharic information retrieval systems are reviewed.

In 2014 Welde [38] applied ontology-based query expansion technique into Amharic information retrieval system with the aim of increasing the performance of Amharic IR systems. The study manually constructed domain-specific ontology in the area of tourism and used the ontology as a source of expansion terms. A corpus size of 195 documents from the tourism domain was collected and used for constructing the ontology. Ten queries and the same document collection used for ontology construction were used for testing the system. The study performed two experiments, the

first experiment was conducted without integration of query expansion technique into the Amharic IR system and registered 44%, 54%, 44% average precision, recall, and F-measure respectively. The second experiment was conducted by integrating ontology-based QE into the Amharic IR system, showed reduced result for precision by 4% and improved result by 32% and 8% for recall and F-measure respectively. Even though using ontology as a source for QE can enhance the performance of IR systems, domain-specific ontologies cannot be applied for general IR systems. Moreover, manually constructing ontology requires domain experts and it is time-consuming.

The other study conducted for enhancing the performance effectiveness of Amharic IR systems was done by Samrawit [33]. The study employed manually constructed WordNet as a source of expansion terms and for disambiguating the ambiguous query terms. The constructed WordNet contains only two pieces of information associated with each term that is the list of synonyms (synset) and gloss definition (definition of the word in that particular sense). For identifying the correct sense, the study used the Lesk algorithm that compares each of the sense definitions of the ambiguous word with the definition of the words surrounding the word to be disambiguated. The study conducted three experiments by expanding the query using three methods: synset expansion, gloss expansion, and a combination of synset and gloss expansion. For evaluating, the system 10 test queries and 300 Amharic News articles were used. The first experiment using synset expansion registered 68% recall, 53% precision, 59% F-measure. The second experiment using gloss expansion registered 68% recall, 27% precision, 36% F-measure, and the last experiment using the combination method registered 70% recall, 25% precision, 33% F-measure.

A recent study conducted to improve the Amharic IR system was conducted by Berihun [54]. The study stated that previous works attempted to construct Amharic thesaurus does not consider contextually similar words. This is because either they used the traditional word count approach or they are limited to predefined vocabulary. The study attempted to construct automatic semantic vocabulary from Amharic text corpus using a distributional semantics approach for information retrieval. To construct semantic vocabulary the author performed the required text pre-processing tasks (such as tokenization, normalization, stop word removal, and stemming), then the author gave the text to word-space modelling, that embed them in vector space by using the Word2Vec CBOW embedding model. Then words were clustered based on their cosine score to construct semantic vocabulary. The study expanded user query by adding five (5) closest (most similar)

terms to each query term from the semantic vocabulary. The study used a corpus size of 8,759 manually collected Amharic documents for semantic vocabulary construction. On the other hand, a total of 90 documents and 9 queries were used for evaluating the system. Then the author conducted two experiments and obtained 69% recall, 44% precision, and 84.23% recall, 23.86% precision without using query expansion and using expanded query respectively.

2.12.3. Query expansion for Tigrigna information retrieval

To enhance retrieval effectiveness by increasing retrieval of relevant documents in response to user queries, Zeray [34] integrated query expansion mechanism into Tigrigna information retrieval. The author identified that Tigrigna retrieval systems are affected by a term mismatching between user query terms and index terms due to synonym and polysemy terms. The study used manually constructed root-based Tigrigna WordNet as a source of expansion terms. It contains the ambiguous terms, synonym terms, and related terms for each sense. For identifying the correct sense of ambiguous query terms and for selecting synonyms, Tigrigna WordNet was used.

Instead of using stemmer, the author prepared a list of word variants together with their root word to reduce all variants of a word to a single form. Then the study has used a context window of three, the preceding, succeeding and the middle ambiguous word to identify the correct sense of the ambiguous term. The context words were compared with related words of each sense and the one with the highest overlap was taken as the correct sense. Then synonyms of the identified sense were selected and added to the query. Finally, the expanded query was used to retrieve a more number of relevant results.

The study evaluated the system by using a set of 300 short documents and ten queries used by previous researchers and conducted two experiments, the first experiment was conducted to show the effect of under-stemming and over-stemming on Tigrigna IR systems by using morphological analysis instead of stemmer and showed that using morphological analysis instead of stemmer increase the performance of Tigrigna IR systems by 9% precision and 1.6 % recall. The second experiment was conducted by expanding the query using synset, the experiment registered an improvement by 12% for precision and 4% for recall on the overall performance.

2.12.4. Query expansion for the English Language

In the attempt to handle the vocabulary mismatch problem Angel F. et.al [55] applied query expansion using similarity thesauri. The study constructed the thesaurus from the matrix of relations between terms. To construct the thesaurus the study takes the vector space model as a base and turns the representation model around. That is the terms of the collection are considered as documents, and the documents are used as index terms. Therefore, each term t_i in the collection of m terms will be represented by a vector of N components in the vector space of documents, $\vec{t} = (p_{i1}, p_{i2}, \dots, p_{iN})$, p_{ij} is a value of the weight of the index document d_j in the representation of the term t_i. The value of p_{ij} was computed by TF-IDF weighting scheme, but the roles of terms and documents were inverted. The weights were normalized to have unit vectors. Scalar product was used to compute the similarity between terms. Calculation of scalar product for all term pairs resulted in similarity thesaurus, which is a matrix with values in a range of 0 and 1. The study expanded the query by adding terms that are more similar to the entire query terms instead of expanding each query term individually. For evaluation, 50 queries and 215,718 documents collected from the EFE news agency were used and the study obtained an improvement of 8.92% average precision for all queries expanded with top 500 related terms.

Vincent et.al, [47] have examined both the construction and evaluation of distributional thesauri by using information retrieval. In the study, the authors used AQUAINT-2 corpus that contains 380 million words out of which those common nouns that occur at least 10 times were considered. This resulted in 25,000 unique nouns for which the contexts of all the occurrences were collected by considering two words at both the right and the left of the target noun for building the thesauri.

For evaluating the constructed thesauri, the authors perform both intrinsic and extrinsic evaluations. Regarding intrinsic evaluation, the study used 12243 nouns with 38 neighbours on average, extracted from WordNet and Moby. On the other hand, for extrinsic evaluation, the study evaluated the thesauri on IR task by using semantic neighbours extracted from the thesaurus for expanding a query (i.e., query expansion). The study experimented with several IR models for building the distributional thesauri and found that adjusted Okapi-BM25 registered the best intrinsic result measured by Mean average precision, R-precision, and Precision at different thresholds against WordNet and Moby reference lexicons.

On the other hand, a corpus size of 170,000 documents and 50 queries, were used for evaluating the thesauri on IR tasks by expanding the query. For each noun query, neighbouring words from the thesaurus were added to expand the query. The IR system used was Indri. The performance of the system was measured by precision at different thresholds, R-precision, and mean average precision with and without expansion. The study found that the automatically built thesaurus is as good as that of reference lexicons.

2.12.5. Query expansion for the Arabic Language

Ashraf et.al [56] proposed semantic Query expansion for Arabic Information Retrieval. An ontology built from Arabic Wikipedia dump and related terms from "A1 Raed" dictionary (that contain 204303 modern Arabic expressions) and "Google WordNet" dictionary (which the authors collected all the words in the English WordNet and translate them into Arabic using Google translate) was used as a resource for obtaining expansion terms. The study performed text pre-processing tasks such as normalization, stop word removal, and stemming. Then the authors fed the documents to Apache Lucene for indexing and retrieval.

For testing the system, the authors have used a dataset that contains 25 queries and 2730 documents from a book entitled "Zad Al Ma`ad". The titles of the book chapters were used as queries. The study conducted three experiments by using the baseline (where no expansion is used), single expanded query, and multiple expanded queries (in which each term is expanded at a time and multiple queries are formulated and the result of them are combined). The system was evaluated using precision, recall, and F-score at five levels (@1, @5, @10, @20, and @30) and the obtained result shows that multiple expanded queries methodology gives the highest result for almost all measures.

2.13. Summary of related works

Table 2.4. Summary of related works

Title	Author, year	Purpose	Approach /methodology	Key findings	Recommendations and future works
Automatic thesaurus construction from Afaan Oromo text [53]	Firomsa Wakjira,2018	To provide automatic thesaurus that can be taken as a good resource for the development of enhanced Afaan Oromo retrieval systems	Term-Term co-occurrence, document clustering	The study found that on average 73.11 % of the generated thesaurus terms are related to their respective term.	The future recommendation: to develop a standard corpus for thesaurus evaluation.
Applications of Information Retrieval for Afaan Oromo text based on Semantic-based Indexing [22].	Berhanu Anbase,2019	To handle the effect of vocabulary problems in the Afaan Oromo IR system.	Latent Semantic Indexing (LSI)	The system obtained on average 67% and 63% precision and recall respectively	The future recommendation: to apply LSI for cross-language retrieval.

Applying Thesaurus-Based Semantic Compression For Afaan Oromo Text Retrieval [23].	Tesfaye Tadele, 2019	To overcome the term-mismatching problem in Afaan Oromo IR.	Manual thesaurus construction, Sematic compression algorithm	The integration of thesaurus-based semantic compression into IR registered an improved performance over the baseline IR system.	The construction of a standard Afaan Oromo thesaurus.
Ontology-Based Query Expansion For Enhancing The Performance Of Amharic Information Retrieval: The Case of Tourism Sector [38]	Welde Janfa, 2014.	To increase the performance of the Amharic IR system.	Domain-specific ontology	Integration of ontology-based QE into Amharic IR system registered reduced result for precision by 4% and improved result by 32% and 8% for recall and F-measure respectively	Automatic or semi-automatic ontology extraction method should be studied and designed

Word Sense Disambiguation using Semantic Similarity for Query Expansion in Amharic Information Retrieval [33]	Samrawit Zewdneh, 2014	To enhance the performance effectiveness of Amharic IR.	Amharic WordNet, Lesk algorithm	Expanding the query by using synset registered 68% recall, 53% precision, 59% F-measure.	Future work: construction of standard Amharic WordNet
Amharic Information Retrieval Using Semantic Vocabulary [54].	Berihun Getnet. 2019	To improve retrieval performance of Amharic IR system	Distributional semantic model: Word2Vec`s CBOW model	The system obtained 69% recall, 44% precision and 84.23% recall, 23.86% precision without using and using QE respectively.	To use semantic vocabulary combined with manually built WordNet for IR systems.

Query Expansion for Tigrigna Information Retrieval [34].	Zeray Tsadu, 2017	To overcome the term-mismatching problem in Tigrigna IR.	Manually constructed Tigrigna WordNet.	Morphological analysis instead of stemmer increase the performance of Tigrigna IR systems by 9% precision and 1.6 %. The system registered an improvement of 12% precision and 4% recall on the overall performance.	To develop a standard corpus, test queries and standard IR system for evaluation.
Reformulation of queries using similarity thesauri [55]	Angel F. and et.al, 2005	To handle vocabulary mismatching problems.	Similarity thesauri (which is constructed by transposition of the term-document matrix)	The study obtained an improvement of 8.92% average precision.	

Distributional Thesauri for Information Retrieval and vice versa [47].	Vincent Claveau & Ewa Kijak	To address the problem of building and evaluating distributional thesauri.	Distributional thesauri	The study found that the result obtained with the thesaurus built automatically is as good as that of reference lexicons.	
Semantic Query Expansion for Arabic Information Retrieval [56].	Ashraf Y and et.al, 2014	To overcome the limitation of keyword-based search	An ontology built from an Arabic Wikipedia dump.	The study found that multiple expanded queries methodology gives the highest result for almost all measures.	

2.14. Executive Summary

Generally query expansion is a method used to reduce the term mismatching problem between short user query and relevant documents. In query expansion the data source used for obtaining expansion terms is very important. A top documents that are retrieved in response to the user`s original query, manually constructed thesaurus, and automatically constructed thesaurus can be used as a source of expansion terms. The most widely used approach for query expansion is by using manual thesaurus such as WordNet and ontologies. However, manual thesaurus construction is very difficult as it is time-consuming, difficult to update, and also requires expert for construction. To overcome this problem a methods that can automatically extracts synonyms and other word relation form large document collections are more preferable. Automatic thesaurus construction methods have the advantage of not involving human in their development and they cover all the terms in the collection.

Some attempts have been made to reduce the effect of term mismatching problem and to enhance the performance of Afaan Oromo IR system. The first study [22] have used co-occurrence based method. Even though co-occurrence based methods are good at capturing word co-occurrence statistics, they are poor at capturing semantic word relationships. The other work [23] have used manually constructed thesaurus but manual thesaurus construction is very challenging, it is time-consuming, and also requires experts. Even though the above studies tried to handle the term mismatching problem, as far as the knowledge of this study no attempt has been made to handle the vocabulary problem by integrating query expansion into Afaan Oromo IR systems. Therefore, this study tries to fill this gap.

Chapter Three: Methodology

3.1. Overview

This chapter discusses the research methodology employed, methods, and techniques that are used to accomplish the objectives of this study. Since the research includes designing a new artifact, evaluation of the artifact, and communicating the result, design science research methodology is a perfect fit for this study. The selected research methodology, the process model, and the procedure followed are discussed in section 3.1 below.

3.2. Research methodology

In this research, design science research methodology (DSRM) is employed. The reason for employing design science is because the study involves the design and development of an artifact. A prototype for constructing automatic Afaan Oromo thesaurus for query expansion to enhance the performance of Afaan Oromo IR system is designed, developed, and evaluated.

This research is focused on a problem-centered approach as shown in figure 3.1, which starts with activity 1. This is for the reason that the idea for this research has resulted from suggested future research from a previous research paper done by Gezehagn [19]. Figure 3.1 shows the DSRM research process that includes six activities. The process model is adopted from [57] and modified for this study.

