Improving *Afaan Oromoo* Information Retrieval using Query Expansion

Ayantu Tamene Daba

A Thesis Submitted to the Department of Computer Science and Engineering School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering Office of Graduate Studies(Artificial Intelligence)

Adama Science and Technology University

Mar 8, 2022

Adama, Ethiopia

Improving *Afaan Oromoo* Information Retrieval using Query Expansion

Ayantu Tamene Daba

Advisor: Dr.Teklu Urgesa(Ph.D)

A Thesis Submitted to The Department of Computer Science and Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering Office of Graduate Studies

Adama Science and Technology University

Mar 8, 2022

Adama, Ethiopia

Approval Page

I/we, the advisors of the thesis entitled **"Improving *Afaan Oromoo* Information Retrieval using Query Expansion"** hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Teklu Urgessa(Ph.D)	_____	_____
Major Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by "**Improving *Afaan Oromoo* Information Retrieval using Query Expansion"** and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in **Computer Science and Engineering**.

_____	_____	_____
Chairperson	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis are contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____	_____	_____
Department Head	Signature	Date
_____	_____	_____
School Dean	Signature	Date
_____	_____	_____
Office of Postgraduate Studies, Dean	Signature	Date

Declaration

I hereby declare that this Master Thesis entitled**"Improving *Afaan Oromoo* Information Retrieval using Query Expansion"** is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Ayantu Tamene Daba	_____	_____
Name of the student	Signature	Date

Recommendation

I/we, the advisor(s) of this thesis, hereby certify that I/we have read the revised version of the thesis entitled **"Improving *Afaan Oromoo* Information Retrieval using Query Expansion"** prepared under my/our guidance by **Ayantu Tamene Daba** submitted in partial fulfillment of the requirements for the degree of Mater's of Science in **Computer Science and Engineering.**

Therefore, I/we recommend the submission of a revised version of the thesis to the department following the applicable procedures.

TekluUrgesa (Ph.D)		
Major Advisor	Signature	Date

ACKNOWLEDGEMENT

First and foremost, I would like to thank God who helped me and make everything possible. Next, I would like to thank my Advisor Dr. Teklu Urgesa(Ph.D) for his encouragement, guidance, and advice. Next, my deepest thanks go to all my family members especially my mom Zenebu Dereje and my two younger sisters for their continuous and endless support throughout my life. Next, I would like to express my heart full thanks to my life partner Fonkaye(M) who helped and supported me in every step while doing this study.

Lastly, I would like to thank Adama Science and Technology University for giving me a chance to attend my post-graduate study for the past two years.

DEDICATION

My greatest dedication goes to the memory of my father Mr. Tamene Daba who passed away one year ago and I would like to say "*Baay'een sii jaaladha Taammii Koo nagaan boqadhu".*

Taammiikoo you will always remain in my heart; I wish I get the chance to thank you for directing me in this way.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

List of Acronyms and Abbreviations

BM25	Best Match 25
BIM	Binary Independence Model
IR	Information Retrieval
QE	Query Expansion
TF	Term Frequency
TF-IDF	Term Frequency Inverse Document
VSM	Vector Space Model
WSD	Word Sense Disambiguation
WWW	World Wide Web

Abstract

In this era technology has a great role in the development of one language and accessing the right information written in a different language is necessary which can be considered as one of the basic human needs. So designing an effective performance of information retrieval system is important to search and retrieve the most relevant document from online or offline. As Afaan Oromoo, all relevant documents cannot be retrieved because of ambiguous words and relevance ranking problems that don't satisfy users' information needs. Afaan Oromoo IR system is developed to improve searching and retrieving of information written in Afaan Oromooand make this language proceed with current technology that satisfies users of this language. In this study, we developed an information retrieval system using one of the probabilistic model approach called BM25 by integrating with an elastic search that makes the searching process fast. However, BM25 improves the search result by optimizing the TF-IDF score for query, word ambiguity decreases the effective performance of Afaan Oromoo information retrieval as one word can have multiple meanings. Thus, the query expansion technique is implemented to improve the retrieval system by disambiguating those words. Afaan Oromoo WordNet is developed with synsets and gloss method construction to reformulate the query and the query is modified using semantic relation through synsets to synsets method. Finally, the query expansion with BM25 is integrated with Afaan Oromoo information retrieval systems to enhance and improve retrieval performance. As per the experimental results carried out, the system registers different values before and after query expansion in terms of recall, precision, and F-measure that is the system registered 87.15%, 82%, and 83.3% respectively before query expansion, and after the query is expanded the retrieval system registered 95.9%, 84.2%, and 89.2% respectively. Generally, the retrieval system is improved by a 5.99% F- measure from the original query. Therefore, developing query expansion with BM25 records better results and improves the performance of the Afaan Oromoo retrieval system. However, the retrieval system performance might be improved better if other semantic relations are included in future work.*

***Keywords**: Information Retrieval, Query Reformulation, BM25, WordNet.*

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the study

The evolution and growth of information retrieval (IR) do not begin with the emergence of the internet; rather it was integrated and prominent over the past ten years. People collected and arranged information for later retrieval usage and retrieval for 400 decades(Baeza-Yates & Riberio-Neto, 1999).

The development and progress of the World Wide Web (WWW) made on search engine technology changed the way how information is collected. The first simple computer-based systems were built in the 1940s(Haigh, 2011), then with the increasing of many automated computer technologies, the potentiality of retrieval systems also raised in both processor speed and storage ability which indicates a rapid improvement of manual-based IR systems to automated methods till current IR system. The emerging and growth of such a system shows improvement of manual techniques of indexing and searching user's information to an automated way which can cover large size data, unlike traditional or earliest method.

Information retrieval is a system that is targeted on how to arrange, store, and retrieve information from a collection of documents. The evolution of the IR system starts with text-based information retrieval which uses a keyword as a query. Many global and local researchers were tried to improve the efficiency of the IR systems for different languages by using different techniques and models even though there was a difference in the number of the relevant documents retrieved(document relevance).

In this day, IR is one of the fast-developing fields (Shiri, 2004) that is mainly related to searching and obtaining relevant information from both structured and unstructured documents. According to research done by Pew internet[1], almost 92% of interested users use a different search engine to search for the information they need, which implies dramatic progress evolved on the development of IR system and become a reason for the evolution and growth of different

[1]https://www.pewresearch.org/topic/internet-technology/

commercial search engines that are well designed with increasing performances such as Bing[2], Yahoo[3], Google[4] , and Amazon[5] are one of influencer IR search engine nowadays. However, Google has a simple interface designed for *Afaan Oromoo* even though the system misses some language-specific aspects as we can see in practice.

Challenges that occurred during retrieving a document are, the user may not access the information he/she is interested that is the document displayed after searching may or may not match a user query or the number of the document retrieved can exceed or below users' need. So, modern information retrieval system employs many techniques that allow us to find relevant information concerning the information need of the user such as the query expansion technique.

Query expansion is a technique of reformulating original query to improve IR performance and it is mainly designed due to ambiguity of words occurring in an all-natural language such as using single terms to represent one information concept or presence of two or more possible meanings in any sentences. Additionally, query expansion also involves a method of searching synonym and polysemy of words, finding other morphological forms of words by stemming, fixing spelling errors, and searching for the corrected form or putting as a suggestion in the result(Azad & Deepak, 2019), therefore perfect IR system only retrieve a relevant document.

Like any other natural language, *Afaan Oromoo* has many set of synonym vocabulary that causes a problem during the information retrieval process. Therefore, the enhanced retrieval system is implemented to overcome this problem by using a query expansion technique that adds more information to satisfy users' needs and retrieve relevant documents.

1.2. Motivation of the Study

For the development of one language, technology is very important. These days the number of documents available on the internet is growing at a very rapid speed and has become quite large over the past few years. To search such documents various types of search engines or information retrieval systems have been performed for numerous languages. The fact that initiated the researchers is enabling the development of Afan Oromo to grow with current

[2] www.Bing.com
[3] www.yahoo.com
[4] www.Google.com
[5] www.Amazon.com

information technology support. When technology grows in one country, the users also choose to use it with their language and need to get relevant information that satisfies their needs. Nowadays, many documents are written in Afan Oromo which is available on the internet. However, because of ambiguity words are there, some relevant documents are not retrieved. This problem and challenge were recommended in the previous researcher's study [6]. Thus, this motivates the researcher to disambiguate the words and enhance the performance of Afan Oromo IR using the probabilistic method.

1.3. Statement of the Problem

Afaan Oromoo is one of the Cushitic languages which is categorized under the Afro-Asiatic language which is spoken by 40 million people in Ethiopia and some other African countries(Ibrahim Bedane, 2015). *Afaan Oromoo* has its writing style and structure. Currently, many books, journals, articles, and magazines are written and available offline and also on the internet. However, accessing a relevant document on a previously designed system is not easy and satisfactory for users of it. Thus, this study is best designed and developed to minimize those problems.

To retrieve one document, the user requests his/her query using a limited set of words. As many previous researchers such as Stefan (Klink, Hust, Junker, & Dengel, 2002) indicates most of the time users use two to three words to write a query on average, because of this it is difficult to identify the exact interest of the user within those words, therefore expected document that fit users need is not retrieved, this is because of those words may have more than one meaning which is called word ambiguity.

To solve such problems, a significant amount of studies are performed on the Afaan Oromoo IR system using various approaches like Isayas W.[43] worked on how to retrieve Afaan Oromoo text by using probabilistic approach and retrieve documents but he didn't apply any algorithm to add synonymous words which help to control the effect of ambiguous words, while Gutema G.[8] make possible retrieval of Afan Oromo text by using vector space model even though he didn't use a modern stemming algorithm and also unable to handle polysemous and synonyms word that affects system performance.

Although some researchers had done on the *Afaan Oromoo* IR system**,** the problem is not solved and it is still challenging to retrieve relevant information. Most of them tried to use a vector space model that is unable to handle uncertainty issues, in which the designed system has an uncertain guess of VSM whether that document contains relevant or non-relevant information concerning user request, therefore due to this problem users may not get a response that satisfies their interest.

Currently, *Afaan Oromoo* becomes the language of research, political welfare, education, and administration many materials are being released timely especially on the internet, and also the users' interest to use those documents is also increasing. To solve the problems mentioned above and meet the need of users, developing an effective information retrieval system is important and necessary. Therefore, based on the problem faced by the users; query expansion with the BM25 approach is developed to improve the performance of *Afaan Oromoo* information retrieval.

1.4. Research Questions

- ✓ RQ1: How to adopt a probabilistic model with query expansion?
- ✓ RQ2: How query expansion technique is applied to reformulate a new query?
- ✓ RQ3: To what extent does the proposed approach enhance the effective performance of the *Afaan Oromo* IR system?

1.5. Objectives of the Study

1.5.1. General Objective

The general objective of this study is to improve *Afaan Oromoo* Information Retrieval using Query Expansion.

1.5.2. Specific objectives

The following specific objectives are formulated through which the general objective had been achieved:

- ✓ To experiment by collecting and organizing *Afaan Oromoo* corpus and query.
- ✓ To design and implement architecture for *Afaan Oromoo* IR system that can retrieve the relevant document from *Afaan Oromoo* text.

- ✓ To apply necessary text preprocessing techniques to make our data easy and understandable.
- ✓ To integrate the BM25 probabilistic model framework with elastic search to increase the number of relevant documents in decreasing order based on users' information needs.
- ✓ To expand query-based semantic relation for better performance of retrieval system.
- ✓ To Evaluate and measure the performance of the proposed prototype using IR measurement techniques such as recall, precision, and F-measure and
- ✓ To recommend the future research direction and put research gaps that are not done in this study.

1.6. Scope and Limitation of the Study

1.6.1. Scope of the Study

This study covers only *Afaan Oromoo* Information Retrieval text-based with the elastic search engine. Image, video, and graphics are not covered in this research. The collection of documents in this corpus is prepared from different political, sport, economic, social, health, education, tourism, and justice. In addition, *Afaan Oromoo* lexical resource is prepared manually to construct senses for query reformulation.

1.6.2. Limitation of the Study

This study is mainly designed to improve the current *Afaan Oromoo* retrieval system even though the following listed issues are a reason for the limitation of this study:-

- ✓ This study is proposed for only text retrieval. Image, video, and graphics are out of this research.
- ✓ The study is designed only based on the Afan Oromo language structure.
- ✓ A limited number of corpus and queries are used as a result of the time factor.

1.7. Significance of the Study

Nowadays, information from different local languages is uploaded online using the internet. The users are also getting this information using technology. Thus, technology has a great role in the growth of one language. As technology grows the speaker and users of that language are also increasing to use technology with their language. So this study is best conducted to put stone for

the future blueprint and project of the *Afaan Oromoo* IR system with this technology. This research also makes it easier for a user to get a relevant document that best matches their requirement and put stone for many researchers who has an intention to improve the performance and efficiency of the current *Afaan Oromoo* developed system in the future.

1.8. Organization of the Thesis

This paper is all about reporting the proposed *Afaan Oromoo* IR system with their experimental results, so it is organized as the following sequence.

Chapter 1- it is an introduction part that covers different topics starting from the background of the study, researcher's motivation, the major problem this proposed system solved with specifying research question, scope, limitation, and objective of the studies also included.