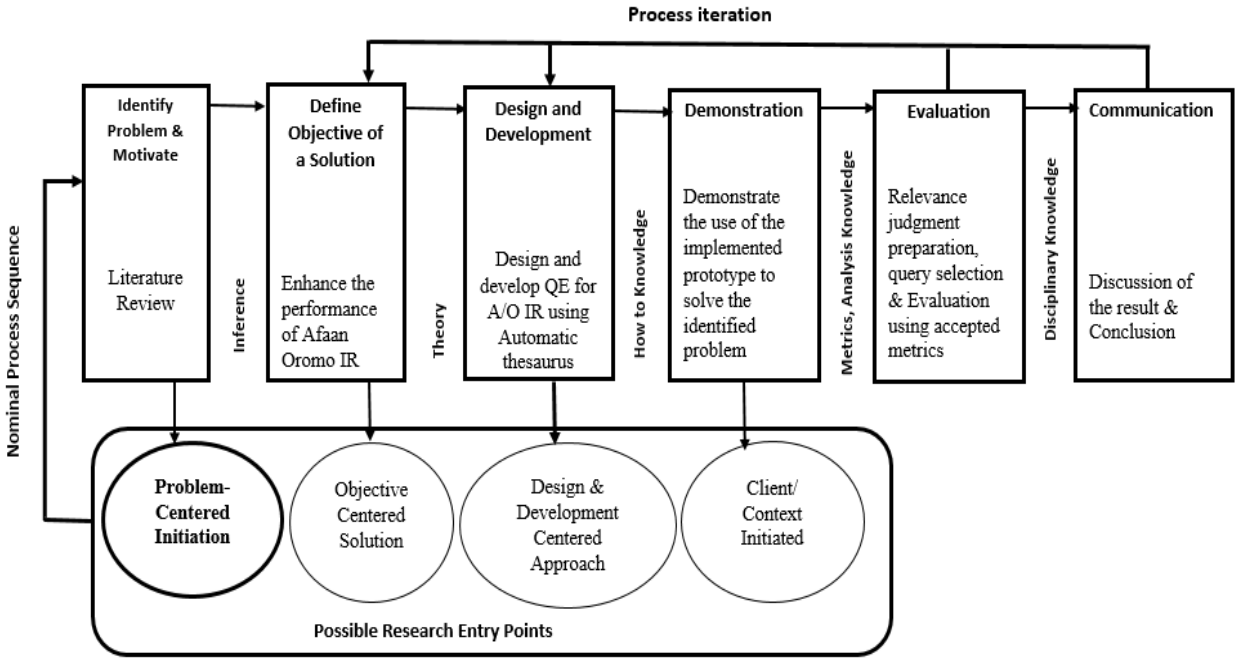


Figure 3.1 DSRM process for QE using automatic thesaurus for Afaan Oromo IR system.

I. Problem identification and motivation

To find a gap from previous studies on Afaan Oromo IR systems and to identify researchable problems that need a solution, a thorough review of prior research works that have been done in the area of Afaan Oromo IR systems was conducted. From reviewed literature, we have identified that Afaan Oromo IR system is affected by term mismatching problem. Term mismatching problem occurs because of the difference in terminology between queries of the user and terms that appear in the document. Terminological variation occurs because the same concept can be expressed by using different terms. This is due to the fact that multiple words can have the same meaning, such words are called synonyms [21]. A Retrieval system that merely depends on term matching is highly affected by vocabulary mismatching problem. Therefore, relevant documents cannot be retrieved in response to a user`s query. As a result, the performance of information retrieval systems is reduced. Prior research work on Afaan Oromo IR system [19] stated that term mismatching problem affected its result and recommends that query expansion can be an ideal solution to reduce the effect of this problem.

II. Definition of the objectives

The objective of this study was to build an automatic Afaan Oromo thesaurus from a manually collected document corpus and to use the thesaurus as a source of expansion terms for expanding user queries in order to enhance the retrieval performance of Afaan Oromo IR system.

III. Design and development

Design and development is the process in which the artifact is created [57]. It includes data collection, architectural design, and the development of the actual artifact that constructs automatic Afaan Oromo thesaurus for query expansion.

a. Data collection

Afaan Oromo is an under-resourced language. It does not have freely available corpora that can be used for IR or natural language processing tasks. Therefore, collecting Afaan Oromo documents is part of the study. Lexical resources such as stop words, affixes, and abbreviations were collected from previous studies and also a book written about the grammar of the language entitled "Seerlugaa Afaan Oromo". The list of stop words and abbreviations used in this study are compiled in Appendix A and B respectively.

On the other hand, to automatically construct Afaan Oromo thesaurus a total of 5,070 documents were manually collected from Afaan Oromo bible, Ethiopian News agency official website[1], Gullalee post magazine, Ethiopian Orthodox Tewahido Church Mahiber Kidusan official website[2], Voice of America Afaan Oromo[3], BBC Afaan Oromo[4] , and Ethiopian press Association Afaan Oromo[5] official websites. Additionally, 120 news articles and 10 queries (an information need) were used for evaluation. The collected documents are mainly news articles except for documents collected from the Afaan Oromo bible. The collected news articles involve different subjects like politics, education, culture, business, history, and other subjects. The details of the sources and the number of documents collected from those sources are summarized in Table 3.1 below.

Table 3. 1. The collected data and their sources

[1] https://www.ena.et/afanoromo/
[2] https://eotcmk.org/ao/
[3] https://www.voaafaanoromoo.com/
[4] https://www.bbc.com/afaanoromoo
[5] http://press.et/afanoromo/

Data source	Type of document	Number of documents
Afaan Oromo Bible	Religious	1,189
Mahiber Kidusan official website	Religious	292
Gaazexaa Bariisaa- Ethiopian Press Agency	General domain news articles	1,006
	Sport domain	193
BBC Afaan Oromo	News	507
VOA Afaan Oromo	News	325
Gullalee Post	News	36
Ethiopian News Agency (ENA)	General domain News articles	1,483
Other sources	Academic books, fictions and so on.	39
Total		5,070

b. Design of the architecture

The architecture of thesaurus-based query expansion for Afaan Oromo information retrieval system is designed. Based upon the architecture the required algorithms are developed. The architecture of the system is composed of four components: text pre-processing, thesaurus construction, term selection, and query expansion. The details about the components are discussed in chapter 4.

c. Prototype development

The prototype that expands queries of the user by using automatically constructed Afaan Oromo thesaurus was implemented on python programming language. Python version 3.8 of 64 bit was used because it offers several libraries. In this study, Python`s NLTK (Natural Language Toolkit) library, is used for text analysis, Gensim library is used for constructing word vector representation. Additionally, Whoosh 2.7 library[1] was used for indexing and searching. The reason for using whoosh over Pylucene is that whoosh is purely implemented in python.

[1] https://whoosh.readthedocs.io/

IV. Demonstration

Demonstration is the process that shows the use of the developed artifact to solve one or more instances of the identified problem. This could include its use in experimentation, simulation, case study, proof, or other appropriate activity [57]. In this research, after the prototype was implemented, to demonstrate the use of the artifact to solve the problem identified in step one, two experimentations were conducted.

Experiment 1

Experiment 1 was conducted to measure the quality of the generated thesaurus terms. There is no ready-made reference lexicon such as WordNet with which the automatically constructed thesaurus is compared. Therefore, to evaluate the quality of the thesaurus, twenty terms were randomly selected and given to language experts together with their corresponding generated thesaurus terms to measure how much the generated terms are related to the given term. Then the experts evaluate how much the generated terms are related to the given term. Language experts give a score that ranges from 1-5 for each thesaurus term based on their relatedness.

Experiment 2

Experiment 2 was conducted to observe the performance effectiveness gained because of the integration of query expansion into the Afaan Oromo IR system over the traditional system, hence it was conducted by measuring the performance of the Afaan Oromo IR system before and after integration of QE into Afaan Oromo IR system.

V. Evaluation

After the prototype of thesaurus-based query expansion is developed, it is important to evaluate its performance. Evaluation is the process of observing and measuring how well the developed artifact supports a solution to the problem identified in step one [57]. This activity includes query selection, relevance judgment, and evaluation.

a. Query selection

In order to evaluate the performance of information retrieval systems, we should have information needs expressed as queries, test document collection, and relevance judgment as discussed in section 2.10. Therefore, for evaluation purposes, 120 news documents and 10 queries (an information need) were used. There is no relevance annotated query-document pair test collection for Afaan Oromo, for this reason, the queries were selected subjectively by the researcher after manually reviewing the content of each news article. The list of the queries used for evaluating the system is depicted in Table 3.2 below.

Table 3. 2. Test queries used for evaluation.

No.	Test query	Short-form
1.	dhibee kooviid	Q1
2.	pirojektota meeggaa	Q2
3.	Sirna Gadaa	Q3
4.	Lola Tigraay	Q4
5.	Ayyaanna Irreecha	Q5
6.	injifannoo Adwaa	Q6
7.	sabaafi sablammootaa	Q7
8.	tola ooltummaa	Q8
9.	Hidha Haaromsaa guddichaa	Q9
10.	Filannoo Biyyaalessaa	Q10

b. Relevance judgment by experts

The standard approach for evaluating the performance of an information retrieval system revolves around the idea of relevance. That is whether a given document is relevant or non-relevant for a given query [1]. Therefore, for a user information need, each document in the test collection is given a binary classification as either relevant or non-relevant. In this study, 120 documents and 10 queries were given to the Afaan Oromo language teacher for relevance judgment. Where for each query the documents are marked as relevant or irrelevant. The summary of the relevance judgment for the selected queries is listed in Table 3.3 below and the complete relevance judgment given by Afaan Oromo language expert is itemized in Appendix C.

Table 3.3. Summary of Relevance judgment

No.	Query	Number of relevant documents	Number of Non-relevant documents
1.	Q1	15	105
2.	Q2	15	105
3.	Q3	22	98
4.	Q4	15	105
5.	Q5	13	107
6.	Q6	11	109
7.	Q7	10	110
8.	Q8	15	105
9.	Q9	15	105
10.	Q10	12	108

c. Evaluation

According to Claveau et.al. [47] Evaluation of automatically generated thesaurus falls into two categories: intrinsic and extrinsic. In intrinsic evaluation, the task is to evaluate the quality of the generated thesaurus by comparing the generated thesaurus to one or more reference lexicons. On the other hand, extrinsic evaluation involves the evaluation of the generated thesauri on downstream tasks such as part of speech tagging, IR, etc., and measure the changes in performance metrics specific to that task.

In this study, both intrinsic and extrinsic evaluations were performed. Intrinsic evaluation was performed by language experts based on experiment 1. On the other hand, extrinsic evaluation was performed by using the generated thesaurus as a source of candidate expansion terms for expanding user query, that is user`s original query is expanded by adding related terms from the automatically constructed thesaurus. Then the performance improvement achieved as a result of expanding user query from the thesaurus was measured against the baseline Afaan Oromo information retrieval system by using the most widely used information retrieval performance effectiveness measures:

precision, recall, and F-measure. The details about information retrieval performance measures were discussed in section 2.10.

VI. Communication

The final step involves summarizing the results of the evaluations and presenting a conclusion and forwarding recommendations for future works.

Chapter Four: System Design and Architecture

4.1. Overview

The primary purpose of an information retrieval system is to fulfil the information need of the users by retrieving as many relevant documents as possible in response to the user query. However, the attempt to satisfy the information need in keyword matching-based systems is highly affected by term mismatching problems. This is because the same concept can be described using different terminology. Due to the terminological variation between the user query and the document, the performance effectiveness of retrieval systems is reduced. To overcome such kind of problem integration of query expansion can be an ideal solution.

In this chapter, the architecture of query expansion using automatic thesaurus for the Afaan Oromo information retrieval system will be discussed.

4.2. Architecture of QE using automatic thesaurus for Afaan Oromo Information Retrieval system.

As shown in figure 4.1 below the overall system architecture is composed of six components: text pre-processing, thesaurus construction, term selection, query expansion, indexing, and searching. In the text-pre-processing component, first Afaan Oromo documents are tokenized into sentences and the sentences are further tokenized into words, then the tokens will be normalized that is converted to lowercase, and also all short forms of compound words will be substituted with their corresponding full word. Then stop word removal is performed and finally stemming is applied to reduce inflected words into their base form. After the pre-processing stage is completed, the stemmed text will be given to the thesaurus construction component wherein each term is represented in multidimensional space to obtain a word-space model.

Once the word vector representation is over and each unique word gets its vector representation, then the word-space model is searched using a term to generate the nearest neighbor to that particular term. By employing similarity measurement metrics such as cosine similarity to compute the similarity between the vector representations, the top n nearest neighbors to a given term are generated. These terms are automatically generated thesaurus terms for that particular term. Once

thesaurus terms are generated, the next step is term selection, which is the task of selecting the thesaurus terms that are to be added to the original query of the user. To select expansion terms one-to-many association approach is employed where the average cosine similarity between each query term in the query and all the generated candidate thesaurus terms, is calculated. Then top k thesaurus terms with the highest score are selected. Finally, the selected thesaurus terms are added to the user's original query and the query is expanded.

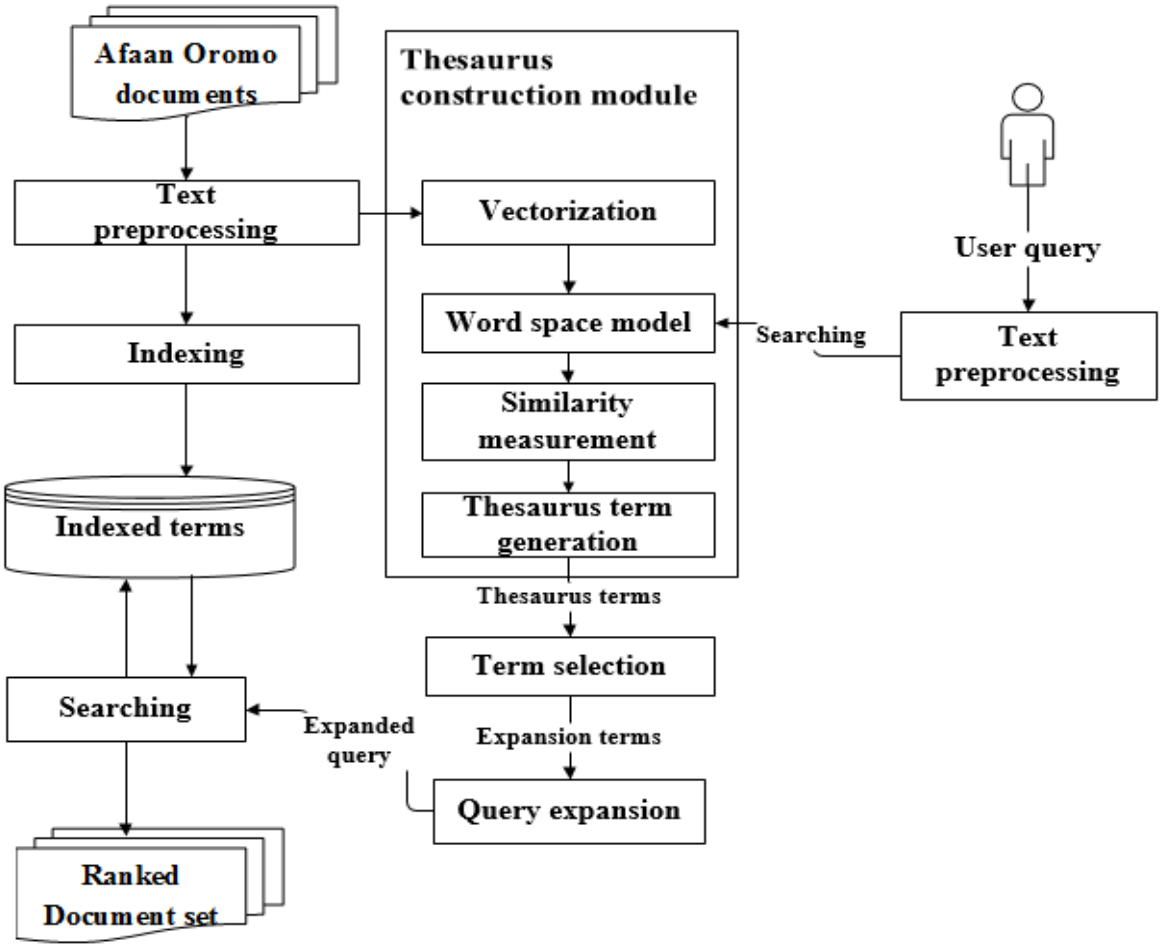


Figure 4.1. The general architecture of the system.