Chapter 2- the second chapter of the study that is concerned with reviewing different literature regarding IR systems such as IR process, IR models, and also about elastic search engine additionally, since this system is developed for *Afaan Oromoo* language, some overview of *Afaan Oromoo* language structure is also included and finally different related works done on different IR language by using IR models are also reviewed.

Chapter 3- is about the research methodology for preparing this proposed system dataset and also different preprocessing steps with their algorithm and at the end, different software and hardware proposed system implementation tools are also covered.

Chapter 4- covered all about the design and implementation of the proposed system, data processing architecture was also discussed.

Chapter 5- this chapter is about the implementation of the system which includes preprocessing step to model implementation with their sample code.

Chapter 6 – this is the last chapter of this report that summarizes the study by describing the experimental result and also this study concluded the study idea by forwarding additional feature works.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Information Retrieval

The base of information sharing is WWW which is emerged in 1994(Junwei, 2010) and has an enormous role in developing and spreading technology. Now a day's information being released and existing in different formats has been increasing exponentially and also the interest of users to access that information is also increased and started to worry about how to handle document overload rather than information shortage, due to this reason the need to store and access large information also increases from time to time (Singhal, n.d.) and become the reason for the evolution of the modern IR system. Information retrieval is the backbone for many search engine existence that plays a significant role in the development of technology throughout human history.

Information retrieval is the process of getting acquired information from different collections of resources. IR is concerned with how to denote, store, organize, and access information items such as text (numeric or data), multimedia (audio or video), web pages, and other related objects and make it easy for users to obtain acquired information. When a user forwards the request the IR system processes and translates it to a query.

2.2. Information Retrieval Process

Information retrieval is a system of searching certain specific documents from the world wide web and document corpus. IR system searches both unstructured and structured documents and ranks documents based on their usage to a user query. IR system basically must have at least three basic steps (process): indexing, query formation, and matching step.

2.2.1. Document Indexing

Indexing is a basic process in the concept of an IR system that performs core functionality in the retrieval process. An index is an offline process that arranges documents in the collection using keywords that do not involve the user intervention directly(Baeza-Yates & Riberio-Neto, 1999). It makes the searching process fast over a large collection of data as per user request. So it increases the efficiency of time for retrieval.

Indexing uses preprocessing techniques like stop word elimination, tokenization, normalization, and stemming to represent documents using keywords which are called index terms. According to Ounis, Iadh et. al(Ounis et al., 2006) indexing the text pass through four basic stages.

- ✓ Specify and arrange collection of documents
- ✓ Perform preprocessing technique
- ✓ Create a logical view by processing terms from document collection
- ✓ Building index data structure

There are different indexing structures are there for building index files such as suffix tree, inverted index, a sequential file, and suffix array. *Suffix tree and suffix array* – as their name indicates they use suffixes of a string for particular best performance of string operation. They are used to solve a string-related problem. *Sequential file* – has no linking pointer and vocabulary but it arranges records one after another(serially). It is easy to use but difficult in case a new record is added to update.

Inverted Index- is one of the popular indexing structures nowadays in which we adopted in this study. This method indexes a document depending on a sorted list of records that have a linkage with documents. Inverted files contain two basic elements: a list of terms(vocabulary) and the location of terms in a document. A list of a term(vocabulary) is used to speed up searching time by storing the terms that exist in the document in their lexicographical order and also for those terms that exist in the vocabulary there is a list of pointers for each document in which the term has existed. In short inverted index is a hashmap that links a term to its location in a document.

2.2.2.Query Formulation

After the document is indexed the next step is the query processing task in which the user request is changed into query language. Initially, the user specifies her/his request by using their natural language through the prepared user interface, then the system directly changes those requests to logical format without user intervention by applying different text operations. Text operation generates a logical view of both user needs and the original document, once text operation is applied system representation of users' needs is created which is called a *query*.

Queries are a collection of words that express users' needs, then this query is further processed to get a relevant document.

2.2.3. Query Matching Processing

Once the query is formulated the next step is the matching step that is carried out to retrieve the ordered relevant document. First, the query is formulated as discussed in section 2.2.2. is compared with the document representation in a database and ordered according to their relevance to the user need(detail process will be explained under another section). So to analyze and select the relevant document from the database different IR algorithms are existed that will be discussed in the next section, then each document's relevance is evaluated and ranked using those algorithms and finally, after the matching process is over then the retrieved documents are sent back to the user.

2.3. Information Retrieval (IR) Models

IR models are a technique of selecting and ranking documents that satisfy users' needs (relevant documents) in correlation with user requests. Mainly IR models support the following concepts: first representation of document and user request and compare those representations of both user query and document for document matching to retrieve ranked relevant document(Choudhary, 2012). Three classical models are there that depend on the statistical approach which is developed to retrieve documents(Manning, Raghavan, & Schutze, 2012)Boolean, probabilistic, and vector model.

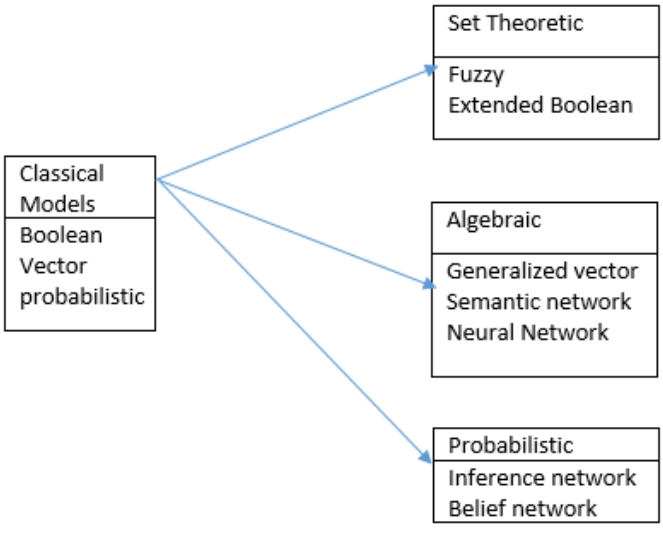


Figure 1: Major IR model classification

The above figure depicts the major IR model classification based on their mathematical basis. In the Boolean model, queries are expressed through a set of index terms therefore the model is expressed as a set-theoretic, whereas in vector space model document and queries are represented by vector t dimension and became a reason for vector space model is called algebraic and also probabilistic model using probabilistic theory for representing document and query. Therefore those models can be used as a blueprint for implementing actual information retrieval systems.

2.3.1. Boolean Model

Boolean model is the simplest and earliest model that depends on Boolean algebra to match user query to a document by matching words found in the query, those words are combined using a Boolean operator such as NOT, AND, and OR(Manning et al., 2012). This model uses binary criteria to determine the relevance of a document which means, a document can be either nonrelevant or relevant without having a relevance degree.

BM= 0, not relevant (term is not found in a document)

1, relevant (term is appeared in a document)

Boolean operators are used in this model to denote queries:

For example, for Q= {q1, q2, q3…….qn} (Queries)

D= {d1, d2, d3…dn} (Documents)

q1 AND q1 denotes both q1 and q2 appeared in document D, q1 OR q2 also denotes either q1 or q2 appeared in document whereas NOT q1 indicates not appeared in the document. This model retrieves exact matches that retrieve either too much or low, unlike best-match retrieval.

Even though the Boolean model is simple to use it has some drawbacks such as retrieving too few or many documents that do not match users need, document retrieved are not ranked, difficult for the user to construct Boolean queries because since user use natural language like AND, OR and NOT operator and those operators may denote different meaning according to the

context they are used in, so to overcome those listed problems and other, number of modification techniques were designed like Extended Boolean model, P- norm and Fuzzy set theory, however because of its less effectiveness other better models are designed.

2.3.2. Vector Space Model

Vector space model is one of the standard models in information retrieval systems that are designed because of drawbacks on the Boolean model and designed by having an idea of denoting documents and queries in terms of a vector of weight(Bakala, 2019). In vector space model vector is used to denote each document or item found in the database whereas a value given indicates how much the term is important by representing the semantic of the document. VSM uses term weighting to identify the similarity of a document with a query called tf*idf. So to calculate tf*idf it depends on two basic concepts term frequency and inverse document frequency.

Term Frequency – is described as the frequency of term t that exists in a document divided by a number of the term found in the document which means it answers how many times a term is used in the entire document. Therefore words that appeared in many documents have larger TFvalue; therefore mathematically it is described as:

$$\mathbf{TF=}\ \boldsymbol{df/D}\ \mathbf{,} \qquad \text{(Equation 1.0.1)}$$

Where D represents a number of the term in a document and d_f – frequency of words

Inverse Document Frequency - According to Salton (Salton & McGill, 1983) idf is used to measure how a word is common in the document to evaluate the importance of a word in the document which means it gives small and large weight for less and most common word existence respectively. So it is described as:-

$$\mathbf{TF\text{-}IDF = log}\ (|\boldsymbol{D}|)/(|\boldsymbol{d}|) \qquad \text{(Equation 1.0.2)}$$

It is the number of total documents over the number of documents that contain term t whereas D indicates the number of documents and d represents the number of the document in which the word exists., then cosine similarity (Munteanu, 2007) will be calculated. which means, the similarity between document D and Query q is calculated by using the cosine angle(cos) of two vectors which is described as the modular product of 2 vectors divided by the modular product of 2 vectors which is mathematically described as:

$$\textbf{Sim (D1,D2)} = \cos \omega = \frac{\sum_{k=1}^{n} wk(D1) * wk(D2)}{\sqrt{(\sum_{k=1}^{n} w2k(D1)) * (\sum_{k=1}^{n} w2k(D2))}} \quad \textbf{(Equation 1.0.3)}$$

Finally, documents with a high score will be selected and taken as relevant.

2.3.3. Probabilistic Model

The probabilistic model is one of the widely used and best performing models that is mainly designed to determine the degree of relevance between document and query better than the Boolean model and Vector space model. VSM does not consider semantic relation between document and query during the matching process is carried out that means, having document and query representation but IR system has uncertain guess whether that document contains relevant or non-relevant information concerning user request, so this situation leads to the existence of probabilistic model concept that provides an idea for such a kind of uncertainty(Kadhim, 2019) and became a common framework for modeling uncertainty nowadays. IR system uncertainty is basically about analyzing and understanding user queries and documents that satisfy user queries. This idea is depicted by the following figure shortly.

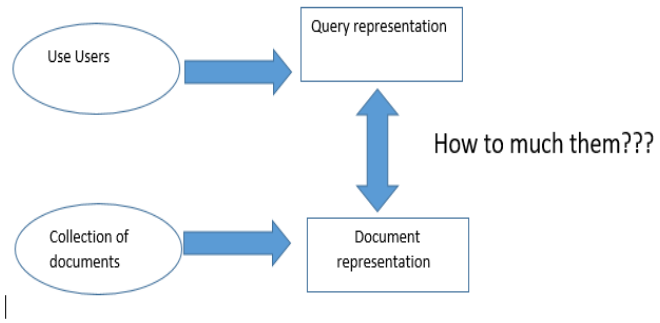


Figure 2: IR system uncertainity

Therefore probabilistic model considers information retrieval problem based on probabilistic theory to determine the probability that a document is relevant to a given user query, so the probabilistic model computes the probability of relevance in which document will get acceptance by the user concerning the user's need and expressed as P (rel|d) for each document and rank those document in their decreasing order.

Generally, probabilistic model ranking steps are:

- ✓ first, get the document collection

- ✓ then get a user request
- ✓ rank documents based on probability of relevance of the document concerning users need which means, P(R=1|d,q)
- ✓ Order the document in their decreasing order (put the best document first, second-best document second.

There are two major approaches are there in probabilistic IR (Fuhr, 1992), one is a classical approach which depends on the user relevance concept in which the user puts his/her judgments to document relevance about the query and the second approach was designed by Van Rijsbergen(Fuhr, 1992) by having an idea of generalizing the database system proof-theoretic model towards uncertainty assumption. Since our study was designed based on a classical approach, we focus on the detail of this model. Different variants of probabilistic approaches are there but the basic classical probabilistic models are Best Match (BM25)(Tamrakar & Vishwakarma, 2016), probability ranking principle(PRP) and Binary Independence Model (BIM), Bayesian Network.

Probability Ranking Principle (PRP): The probability ranking principle describes that the IR system will obtain the best efficiency (optimal retrieval) if and only if the ranking of documents is carried out in a decreasing manner to their probability of relevance that can be described as the following.

Let - d indicates documents

- R indicates relevance of certain document, while

  - ✓ R=1 indicates relevant document
  - ✓ R=0 indicates non relevant document

Therefore, the probability of retrieving relevant and non-relevant documents is computed as the following respectively: -

$$\mathbf{P(R=1/d)}=\frac{\boldsymbol{P(d|R=1)P(R=1)}}{\boldsymbol{P(d)}} \qquad \text{(Equation 1.0.4)}$$

$$\mathbf{P(R=0/d)} = \frac{\boldsymbol{P(d|R=0)P(R=0)}}{\boldsymbol{P(d)}} \qquad \text{(Equation 1.0.5)}$$

And the sum of both equations become 1 which means, P(R=1/d) + P(R=0/d)=1

Binary Independence Model (BIM): The idea of the binary independence model first emerged in 1967(Baeza-Yates & Riberio-Neto, 1999) which is based on probabilistic information retrieval by having an assumption of taking a document as binary vectors. The word binary indicates that queries and documents are represented as the binary which is equivalent to a Boolean representation of terms in vector form which means. each term found in document or queries are represented by one Boolean element, as a result of this assumption, many documents can have the same vector representation. For example for document d= {d1, d2, d3…dn} q1=1 if the term exist in a document and q1=0 if it is not exist in a document. The word independence indicates that terms exist in a document independently, which means there is no association between terms under consideration, and also the probability of word existence in the relevant document does not affect the probability of another word's existence in the relevant document, therefore BIM records presence or absence of words (terms) in documents which are represented as d=1 or d=0 respectively. BIM describes document relevance in terms of the probabilistic model as described in the following:-

$$P(R|d,q) = \frac{\mathbf{P(d,q|R)P(R)}}{\boldsymbol{P(d,q)}} \qquad \text{(Equation 1.0.6)}$$

$$P(T|d,q) = \frac{\boldsymbol{P(d|R=0)P(R=0)}}{\boldsymbol{P(d)}} \qquad \text{(Equation 1.0.7)}$$

Where – R indicates relevant documents

- T indicates non-relevant documents
- d and q indicate document and query respectively.

According to Solomon Regassa's experiment (“SOLOMON REGASSA GUDA July, 2016 Adama, Ethiopia,” 2016), BIM does not give us a better result that has good performance since the BIM assumption is far from the correct ideology as described above.

BestMatch25 (BM25, Okapi): BM25 is one of the robust and widely used non-binary probabilistic approach models. It was first designed in 1994 by Stephen and Karen Sparck(Jones, 1979)in the context of the OOKAPI system that was implemented in the 1980 and 1990s and later started to be adopted by other systems. BM25 is one o the ranking function that is used to rank document that exists in a collection based on user query by computing the probability of a term in a document, so it correlated document length and term frequency.

Currently BM25 model is popular because of its efficiency and it performs well in many retrieval tasks(Svore & Burges, 2009) and it depends on two big components TF and IDF:-

- ✓ TF (Term Frequency)- indicates the number of a term in a document.
- ✓ IDF (inverse document frequency) – indicates the number of the document that includes the term.

So BM25 operates based on the above two components. So general equation of BM25 is

Score (q,d) = $\sum_i^n W.R(q,d)$ whereas – W- represents a weight

- R (q, d)- relevance of a document to a given query

but since the weight(W) of the relevance of a word is first must be calculated, it uses IDF.

$$IDF(q) = \log\frac{\boldsymbol{N-n(q)+0.5}}{\boldsymbol{n(q)+0.5}} \quad \text{(Equation 1.0.8)}$$

Therefore, the correlation score between a document d and query q formula for BM25 is summarized as:

$$Score(Q,d) = \sum_{\mathbf{i}}^{\mathbf{n}} \mathbf{IDF(q)}.\frac{\mathbf{fi.(k+1)}}{\mathbf{fi+k.(1-b+b.\frac{dl}{avgdl})}} \quad \text{(Equation 1.0.9)}$$

Where both parameters (b and k1) are adjustment factors, f*i* represents of query that exists in document d, d*i*- length of document or number of words in a document that is identified during we tokenize a document then we can count the number of words, avgdl- the average length of the whole document.

A tuning parameter in BM25:b and k are two parameters in BM25 which are used to adjust the relevance score, b parameter controls the amount of length normalization which ranges from 0 to 1 value, where b=0 indicates no document length and b=1 means full normalization exist in full effect, so it controls how much we make document length fair, as many experiments indicate the optimal saturation value exists between 0.3 to 0.9 and mostly works well for many corpora when b=0.7, where as k parameter indicates saturation level which controls term frequency and ranges from 0 to 3. According to many experiments, the value between 0.5 to 2.0 can be taken as optimal saturation level and mostly works well at k=1.5.

2.4. Elastic Search Engine

Elastic search is one open-source engine that helps us to store, analyze and search a large volume of data within a small-time (quickly). It depends on Apache Lucene(Trotman, Clarke, & Culpepper, 2012) full-text search engine which is developed in java. Nowadays to access any information we should look for many huge documents that existed either online or offline, so as

the data volume increases it became time-consuming and complex. An elastic search is a tool that is designed for this kind of big data search. Additionally, elastic search allows us to consolidate structured search, full-text search that correlates with their location existence.

Elastic search as the following backend components: -

- ✓ **Node –** It is a single server that is used to store our data and perform cluster indexing and search capability. It is a running instance of Elastic search(Thacker, Pandey, & Rautaray, 2016).
- ✓ **Cluster-** it is a combination of one or more nodes that are used to facilitate indexing and searching capabilities.
- ✓ **Index –** index in elastic search is the combination of similar documents that have similar characteristics and they have unique Id which plays a great role during searching, deleting, and storing operations. However elastic search uses an inverted index that indexes all unique terms that exist in a document, so it allows very fast data retrieval as well as data insertion.
- ✓ **Document –** It is a basic component in elastic search that can be indexed. This document operates through JSON[6].

So elastic search works on the concept of the above components to fetch and store a large volume of data. Including elastic search, the backbone of every search engine is relevancy string that is why elastic search use TFIDF before version 5.0 and BM25 later version, because BM25 was designed to overcome the shortcoming of TFIDF such as lack considering document length and saturation. Elastic search has the following features(Zheng, Yicheng1. Zheng Y, Deng F, Zhu Q, Deng, Zhu, & Deng, 2014).

Distributed – it indicates that an elastic search server can start with a single node and increased horizontally depending on the requirement needed.

High availability – elastic search makes sure that data is safe and accessible for information retrieval.

Full-text search – indicates the capability of elastic search that full query-based search

Document oriented– elastic search data is found in JSON format

[6]JSON – it is a text based representation of structured data that is mainly used for sharing data in web application by storing the data in the form of array in order to make data transfer easy.

2.4.1. Indexing in Elastic Search

Indexing is a basic small unit of elastic search. As mentioned above elastic search uses one of the indexing types called an *inverted index*, which divides the index into multiple small pieces called shards, which is fully functional and can independently be hosted on any node in the cluster. Depending on the Lucene version used the maximum capability of a number of the document is fixed (Thacker et al., 2016). To create elastic search indexing, the number of shards is mentioned first. These shards are mainly used to divide the data horizontally to increase performance. Elastic search inverted index contains field name with its corresponding value and also a pointer to the document and also it contains information which is used for computing relevance such as a number of the document in which a term appears and a number of the term exist in a document. Therefore, when we perform a search we are not searching documents themselves rather inverted indexes.

The general workflow of elastic search is depicted as the following: -

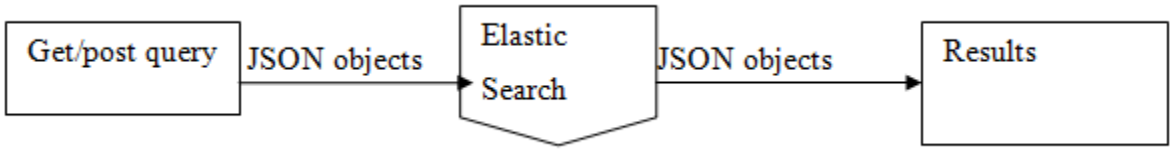


Figure 2.3: Elastic search workflow

Why does Elastic search use BM25 instead of TFIDF?

Today's scoring algorithm for Elastic search especially after Lucene 6 is BM25, because of TFIDF's shortcoming and to increase the performance. It began to use one probabilistic ranking model called BM25 because TFIDF does not consider document length into account and also lacks term frequency saturation (which controls the effect of the single query term in a document). So to control the above-mentioned problem additional parameter (b and k) is included in BM25 as described in the above section. Therefore, BM25 performs better than TFIDF that is why Elastic search uses BM25.

2.5. Text Processing in IRs

2.5.1. Tokenization

Tokenization is one preliminary step in the natural language processing task which is concerned with a splitting or breakdown sequence of text into semantically meaning full small units called tokens and further other issues are also included such as identification of date, accent, and abbreviation. Tokenization also can interpret the meaning of words by analyzing those words that exist in the text given, therefore tokenization is useful both in computer science as well as in linguistics, where it forms part of lexical analysis.

For example: for the sentence “Meetii fi Meerobobbolaadha.” (Meti and Merob are sisters). The tokens will be “Meetii”, “fi”, “Meerob”, “obbolaadha”.

2.5.2. Normalization

Text normalization is the process of transforming a text into a standard (normal) form. It includes removal of punctuation and multiple spaces to just one space between each word uppercase to lowercase letters, etc. For example, the word “gooood” and “gud” can be transformed to “good.

2.5.3. Stop word Removal

Stop words are common words that existed in the document that has no value and have less important meaning than keywords and thus are excluded from the dictionary or index to save memory,

Stop word removal is the process of removing (omitting) the most common and highly frequent words found in the document**.** As a preprocessing task, it must be removed to ease further tasks and speed up the core task in text processing. Sometimes removing stop words may change semantic meanings of the word that may decrease recall.

2.5.4. Stemming

A stemming is a process that reduces all words with the same stem to a common form, usually by stripping each word of its derivational and inflectional suffixes [15]. The main purpose of a steaming process is to reduce the inflectional or derivational word into its root form. But it may cause difficult to choose a good algorithm that is used to remove affixes since every language has its own unique rule which is also applied for Afan Oromo. Different stemmers are available for stemming namely Porter's Stemmer, Snowball Stemmer, etc.

2.6. Word Ambiguity

Words that have more than one sense (meaning) can lead to ambiguity. Ambiguity is the ability of one word to have more than one meaning. All-natural language including *Afaan Oromoo*is

ambiguous because these machines cannot understand the way human understands. Therefore this can be a reason for poor performance in IR system unless it is disambiguated(K & Anto, 2007)

There are different ambiguity occurrences levels are there in NLP such as Lexical, Syntactic, Semantic, Pragmatic, etc.

Lexical Ambiguity-The ambiguity of a single word is called lexical ambiguity. For example,

a. Ayyaantuun kitaaba ishee bade **soquu** deemte (Ayyaantuun went to find her lost book).

b. Meerob lafa irraa marga **soquu** deemte (Meerob went to scour grass from the land).

The occurrences of the word "**soquu**" in the two sentences denote different meanings contextually that as "looking carefully to find something lost" and "cleaning or removing something unwanted" respectively.

Syntactic Ambiguity-This ambiguity can occur when one sentence is interpreted in different ways.

Semantic Ambiguity occurred - This kind of ambiguity occurred because of incorrect interpretation of the meaning of words in a sentence, which means giving unrelated definitions.

Anaphoric ambiguity- This kind of ambiguity arises due to the use of anaphora entities in discourse.

Pragmatic ambiguity- Such kind of ambiguity refers to the situation where the context of a phrase gives it multiple interpretations.

2.6.1. Semantic ambiguity

Semantic ambiguity has occurred when one word, phrase, and a statement is understood and interpreted in more than one way(Vasiloaia, 2020). This problem is a key problem in most natural languages including *Afaan Oromoo* in which this system is designed to resolve this issue.

For example, for the following sentence " Ittoonisheehar'abadaamiti", the word "badaa" represents two different ideas which are "small" represents quantity while the second meaning "does not taste good" represents quality. Therefore, since the word "badaa" lacks detail they are subject to multiple interpretations which is a reason for word ambiguity.

2.6.2. Measures of word semantic similarity

Semantic similarity measurements of words are measurements performed based on the IS-A relationship defined between synsets found in WordNet. To say two words are similar when some sense of one word exists in another word, they are also called synonymous words. therefore words that are semantically related are extracted from a list of similar synsets(Banerjee & Pedersen, 2003), then their frequency of those words is obtained from the index file.

For example, in *Afaan Oromoo* the word "hori" has a different meaning in the following two contexts.

- "hori"- loon(cattle)
- "hori"- qabeenya (Wealth)

The algorithm bases its disambiguation decision on the semantic similarity, therefore semantic similarity between words and sentences is a very important issue in the NLP field because it is used to assess the level of relatedness between them based on their meanings. WordNet is one lexical database that is constructed manually to measure semantic similarity between two words or sentences.

2.7. Word Sense Disambiguation

WSD is a technique of identifying the meaning of a sense in a given context. Starting from the 1960's it has been a concern and an interest(Arukgoda, Bandara, Bashani, Gamage, & Wimalasuriya, 2015) until now. In all-natural language solving words, ambiguity is one of the biggest challenging tasks that can be taken as a reason for retrieving unnecessary documents. so to solve unless it is disambiguated. For example in the sentence "I have a pen", the meaning of the word "pen" must be disambiguated, because the meaning of pen can be either "female swan" or "material used for writing purpose which is made from different color inks" therefore to solve the problem each sense of term found in a document should be separated and a word must be associated with correct senses.

According to Aguirre and Edmonds basically, the most known word sense disambiguations are machine learning that includes supervised corpus-based and unsupervised corpus-based.