4.2.1. Preprocessing module

After completing the task of collecting the required document collection from different sources, the next step is to apply the necessary text pre-processing task to the collected document. At this stage, noisy symbols such as punctuation marks, special characters, and tags are removed from the

text. Text pre-processing tasks that are used in this study are depicted in figure 4.2 below. The details of text pre-processing tasks that are applied in this research are discussed as follows.

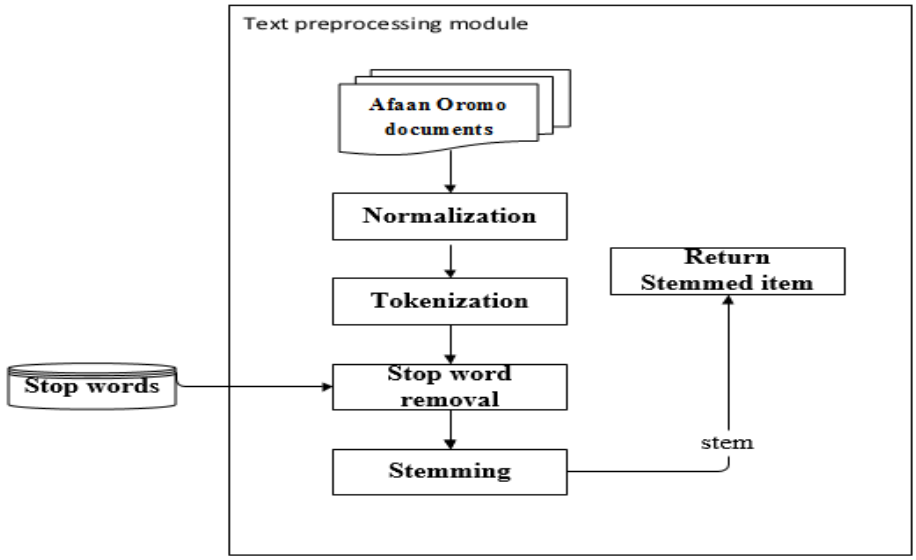


Figure 4. 2. The Text preprocessing module.

4.2.1.1. Tokenization

In distributional semantic models, the meaning of a word is derived from the context in which it appears. To obtain contexts of a certain word, and its vector representation, tokenization should be performed. Tokenization also called word segmentation is the process of splitting out sequential texts into a list of pieces or tokens. Therefore, a word is a token in a sentence and a sentence is a token in a paragraph [58]. Sentence tokenization is a process of splitting paragraphs into a list of sentences, whereas word tokenization is a process of splitting sentences into words. In this study, both sentence and word tokenization tasks were performed.

Sentence tokenization is performed by looking at the sentence boundary. In Afaan Oromo, sentences end up with period (**.**), question mark (?), or exclamation mark (!). If one of the punctuations that shows sentence boundary is encountered in running text, then the text is tokenized into a sentence. The other tokenization task is word tokenization. It is performed to obtain individual words by splitting the sentence into words using white space.

4.2.1.2. Normalization

In this study, two normalization tasks are performed. The first one is case folding, which is done in order to convert different forms of writing that are uppercase, lower case, and mixed-use into a single form. Therefore, all the terms are converted into lower case. The second normalization task is performed for converting abbreviations into their long-form. In Afaan Oromo language abbreviations are usually written by combining two alphabets that are either separated by a full stop (.) or forward-slash (/). Therefore, words hash tables that map key-value pairs is used for this task. In python, the implementation of hash tables uses the built-in dictionary data type. Dictionary data structure is also called a hash map or associative array [14]. In this study, the short form of words are stored as keys and their corresponding long-form are stored as values in a dictionary. For instance, the abbreviation W/A is stored in a dictionary as a key and its corresponding long-form "Waldaa Aksiyoonaa" is stored as a value. The complete list of abbreviations and their corresponding expanded form are listed in Appendix A. The algorithm for tokenization and normalization is depicted in figure 4.3 below.

```
Start
  input: Afaan Oromo text file
       : Abbreviation dictionary
1      Open and read a file
2      Convert the text into lower case
3      Perform sentence tokenization
4      If the sentences contain unnecessary symbols
5          Replace the symbol with white space.
6      Perform word tokenization
7      If token exists in the keys of abbreviation dictionary
8          then replace the token with the value of that key from the dictionary.
9      else
10         return token
  output: List of sentences
End
```

Figure 4.3. Tokenization and Normalization algorithm.

The algorithm shown in figure 4.3 above accepts text by opening and reading one Afaan Oromo text file at a time, then texts obtained are converted to lower case to normalize all words into the same form (line 1-3). The algorithm then uses the normalized text to perform sentence tokenization. To perform sentence tokenization the algorithm looks for a full stop, exclamation, or question marks. The reason why sentence segmentation is needed is that because the word2vec algorithm takes one sentence at a time. Then unnecessary symbols (such as punctuation marks, special characters, and HTML tags) are removed and replaced with white space (line 4). Then the next step is to perform word tokenization (line 5). After the sentences are tokenized into separate tokens, the next step is to normalize abbreviations. The algorithm looks for either full stop (.) or forward-slash (/) separated alphabets in the text and if they match with the keys of a dictionary, then it replaces them with their corresponding value from the dictionary (line 6). Finally, the algorithm produces a list of preprocessed sentences as output.

4.2.1.3. Stop word Removal

Stop words are those words that appear very frequently in a document. These words can be prepositions, conjunctions, and articles. Even though, stop words appear very frequently, their contribution in describing the content of the document is very little. In information retrieval, stop words have a minor effect in selecting documents matching user queries [1]. Therefore, these words are usually discarded during indexing. This helps to reduce the storage space requirement. To make use of these advantages, stop word removal is included as one of the preprocessing tasks in this study. Afaan Oromo stop words that are used in this study are listed in Appendix B.

```
Start
        input: Afaan Oromo stop word file
                : preprocessed document
1       Read stop word file
2       Read preprocessed document
3       while end of file is reached
4               if token exists in stop word file
5                       remove the token from the document
        output: document without stop word
End
```

Figure 4.4. Stop word removal algorithm

The algorithm for removing stop words takes preprocessed document (which is normalized and tokenized) and Afaan Oromo stop word file as an input. Then the algorithm opens both preprocessed document and stop word files and until the end of the file is reached, the algorithm iteratively checks whether a token is found in the stop word list or not. If it is found, then the algorithm will remove it and return a document without stop words.

4.2.1.4. Stemming

A stemming is a process of chopping affixations from words that have the same stem and reduce them to a common form. It is widely used in modern-day search engines and retrieval systems to make the search result broader. Stemming improves the performance of IR systems by increasing the number of relevant matches. This is because it brings variant forms of a word that share a common meaning under one heading. Stemming can also reduce the storage space by representing every morphological variation of a word by using single term rather than storing them separately [58]. This in turn helps during indexing.

The result of stemming might be difficult to recognize to which word the resulting stem belongs. To avoid this problem, the study tried to use HornMorpho[1] for obtaining the root words

[1] https://github.com/hltdi/HornMorpho

unfortunately, we found that most of the words are not known by HornMorpho. The result of HornMorpho for some Afaan Oromo words is as shown in Table 4.5 below. Therefore we have decided to implement a rule-based stemming algorithm designed by Debela [48] which works by stripping inflections of a word based on a given set of rules.

Table 4.1. Afaan Oromo words known and unknown by HornMorpho

Known by HornMorpho	Unknown by HornMorpho
Fedhii, ergamaa, qulqulloota, amanan, abbaa, nagaafi, naannichaa, mirkaneessuu, hojjechuuf, fedhii, qabdi, jedhan	Phaawuloos, Waaqayyootiin, Kiristoos, Yesuus, ja'a, Yesuusitti, kanneen, efeeson, jiraataniitti, Waaqayyoo, Gooftaa, ayyaanniifi, nageenni, tasgabbii, Ameerikaan, Keeniyaa, waliin

In this study, the rule-based stemming algorithm designed by Debela [48] is implemented with a slight modification. The stemming algorithm is shown in figure 4.5 below. The algorithm takes a tokenized document as input and until all the tokens are covered it checks if a match is found from the rules. If a match is found, it removes the suffix and performs the necessary adjustment. If not the algorithm removes the vowel at the end of the term and return the result.

```
Start
      Input : tokenized document

1.          Read the next word to be stemmed
2.          If end of file not reached
3.              If word matches with one of the rules
4.                  Remove the suffix and do the necessary adjustments
5.                  Return word
5.                  Go back to 1
6.              Else
7.                  Go to 10
8.          Else
9.               Stop processing
10.        IF there is no applicable condition and action exists
12.             Remove vowel and return the word
                Go to 1
       Output: list of stemmed word
End.
```

Figure 4.5. Stemming algorithm.

4.2.2. Thesaurus construction module

After the necessary text preprocessing tasks such as normalization, tokenization, stop word removal, and stemming are completed then, the output of preprocessing module is given as an input to the thesaurus construction module. The tasks in this module include vectorization, similarity measurement, and thesaurus construction.

4.2.2.1. Vectorization

To construct thesaurus automatically, the first step is to represent all the terms in a vector space. This task is known as vectorization. It is a task of numerically representing each vocabulary term in the multi-dimensional space. As a sematic relationship is more important for this study Word2Vec`s skip-gram model is used for word embedding. This is because literature works have testified that skip-gram architecture is better for capturing semantic relationships than CBOW [11]. After all the necessary preprocessing tasks are completed then the output of preprocessing module is given as input to Word2Vec`s skip-gram algorithm. The algorithm trains word representations

in such a way that words are embedded in space along with similar words based on their context. The details of how Word2Vec is used for automatically constructing thesaurus were discussed in section 2.8. To build the word space model, the Word2vec algorithm takes several parameters that affect both training speed and the quality of the word representation. The parameters considered in this study are discussed as follows:

- **Training architecture:** because the objective of the study is to build word vector representation that can capture better semantic similarity over syntactic similarity, the study implemented skip-gram architecture
- **Vector size:** the size of vector refers to the number of dimensions that the word vector has. In this study, the vector size is set to 300.
- **Window size:** refers to the maximum distance between the current and predicted word within a sentence. In this study, the window size is set to 5.
- **Number of workers**: is the parameter to set the number of worker threads or the number of CPU cores used for the training. The study dynamically set the number of workers by importing the multiprocessing package and "multiprocessing.cpu_count()" method that returns the number of threads of the system.
- **Minimum count**: is a parameter set to integer number to tell the model to ignore words that have total frequency lower than the given integer number. This parameter is set to 10.
- **Sub-sampling**: is the threshold for configuring which higher-frequency words are randomly down sampled, the useful range is (0, 1e-5). It is the subsampling rate for frequent terms. To reduce the influence of frequent words, words are sampled during training in inverse proportion to their frequency. The study set the sample to the default 1e-3.
- **Sorted vocabulary: -** if 1, sort the vocabulary by descending frequency before assigning word indexes.

4.2.2.2. Similarity measurement

After the training of word representation, is completed and all the vocabulary words get their vector representation, then the next step is to apply similarity measurement to compute the similarity between terms. This study used cosine similarity. The calculation of the cosine similarity between vectors of a given two terms corresponds to the similarity between the two terms. Given two terms

that are represented in their vector representation, then term vectors that have similar orientation will have a cosine score closer to 1 ($\cos 0^0$), which shows the terms are related to each other. On the other hand, term vectors with a cosine score closer to 0 ($\cos 90^0$), indicates that the terms are unrelated.

4.2.2.3. Thesaurus term generation

Once the word vector representation is over and all vocabulary terms get their corresponding vector representation, then to generate thesaurus for a given query, the following steps are followed. First, the query is preprocessed (that is tokenized, normalized, stop words are removed, and stemmed). Next for each query term, cosine similarity between the vector of the query term and all vectors in the space is computed. Then top k similar or related terms are generated as a thesaurus for each query term. The Thesaurus construction algorithm used in this study is illustrated in the following figure.

```
Start
      Input: list of query terms
             : distributional semantic model file
1     Accept user query
2     preprocess the query
3     open the distributional semantic model file
4     for each term in the query
5          if term exist in the distributional semantic model file
6               then calculate similarity between the term and all terms in the space
7                    return top k related terms
8          else
9               Return term not found message.
      Output: list of k thesaurus term for each query terms.
End
```

Figure 4.6. Thesaurus construction algorithm.

The algorithm for thesaurus construction starts working by accepting a query expressed as either a single term or a collection of terms. The necessary preprocessing tasks (such as normalization and tokenization) are performed on the query (lines 1-2). Then the algorithm opens the distributional semantic model file (line 3). If the terms exist in the file, then cosine similarity between the vector of that particular term and vectors of all terms in the model is calculated. Based on the similarity score obtained, top k related terms are generated as thesaurus terms. The loop continues until the last term of the query is reached (lines 4-9).

4.2.3. Term selection

As discussed in Section 2.2.1 after obtaining top k thesaurus terms for each query term, thesaurus terms are selected as expansion terms based on their relationship with the whole query instead of their relationship with individual query terms. Therefore this study followed a one-to-many association approach for term selection. In one-to-many association, expansion terms are selected based on their similarity with the entire query. In this approach, the first step is to compute the similarity between each query term in the query and all the generated candidate thesaurus terms. Let the query be Q $Q = \{q_1, q_2, , \ldots \ldots q_n\}$ and candidate expansion terms C, then the average cosine similarity for each term t in C and all the terms in Q is given by the following formula.

$$sim(t, q) = \frac{1}{|Q|} \sum_{q_i \in Q} t. q_i \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (4.1)$$

Then terms in C are sorted according to their average cosine similarity score and top n thesaurus terms with the highest score are selected as expansion terms.

4.2.4. Query expansion

To reduce the effect of term mismatching problem between the user query and index terms that represent the document in the collection and increase the performance effectiveness of the Afaan Oromo Information retrieval system, the user query is expanded. Once expansion terms are obtained from automatically generated thesaurus by following the term selection stage, then the selected expansion terms are added to the original query of the user, and the query is expanded.