Supervised approach- this model is used to disambiguate and it depends on sense annotated corpora since machines need some additional protocol in memory to decide ambiguous situations

(Kokane & Babar, 2019). So some learning set is prepared to predict word senses which means supervised approach response if and if only information has existed in the predefined database.

Unsupervised approach- this approach is used to improve the efficiency of than supervised one by taking an online dictionary as a learning set, unlike the supervised one. It depends on unnoted corpora, this implies that the sense of words is extracted from the input text and the occurrence of words are clustered together.

The other word sense disambiguation approach is the knowledge-based approach in which we for designing our system because of its high better performance than the other machine learning approaches.

2.7.1. Knowledge-based approaches of word sense disambiguation

Knowledge-based methods use machine-readable external resources to get word senses in the context such as a dictionary, thesaurus, etc. As its name indicates in this approach knowledge of words is a basic ingredient to identify the sense of the word and critical issue for further tasks (Sharma & Joshi, 2019) so words that share common meaning (sense) are selected according to this approach.

The knowledge-based approach uses the following technique, first overlap of sense definition, restriction regarding selection, and structural approaches. So to perform the following technique, knowledge-based use a different algorithm for similarity measures, such as Walker's algorithm, random walk algorithm, and Lesk algorithm.

2.7.1.1. Walker's algorithm

In 1987 a person whose name is called Walker proposed this algorithm for word sense disambiguation. Since this algorithm is one of the knowledge-based approaches it uses thesaurus as the knowledge resource. The first synonym of the target word is identified and then by applying words of the context compute the result for every sense(Aliwy & Taher, 2019).

2.7.1.2. Random Walk Algorithm

Most of the time many words that have different senses can exist in a sentence, so to disambiguate those ambiguous word random walk algorithm create a vertex for every sense of the word, then each score of the vector is calculated through a graph-based ranking algorithm and larger vertex result is taken as appropriate sense.

2.7.1.3. Lesk Algorithm

This algorithm is introduced in 1980 and currently, it is one of the famous unsupervised word sense disambiguation algorithms used(Arukgoda et al., 2015). In this algorithm the meaning of the target word is compared with the glosses of every word in the document, this implies that the meaning of words(gloss) that share the largest common number of words is selected as the correct senses, therefore the idea behind this algorithm is comparing the meaning of ambiguous word found in dictionary with the terms exist with its neighbors by counting common words exist in dictionary and neighbors of that words, then the sense with the largest number will be chosen.

So for n number of senses of the target word, n number of comparisons will be carried out(Ayetiran & Agbele, 2018)(Arukgoda et al., 2015). Therefore, the Lesk algorithm uses the following approaches: -

- ✓ Identify the gloss of senses of the target word
- ✓ Compare the gloss of each meaning of the target word with other meanings of every word that exists in a document.
- ✓ Count number of commonly shared words(overlap)
- ✓ Finally, a high number of overlaps will be selected as the appropriate sense.

2.8. Query Expansion

The idea of query expansion was first implied in 1960 by Moron et al. It is a technique used to enhance the effectiveness of information retrieval by reformulating and expanding users' original queries. So this technique improves the chance of retrieving not retrieved relevant documents by using the original queries. So to expand the original query it extracts terms from the retrieved document initially(Yue, Chen, Lu, Lin, & Liu, 2005).

Query expansion technique is used to resolve the problem that occurred due to words ambiguity that occurred in many natural languages. So as many researchers indicate adding extra terms to the original query increase precision and recall because the newly added query contains more terms so the chance of matching the new terms in a relevant document also increases. Therefore this technique is mainly designed to improve search results. There are two important strategies

are used in the query expansion technique, those are global analysis and local analysis (Imran & Sharan, 2009).

In the global analysis technique, new words are added before searching to a user query, so to perform this process global analysis needs external resources such as treasure, WordNet, etc. While in local analysis new query is created based on some retrieved document, so unlike global analysis, local analysis need semantic vocabulary which is created based on assumption that different word form can have a similar meaning. Therefore, in this method, the first indexing document is performed and the user gives the query by expanding the term automatically from this semantic vocabulary then these expanded queries are searched.

2.8.1. Different Query Expansion Techniques

Currently, different approaches are there that is used during designing query expansion.

2.8.1.1. External resource-based

In these approaches, the query is expanded using some external resource like WordNet, lexical dictionaries, or thesaurus. Most of the time those resources are built manually due to a lack of well-crafted and prepared resources before that maps terms to their relevant sense. This technique involves looking up such resources and adding the related terms to the query. Some of the examples used under external resources are:-

WordNet

WordNet is one of the lexical databases that was first developed by Princeton for the English language (Banerjee & Pedersen, 2003) and it became one part of NLTK later. Therefore English WordNet groups different parts of speech like a verb, adjective, noun, and adverb into a set of synonym words that describe a distinct idea. However currently besides English WordNet different WordNet are prepared for many languages based on Princeton WordNet. So WordNet is also a dictionary that group synonym words together called synsets with their short definition called gloss and describe relationships among these synonyms. Those relations are defined explicitly in WordNet that is used to link synsets to each other, such as hypernym, meronym, holonym, hyponym,(Banerjee & Pedersen, 2003).

- ✓ Hypernym - relate more general synsets to a specific one, animal --→ dog, means an animal is the general name of the whole animal, while a dog is a specific kind of animal.
- ✓ Meronym- describe part, leg---→ human body, means that leg is one part of our body

- ✓ Holonym- is the inverse of meronym that describes the whole which means from the above example human body is a holonym of the leg.
- ✓ Hyponym- this relation is the inverse of hypernym that describes a specific thing, so the dog is a hyponym of the animal.

To summarize the relation between them, the following taxonomy is where every node is a set of other nodes:

- ✓ Hypernym ⇨ hyponym (Has-a/ Is-a)
- ✓ Holonym ⇨ meronym(Has-part/ part- of)
- ✓ Meronym ⇨ holonym (member-of/has-member) or
- ✓ Meronym ⇨ holonym(substance of/ has-substance)

There are also many other relations are there in WordNet as explained by Banerjee and Pederson those mentioned above are most frequently used and have a great impact on the relation. According to Diparsee(Pal, Mitra, & Datta, 2014), WordNet is used for both query expansion and disambiguation purposes.

Thesaurus

Thesaurus is a data structure that provides a limited set of semantic relations between different concepts. Words are listed based on similarity of their meaning, unlike dictionaries that list words alphabetically by providing words definition. So thesaurus supports both automatic indexing and retrieval, Because of this, it is used as one technique to expand the query.

2.8.1.2. Relevance feedback query expansion

Relevance feedback is a feature that exists in some IR systems in which users give feedback regarding the IR system and is used to evaluate relevancy for performing and executing another new query. Therefore, relevant feedback defines how much the retrieved document meets users' needs.

In addition, relevance feedback is a mechanism used to modify original queries from a previously top-ranked retrieved document in which the user identified as a relevant document, therefore relevance feedback plays a great role for the retrieval accuracy(Hamid, 2017). In the relevance feedback approach first user sends feedback and presents the user request to the IR system. IR system uses this information in two ways which is the quantitative approach which

retrieves more relevant documents and the qualitative approach that retrieve similar relevant documents. Mainly there are two major relevant feedbacks are there, implicit feedback(IF) and explicit feedback (EF)(White, Jose, & Ruthven, 2002).

Implicit feedback (IF): As its name indicates in this type of relevant feedback users feedback is taken from user actions, which means in this method users feedback is not indicated by users behavior because it is taken implicitly from users actions such as the time user spent on document viewing, searching actions and other related actions are used.

Explicit feedback (EF): Users' feedback is described explicitly by a query, so most explicit feedback had an interface that contains checkboxes that help users to mark document retrieval explicitly. In this type of feedback, two ways are there which are used to indicate document relevance; those are binary and graded relevance feedback. In binary relevance feedback document relevance is indicated through binary feedback, while in graded relevance feedback document relevance is described by giving some description about relevance.

2.9. *Afaan Oromoo* Language

Afaan Oromoo is also called Oromiffaa is a member of the Afro-Asiatic family that is categorized under the Cushitic language. It is spoken by more than 30 million people living all over the world especially those people living in Somalia, Kenya, Ethiopia, and Egypt and it is the third most widely spoken language next to Hausa and Kiswahali (Ibrahim Bedane, 2015). Currently, many governmental and private colleges and universities give training in *Afaan Oromoo* as a major field of study.

2.9.1. Overview of Afan Oromo Script[7]

Afan Oromo uses Latin based alphabet called *Qubee* which consists of 33 letters. The exact time when *Afaan Oromoo* started to use the Latin alphabet is not known, but it is adopted on November 3, 1991, until 1991 it was written by Ge'ez script and it became a language of media, religion, and education nowadays.

Afaan Oromoo uses the English alphabet even though it sounds different. There are 31 letters are there that include five vowels, twenty-four consonants, and out of which seven are paired letters

[7]https://www.researchgate.net/publication/333396759_investigating-afan-oromo-language-structure-and-developing-effective-file-editing-tool-as-plugin-into-ms-word-to-support-text-entr

such as CH', 'DH', 'NY', 'SH', 'TS'. All upper and lowercase alphabets including their pronounced sounds are available in Appendix- I.

Vowels (Dubbachistu) and Consonants (Dubbifamaa)

Similar to the English language *Afaan Oromoo* vowels use similar letters even though it sounds different, those are A, E, I, O, and U and it is pronounced similarly throughout *Afaan Oromoo* literature. For example,

- ✓ A- Abdii, Ayyaantu
- ✓ E- Eegan, Eebla
- ✓ I- Ililli, ilma
- ✓ O- Okkote, obbolessa
- ✓ U- Urji, uffata

Afaan Oromoo has 24 consonant phonemes such as B, F, G, and other letters that sounds make a difference in word meaning. Like other natural languages, *Afaan Oromoo* has its structure and rules which means *Afaan Oromoo* has its format to describe gender, adjective, adverb, and other necessary structures.

2.10. Related Works

There is a lot of work that has been done by many global and local scholars related to information retrieval for different languages by using different IR models.

2.10.1. Global researcher

Imran (Imran & Sharan, 2009)proposed how biomedical literature search queries improve performance by expanding queries. He used an enhanced BM25 approach to retrieve the most relevant literature from clustered biomedical literature with expanding queries. So he proved that his retrieval performance is improved using query expansion.

Khan J.A(Khan, 2014)developed an experiment to compare the three known IR models(binary independence, probabilistic and Boolean model) to analyze the performance of those models by evaluating the recall and precision of those models respectively. So his experiment of those model comparisons was based on certain significant parameters and finally, he put the experimental results. For example, since IR effectiveness is mostly measured by recall and

precision the result of these two parameters indicates that the probabilistic model has better recall result when compared to other models which means the probabilistic model is better in most aspects of the parameter, and at the end, he concluded that better result improves search result.

Khaled, Sinan, Farhad, et al. (Shaalan, Al-Sheikh, & Oroumchian, 2012)worked on query expansion on Arabic Information Retrieval using Expectation-Maximization (EM). They tested their algorithm on INFILE test collection of CLLEF2009, and the experiments show that query expansion that considers the similarity of terms both improves precision and retrieves more relevant documents. The main finding of their research is that; increase the recall while keeping the precision at the same level by this method.

Rivas *et al*.(Rivas, Iglesias, & Borrajo, 2014)developed and evaluated preprocessing and query expansion techniques for retrieving documents in several fields of biomedical articles belonging to the corpus Cystic Fibrosis, a corpus of MEDLINE documents. The Studies were carried out to compare the weighting algorithms Okapi BM25 and TF-IDF available in the Lemur tool, concluding that TF-IDF with TF formula given by BM25 approximation is superior in its results.

2.10.2 Local researcher

Several IR systems developed so far for retrieving *Afaan Oromoo* texts **Berhanu A.**(Anbase, 2019)Worked on how to retrieve *Afaan Oromoo* language based on semantic-based indexing. He use vector space to represent documents and queries and he used 70 *Afaan Oromoo* corpus and 9 queries. Finally, he got a performance result of 67 %(0.67) precision and 63%(0.63) recall, but he didn't use stemming that affected retrieval result.

Solomon R.("SOLOMON REGASSA GUDA July, 2016 Adama, Ethiopia," 2016)proposed *Afaan Oromoo* information retrieval system by using one probabilistic model which is called BIM, so he calculated relevant documents based on the number of queries that occurred in the document. Unlike VSM this method cannot count term frequency to determine document relevance rather it determines based on the existence of the term in a corpus which means if the term exists in a document the model judges the relevance of the document and the irrelevant if the term does not exist. But like Berhanu, Solomon didn't include a stemming algorithm in his experiment even though his experiment shows a better result when compared to Berhanu(Anbase, 2019).

Gutema G.(Eggi, 2012) makes possible retrieval of Afan Oromo text documents by applying techniques of modern information retrieval system. He uses Vector Space Model and his experiment shows that the performance is on average 0.575(57.5%) precision and 0.6264(62.64%) recall. And he recommended that the performance of the system can be increased if the stemming algorithm is improved, and thesaurus is used to handle polysemy and synonymy words in the language.

However, most information retrieval system works have been done by using different Ethiopian languages such as Amharic, Silte, Tigrigna, and others, even though most of them are based on the Amharic language.

Amanuel H.(Choudhary, 2012)worked on the Amharic IR system based on the probabilistic model and he used 300 documents and 10 queries were selected to test system performance. Amanuel also used the BIM model for the relevance judgment, so by using this model he tested this system two times to compare and identify better results which are before and after user relevance feedback. So he got precision of 42% and recall 88% on average before user feedback and he proved this result was increased by 52% precision but recall is decreased by 20% and finally, he recommends synonym and polysemy words should be handled for the future `work.

Abdulmalik J.(Johar, 2020) proposed an IR system for Silte language based on the probabilistic model by using the BM25 model. The system he proposed also includes both searching and indexing parts and finally he registered 88% mean average precision. Therefore, the following table summarizes the above-related works mentioned.

Isayas W.(Wakgari, Advisor, & Assefa, 2016)worked on "*Afaan Oromoo* Text Retrieval System using Probabilistic approach" and he implemented the system by using Java language and better number of retrieved relevant documents are registered when we compared with the research done before by previous *Afaan Oromoo* scholars, however, he didn't use any algorithm for adding synonyms that decreases his system performance.

Table 1: Summary of related works

No	*Title*	*Author*	*Objective*	*Gap*	*Algorithm*
1.	Query Expansion Based-on Similarity of Terms For Improving Arabic Information Retrieval	Khaled Shaalan1, Sinan Al-Sheikh, and FarhadOroumchian	Suggests method for query expansion on Arabic Information Retrieval	Precision is not increased.	Expectation Maximization (EM)
2	Application of information retrieval for *Afaan Oromoo* text based on semantic-based indexing	Birhanu A.	He worked on how to retrieve the *Afaan Oromoo* document semantically	He didn't use stemming	VSM
3	Probabilistic *Afaan Oromoo* IR system	Solomon R.	He proposed how to retrieve *Afaan Oromoo*document using BIM	No stemming algorithm is used and less number of relevant document is retrieved	Probabilistic Model
4	Probabilistic IR system for the Amharic language	Amanuel H.	Retrieving Amharic text using probabilistic model	Synonym and polysemous words are not handled	Probabilistic model
5	Information retrieval system for Silte language using BM25 weighting	Abdulmalik J.	Worked on how to retrieve documents written in the Silte language.	Because of word ambiguity, some words are not retrieved	Probabilistic model
6	*Afaan Oromoo* Text Retrieval System using Probabilistic approach	Isayas Wakgari	Retrieving relevant *Afaan Oromoo* documents	Manual use of adding extra words	Probabilistic model

In short, we tried to review different articles done by both local and global researchers regarding IR systems based on different language structures and we tried to identify major gaps seen in the Afaan Oromoo retrieval system those works to solve those problems and many of them were focused on the VSM which is an old and low performing model since it doesn't have a method to control uncertainty in information retrieval and also they didn't work on how to control word ambiguity existed in the document, that can cause a decrease in the performance of the system and doesn't fit user interest, therefore, to minimize such problems, another technique is required which is used to satisfy users' needs by adding other descriptive terms

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Overview of research design

To conduct the *Afaan Oromoo* IR system we passed through different steps. Those steps are crucial for the final implementation of this study.

- ✓ The earliest step is searching different sources to identify the research area.
- ✓ Then based on the selected research area, the next step will be searching and analyzing different papers and books to identify the research gap that is mentioned as the title of the study.
- ✓ After the research gap is identified different research questions are formulated which are answered under this study.
- ✓ The next step is collecting and organizing corpora[8] from different sources which are the backbone for developing this system.
- ✓ After that general design of the system that will answer the research question is prepared.
- ✓ Then the process of implementing the design is started so that it is possible to solve the research gap identified.
- ✓ After the implementation step is finished different evaluation techniques are carried out so as to check and evaluate our system performance.
- ✓ Finally, the result of this whole step is written and explained.

  So the above steps are explained shortly as the following diagram

[8] Corpuses are collection of documents in which different words are found

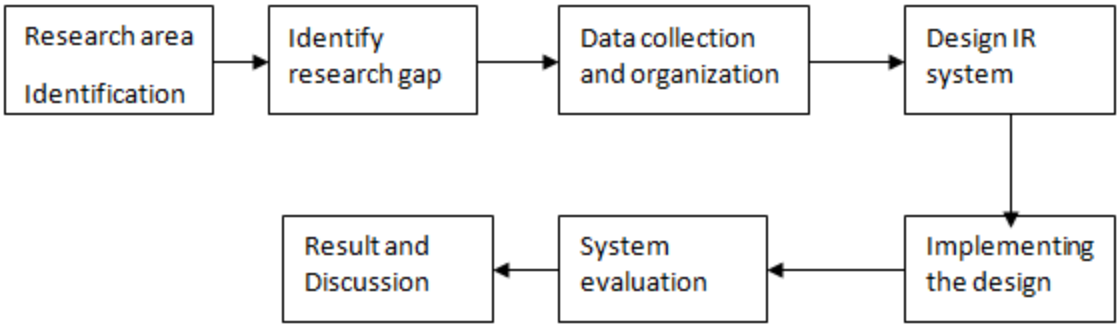


Figure 3.1: Overview of a research design diagram

3.2. Data Collection

Preparing a standard and large size corpus for the *Afaan Oromoo* language is one of the challenges faced in this research. The state of the art in the area of text processing indicates that there is no developed standard corpus for the *Afaan Oromoo* language and also it takes a lot of time to prepare large size of the corpus with many documents. So the researcher uses a small size corpus (100 corpora).

The data collected for this study was both primary data which is carried out by interviewing some linguistic experts in WordNet preparation and also secondary data from different social media, books, magazine and also most of the data were collected from the Oromo Cultural Center were stored at different times. The collected data is depending on different aspects so that it is possible to address political, cultural, religious, and other related ideas.

So we gathered and prepared more than 100 corpora and categorized them according to the following target groups and metrics so that it is possible to include various aspects of information so that it is possible to address users' needs.

Table 2:Sampled Corpuses

	Types of News	***Number of corpora***
1.	Health	15
2.	Cultural	10
3.	Sports	18
4.	Religious	20
5.	Politics	12
6.	Social	12
7.	Economic	13
	Total	100

3.3. Corpus Preparation

Even though *Afaan Oromoo* is an official language and many resources is existed in paper and digital form, but still, there is a lack of resources for the researcher. Since the researcher is intended to do an *Afaan Oromoo* information retrieval system it must include different aspects of information, i.e. sport, politics, culture, religion, and others which are existed in *Afaan Oromoo*. So building new data set is needed since there is no published dataset for this purpose yet.

The process of constructing corpus for *Afaan Oromoo* information retrieval includes the following steps: -

- ✓ Gathering *Afaan Oromoo* resources from different sources like websites, books, magazines, newspapers, and other sources.
- ✓ Preparing, filtering, or consolidating gathered data into one file dataset.
- ✓ Identify the query needed to test the experiment.

After the data was collected a preparation step follows, which are filtering, cleaning, and consolidating data into one file which may have a great impact on the quality of data collected. So the following tasks were performed to prepare the corpus.

- ✓ Removing all non-*Afaan Oromoo* data
- ✓ Removing all null, blank values, and whitespace
- ✓ Removing duplication to ensure the uniqueness of each text in the dataset
- ✓ Filtering and cleaning poorly formatted entries.

3.4. Data Source

To develop query expansion for Afan Oromo information retrieval, a standard and representative document corpus was selected from different sources that express different aspects, and also test data which is used to evaluate the system is selected appropriately. Most of the data were gathered from the Oromo cultural center which is found in Addis Ababa, Kirkos sub-city and it was first recognized to represent the vast region of Oromia in a single cultural complex that depicts the culture, tradition custom, art, history, and dress of Oromo people. Different media, magazines, and books were also used as a source for data collection additionally.

Additionally, to test the experiment and to evaluate the performance of the system ten (10) queries are identified that are selected depending on the content of a file in the corpus.

3.5. Information retrieval Modeling

3.5.1. Data Preprocessing

Data Preprocessing is one task of NLP which is used to make our text *predictable* and *examined* for our task. After data is collected it passes through different procedures to index and use the retrieval system. So our system uses a data mining text preprocessing technique which is used to make the dataset ready for the next process.

3.5.1.1. Tokenization

Tokenization is the first step of preprocessing techniques in NLP and it is the process of dividing a stream of text into a unit called tokens. A machine cannot understand the raw text as it is, so we

need to split our text. We applied a tokenization algorithm in *Afaan Oromoo* text depending on the space or punctuation marks that existed in the text. For example

Raw text **: "***Ati akka namicha sooressa sanaa ta'uu hin barbaaddu mitii?"*

Tokenized text: [‘ati’, ‘akka’, ‘namicha’, ‘sooressa’, ‘sanaa’, ‘ta’uu’, ‘hin’, ‘barbaaddu’, ‘mitii’, ‘?’]

INPUT: - dataset

OUTPUT: - token

get data

define tokenized function

split text to words

return tokenized words

end

Algorithm 1: Tokenization Algorithm

3.5.1.2. Text Normalization

Text normalization is the process of transforming a text into a standard (normal) form. It includes removal of punctuation and multiple spaces to just one space between each word uppercase to lowercase letters, etc.

3.5.1.3. Stop word Removal

It is the process of omitting the most common and highly frequent words found in the document. Sometimes removing stop words may change semantic meanings of the word that may decrease recall. As Greengrass(“Information Retrieval : A Survey by Ed Greengrass,” 2000) described many terms occurred with low frequency, medium number of the term occurs with medium frequency, and also few terms occurred most frequently. So the following figure indicates the stop word detector and removal algorithm we used to remove the most frequent words.

```
INPUT: - token
OUTPUT: - stop word removed text
get tokenized dataset
open stop word list
for each word in tokens
        IF a word is in {stop word} list then
         remove word
     ELSE
        Add to non- stop word list
end
```

Algorithm 2: Stop word removal algorithm

3.5.1.4. Stemming

It is a process that reduces all words to root form by reducing inflectional and derivational words. As Nega(Alemayehu, 2003) explained context-free stemmer is a type of stemmer in which words are stemmed through removing words that are the same with suffix and prefix without consideration of other remaining stems. Algorithm 3.3. depicts stemming algorithm according to Debela and Ermias (Gemechu & Abebe, 2010)that design stemming algorithm based on Porter Stemmer rule that we applied for this study.

```
INPUT: - un stemmed tokens
OUTPUT:-stemmed tokens
get tokenized data set
read word to be stemmed
        IF word match rules
                remove suffix
                return suffix
        ELSE  leave as it is
end
```

Algorithm 3: Algorithm for Stemming

3.6. Model Election Technique

Many models have been designed and developed in information Retrieval. Those are a statistical model in which both probabilistic and vector space model is included, Boolean model, knowledge-based and linguistic model.

To select a model for this work we compare each model based on the number of the relevant document retrieved. So we select the probabilistic model because it gives a high and better number of relevant documents than the vector space model and the rest.

3.6.1 Probabilistic model

This model order and rank document according to decreasing their relevance to a given query and also it depends on probability theory, so it is well understood and simple to extend. There are three probabilistic retrieval models, those are Probability Ranking Principle (PRP), Binary Independence Model (BIM), BestMatch25 (BM25, Okapi). BIM as its name indicates the word binary shows that this model represents both document and query as binary term incidence vector whereas the word independence indicates that there Is no association between each term.

3.6.2 Boolean Model

it is considered as the weakest classic model since partial matching (no term weighting) is not provided in this model and also there is no ordering of documents since there is no counting of

term frequency, so as a consequence this model most of the time returns either too many or too fewer documents in response to a user query.

3.6.3 Vector Space Model

Vector space model depends on linear algebra which is documented according to their relevance and also unlike the Binary model it allows partial matching but it assumes that terms are independent statistically. VSM uses term weight to compute a degree of similarity between each document and query which means, this technique gives a low weight to the most frequent term that occurred in many documents and gives weight to less frequent terms however VSM doesn't check whether the document is good or not to a user.

In the vector space model term weights are used to calculate a degree of similarity between document and query then rank those documents in their decreasing order. So VSM uses the following steps to find relevant documents for a query.

First, both document and queries are mapped into term document VSM, then those queries and documents are indicated by weighted vectors (Vi) then the closeness of vector to query identified through cosine similarity measurements for the propose of ranking documents based on cosine similarity measure.

Generally, according to Khan's (Khan, 2014) experiment, the following table shows their difference by comparing those models based on different parameters as a summary which we used as a model selection purpose.

Table 3: Comparison of IR model

Basic Parameter	Information retrieval model		
	Boolean Model	Vector Space model	Probabilistic model
User query	Query oriented match	Partial match	Partial match
Precision	all or none match user request	Retrieve document based on document weight	Retrieve documents based on their probability of occurrence
Recall	Since it is based on exact match concept all or none match document	Better recall when compared to Boolean model	Have better recall than both VSM and Boolean model

3.7. Lesk Algorithm

Information retrieval is an integration of different algorithms that makes the process of searching relevant documents smoothly. Many IR systems are designed based on Boolean, vector space, and probabilistic models. So all those model have their way to represent queries, documents, and algorithms to compute and evaluate document relevance, but while retrieving documents there may cause a disproportion between user needs and documents retrieved. So one of the biggest methods to solve this problem is query expansion methods, which is a method of adding a new match query to the original user query.