4.2.5. Document indexing

Indexing is the process of organizing documents using keywords or content bearing terms extracted from the document collections. Indexing makes the searching process efficient. In order to avoid linear scanning of the text documents for each query, it is necessary to index the documents. Indexing maps terms to the documents that contain them. Therefore, for each term, we have a list that records in which document the term occurs. It is useful for simplifying the retrieval process [59]. In this study, an inverted index is built by using Whoosh[1], which is a fast and pure python search engine library. Unlike Lucene, Whoosh is not a search engine; it is a programmer library for creating a search engine. In addition to indexing, Whoosh allows us to quickly find matching documents based on simple or complex search criteria.

To build an inverted index, first, the document collections are identified, followed by the necessary preprocessing tasks such as tokenization, normalization, stop word removal, and stemming are performed. Then the preprocessed terms are given to the indexer to build an inverted index or index file. The index file contains a dictionary file that stores alphabetically sorted vocabulary terms and has a pointer to the posting list for each term. Whereas the posting list stores a list of documents, in which a term occurs.

```
    Start.
        Input: set of documents to be indexed
1           Walk through the directory
2           for each document in the directory
3               Open the Document file one by one
4               If read is successful:
5                   Apply preprocessing on the Document
6                   write the document into the indexer
7           Close the indexer by committing.
        Output: index file
    End.
```

[1] https://whoosh.readthedocs.io/

Figure 4.7. Document indexing algorithm.

The indexing algorithm shown above works by taking a set of documents to be indexed as input. By walking through the directory the algorithm open and read each document one by one (line 1&2). If reading is successful, then the necessary text preprocessing tasks such as tokenization, normalization, stop word removal, and stemming are performed on the document. The preprocessed documents are written to the indexer one after the other. This process continues until all the documents in the directory are covered (lines 2-6). Lastly, the algorithm closes the indexer by committing.

4.2.6. Document searching

Searching is the process of finding relevant documents in the index list. To search and retrieve relevant documents in response to a given query, the information need of the user expressed in terms of a query goes through the same text preprocessing tasks that were applied to the indexed document collections. Hence, tokenization, normalization, stop word removal, and stemming tasks are performed on the query of the user. Once the preprocessing task is completed for each query term the set of related thesaurus terms are obtained to expand the query. Then the next step is to use the expanded query to search and retrieve more relevant documents. The algorithm for searching from document collection is depicted in figure 4.8 below.

```
Start
    Input: user query
1       Accept user query
2       Preprocess the query
3       if query expansion is selected
4           Open Afaan Oromo thesauri
5           For each query term
6             If the term exists in the thesauri
7               Retrieve top n related terms
8           Calculate the average cosine similarity between each
            candidate thesaurus term and user query
9           Select top 5 thesaurus terms
10          Expand the query by using generated thesaurus terms
11          Search using expanded query.
12      else
13          Search using original query
14      Rank the document
15      Display ranked document
    Output: Ranked document set
End.
```

Figure 4.8. Document searching algorithm.

The document searching algorithm works as follows. First, the algorithm accepts user information need expressed as a query and apply all text preprocessing tasks that were performed during indexing (line 1 & 2). Once the query is preprocessed, then if query expansion is selected the algorithm opens Afaan Oromo thesaurus and retrieves top k related terms for each query term. The process proceeds until all the query terms are covered. After thesaurus terms are generated for all the terms in the query, to select expansion terms from the candidate thesaurus terms, the average cosine similarity between all the candidate thesaurus terms and each query term is calculated. Then top n thesaurus terms with the highest average similarity score are selected as expansion terms. Finally, the selected thesaurus terms are added to the original query to reformulate the query. The expanded query is used to search from the index file (lines 3-9). Otherwise, the original query is used to search from the index file (10 &11). Finally, a ranked set of documents are retrieved and displayed in response to the query (line 12-13).

Chapter Five

Experimentation and Discussion of the Result

5.1. Overview

This study aims to come up with an automatically built Afaan Oromo thesaurus that can serve as a source of terms for expanding user queries. Then the expanded query is used to search and retrieve more relevant documents and therefore, it enhances the retrieval performance of the Afaan Oromo text retrieval system. This chapter discusses the implementation, set of experiments conducted, both intrinsic and extrinsic evaluations, and result and discussion of the system designed in chapter four.

5.2. Implementation

5.2.1. Text preprocessing

Text preprocessing tasks are important not only for constructing high-quality thesaurus but also for implementing information retrieval systems. In this study, Text preprocessing tasks such as tokenization, normalization, stop word removal, and stemming are performed. The preprocessing activities are applied to both the document collections used in thesaurus generation and also indexed documents used for evaluation purposes. Except for sentence tokenization, the rest preprocessing activities discussed in section 4.2.1 are also applied to the query of the user.

5.2.2. Thesaurus construction

In this study, an automatically constructed thesaurus is used as a source of expansion terms. Therefore, for each query term, a set of N thesaurus terms are generated as candidate expansion terms from the thesaurus. The steps to construct the thesaurus include vectorization, similarity measurement, and thesaurus generation.

a. Vectorization

After the necessary preprocessing activities (such as sentence and word tokenization, normalization, and stop word removal) are performed on the collected document, all vocabulary words are represented in multidimensional space using word2Vec`s Skip-gram model that works

based on the concept of distributional semantics. The words obtain their vector representation in such a way that words that occur in similar contexts have similar representation. This is with the assumption that words that appear in similar contexts are somewhat related in meaning. In this study, every vocabulary word is represented in a vector having 300 dimensions. The vector representation for sample Afaan Oromo word “ilmaa” which means “son” in English is depicted in Figure 5.10 as shown below.

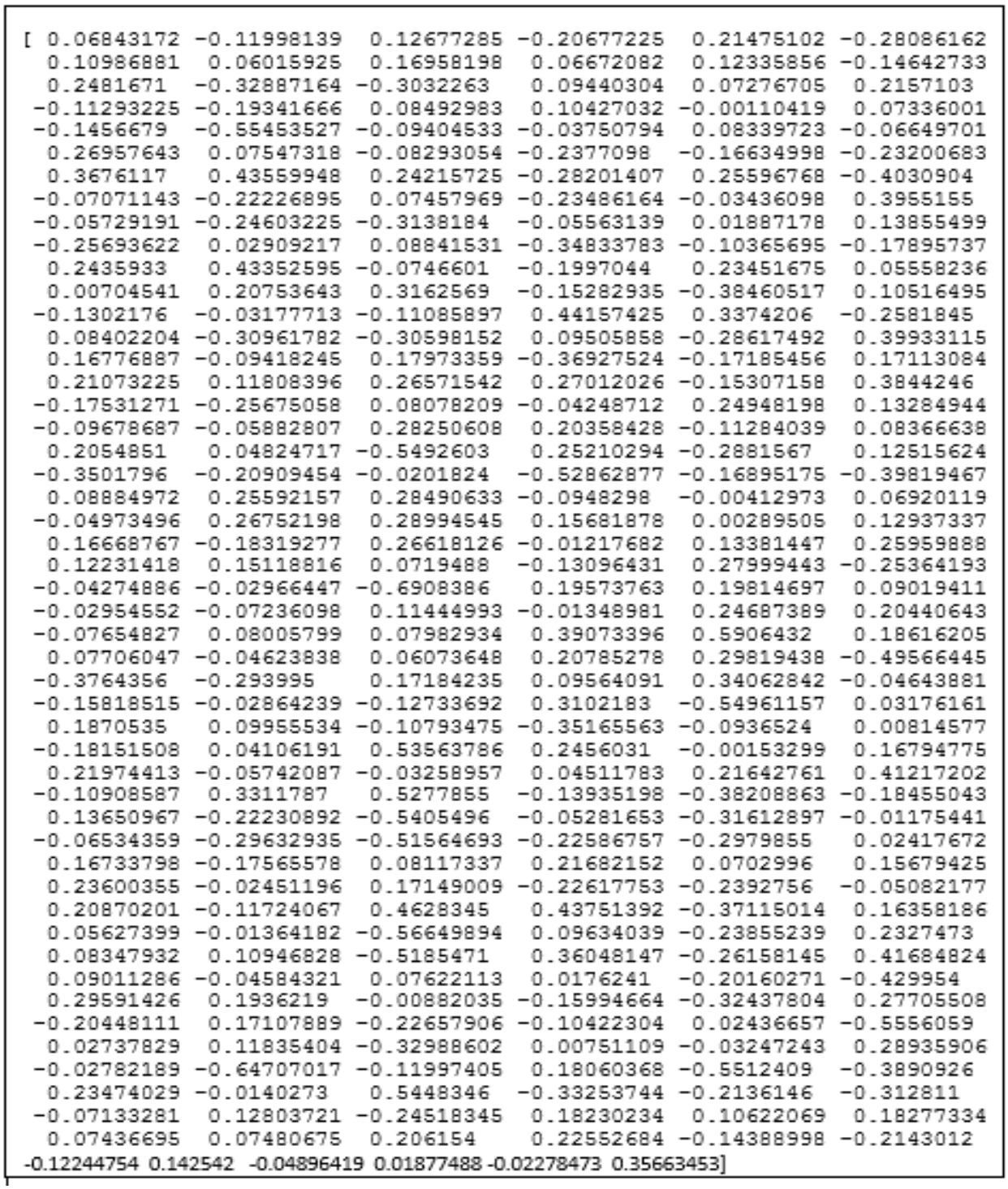


Figure 5.1. Vector Representation of Afaan Oromo "ilmaa

b. Similarity measurement

Once all the vocabulary words get their respective vector representation, to obtain the similarity between terms similarity calculation is performed among the vectors of the terms by using a cosine similarity measure. The similarity score approaching to 1 shows that the terms are highly related to each other. On the other hand, a similarity score approaching to 0 shows the terms are unrelated.

c. Thesaurus generation

To generate thesaurus terms for a given term, the first step is to preprocess the term for which thesaurus terms are going to be generated. The same preprocessing activities discussed in section (4.2.1) are applied. Once the necessary preprocessing activities are performed, thesaurus terms are generated based on the similarity score between the term for which thesaurus is generated and all the terms in the word space. Therefore, cosine similarity between the given term and each vocabulary term in the word space is calculated and the top K nearest neighboring terms are generated as thesaurus terms. Meaning that terms with the highest similarity score are generated. Thesaurus terms generated together with their similarity score, for example, Afaan Oromo word "*dhukkuba*" which means disease in English is depicted as shown in the prototype below.

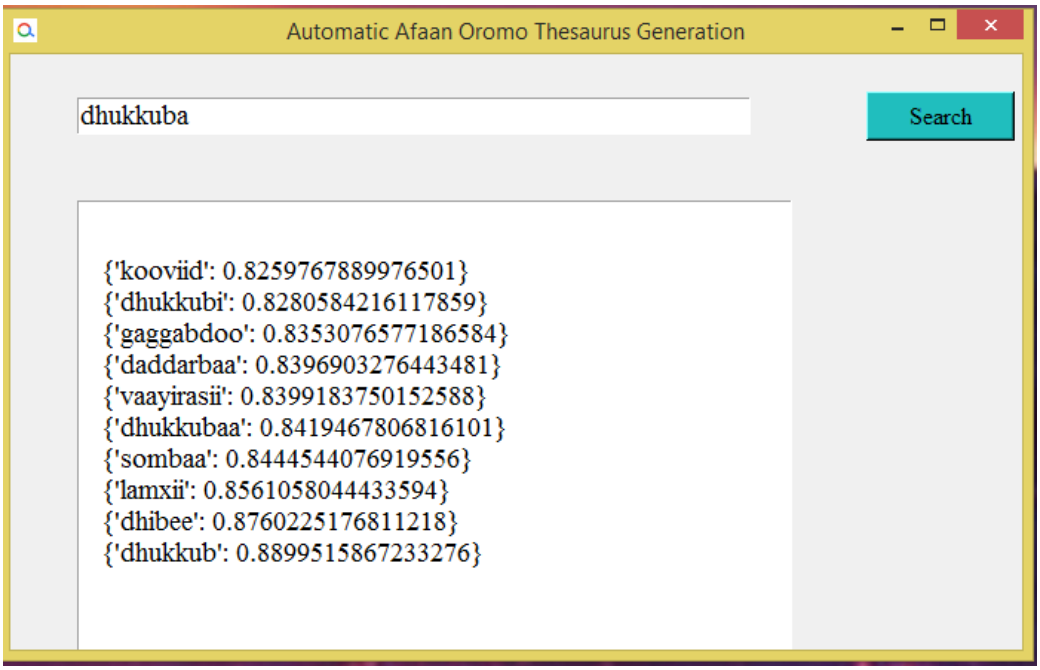


Figure 5.2. Thesaurus terms generated with their similarity score for word "dhukkuba"

As shown in Figure 5.10 above ten thesaurus terms are generated based on their corresponding similarity score with the term “dhukkuba”. In order to show how much the generated terms are related to the given query term, the English translation of the generated thesaurus terms is given in the table below.

Table 5.1. Generated thesaurus terms English translation

Query term	Generated thesaurus term	English translation
dhukkuba	kooviid	Covid
	dhukkubi	disease
	gaggabdoo	epilepsy
	daddarbaa	Communicable or contagious
	vaayirasii	virus
	dhukkubaa	disease
	sombaa	lung
	lamxii	vitiligo
	dhibee	disease
	dhukkub	disease

As shown in the table above, almost all of the generated thesaurus terms are related to the query term “dhukkuba” which means “disease” in English. The terms dhukkubi, dhukkubaa and dhukkub are morphological variants of the word dhukkuba but, they are generated as thesaurus terms because of spelling error and lack of proper stemmer.

5.3. Term selection

After the necessary preprocessing tasks are performed on the query of the user, then for each query term top ten related thesaurus terms from the word-space model are generated based on the similarity score of the terms with the given query term. The generated nearest neighboring terms are then taken as candidate expansion terms. The candidate expansion terms together with their similarity score for example query “dhibee kooviid” is as shown in the table below.

Table 5.2. Candidate expansion terms with their similarity score

Query term	Candidate expansion term with their score
dhibee	{'dhukkub': 0.8974599242210388} {'vaayirasii': 0.8913322687149048} {'dhukkuba': 0.8804500102996826} {'vaayirasi': 0.8722659349441528} {'koronaa': 0.8719441890716553} {'gifiraa': 0.8696415424346924} {'daddarboo': 0.8660792112350464} {'dhukkubi': 0.8612662553787231} {"tatamsa'i": 0.8603622913360596} {'sombaa': 0.8586907982826233}
kooviid	{'baakteeriyaa': 0.9647431373596191} {'mallatto': 0.965191662311554} {'jarmii': 0.9656888842582703} {"laka'amu": 0.9666047692298889} {'ittifamu': 0.9670385718345642} {'gifirri': 0.9679794907569885} {'itituu': 0.9686122536659241} {'ilbi': 0.9692367911338806} {'dhibamt': 0.969592273235321} {'vaayirasi': 0.8722659349441528}

For certain candidate expansion term to be used for expanding the query, that candidate terms must be somewhat related to the entire query rather than related to a single term of the query. Therefore, since this study follows a one-to-many association for term selection, the average similarity of each candidate expansion term with the whole query is calculated. Table 5.3 depicts the average similarity score of each candidate term with the entire query.

Table 5. 3. The average similarity score of each candidate terms with the entire query.

```
[('dhibee', 0.825494796037674), ('kooviid', 0.825494796037674),
('dhibamt', 0.8072588741779327), ('sombaa', 0.8043002188205719),
('daddarboo', 0.7971569001674652), ('vaayirasi', 0.7255832850933075),
('gifiraa', 0.7765467464923859), ('gifirri', 0.772464245557785),
('itituu', 0.7672955393791199), ("laka'amu", 0.7572628557682037),
("tatamsa'i", 0.7572177350521088), ('baakteeriyaa', 0.7559666335582733),
('ilbi', 0.7529005408287048), ('ittifamu', 0.7509682774543762),
('vaayirasii', 0.7506028711795807), ('jarmii', 0.7495947182178497),
('mallatto', 0.7447139918804169), ('dhukkub', 0.7354255020618439),
('dhukkubi', 0.730404257774353),('koronaa', 0.7176231145858765),
('dhukkuba', 0.6918947100639343)]
```

After the average similarity of each candidate expansion term with the entire query terms is calculated then top k candidate terms with the highest average similarity score are taken as expansion terms.

5.4. Query expansion

Once thesaurus terms are selected as expansion terms based on their average similarity score with the whole query, then the selected terms are added to the original query to reformulate the query. The final expanded query for our example query is as shown in Figure 5.3 below. Then the reformulated query is used to retrieve more relevant documents and hence increase the performance effectiveness of the Afaan Oromo information retrieval system.

```
['dhibee', 'kooviid', 'dhibamt', 'sombaa', 'daddarboo', ' vaayirasi', 'gifiraa']
```

Figure 5.3. The final expanded query for our example query.

5.5. Experimentations

In this section, the way experiments are conducted to evaluate the proposed query expansion for the Afaan Oromo Information retrieval system is discussed. As discussed in the earlier sections, the proposed query expansion system makes use of automatically constructed thesaurus as a source of expansion terms. Therefore, automatic thesaurus construction is also part of this work. In this study, two experimentations are conducted. The first experimentation was conducted to evaluate the quality of the automatically constructed thesaurus. The second experimentation was conducted to observe the effect of the proposed query expansion approach on the Afaan Oromo information retrieval system. That is to measure the performance effectiveness obtained as a result of the integration of query expansion into the Afaan Oromo IR system. Hence, evaluation of the Afaan Oromo IR system with and without query expansion is conducted.

5.6. Performance evaluation

Performance evaluation is essential for measuring how much a certain stated objective is achieved. As discussed in Section 2.10 the performance evaluation methods for automatic thesaurus falls into two categories: namely intrinsic and extrinsic evaluation. Intrinsic evaluation is the task of evaluating the performance of the thesaurus against existing reference lexicons such as WordNet. Whereas extrinsic evaluation is the evaluation of the thesaurus on downstream tasks such as information retrieval. Therefore, the following subsections discuss the result of both intrinsic and extrinsic evaluations.

5.6.1. Intrinsic evaluation

The Intrinsic evaluation is conducted to measure the quality of the automatically generated thesaurus. Intrinsic evaluation of automatically constructed thesaurus is done by comparing the system generated thesaurus terms against existing reference lexicons. However, in the Afaan Oromo language, we don't have such kinds of reference lexicon. As a result, the study performed intrinsic evaluation by measuring the degree of similarity that each thesaurus terms have with the term for which the thesaurus terms are generated. That can be taken as a measure of the relatedness of the terms.

To perform intrinsic evaluation twenty words were randomly selected. For each of the selected terms top five, thesaurus terms were generated by the system. The selected terms together with their corresponding thesaurus terms were given to three language experts. The language experts give a score that ranges from 1 up to 5 for each system generated thesaurus term based on their relatedness. Where the grade level 5 presents Very Good, 4 represents Good, 3 represents Fair, 2 represents Poor, and 1 represents Very Poor. To get the average evaluation result of the relatedness of each thesaurus term, the score given by the three evaluators for that particular term is added and the result is divided by 15 (which is the sum of the maximum score of the three language experts), then the result is multiplied by 100 to show the result in percentage. Then the average of the result of the hundred thesaurus terms is calculated to show how much the automatically constructed thesaurus is good in generating related terms. Based on the judgment given by the language experts, the automatically generated thesaurus generates related terms with an average accuracy of 62.1%. The detailed result for intrinsic evaluation together with the similarity score given by the three language experts is depicted in Appendix E.

As it can be seen from the result obtained from the above experimentation, we can deduce that the constructed automatic thesaurus can generate terms that are related to a given term with average relatedness accuracy of 61.33%. A better result would have been expected if all the variations of morphologically the same words are reduced to their root.

5.6.2. Extrinsic evaluation

Extrinsic evaluation is the other evaluation approach for evaluating the quality of the automatically generated thesaurus. In this evaluation approach, the thesaurus is evaluated by applying it to downstream applications such as part of speech tagging, information retrieval, and so on.

This research aims to integrate query expansion into the Afaan Oromo IR system and hence enhance the performance effectiveness of the Afaan Oromo information retrieval system. To show the performance improvement gained because of the integration of query expansion into the Afaan Oromo IR system, the evaluation of the IR system without query expansion and with the integration of query expansion is performed. For this evaluation, ten queries and 120 indexed documents for which relevance judgment is given by a language expert are used. The result of the evaluations is measured by precision, recall, and F-measure IR performance effectiveness metrics. The result obtained for the two experimentations are discussed in the following subsections.

I. Evaluation of Afaan Oromo IR system without query expansion

As the objective of this research is to integrate query expansion into The Afaan Oromo IR system, it is necessary to evaluate the IR system before the integration of query expansion. This helps to compare the resulting effectiveness of the system before and after integration of query expansion. Figure 5.4 shown below depicts the result of the Afaan Oromo IR system applying without query expansion for query “dhibee kooviid”.

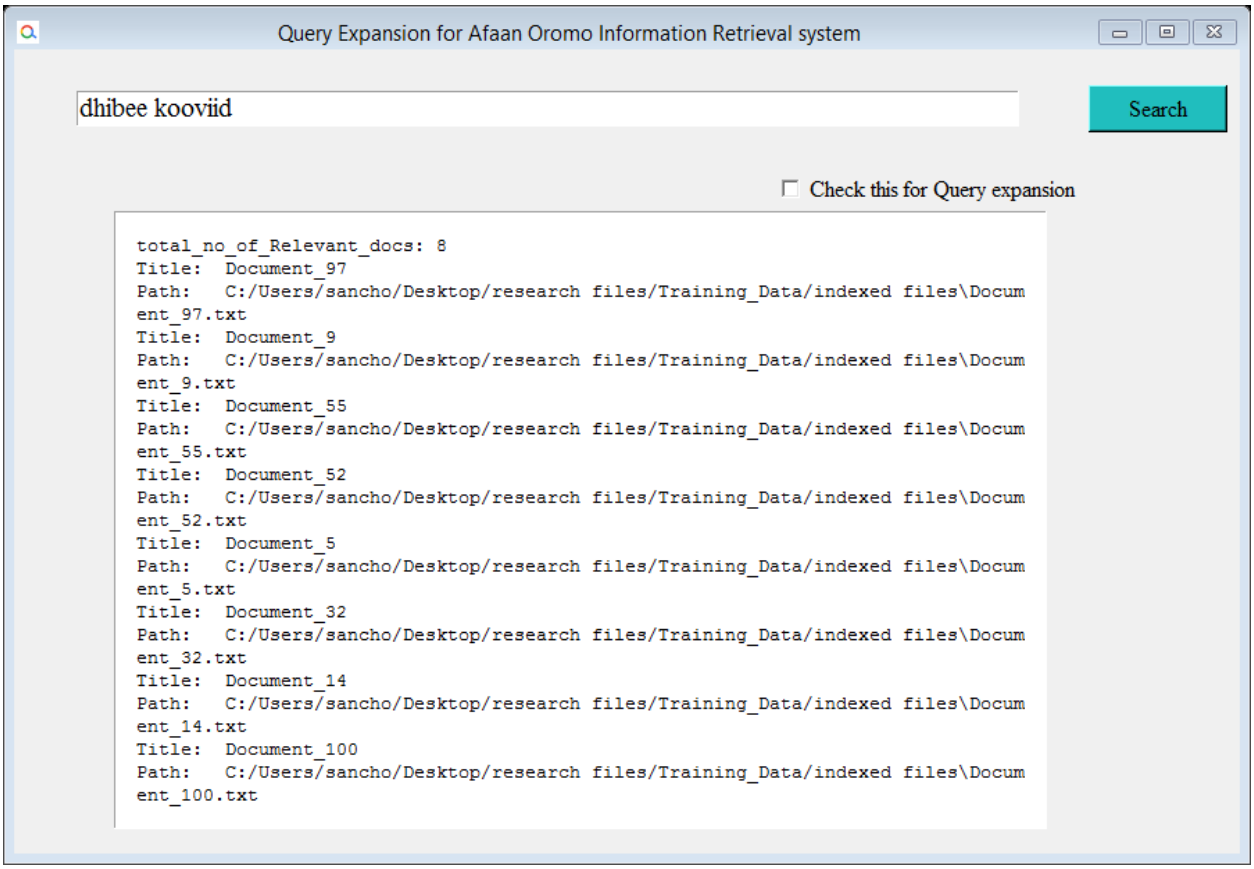


Figure 5.4. The result of Afaan Oromo IR system without QE for example query.

The result in the above figure shows that 8 documents are retrieved as relevant while in our corpus of indexed documents the number of relevant documents for this query according to the relevance judgment given by expert is 15. This means that 7 relevant documents are left without being retrieved. The experimental result for ten queries without query expansion is presented in the following table**.**

Table 5.4. Experimental result of Afaan Oromo IR without query expansion

Query	No of Relevant documents	No of Retrieved Documents	No of Relevant Retrieved	precision	Recall	F-measure
Q1	15	8	3	0.375	0.2	0.260
Q2	15	15	11	0.733	0.733	0.733
Q3	22	27	20	0.740	0.909	0.816
Q4	15	13	9	0.692	0.6	0.642
Q5	13	19	13	0.684	1	0.812
Q6	8	17	6	0.352	0.75	0.48
Q7	11	22	4	0.181	0.363	0.242

Q8	15	14	13	0.928	0.866	0.896
Q9	15	29	15	0.517	1	0.681
Q10	14	32	13	0.406	0.928	0.565
Average				0.561	0.735	0.613

The above table summarizes the result of the Afaan Oromo IR system without query expansion. As shown in the above table the retrieval system without query expansion for ten queries registered an average result of 56.1 % precision, 73.5% recall, and 61.3% F- measure.

II. Evaluation of Afaan Oromo IR with query expansion

In this study, an automatically constructed Afaan Oromo thesaurus is used as a source of expansion terms for expanding short queries of the user. The queries are reformulated by adding the top five related terms from the thesaurus. The expansion terms are selected based on their similarity with the entire query. The number of expansion terms is restricted to five to reduce the complexity of the evaluation and also to expand the query with the terms that are highly related to the whole query. To use query expansion functionality, the checkbox of the developed prototype should be checked. Once the user checked the checkbox, the top five terms that are highly related to the entire query are automatically added to the query. Then the expanded query is sent to the retrieval system. The result of the retrieval system with query expansion for the same query used above is as shown in the figure below.

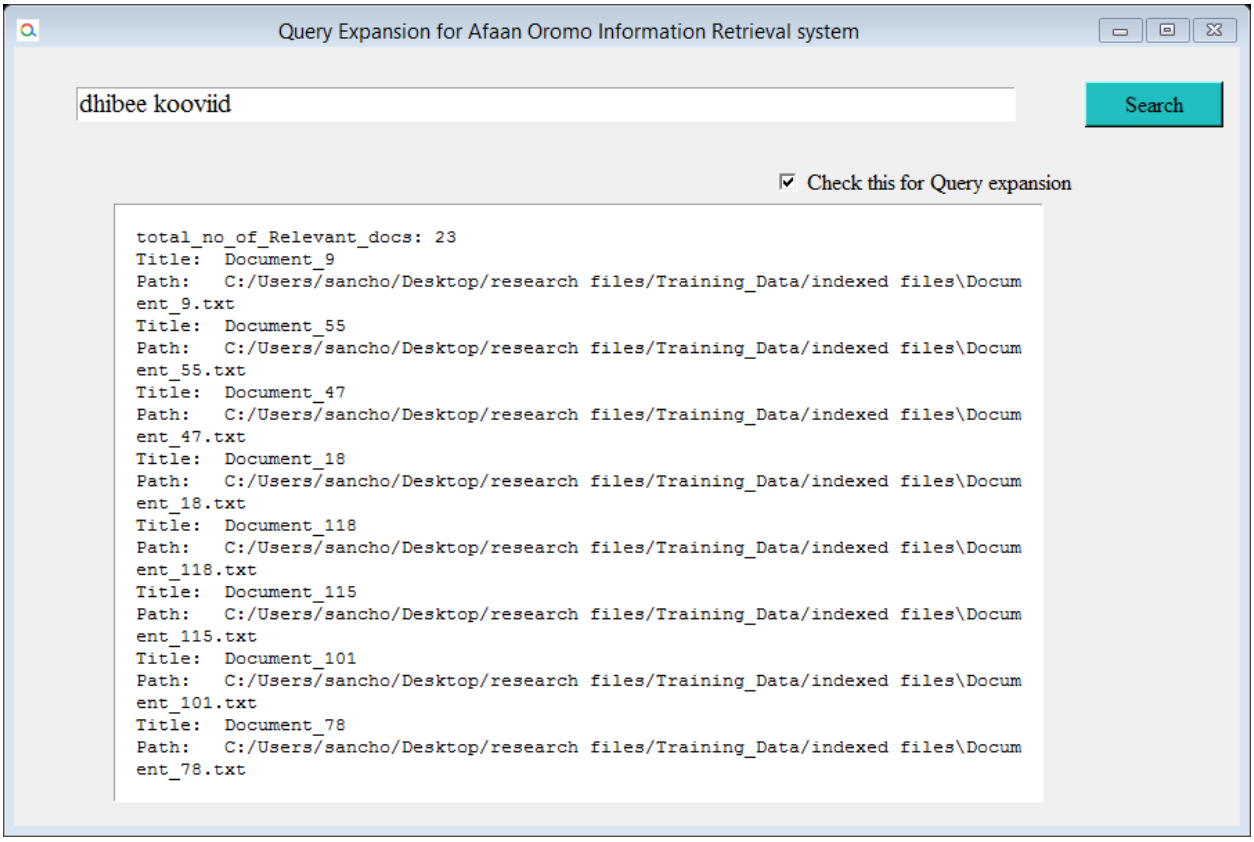


Figure 5.5. The result of Afaan Oromo IR system with QE for example query.