Query expansion uses different algorithms for adding the new query as discussed in chapter 2 (Sharma & Joshi, 2019). Therefore this study focuses on one of the known query expansion algorithms called the Lesk algorithm(Ayetiran & Agbele, 2018)

3.8. Elastic Search

Currently, the biggest company like Amazon and Google use the most popular search engine called an elastic search for searching, storing, and organizing information which makes it a lot easier to implement those tasks and also this engine makes it easy to use search API by making it

easy and powerful searchability. Elastic search as specified in chapter-2 is a technique of organizing different data and making them easily accessible. It uses documents to the index for searching purposes, this document can be more than a text which means, it may include other structured data like dates, numbers, and strings.

Documents that have similar characteristics are called indexes. Index in elastic search is used to store documents and access them from it so it is a technique by which search engine operates. Thus, elastic search manages indexes by default through shards that are small elements of indexes. So that it is easy to be searchable because elastic search searches the index instead of searching directly the text.

3.9. *Afaan Oromoo* WordNet

WordNet is a lexical resource used in IR, machine translation, and other applications of NLP, so python contains a library for WordNet in the NLTK package that is prepared for the English language, therefore we construct *Afaan Oromoo* WordNet manually by using different *Afaan Oromoo* dictionaries and discussing with some *Afaan Oromoo* language experts. So in this *Afaan Oromoo* WordNet, there are two major parts are there that is a list of *Afaan Oromoo* synonym words that have similar senses called *Synsets*and *gloss* those words definition(meaning).

According to Banerjee and Pederson(Banerjee & Pedersen, 2003), there are two approaches are there while constructing any WordNet by using different language those are “merge approach” which explains that first defining synsets and their relationship by our language then correlate our WordNet with Princeton WordNet, while the second approach which is called “extended approach” says translate the synsets exist in Princeton WordNet to our language including their relationship. Therefore, in this study, we selected and constructed *Afaan Oromoo* WordNet based on an extended approach.

Lesk algorithm performs word disambiguation by computing word overlap between the target word and words that exist in this dictionary and identifying the highest overlap. This sense overlap can be either synsets overlap which is a group of words in WordNet that are synonym and share the same concept or gloss overlap which is the definition of the target word with the meaning of the word that exists in WordNet. So we selected gloss overlap or synset to synset definition to prepare a sample of *Afaan Oromoo* WordNet.

3.10. Model Evaluation

After retrieving the document, the performance and effectiveness of the system must be evaluated to measure how efficient and effective the system is and the extent to which the users are satisfied, mainly the main aspects of evaluation in IR systems are efficiency and effectiveness.

Effectiveness – this evaluation technique is related to measuring the accuracy of search results, so it evaluates retrieved documents that are either relevant or not. To evaluate the accuracy of the system, it is determined in terms of precision and recall.

- **Precision** is defined as the proportion of a retrieved set of documents that is relevant to the query

$$\text{Precision} = \frac{\text{number of relevant document retrieved}}{\text{Number of relevant document}}$$

- **A recall** is defined as the ratio between all relevant documents and a query that is retrieved

$$\text{Recall} = \frac{\text{number of relevant document retrieved}}{\text{Number of document retrieved}}$$

- **F-Measure** is the harmonic mean of precision and recall

$$\text{F-measure} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision} + \text{Recall}}$$

So we used a matrix table called confusion matrix to represent the performance of the model and to determine the number of the relevant documents retrieved and not retrieved as well as the number of non-relevant documents retrieved. So this idea is depicted in the following confusion matrix table.

Table 4: Confusion Matrix

	P	***R***	***T***
R	doc1	doc2	doc1+ doc2
NR	doc3	doc4	doc3+doc4
T	doc1+doc3	doc2+ doc4	doc1+doc2+doc3+doc4

Where P and R denote relevant and irrelevant documents while T denotes a total number

-Whereas, R, NR denotes retrieved and not retrieved documents.

For example, we can describe doc1 and doc3 as the following based on the above confusion matrix table

- doc1 represent a relevant document that is retrieved whereas
- doc3 represent a relevant document that is not retrieved

Therefore, from the above table, we can compute accuracy as the following: -

Accuracy= $\frac{doc1+doc4}{doc1+doc2+doc3+doc4}$ **x 100%**

Precision = $\frac{doc\,1}{doc\,1+\text{doc}\,3}$ x 100%

Recall = $\frac{doc\,1}{doc\,1+\text{doc}\,2}$ x 100%

Generally, to call IR system is effective, should have:

- ✓ high accuracy value which is close to 100%
- ✓ recall and precision value have equal

Efficiency - this is the evaluation technique that is concerned with evaluating resources used during designing IR systems such as memory space, time-space, and other related resources.

3.11. Implementation Tools

We use the python programming language for implementing purposes because it is a simple, integrated, and powerful tool currently.

3.11.1. Python IDE Tools

Python programming language has different IDE's which are used to create a different program by making developers work easier. In this study, we use a different IDE tool that helps us to develop this system. Every IDE tools have built-in packages which are used in developing a different program. Python tools used in this study are described as the following: -

- ✓ **Anaconda Navigator –** Anaconda is one of GUI (graphical user interfaces) which is used to launch and manage python application, environment, and Conda packages, whereas **Navigator** indicates searching packages on Anaconda.
- ✓ **Jupiter Notebook –** It is one of the python web applications that is used to create a document that contains equations, visualization, and live code.
- ✓ **Spyder -** It is one of free and open-source python IDE which is designed for scientific programming.

3.11.2. Designing Tool

To explain how this proposed system works and also to describe the flow process we design it using different tools. So for designing our system we use Edraw Max and Enterprise Architect application.

- ✓ **Edraw max -**it is one of the designing tools and diagramming software that is mainly used to create a workflow diagram, network diagrams, flowcharts, and other necessary diagrams which is used to design one system. This software also has a good selection of different libraries which simplify our work.
- ✓ **Enterprise Architect -** It is one of the UML tools that is used for designing, building, and deploying a system. This software performs better and loads large models within a second when compared to Edraw Max.

Frameworks used in Python Package

- ✓ **NLTK –** It is one of the NLP (Natural language processing) library that contains different module and program which is used for language processing tasks so that it is possible to understand human language. It includes different packages which are used for different purposes such as for tokenizing a word, stop word removal, stemming, and other necessary

processes. NLTK contains more than 50 corpora and also additionally it includes many lexical resources which contain different data such as WordNet.

- ✓ **Pandas -** It is an open-source data analysis tool that is used in case we want to work with data set for data manipulation, analysis and make those data clear and readable. This library also allows us to import different data that exist in various formats.
- ✓ **Flask –** It is one of the python modules which helps us during web application development and it provides the necessary feature and tools so that it is possible to create web applications easier.

3.11.3. Prepared interface for *Afaan Oromoo* Information retrieval

Interface is a means in which user interact with the system designed for the matter of effective operation so that the user has the ability to control and manage a system or hard ware devices. So it can be taken as a get way for one user to access the system and operate further. So the below figure indicates how a user interact with system designed.

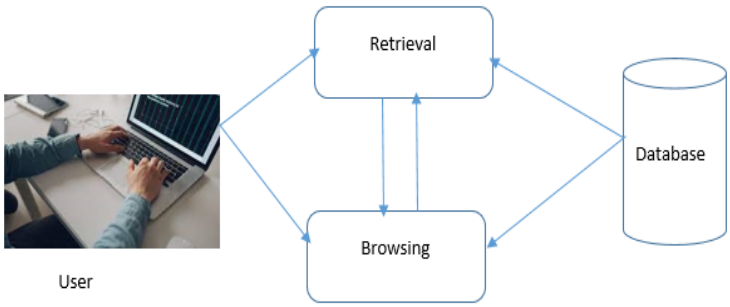


Figure 4: User interaction with the system

3.11.4. Hardware Tools

The designed system had been deployed on a personal computer (PC) with a processor of Intel(R) Core(TM) i5-4210U CPU@1.70GHz 2.40 GHz with the operating system is Windows 10pro, 64-bit operating system, x64-based processor and requires 500 Gigabyte hard disk storage.

CHAPTER FOUR

4. DESIGN AND IMPLEMENTATION OF THE PROPOSED SYSTEM

4.1. Proposed model Architecture

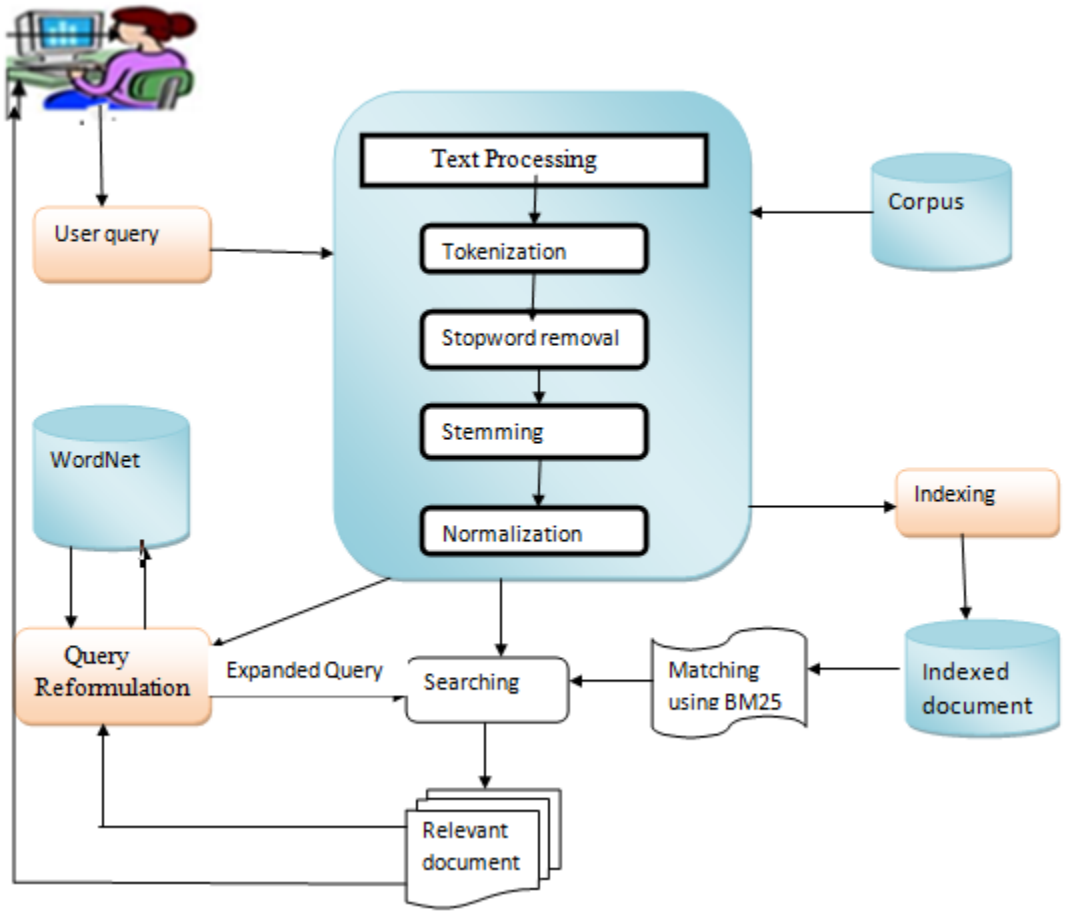


Figure 5: *Afaan Oromoo* IR system Architecture

First, user requests his/her query through the prepared user interface then the next processes are continued. Therefore, the above model is described in short as the following: -

Get the document and query and then apply preprocessing steps like tokenization, stop word removal, normalization, stemming, and also cleaning unnecessary things like symbols, numbers, punctuations, and others to increase the quality of both documents and query.

After the major preprocessing step is finalized the next step is the indexing process in which preprocessed documents and queries are indexed to get informative keywords by reducing documents and elastic search is used to index documents to make prepared corpus parsed and stored to make retrieval fast and accurate. The elastic search takes unstructured data to store and indexes according to user-specified or automatically from data to make our searching faster and better; this is because elastic search uses an inverted index to get keywords.

Then matching step is continued to calculate the similarity between the user query and our document based on weighting terms. Therefore, well known and a best algorithm called BM25 is used to rank documents in response to a user query in decreasing order, since BM25 depends on a probabilistic approach, then ranked documents are retrieved.

Finally, the process of expanding terms will be performed by formulating a new query from WordNet and earlier retrieved document in step 3 to formulate new terms and re-rank relevant documents, then finalized ranked relevant document is back to the user through user interface prepared.

Generally, this study focused on retrieving relevant documents based on query expansion and the Lesk algorithm is used to disambiguate words by calculating sense overlaps (disambiguating semantically). So we use synsets to synsets overlap for expanding and reformulate new terms for designing our system.

4.2. Proposed data preprocessing technique

In this study, we gathered and organized data from different sources as described in section 3, but we cannot use data collected as it is, so the gathered data must pass through necessary preprocessing steps to make gathered data suitable for the model we built.

Tokenization: to make a better environment for our model, the full sentence must be broken down into a sequence of words called token. So we tokenized the data by using the word Tokenizer.