As it is shown in the above figure the retrieval system with QE retrieves 23 documents. Among the retrieved documents, 13 documents are marked as relevant whereas 10 documents are marked as irrelevant in the corpus of indexed documents. This shows that integration of query expansion into the Afaan Oromo Information retrieval system improves the search space and enhances the retrieval effectiveness over searching by using the original user query alone. The complete evaluation result for ten queries is presented in the table below.

Table 5.5. Experimental result of Afaan Oromo IR with query expansion

Query	No of Relevant documents	No of Retrieved Documents	No of Relevant Retrieved	Precision	Recall	F-measure
Q1	15	23	13	0.565	0.867	0.684
Q2	15	24	13	0.541	0.867	0.667
Q3	22	30	20	0.667	0.909	0.769
Q4	15	20	11	0.55	0.733	0.628
Q5	13	26	13	0.5	1	0.667
Q6	8	20	8	0.4	1	0.571

Q7	11	28	6	0.214	0.545	0.307
Q8	15	17	14	0.823	0.933	0.875
Q9	15	32	15	0.468	1	0.638
Q10	14	39	13	0.333	0.928	0.491
Average				0.506	0.878	0.642

Table 5.5 shows the result of the Afaan Oromo IR system with query expansion. As it is clearly shown in the above table the retrieval system registered an average result of 50.6 % precision, 87.8% recall, and 64.2 % F-measure. The result obtained by integrating query expansion into the Afaan Oromo IR system outperforms the result obtained by the retrieval system without QE for all measures except precision. Since this research aimed to improve the performance effectiveness of the Afaan Oromo IR system, based on the above evaluation, we can conclude QE can improve the retrieval effectiveness of the Afaan Oromo IR system.

5.7. Discussion

In this study, both intrinsic and extrinsic evaluations are conducted. Intrinsic evaluation has been conducted to measure the quality of the constructed thesaurus. To perform this evaluation three language experts were involved to give a score that ranges from 1 to 5 for each term that the system generates as a related term to a given term. For each test term, five thesaurus terms were given a score. The reason for performing intrinsic evaluation based on the score given by language experts was because of the absence of a reference lexicon with which the generated thesaurus terms are compared. Based on the relatedness score given by language experts, the obtained average relatedness result for 50 terms generated for the ten terms achieved a percentage performance of 61.33%. This implies that the automatically generated thesaurus is promising in generating related terms and enhancing retrieval systems.

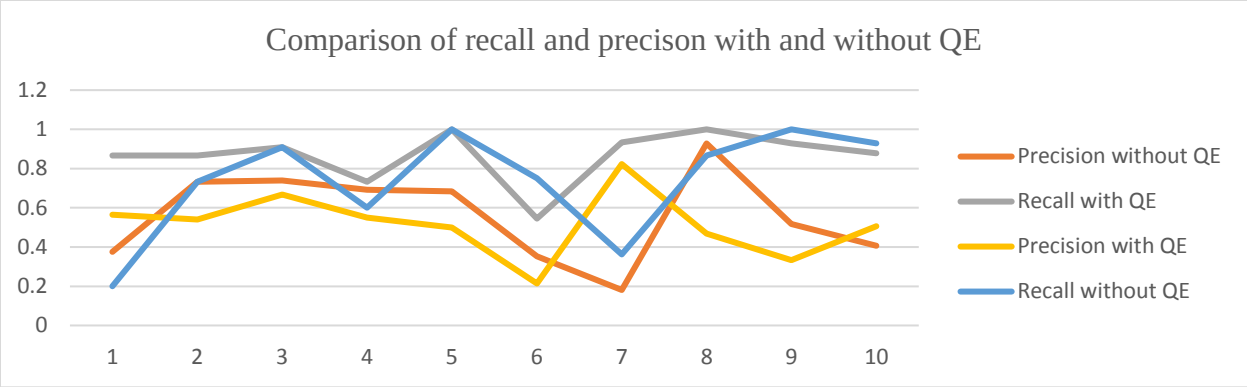


Figure 5.6. Comparison of retrieval results with and without QE.

On the other hand, in extrinsic evaluation, the result for evaluation of Afaan Oromo IR system without QE, registered an average result of 56.1 % precision, 73.5% recall, and 61.3% F- measure. Whereas Afaan Oromo IR system with QE showed an average result of 50.6 % precision, 87.8% recall, and 64.2 % F-measure. This means the result obtained by integrating query expansion into the Afaan Oromo IR system shows an improvement by 14.3 % recall and 2.9 % F-measure. On the other hand, the result for precision shows a decrement by 5.5 %. The reason why the result of precision shows decrement is that, as the number of retrieved documents increases or recall increases, precision always decreases. Therefore, the obtained result tells us that integration of query expansion can improve the performance of retrieval systems where recall is more important. For example paralegals and intelligence analysts are very concerned with trying to get as high recall as possible. Peoples searching their hard disks are usually interested in high recall searches [1]. In addition, the result of the above evaluation shows that an automatically constructed thesaurus can be used as a good source of expansion terms for query expansion. This is because the constructed thesaurus is effective at generating similar or related terms. The quality of the thesaurus would have been better if the number of documents used in the training increases and also good stemmer is applied to reduce morphologically the same words into single stem.

Chapter Six: Conclusion and Future Works

6.1. Conclusion

Information has an important role in the day-to-day activities of the people. Since the development of the World Wide Web, the number of textual information shared in various domains and different languages increase dynamically. Due to the explosion of information over the internet, whenever the users want to retrieve relevant information, it is difficult to retrieve and satisfy one`s own information need. This is because of the inability of the users to formulate a good query and also the terminological variation among the world of readers and the world of authors. The reason for the terminological variation is because the same concepts can be described by different terms. Therefore, retrieval system that works based on keyword matching, face difficulty in retrieving relevant documents that are written in different terminology. This is because of the occurrence of the term mismatching problem. This problem greatly reduces the performance effectiveness of IR systems.

To deal with the term mismatching problem and to increase the performance of the retrieval systems, query expansion is one of the ideal solutions. Query expansion, aimed at reformulating the user's original query by adding related terms. This study tried to develop query expansion for Afaan Oromo IR using an automatic thesaurus. To achieve the objective of this study various tasks were performed, the first activity was corpus development. Here a list of stop words, abbreviations, and affixes was collected. In addition to this, a total of 5,070 heterogeneous documents were manually collected for training the word2vec model. On the other hand, 120 documents and 10 queries were selected for evaluation purposes. Relevance judgment was made by a language expert for each query-document pair in the test collection.

After the completion of corpus preparation, the collected documents were preprocessed. The preprocessing activities include tokenization (both sentence and word), normalization, stop word removal, and stemming. Then the next step was to build an automatic thesaurus. To construct automatic thesaurus, the first step was vectorization, here the preprocessed sentences were given to word2Vec`s skip-gram model where each vocabulary terms get their vector representation. The vector representation of the terms was used to calculate the similarity between terms. For a given

term, the top 10 thesaurus terms were generated based on their cosine similarity score with the given term. For expansion term selection, the study followed a one-to-many association approach. Where the thesaurus terms are selected based on their relatedness with the entire query rather than the individual query terms. Therefore top five thesaurus terms with the highest score are added to the query of the user to expand it. Then the reformulated query was used to retrieve more relevant documents and hence, enhance the performance effectiveness of the Afaan Oromo IR system.

To measure the performance of the proposed system both intrinsic and extrinsic evaluations were performed. Intrinsic evaluation was performed to measure the quality of the constructed thesaurus. Three language experts were involved to judge the quality of the constructed thesaurus. On the other hand, extrinsic evaluation was performed to observe the resulting difference in the effectiveness of the retrieval system because of the integration of QE into the Afaan Oromo IR system.

This study tried to answer two research questions. The first one was related to the extent to which the automatically generated thesaurus terms are related to the given query term. To answer this question, an intrinsic evaluation was performed. Because of the lack of ready-made reference lexicons with which the automatically constructed thesaurus is compared, the evaluation was performed by giving 20 randomly selected terms together with their top five thesaurus terms from the thesaurus to language experts. Then for each thesaurus term, the experts gave a score that ranges from 1 to 5 based on their relatedness, and the percentage relatedness accuracy of 62.1%. % was achieved. This means an automatically generated thesaurus can generate thesaurus terms that are related to the term for which they are generated with an average accuracy of 62.1%. The other question was related to the extent to which QE improves the performance effectiveness of the Afaan Oromo IR system. To answer this question, an evaluation of the Afaan Oromo IR system before and after integration of QE was performed. Then the obtained result showed that integrating query expansion into the Afaan Oromo IR system registered performance improvement by 14.3 % recall, 2.9 % F-measure, and performance decrement of 5.5% for precision.

To conclude, the performed experimentation testified that automatically constructed thesaurus is good in generating semantically similar terms and can also be used as a good source for obtaining expansion terms. Moreover, applying QE into the Afaan Oromo IR system is a good approach for

enhancing the performance of the Afaan Oromo retrieval system especially at increasing the number of relevant documents retrieved.

6.2. Contribution of the thesis

The contributions of this study are summarized as follows:

- We proposed the general architecture of thesaurus-based QE for the Afaan Oromo IR system.
- Prior work on Afaan Oromo text retrieval stated that the performance of his study was affected by the term mismatching problem and recommended that query expansion can handle this problem. Hence, this study filled this gap by integrating QE into Afaan Oromo IR and tried to reduce the effect of the term mismatching problem.
- This study constructed an automatic thesaurus for Afaan Oromo.

6.3. Recommendation and future works

Based on the knowledge gained and the findings of this thesis, the following recommendations are forwarded for future work:

- If ambiguous terms are found in the query, to know the exact sense of the term and expand the query based on its correct sense, integrating query expansion with query disambiguation might be considered.
- The absence of a standard corpus for evaluation makes the evaluation of the Afaan Oromo IR system very challenging. Therefore, the development of a well-designed corpus of Afaan Oromo document-query pairs with its relevance judgment can solve this problem.
- To evaluate the quality of the generated thesaurus, the standard evaluation mechanism is to compare the generated thesaurus terms against existing reference lexicons. However, there is no ready-made reference lexicon for Afaan Oromo. Therefore, the development of reference lexicons such as WordNet can be an ideal solution.

References

[1] C. D. Manning, P. Raghavan and H. Schütze, An Introduction to Information Retrieval, Cambridge: Cambridge University Press, 2009.

[2] A. Spink, D. Wolfram, Jansen and T. Saracevic, "Searching the web: The public and their queries.," *Journal of the American Society for Information Science and Technology,* vol. III, no. 52, pp. 226-234, 2001.

[3] Deepak, K. Hiteshwar and A. Azad, "Query Expansion Techniques for Information Retrieval: a Survey," *ELSEVIER,* no. 56, pp. 1698-1735, 2019.

[4] C. Claudio and R. Giovanni, "A survey of automatic query expansion in information retrieval," *Acm Computing Surveys (CSUR),* vol. 44, no. 1, pp. 1-50, 2012.

[5] C. Hang, W. Ji-Rong, N. Jian-Yun and M. and Wei-Ying, "Query Expansion by Mining User Logs," *IEEE,* vol. 15, no. 4, pp. 829-839, 2003.

[6] J. X. Croft and W. Bruce, "Query Expansion Using Local And Global Document Analysis," *ACM SIGIR Forum,* vol. 51, no. 2, pp. 168-175, 2017.

[7] Pedersen, s. HIintich and J. 0., "A Co-occurrence-Based Thesaurus and Two Applications to Information Retrieval," *Else& Science Ltd,* no. 3, pp. 307-318, 1997.

[8] Baeza-Yates, F. William and Ricardo, Information Retrieval: Data Structures & Algorithms, Prentice Hall, 1992.

[9] J. Martin and J. H, Speech and Language Processing: An introduction to natural language processing, Computational linguistics, and speech recognition., 2007.

[10] A. Lenci, "Distributional semantics in linguistic and cognitive research," *Rivista di Linguistica* , vol. I, no. 20, pp. 1-31, 2008.

[11] U. Kamath, J. Liu and J. Whitaker, Deep Learning for NLP and Speech Recognition, Gewerbestrasse: Springer, 2019.

[12] G. Sahar, F. Benoit, E. Yannick and C. Nathalie, "Word Embeddings Evaluation and Combination," *ACL Anthology*, pp. 300-306, 2016.

[13] M. Tomas, S. Ilya, C. Kai, C. Greg and D. Jeffrey, "Distributed Representations of Words and Phrases and their Compositionality," *Advances in Neural Information Processing Systems 26.*, pp. 1-9, 2013.

[14] X. Rong, "word2vec Parameter Learning Explained," 2016.

[15] B. Benjamin, B. Rebecca and O. Tony, Applied Text Analysis with Python, Boston: O'Reilly Media, Inc.,, 2018.

[16] B. I., "The Origin of Afaan Oromo: Mother Lanugage," vol. 15, no. 12, 2015.

[17] T. Gamta, "Qube Afaan Oromo: Reasons for Choosing the Latin Script for Developing an Oromo Alphabet," *The Journal of Oromo Studies*, no. 1, 1993.

[18] H. Mekonnen, "Lexical Standardization in Oromo," *Masters Thesis, School of Graduate Studies, Addis Ababa*, 2002.

[19] G. Gezehagn, "Afaan Oromo Text Retrieval System," *Masters thesis Department of Information Science, Addis Ababa University*, 2012.

[20] T. Debela, "Afaan Oromo Search Engine," *Masters Thesis, Department of Computer science, Addis Ababa University,* 2010.

[21] G. A. Miller, "WordNet: A Lexical Database for English," *Communication of the ACM*, vol. 38, no. 11, pp. 39-41, 1995.

[22] B. Anbase, "Applications of Information Retrieval for Afaan Oromo text based on Semantic-based Indexing," *Masters Thesis, School of Computing, Jimma University,* 2019.

[23] T. Tadele, "Applying Thesaurus Based Semantic Compression For Afaan Oromo Text Retrieval," *Masters Thesis, School of Computing, Jimma University,* 2019.

[24] A. Singhal, "Modern Information Retrieval: A Brief Overview," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering,* vol. IV, no. 24, pp. 35-43, 2001.

[25] B.-Y. Ricardo and R.-N. Berthier, Modern Information Retrieval, the concepts and technology behind search, Harlow, England: Addison Wesley, 2011.

[26] A. Zazo, C. Figuerola, J. Berrocal, L. Alonso and E. Rodriguez, "Reformulation of queries using similarity thesauri," *Information Processing and Management,* no. 41, p. 1163–1173, 2005.

[27] B. Croft and J. Lafferty, "Language modeling for information retrieval.," *Springer Science & Business Media.,* no. 13, pp. 11-56, 2013.

[28] Yang, C. Carolyn and Bokyung, "Experiments in automatic statistical thesaurus construction," in *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, Minnesota,, June 1992.

[29] Y. Croft and W. Bruce, "An Association Thesaurus for Information Retrieval," in *Proceedings of RIAO*, New York: Rockefeller University., 1994.

[30] Y. Choi, C. Park and Key-Sun, "Automatic Thesaurus Construction using Bayesian Networks," *lnformation Processing & Management,* no. 5, pp. 543-553, 1996.

[31] J. Martin and J. H, "WordNet: Word Relations, Senses, and Disambiguation," in *Speech and Language Processing.*, 2018, pp. 1-26.

[32] P. Dipasree, M. Mandar and D. Kalyankumar, "Improving Query Expansion Using WordNet," *Journal Of The Association For Information Science And Technology,* pp. 1-10, 2014.

[33] S. Zewdneh, "Word Sense Disambiguation using Semantic Similarity for Query Expansion in Amharic Information Retrieval," *Masters Thesis, School Of Information Science , Addis Ababa Univeristy,* 2014.

[34] T. Zeray, "Query Expansion for Tigrigna Information Retrieval," *Masters thesis, Department of Computer Science, Addis Ababa University,* 2017.

[35] A. Rayner, K. C. A. Patricia, S. Phang, L. I. Tan, L. Leow and S. Gan, "Ontology-Based Query Expansion for Supporting Information Retrieval in Agriculture," in *The 8th International Conference on Knowledge Management in Organizations*, Dordrecht, 2014.

[36] T. Ming, Teng-yang and Zhao, "An Ontology-Based Information Retrieval Model for Vegetables E-Commerce," *Journal of Integrative Agriculture,* vol. V, no. 11, pp. 800-807, 2012.

[37] J. Bhogal, A. Macfarlane and P. Smith, "A review of ontology based query expansion," *Information Processing and Management,* no. 43, p. 866–886, 2007.

[38] W. Janfa, "Ontology Based Query Expansion For Enhancing The Performance Of Amharic Information Retrieval : The Case of Tourism Sector," *Masters Thesis, School of Information Science, Addis Ababa University,* 2014.

[39] L. Hobson, H. Cole and M. P. Hannes, Natural language processing in action, Shelter Island: Manning, 2019.

[40] G. Sahar, F. Benoit, E. Yannick and C. Nathalie, "Word Embeddings Evaluation and Combination," *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16),* pp. 300-306, 2016.

[41] G. Z. M. Wen-tau Yih, "Linguistic Regularities in Continuous Space Word Representations," *Association for Computational Linguistics,* p. 746–751, 2013.

[42] D. Sarkar, "Text Similarity and Clustering," in *Text Analytics with Python*, 2016, pp. 256-317.

[43] M. Tomas, Y. Wen-tau and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," in *Proceedings of NAACL-HLT 2013, Association for Computational Linguistics*, Atlanta, Georgia, 2013.

[44] Vijaymeena and Kavitha, "A Survey on Similarity Measures In Text Mining," *Machine Learning and Applications: An International Journal (MLAIJ),* vol. III, no. 1, pp. 19-28, 2016.

[45] G. Wael and F. Aly, "A Survey of Text Similarity Approaches," *International Journal of Computer Applications*, vol. 68, no. 13, pp. 13-18, 2013.

[46] A. Huang, "Similarity Measures for Text Document Clustering," in *New Zealand Computer Science Research, Student Conference*, Christchurch, 2008.

[47] C. Vincent and K. Ewa, "Distributional Thesauri for Information Retrieval and vice versa," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Portorož, Slovenia, 2016.

[48] D. Tesfaye, "Designing a Stemmer for Afaan Oromo Text: A Hybrid Approach," *Masteres Thesis, Addis Ababa University,* 2010.

[49] A. Mideksa, "Statistical Afaan Oromo Grammar Checker," *Masters thesis, Addis Ababa University, School of Information Science,* 2015.

[50] C. Fita, "Afaan Oromo List, Definition and Description Question Answering System," *Masters Thesis, Addis Ababa University, Department of Computer Science,* 2016.

[51] K. Hordofa, "Towards the Genetic Classification of the Afaan Oromoo Dialects," *Department of Linguistics, and Scandinavian Studies: The University of Oslo,* 2009.

[52] A. Raga and Samuel Adola, "Homonymy as a barrier to mutual intelligibility among speakers of various dialects of Afan Oromo," *Journal of Language and Culture,* vol. III, no. 2, pp. 32-43, 2012.

[53] F. Wakjira, "Automatic Thesaurus Construction From Afan Oromo Text," *Masters Thesis, Bahir Dar University,Institute of Technology, Faculty of Computing,* 2018.

[54] G. Berihun, "Amharic Information Retrieval Using Semantic Vocabulary," *Masters Thesis, Department of Computer Science, Addis Ababa Univeristy,* 2019.

[55] Z. Angel, F. Carlos, B. Jose and E. R. ´guez, "Reformulation of queries using similarity thesauri," *Information Processing and Management,* no. 41, p. 1163–1173, 2005.

[56] A. Mahgoub, M. Rashwan, H. Raafat and M. Zahran, "Semantic Query Expansion for Arabic Information Retrieval," in *Association of Computational Linguistics*, Doha, 2014.

[57] P. Ken, T. Tuure, R. Marcus and C. Samir, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems,* vol. 3, no. 24, pp. 45-77, 2007.

[58] J. Perkins, python 3 Text Processing with NLTK 3 cookbook, Birmingham : PUCKT Publishing Ltd., 2014.

[59] K. Harpreet and G. Vishal, "Indexing Process Insight and Evaluation," in *International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, 2016.

[60] W. Tegegne, "The Development of Written Afan Oromo and the Appropriateness of Qubee, Latin Script, for Afan Oromo Writing," *Historical Research Letter,* vol. 28, pp. 8-14, 2016.

[61] M. Tomas, C. Kai, C. Greg and D. Jeffrey, "Efficient Estimation of Word Representations in Vector space," pp. 1-12, 2013.

List of Appendixes.

Appendix A. List of Abbreviations and their expanded form

Abbreviation	Expanded Form	Abbreviation	Expanded Form
Obb.	Obboo	W.B.	Waree booda
Add.	Addee	W/B	Waaree Booda
Fkn.	Fakkeenyaaf	Ado.	Adoolleessa
Hub.	Hubachiisa	W.D.	Waaree dura
w.k.f	Waan kana fakkaatan	W/D	Waaree Dura
w/k/f	Waan kana fakkaatan	Bil.	Bilbila
k.k.f	Kan kana fakkaatan	Qar.	Qarshii
k/k/f	Kan kana fakkaatan	Bill.	Billiyoona
Ful.	Fulbaana	Mill.	Milliyoona
Sad.	Sadaasa	M/B	Mana Barumsaa
Ama.	Amajjii	M.B	Mana Barumsaa
Onk.	Onkololeessa	I/G	Itti Gaafatamaa
Bit.	Biteetossa	I.G	Itti Gaafatamaa
Mud.	Muddee	M/Murtii	Mana Murtii
Wax.	Waxabajjii	M.Murtii	Mana Murtii
Gur.	Guraandhala	Hosp.	Hoospitaala
Hag.	Hagayya	Lakk.	Lakkofsa
Ebl.	Eebla	Hogg.	Hooganaa
Pirof.	Pirofeesara	Dr.	Doktara
Ykn	Yookiin	G/L/Sha	Godina Lixa shawaa
G/ad/F	Godina Addaa Finfinnee	A.L.A	Akka Lakkoofsa Awurooppa
A/L/A	Akka Lakkoofsa Awurooppa	A.L.I	Akka Lakkoofsa Itoophiyaa
A/L/I	Akka Lakkoofsa Itoophiyaa	Dh/K/B	Dhaloota Kiriistoosin Booda

M.M	Muummee Ministeeraa	H/Bulaa	Hoorsisee Bulaa
Q/Bulaa	Qoonnaan Bulaa	Dh.K.D	Dhaloota Kiriistoosiin Dura
Dh/K/D	Dhaloota Kiriistoosiin Dura	M/Ministeeraa	Muummee Ministeeraa
Dh.K.B	Dhaloota Kiriistoosin Booda		

Appendix B. List of stop words in Afaan Oromo

aaf	chiisee	ifte	jechaan	otoo
aas	chiisna	iifis	jechuu	otta
aati	chiisne	iifuu	jechuun	otumallee
aatii	chiista	iin	kan	otuu
aatu	chisiisa	illee	kanaaf	otuullee
achiisan	chisiisan	immoo	kanaafi	rra
achiise	chisiista	ine	kanaafuu	saaf
achisa	chisiistan	ini	kee	saniif
achuuf	chisiiste	innaa	koo	silaa
akka	chisiistu	inu	kun	simmoo
akkam	dachiisaa	iraanis	lee	sun
akkasumas	dani	irraa	malee	ta`ullee
akkum	dha	irraahuu	moo	tahullee
akkuma	dhe	irraan	nan	tanaaf
aman	dhu	irraanuu	ni	tanaafi
amani	dura	irrattillee	odoo	tanaafuu
amanii	eega	irrattis	ofii	tawullee
ammo	eegana	irrattuu	oggaa	umaaatti
amne	eegasii	isa	oo	umaafdhaa
amni	eetii	isaa	ooftan	utuu
amoo	ennaa	isaan	oolee	waan
amte	erga	isan	ooleedhaan	waggaa
amti	f	ise	ooleef	woo

amtu	fi	iseen	ooleewwan	wwaan
ani	garuu	isisa	ooliidhan	yammuu
anna	hanga	itti	ooliif	yemmuu
anne	henna	ittii	ooliiwwan	yeroo
ata	hin	ittiin	ooma	yommii
atani	hoggaa	ittillee	oota	yommuu
ate	hogguu	ittis	ootaaf	yoo
atti	hoo	ittuu	ootaan	yookaan
booda	icha	itumallee	ootawwan	yookiin
booddee	ichi	ituu	oottan	yookinimoo
chiisan	ifna	ituullee	osoo	yoom

Appendix C. Relevance judgment by language expert

Document/query	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Document_1	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_2	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_3	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_4	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Document_5	R	NR	NR	NR	NR	NR	NR	R	NR	NR
Document_6	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_7	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_8	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Document_9	NR	R	NR	NR	NR	NR	NR	NR	R	NR
Document_10	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Document_11	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Document_12	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Document_13	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Document_14	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Document_15	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_16	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_17	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_18	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Document_19	NR	NR	R	NR	R	NR	NR	NR	NR	NR
Document_20	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Document_21	NR	R	NR	NR	NR	NR	NR	NR	R	NR
Document_22	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Document_23	NR	NR	R	NR	R	NR	NR	NR	NR	NR
Document_24	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Document_25	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Document_26	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Document_27	NR	NR	NR	NR	NR	NR	R	NR	R	NR
Document_28	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Document_29	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Document_30	NR	NR	NR	NR	NR	R	R	NR	NR	NR
Document_31	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Document_32	R	NR	NR	NR	NR	NR	NR	R	NR	NR
Document_33	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Document_34	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Document_35	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Document_36	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Document_37	NR	NR	R	NR	R	R	NR	NR	NR	NR

Document_38	NR	NR	NR	NR	R	R	NR	NR	NR	NR
Document_39	NR	NR	NR	NR	R	R	NR	NR	NR	NR
Document_40	NR	NR	R	NR	R	R	NR	NR	NR	NR
Document_41	NR	NR	R	NR	R	NR	NR	NR	NR	NR
Document_42	NR	NR	R	NR	R	NR	NR	NR	NR	NR
Document_43	NR	NR	R	NR	R	R	NR	NR	NR	NR
Document_44	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Document_45	NR	R	NR	NR	NR	NR	NR	NR	R	NR
Document_46	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Document_47	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_48	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Document_49	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_50	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_51	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Document_52	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_53	NR	NR	R	NR	NR	R	NR	NR	NR	R
Document_54	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Document_55	NR	R	NR	NR	NR	NR	NR	NR	NR	NR
Document_56	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Document_57	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Document_58	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_59	NR	NR	NR	NR	R	NR	NR	NR	NR	NR
Document_60	NR	NR	NR	NR	NR	R	R	NR	NR	NR
Document_61	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Document_62	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Document_63	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Document_64	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_65	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_66	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Document_67	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Document_68	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Document_69	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Document_70	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Document_71	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Document_72	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Document_73	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Document_74	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Document_75	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_76	NR	NR	R	NR	NR	NR	NR	NR	NR	NR

Document_77	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Document_78	R	NR	NR	NR	NR	NR	NR	NR	NR	R
Document_79	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Document_80	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Document_81	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_82	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Document_83	NR	NR	R	NR	R	NR	NR	NR	NR	NR
Document_84	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_85	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_86	NR	NR	NR	R	NR	NR	NR	NR	NR	R
Document_87	NR	R	NR	NR	NR	NR	NR	NR	R	NR
Document_88	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_89	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Document_90	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Document_91	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Document_92	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Document_93	NR	NR	NR	NR	NR	R	R	NR	NR	NR
Document_94	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Document_95	NR	NR	NR	NR	NR	R	NR	NR	NR	NR
Document_96	NR	NR	R	NR	R	R	NR	NR	NR	NR
Document_97	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Document_98	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_99	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Document_100	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Document_101	R	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_102	NR	NR	NR	R	NR	NR	NR	NR	NR	R
Document_103	NR	NR	NR	NR	NR	NR	NR	NR	NR	R
Document_104	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_105	NR	NR	R	NR	NR	NR	NR	R	NR	NR
Document_106	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Document_107	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Document_108	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Document_109	NR	R	NR	NR	NR	NR	NR	R	NR	NR
Document_110	NR	NR	NR	NR	NR	NR	R	NR	NR	NR
Document_111	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
Document_112	NR	NR	NR	NR	NR	NR	NR	R	NR	NR
Document_113	NR	NR	R	NR	NR	NR	NR	NR	NR	NR
Document_114	NR	NR	NR	NR	NR	NR	NR	NR	R	NR
Document_115	NR	NR	NR	NR	NR	NR	NR	R	NR	NR

Document_116	NR	NR	NR	R	NR	NR	NR	NR	NR	NR
Document_117	R	NR	NR	R	NR	NR	NR	NR	NR	NR
Document_118	R	NR	NR	NR	NR	NR	NR	R	NR	NR
Document_119	R	NR	NR	R	NR	NR	NR	NR	R	R
Document_120	NR	R	NR	NR	NR	NR	R	NR	NR	NR

Key: Q: Query, R: Relevant, NR: Non-Relevant

Appendix D. Thesaurus Evaluation Guideline

Dear Evaluator,

This questioner aims to evaluate the quality of automatic thesaurus for Afaan Oromo, thesaurus is a book of words together with their synonyms. But the words need not necessarily be synonyms, if they are somehow related in meaning they can be considered as thesaurus terms. This thesaurus can be built manually or automatically. But since manual construction is tedious, automatic construction is preferable. Hence, an automatic thesaurus gives a set of related terms to a given term. For evaluation purpose top five thesaurus terms together with the term for which the related terms are generated is provided to you. Then you are kindly requested to provide your evaluation based on how much the generated terms are related. The evaluation score ranges from 1 to 5. A score of 5 = Very Good, 4 = Good, 3 = Fair, 2 = Poor, 1 = Very Poor. Please put your answer either by marking or by writing scores in the given Table. For instance, if you judge the semantic relatedness (meaning similarity) of the two terms as very strong, so you can answer it either by writing 5 or marking under 5 in the Table as shown below.