Stop word removal: words that do not add any value in this study were removed. So we removed the most frequent words in the document since they didn't add anything to analyze the data, therefore based on prepared *Afaan Oromoo* stop word lists we removed those unnecessary words.

Normalization: Additionally we remove other unnecessary characteristics like symbols, numbers, punctuation, and special character to clean our data. Lowering the text is also carried out which is used to convert document and user query to lower case if it has existed in the upper one.

Stemming: we stemmed *Afaan Oromoo* words to their root to increase the system efficiency.

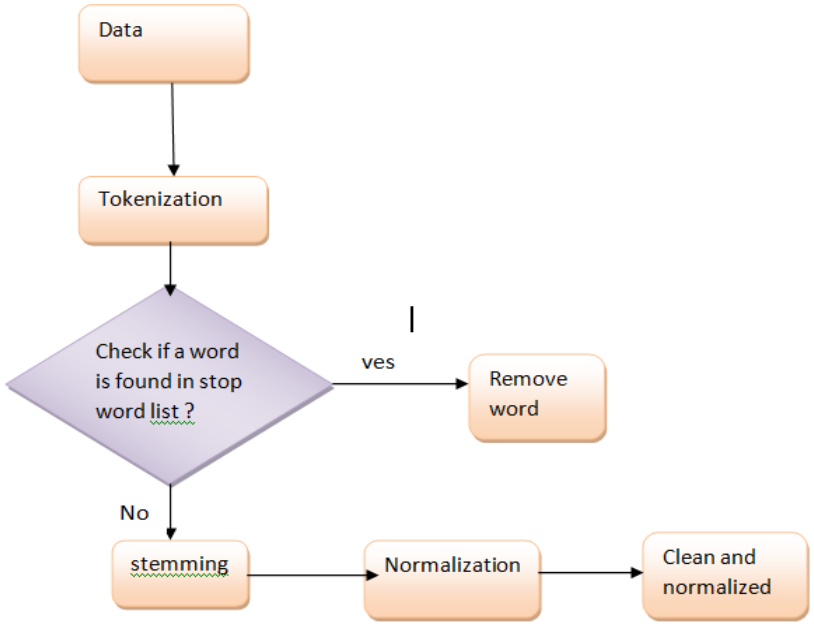


Figure 6: Preprocessing steps

4.3. Proposed Probabilistic IR System using BM25 Model

A probabilistic model is one of the IR models that is designed based on a probabilistic framework to rank documents in decreasing manner based on their relevance score. There are many probabilistic models are there such as the binary independence model, BM25, and others, therefore we selected and propose one of the robust and best probabilistic models used for searching called BM25. BM25 is the ranking function used to estimate document relevance for user query by computing document and user query. So this model uses both TF and IDF and the overall process used in BM25 is depicted as the following diagram.

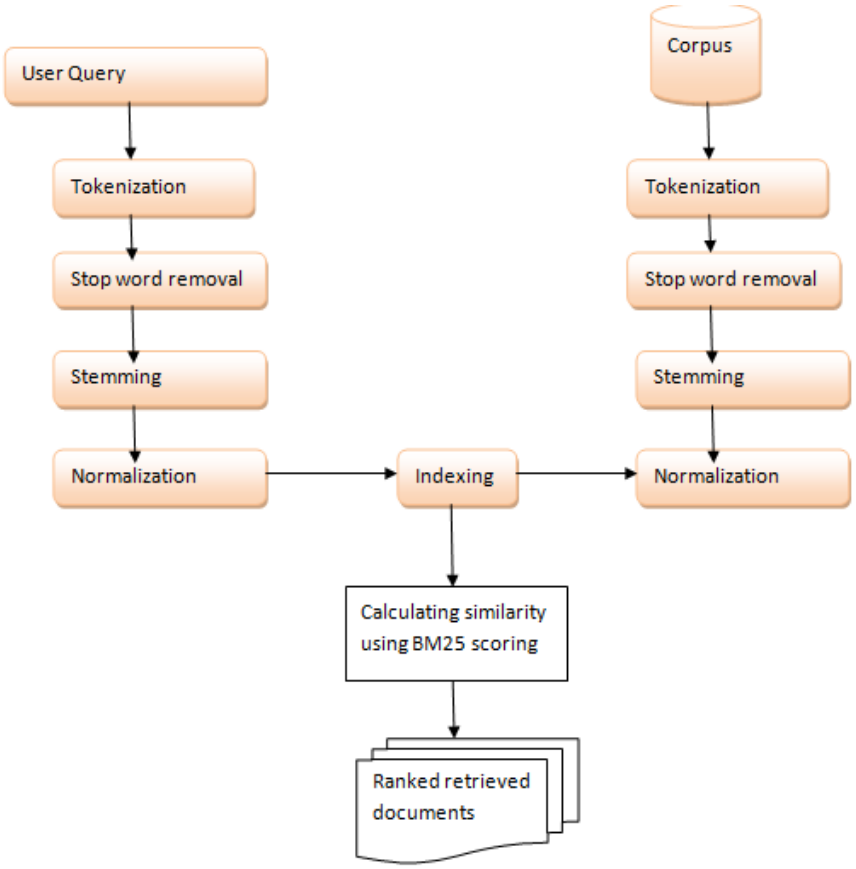


Figure 7: Proposed Probabilistic IR system using BM25 model

4.4. Similarity scoring using Elastic Search

The elastic search uses TF-IDF before version 5.0 and then BM25 similarity scoring function because BM25 existed to handle the shortcoming of TF-IDF function to get a better and more efficient relevant document. Since BM25 is one of the probabilistic models as specified in chapter-2, their relevance score was calculated and documents are ordered depending on the value calculated. So the first index is created which is used to store our document and then give a query against the created index to retrieve relevant documents in decreasing order based on their relevance score by depending on the BM25 model.

4.5. Proposed *Afaan Oromoo* word Sense disambiguation using Lesk algorithm

This study mainly focused on designing and developing the *Afaan Oromoo* IR system for improving the performance by using query expansion technique to decrease the ambiguity of words. Word disambiguate is a technique of finding an appropriate sense of word found in a given sentence. So for this study, we proposed to implement one of the word sense disambiguation techniques called the Lesk algorithm that uses dictionary meaning to disambiguate ambiguous words that exist in the sentence. So we prepared *Afaan Oromoo* WordNet which contains words with their definition as dictionary and then Lesk algorithm count number of words that share the same meanings(synonymous word) between the gloss of word found in the WordNet with a sense of target word, then most appropriate sense will be selected as an additional term for expansion purpose, finally expanded term will be researched for further document retrieval.

Therefore Lesk algorithm states that the more overlapping the words, the more related the senses are, for instance, the following example describes how this proposed system disambiguate ambiguous words: -

The word “pine” has the following two meanings.

- Sense 1: kind of evergreen tree with needle-shaped leaves.
- Sense 2: waste away through sorrow or illness.

While the word "cone" has three senses:

- Sense 1: a solid body that narrows to a point.
- Sense 2: something of this shape, whether solid or hollow.
- Sense 3: the fruit of a certain evergreen tree.

So this algorithm bases its disambiguation decision on the semantic similarity and uses the technique of identifying the relatedness between the meaning of an ambiguous word and the meaning of a word of its context, for instance, for a definition of a word that exists in the dictionary says {d1,d2,d2…dn} and Si gloss definition of the ambiguous word corresponding to

ith sense and A(s) is context definition of an ambiguous word and D(Si) indicates the possible definition of the ambiguous word.

So based on the above-identified element this algorithm countsthe number of words shared between the above two glosses Therefore the process of how these algorithms work are described as the following diagram.

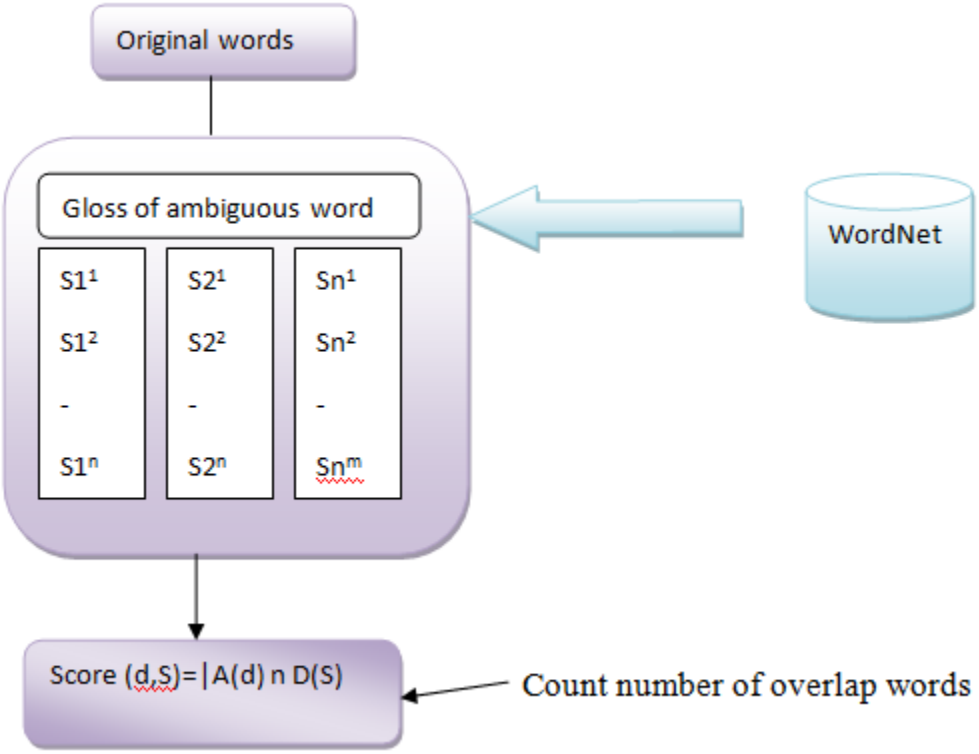


Figure 8: Proposed *Afaan Oromoo* word disambiguation using Lesk algorithm

So depending on the above description of the algorithm, appropriate senses are selected by comparing each of the two glosses of the word “pine” and “cone” then the word “evergreen tree” is selected.

4.6. Preparation of *Afaan Oromoo* WordNet

The main objective of this study is to apply query expansion to improve the performance of the Afan Oromo text retrieval system. Thus, to reformulate the query, the senses of the queries must be prepared from Afan Oromo WordNet. The sample of WordNet is constructed using the Afan Oromo to Afan Oromo dictionary (Tesfaye, 2013). For example, the following sentence shows the sample of Afan Oromo WordNet.

abdii@waan gara fuulduraatti argadha jedhanii eeggatani:ollaa abdatteeti dhirsa heexoo obafte

ayyaana@ guyyaa hojii kan hintaane:guyyoota akka Cuuphaa: Gubaa; Irreechaafi Iidaa kan hawaasti waliin kabajatu: guyyaa laguu akkuma aadaa hawaasaa keessatti ibsamu;afuura /ayyandaa keessa namaatti namatti dhagahamufi nama gajeelcha yn nama eega jedhamee amanamu

Odaa@ gosa mukaa dameefi jirma heddu qabu: maqaa namaa fakkoommii idda dhalootaa heddummaachuu bakka bu'u; idda maqaalee wiirtuu tumaa gadaa waliigalaa kan akka Odaa Namee: Odaa Bultum.

qabeenya@horii:lafa;qabeenna:namni hore qabu.

The WordNet contains the term, synonym term, and the definition of the synonyms. The term before'@' is a reference term about which the different senses are given, for example, the term " qabeenyaa". The synsets are defined between '@' and ':' in this case, the first synset for the word"qabeenya" is "Horii" then followed with gloss definition which is " lafa".If the given term has multiple senses, it is separated with ';' on the WordNet. The second sense for the term "qabeenya"is "qabeenna" is the second sense and the rest is the gloss definition.

The format for the term, synsets, gloss, and sense on WordNet are root word, synsets of the first sense, gloss of first sense, synsets of second sense, and gloss of second sense. For example, the format for qabeenya@horii:lafa;qabeenna:namni hore yookiin qabu:

Root word: qabeenya

Synsets of first sense: Horii

A gloss of first sense: lafa

Synsets of second sense: qabeenna

A gloss of second sense: namni hore yookiin qabu

4.7. Proposed query expansion technique

To retrieve missed documents we proposed to apply the query reformulation technique in which the transformed query is expanded and modified afterword sense disambiguation algorithm is applied and the synonyms of queries words are identified in WordNet. The expanded and modified query is then processed to obtain the set of retrieved documents, which is composed of documents that contain the query terms. Fast query processing is made possible by the index structure previously built. The steps required to produce the set of retrieved documents constitute the retrieval process. Next, the retrieved documents are displayed as search results in ranked order, which is depicted in the following diagram.