No	**Query term**	Thesaurus term	**1**	**2**	**3**	**4**		**5**
1	Term 1	Thesaurus term 1						X

Thank You Very Much for Your Time!

Appendix E. Thesaurus evaluation result

No		Generated thesaurus terms	Evaluator-1	Evaluator-2	Evaluator-3	Total score	Average score	Percentage (%)
1	waljaalachuu	waliiyaaduu	5	4	5	14	0.933	93.33333
		walutubuu	3	4	3	10	0.667	66.66667
		walkabajuu	4	3	3	10	0.667	66.66667
		aadaasaa	1	1	1	3	0.2	20
		walooma	3	2	3	8	0.533	53.33333
2	jinnii	bi'eelzebuulii	4	5	5	14	0.933	93.33333
		milikkitaa	1	2	1	4	0.267	26.66667
		seexanootaa	5	5	5	15	1	100
		xuraawoo	3	3	4	10	0.667	66.66667
		fayye	1	1	1	3	0.2	20
3	qajeelfama	qajeelfam	1	4	1	6	0.4	40
		dambii	5	5	5	15	1	100
		qajeelfamootaa	5	5	5	15	1	100
		baastuu	1	1	1	3	0.2	20
		hojmaataa	5	5	5	15	1	100
4	tawaahidoo	ortodoksi	5	5	5	15	1	100
		kiristaana	5	5	5	15	1	100
		kiristiyaanaa	5	5	5	15	1	100
		phaaphaasii	3	4	3	10	0.667	66.66667
		maatiyaas	2	1	1	4	0.267	26.66667
5	dharaa	hatin	2	1	2	5	0.333	33.33333
		ejjin	1	1	2	4	0.267	26.66667
		sobduu	5	5	5	15	1	100
		sobanii	5	5	5	15	1	100
		dhara	5	5	5	15	1	100
6	qarshii	birrii	5	5	5	15	1	100
		biliyoona	3	2	2	7	0.467	46.66667
		doola	1	1	1	3	0.2	20
		biiliyoona	3	2	2	7	0.467	46.66667
		oliin	1	1	1	3	0.2	20
7	godambaa	muuziyeemii	5	5	5	15	1	100
		eebbisisee	1	1	1	3	0.2	20

		turtii	1	1	1	3	0.2	20
		qarodhabeeyyii	1	1	1	3	0.2	20
		ammayaa	1	1	1	3	0.2	20
8	hawaasa	ummata	5	5	5	15	1	100
		hawaas	3	1	1	5	0.333	33.33333
		dargaggoot	2	1	2	5	0.333	33.33333
		muslimaa	1	1	1	3	0.2	20
		babal'atanii	1	1	1	3	0.2	20
9	hayyuu	piroofeesar	4	5	4	13	0.867	86.66667
		xiinxalaa	5	4	4	13	0.867	86.66667
		dubbisnee	1	1	1	3	0.2	20
		gameessi	5	3	4	12	0.8	80
		dhaloon	1	1	1	3	0.2	20
10	koronaa	koroonaa	5	5	5	15	1	100
		vaayirasii	3	4	3	10	0.667	66.66667
		koviid	5	5	5	15	1	100
		koronaavaayirasii	5	5	5	15	1	100
		vaayrasii	3	4	3	10	0.667	66.66667
11	intala	durba	5	4	5	14	0.933	93.3333
		da`ee	3	5	3	11	0.733	73.3333
		obboleettii	4	4	3	11	0.733	73.3333
		deessee	3	5	3	11	0.733	73.3333
		taamaar	1	1	1	3	0.2	20
12	bushaayee	harree	2	1	2	5	0.333	33.3333
		saawwa	3	4	3	10	0.667	66.6667
		coccoomoo	3	3	4	10	0.667	66.6667
		qalamanii	1	2	1	4	0.267	26.6667
		saa	3	3	4	10	0.667	66.6667
13	dhukkuba	dhukkub	5	4	5	14	0.933	93.3333
		dhibee	5	5	5	15	1	100
		teetaanasii	4	4	5	13	0.867	86.6667
		lamxii	4	4	5	13	0.867	86.6667
		dhukkubaa	5	5	5	15	1	100
14	jal`ina	hammina	4	3	4	11	0.733	73.3333
		haxxummaa	3	4	4	11	0.733	73.3333
		ejjummaa	2	3	2	7	0.467	46.6667
		aarin	1	1	1	3	0.2	20
		ciigga'a	3	3	4	10	0.667	66.6667
15	dhirsa	miinjee	1	1	2	4	0.267	26.6667

		dhirsi	4	4	5	13	0.867	86.6667
		misirroo	5	4	4	13	0.867	86.6667
		iyyitee	1	1	1	3	0.2	20
		dibbee	1	1	1	3	0.2	20
16	shororkeessummaa	shororkaa	5	5	4	14	0.933	93.3333
		shororkeessitootaa	5	5	5	15	1	100
		shanee	4	3	5	12	0.8	80
		shororkeessit	2	3	1	6	0.4	40
		al-shabaab	5	5	5	15	1	100
17	aadaa	duudhaa	5	5	5	15	1	100
		ammayyummaa	1	1	1	3	0.2	20
		calaqqee	2	4	3	9	0.6	60
		tuurizimii	3	2	3	8	0.533	53.3333
		heddummina	1	1	1	3	0.2	20
18	atileetiksii	federeeshina	3	2	3	8	0.533	53.3333
		olompikii	4	5	4	13	0.867	86.6667
		shaampiyoonaa	4	4	3	11	0.733	73.3333
		paaraalompikii	4	3	4	11	0.733	73.3333
		ispoorteessitoota	4	4	3	11	0.733	73.3333
19	dinqii	milikkitaa	5	5	5	15	1	100
		argisiisee	3	4	5	12	0.8	80
		jajamaa	4	4	3	11	0.733	73.3333
		gugurdaa	1	1	1	3	0.2	20
		hojjetes	1	1	1	3	0.2	20
20	dorgommi	liivarpuul	1	2	1	4	0.267	26.6667
		miiniimaa	2	3	2	7	0.467	46.6667
		jaanmeedaa	2	2	3	7	0.467	46.6667
		teekuwaandoo	3	4	3	10	0.667	66.6667
		mo'annaa	2	4	2	8	0.533	53.3333
Average Relatedness Score of all Thesaurus terms								62.1%

Appendix F. Sample code

```
from __future__ import division

from whoosh import index

from whoosh.fields import Schema, ID, TEXT
```

```
from whoosh.writing import AsyncWriter
from whoosh.qparser import QueryParser,OrGroup
from whoosh import scoring,qparser
from whoosh.index import open_dir
from gensim.models.keyedvectors import KeyedVectors
from whoosh.qparser import QueryParser
from whoosh.qparser.dateparse import DateParserPlugin
from nltk.tokenize import sent_tokenize, word_tokenize
from gensim.models import Word2Vec
from gensim.models.keyedvectors import KeyedVectors
from Afaan_Oromo_Stemmer import Afaan_Oromo_Words_Stemmer
from Afaan_Oromo_light_weight_stemmer import Afaan_Oromo_Stemmer
from Afaan_Oromo_light_weight_stemmer import Ao_stemmer
from gensim.models import Phrases
from whoosh import *
import multiprocessing
import tkinter as tk
import os.path
import string
import re
# preprocessing
def doc_preprocessing(doc):
    punctuations ='"!"#$%&()*+,./:;<=">?@`[\]^_{|~}“”~0123456789"'
    length=len(punctuations)
    doc = re.sub(r'\b[0-9]+\b\s*', '', doc)
```

```
    doc_token=doc.split()
    doc_lowered = [term.lower() for term in doc_token]
    normalized_doc=normalization(doc_lowered)
    stop_removed_doc=stopword_removal(normalized_doc)
    doc =[term.translate(str.maketrans(punctuations,' '*length)).replace(' '*4, ' ').replace(' '*3,
     ' ').replace(' '*2, ' ').replace(' '*2, ' ').strip() for term in stop_removed_doc]
    stemmed_docs = stemmer(doc)
    string_doc=' '.join(stemmed_docs)
    return string_doc
```

 # Thesaurus generation

```
def thesaurus_generation (token_query):
    model_name = "C:/Users/sancho/Desktop/research files/Training_Data/religious
domain/religious_model.vec"
    model =KeyedVectors.load_word2vec_format(model_name,unicode_errors='strict')
    vocabulary_list=list(model.vocab)
    sp =sorted(model.vocab.keys())
    tokenized_query=token_query.split()
    term_list=[]
    non_vacab_list=[]
    candidate_set=[]
    out_of_vocab=[]
    word_in_vocab=[]
    expandeded_query = []
    global similar_words
    for terms in tokenized_query:
        if terms not in vocabulary_list:
```

```
            textbox.insert(1.0,"The term doesnot exist in the vocabulary list:\n")
        else:
            word_in_vocab.append(terms)
    textbox=tk.Text(root,  width=50,height=15,padx=15,pady=15,font=('Times New Roman',14))
    textbox.grid(column=1, row=2,padx=45, pady=15)
    for search_term in word_in_vocab:
        for item in model.most_similar_cosmul(search_term,topn=10):
            similar_words={item[0]:item[1]}
            textbox.insert(1.0,similar_words)
            textbox.insert(1.0,"\n")
```

 # Term selection and query expansion

```
def concept_expansion(preprocessed_query):
    topk=5
    model_name = "C:/Users/sancho/Desktop/research files/Training_Data/religious
domain/religious_model.vec"
    model =KeyedVectors.load_word2vec_format(model_name,unicode_errors='strict')
    vocabulary_list=list(model.vocab)
    candidate_terms = query_expansion(preprocessed_query)
    candidate_set = candidate_terms.split()
    tokenized_query = preprocessed_query.split()
    length_of_query = len(tokenized_query)
    concept_sim_score=0
    candidate_terms_avg_score = {}
    dict_of_candidates_with_avg_score={}
    out_of_voc = []
    in_voc_list=[]
```

```
    expanded_query=[]
    for item in tokenized_query:
        if item not in vocabulary_list:
            out_of_voc.append(item)
        else:
            in_voc_list.append(item)
    for candidate in candidate_set:
        for item in in_voc_list:
            similarity_result = model.similarity(item, candidate)
            concept_sim_score +=similarity_result
            average_score = concept_sim_score/length_of_query
        average_sim_score = {candidate:average_score}
        dict_of_candidates_with_avg_score[candidate]=average_score
        concept_sim_score=0
    sort_candidates = sorted(dict_of_candidates_with_avg_score.items(), key=lambda x:
x[1],reverse=True)
    topk_expansion_terms = sorted(dict_of_candidates_with_avg_score,
key=dict_of_candidates_with_avg_score.get, reverse=True)[:topk]
    expanded_query.append(' '.join(topk_expansion_terms))
    expanded_query.append(' '.join(out_of_voc))
    final_expanded_query=' '.join(expanded_query)
    return final_expanded_query
```

 # indexing

```
def create_index(root):
    if not os.path.exists("index_bible"):
        os.mkdir("index_bible")
```

```
    ix = index.create_in("index_bible", schema=get_schema())
    writer = AsyncWriter(ix)
     file_root= u"C:/Users/sancho/Desktop/research files/Training_Data/indexed files"
    filepaths = [os.path.join(file_root,paths) for paths in os.listdir(file_root)]
    for path in filepaths:
        add_doc(writer, path)
    writer.commit()
def get_schema():
    return
Schema(title=TEXT(stored=True),path=ID(stored=True,unique=True),content=TEXT,textdata=
TEXT(stored=True))
def add_doc(writer, path):
    fileobj = open(path, "r", encoding='mbcs')
    content = fileobj.read()
    content=doc_preprocessing(content)
    path= str(path)
    fileobj.close()
    split_file = os.path.split(path)
    file_name =split_file[1]
    if file_name.endswith('.txt'):
        file_name = file_name[:-4]
    file_name = str(file_name)
    writer.add_document(title=file_name,path=path, content=content)
root= r"C:/Users/sancho/Desktop/research files/"
create_index(root)
```

 # Source code of the GUI using Tkinter

```
import tkinter as tk

root=tk.Tk()

root.title("Query Expansion for Afaan Oromo Information Retrieval system")

root.iconbitmap('C:/Users/sancho/Desktop/research files/search_logo.ico')

def main_screen():

    canvas = tk.Canvas(root,width=600,height=300)

    canvas.grid(columnspan=3, rowspan=3,padx=5,pady=5) #if not working delete the padding

    global query

    global options

    query = tk.StringVar()

    keyword =tk.Entry(root, width=75,textvariable=query,font=('Times New Roman',14))

    keyword.grid(columnspan=2,column=0,row=0,padx=45, pady=15,sticky=tk.W)

    keyword.insert(0,"Enter your query")

    query_content= keyword.get()

    print(query_content)

search_button=tk.Button(root,text="Search",width=10,height=1,bg="#20bebe",command=displa
y_function,font=('Times New Roman',12))

    search_button.grid(column=2,row=0,sticky=tk.W,padx=5,pady=25)

    options = tk.IntVar()

    match_case_check = tk.Checkbutton(root,text='Check this for Query
expansion',variable=options,font=('Times New Roman',12),command=lambda:options.get())

    match_case_check.grid(column=1, row=1, sticky=tk.NE)

    canvas = tk.Canvas(root,width=800,height=200)

    canvas.grid(columnspan=3)

    root.mainloop()
```