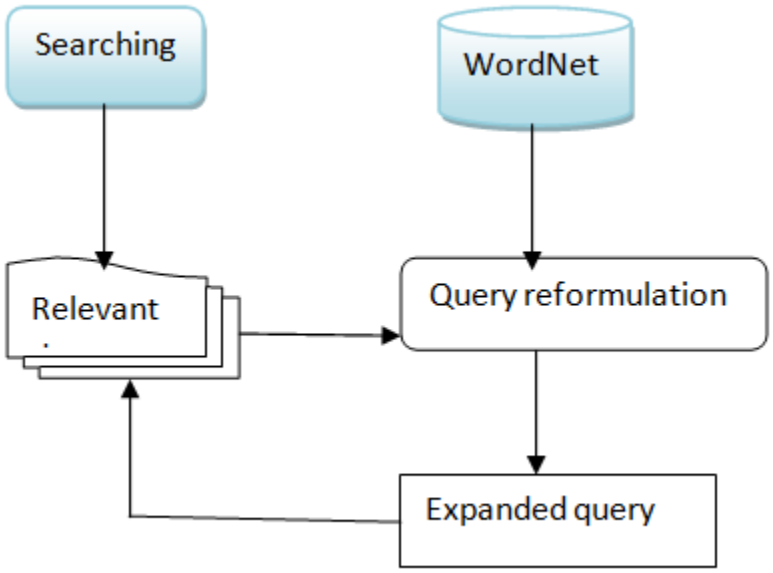


Figure 9: Proposed query expansion technique

Therefore in this mechanism, a new query is formulated from an original query to clarify ambiguous words, so in this study new query is reformulated based on the synset to sense method (semantically related words) that is constructed on *Afaan Oromoo* Wordnet as described in appendix-II.

For example, for the original query: Daandii waaqayyoo

For Daandii the senses are:

1. Bakka namni irra deemu; karaa namoonni irra darban
2. Daandii walqunnamsiistuu: karaa itti namoonni uumaa saanii waliin walqunnaman

For waaqayyo the senses are:

1. uumaa: waaqa uumama hunda uume

2. gooftaa: waaqayyoo gooftaa namootaati

The original query ‘daandii waaqayyoo’ has two words and each word has two senses with a total of four comparisons required to identify similar sense and synset to synset is applied using Lesk algorithm. From the above example, the common words found in each sense definition are taken from the intersection of each definition at the synsets. Then, the query with a common intersection is identified for the expansion. The same process is done for each term of the query and the one with a common intersection is assigned as the sense of the word based on the given query context.

4.9. Proposed Model Evaluation and Testing

As explained in the previous chapter we evaluate our system using effective measures to evaluate whether we developed an expected system that satisfies the user’s need or not. This system is evaluated before the query is expanded and after the query is expanded to compare and evaluate system performance additionally it helps us to identify the effect of this technique on document retrieval. So we use *confusion matrix* – which is a matrix form table that can be n x n in size which is used in case of performance evaluation by summarizing results in short and also this table help us by simplifying other evaluation processes by identifying entity needed such as while calculating precision, recall, and F- measure as explained under chapter 3.

CHAPTER FIVE

5. IMPLEMENTATION OF THE SYSTEM

5.1. Corpus Description

Corpus is the collection of documents in which different terms are found in each document. For *Afaan Oromoo* IR, a sample of hundred documents and ten queries are used to evaluate the retrieval system. These 10 queries are selected on average for 100 documents in a way it includes all aspects of collected data to test the system performance.

5.2 Data preprocessing

To use text operations and import different Natural Language Toolkit, the python library packages are implemented.

```
import math
import requests
import json
import os
from document2 import corpus
```

Figure 10: Data processing implementation

5.2.1. Data Cleaning

Removing different symbols and characters is used to increase the retrieval process while searching relevant documents.

```
def remove_puc(v_String):
 for i in lp:
     v_String=v_String.replace(i,'')
 v_String= re.sub('[\d+]','', v_String)
 return v_String
characters = " .,!#$%^&*();: \n\t\\\"?!{}[]<>0123456789"
def tokenize(document):
 characters = ".,!#$%^&*();: \n\t\\\"?!{}[]<>0123456789"
def tokenize(document):
 terms = document.lower().split()
 return [term.strip(characters) for term in terms]
```

Figure 11.2:Data cleaning Implementation

5.2.2. Stop word Removal

Stop words have insignificant values in IR and decrease the speed and performance of the system while retrieving relevant documents, so based on prepared *Afaan Oromoo* stop word lists, user's queries are sorted as a sample of implementation is described in the following.

```
s=open('stopwords.txt','r')
stopword=s.read()
texts = [
    [word for word in document.lower().split() if word not in stopwords]
    for document in corpus
stopword.close()
]
```

Figure 12: Stop word removal implementation

5.2.4. Stemming

The following sample code implementation implies how *Afaan Oromoo* words are stemmed by removing affixes to get the root or base of words which is further used for indexing purposes.

```
def thestemmer(token):
  if dic.has_key(token):
    return dic.values()

  elif token.endswith('olee'):
       stem=token.replace('olee','')
       if count_m(stem)>=1:
           return stem
  elif token.endswith('olii'):
       stem=token.replace('olii','')
       if count_m(stem)>=1:
          return stem
  elif token.endswith('oolii'):
       stem=token.replace('oolii','')
       if count_m(stem)>=1:
          return stem
  elif token.endswith('ota'):
       stem=token.replace('ota','')
       if count_m(stem)>=1:
          return stem
  elif token.endswith('oolee'):
       stem=token.replace('oolee','')
       if count_m(stem)>=1:
          return stem
  elif token.endswith('oota'):
       stem=token.replace('oota','')
       if count_m(stem)>=1:
          return stem
  elif token.endswith('icha'):
       stem=token.replace('icha','')
       if count_m(stem)>=1:
          return stem
```

Figure 13: Stemming Implementation

5.3. Model used to implement *Afaan Oromoo* Retrieval System

To implement this system, we used a probabilistic model framework. BM25 is used in this study, to optimize the score of queries in TF-ID. The optimization is improved by using BM25 parameters (k=1.5 and b=0.75). These parameters are used to control term saturation and document length to reduce insignificant terms in the documents.

5.3.1. Implementation with BM25

This system is designed based on a probabilistic idea and BM25 to calculate the similarity between user queries and documents based on term weights, which means it takes information from documents through TF and IDF, and then by integrating with the above-mentioned parameters it ranks them decreasingly based on terms appearing in each document. The sample of implementation is depicted as the following.

```
class BM25:
    def __init__(self, k=1.5, b=0.75):
        self.b = b
        self.k = k
    def fit(self, corpus):
        tf = []
        df = {}
        idf = {}
        doc_len = []
        corpus_size = 0
        for document in corpus:
            corpus_size += 1
            doc_len.append(len(document))
            frequencies = {}
            for term in document:
                term_count = frequencies.get(term, 0) + 1
                frequencies[term] = term_count
            tf.append(frequencies)
            for term, _ in frequencies.items():
                df_count = df.get(term, 0) + 1
                df[term] = df_count
        for term, freq in df.items():
            idf[term] = math.log(1 + (corpus_size - freq + 0.5) / (freq + 0.5))
        self.tf_ = tf
        self.df_ = df
        self.idf_ = idf
        self.doc_len_ = doc_len
        self.corpus_ = corpus
        self.corpus_size_ = corpus_size
        self.avg_doc_len_ = sum(doc_len) / corpus_size
        return self
    def search(self, query):
        scores = [self._score(query, index) for index in range(self.corpus_size_)]
        return scores
```

```
def _score(self, query, index):
    score = 0.0
    doc_len = self.doc_len_[index]
    frequencies = self.tf_[index]
    for term in query:
        if term not in frequencies:
            continue
        freq = frequencies[term]
        numerator = self.idf_[term] * freq * (self.k + 1)
        denominator = freq + self.k * (1 - self.b + self.b * doc_len / self.avg_doc_len_)
        score += (numerator / denominator)
    return score
```

Figure 14: BM25 implementation

5.4. Query Reformulation

To expand a word we formulate a new word query by identifying the synonyms of query words found in WordNet, so the following code shows query reformulation implementation.

```
        print("Barbaachi Erga jijjirame (After Query Reformulation)")
        aa=int(l3[xx][0][0])
        ab=int(l3[xx][0][1])
        ac=int(l3[xx][0][2])
        lens=len(sense[aa][ab][ac][1])
        aaa=int(l3[xx][1][0])
        aab=int(l3[xx][1][1])
        aac=int(l3[xx][1][2])
        leng=len(sense[aaa][aab][aac][1])
        nf1=(w[xx]/lens)*(lens/leng)
        case1.append(l3[xx])
        case1.append(nf1)
        expandedQ=expandedQ+sense[aa][ab][ac][1]
        temp=l3[0][0]
        l=len(l3)
        for j in range(len(l3)):
            for i in range(len(l3)-1):
                if l3[i][0]==temp:
                    del l3[i]
                    break
key=0
for j in range(len(expandedQ)):
        for i in range(key+1,len(expandedQ)-1):
            if expandedQ[key]==expandedQ[i]:
                del expandedQ[i]
                break
        key=key+1
```

Figure 15: Query reformulation implementation

5.4.1. *Afaan Oromoo* WordNet Implementation

Sample WordNet is developed manually because there is no standard *Afaan Oromoo* WordNet. Then the word senses are accessed by the Lesk algorithm from lexical resources to reformulate new queries.

```
import re
import os
import math
import sys
from elasticsearches import *
from operator import itemgetter
lists=[]
collection=0
wrdnet=open('wordnet.txt','r')
while wordnet.readline()!='':
     collection=collection+1
wrdnet.close()
wrdnet1=open('wordnet.txt','r')
wrdnet1.readline()
for i in range(c):
     list1=wordnetab.readline()
     g.append(list1)
wrdnet1=[]
wrdnet2=[]
for i in range(len(lists)):
    wrdnet1.append(lists[i])
     wrdnet2.append(wrdnet1)
    wrdnet1=[]
wordnetsynset=[]
word=""
for i in range (len(wrdnet2)):
     word=wrdnet2[i][0]
     wrdnet1=word.split('@')
     worndnetsynset.append(wrdnet1)
for i in range(len(wordnetsynset)-1):
     word=wordnetsynset[i][1]
     wrdnet1=word.split(';')
     wordnetsynset[i][1]=wrdnet1
```

Figure 16: *Afaan Oromoo* WordNet implementation

5.5. Evaluation of *Afaan Oromoo* IR System

IR measurement techniques (recall, precision and, F-measure) are used to evaluate the performance of the retrieval system and to what extent the query expansion techniques satisfy the users to retrieve relevant documents.

CHAPTER SIX

6. EVALUATION RESULTS AND DISCUSSION

6.1. Document Retrieval Result

The attempt this study focused is mainly on searching relevant documents from Afan Oromo text by giving queries that satisfy the information need of the users. In this study, the system is developed with three important mechanisms. Firstly, Afan Oromo WordNet is used as a lexical resource to know the meaning of words. Secondly, query reformulation is applied to expand the query with the match of senses using word sense disambiguation from WordNet. Finally, query expansion with BM25 is integrated to improve the performance of *Afaan Oromoo* Information Retrieval.

6.1.1. List of Select Queries with their Relevant Judgment

We evaluate document relevance by using BM25 to get a ranked document, a number of the document retrieved for each query is identified. Therefore, we summarize the searching result of all those 10 queries in the following table.

Table 5: Document retrieved judgment result

S.No.	***Barbaacha (Query)***	***Relevant Documents in the corpus***	***Non Relevant document in the corpus***	***Relevant documents retrieved for a query***
Q1	DaandiiWaaqayyoo (God's Way)	10	90	doc16,doc3,doc36, doc100,doc60,doc82,doc97,doc23
Q2	SirnaGadaa (Gadaa System)	9	91	doc88,doc69,doc77,doc7,doc25,doc32,doc61,doc89,doc70
Q3	WaldaaIspoortii (Sport Team)	8	92	doc99,doc22,doc19,doc91,doc98,doc41,doc49,doc50
Q4	RakkooNageenyaa (The Threat of Peace)	8	92	doc43,doc47,doc40,doc13,doc78,doc79,doc80
Q5	BulchiinsaGaarii (Good Governance)	6	94	doc53,doc7,doc9,doc10, doc13,doc14
Q6	BirmadummaaBiyaa (The Sovereignty Country)	7	93	doc44,doc64,doc65,doc68,doc90,doc92
Q7	ManneenBarnootaa (Schools)	11	89	doc53,doc54,doc55,doc56,doc91,doc57,doc76,78
Q8	QabeenyaUumamaa (Natural Resource)	8	82	doc1,doc38,doc58,doc51,doc42
Q9	MasaraaMootummaa (Palace)	7	93	doc74,doc96,doc30,doc59,doc62
Q10	QulqullinaBarnootaa	10	90	doc35,doc31,doc37,doc76,doc27,doc38,doc13

6.2. Scoring of Queries Using TF-IDF and BM25 Algorithms

This study is designed based on a probabilistic framework by using an improved version of the TF-IDF approach called BM25. Elastic search use both TF-IDF and BM25 scoring function even

though currently used elastic search version 5 only use the improved version of TF-IDF. So we experiment by using the most frequently applied algorithm TF-IDF and BM25 on our prepared corpus to compare and evaluate which algorithm order documents better that fit with user interest, therefore both algorithms registered their scoring weight for each query in every document. So to show their scoring weight difference we select one retrieved document per query from all retrieved documents as an example as described in the following table.

Table 6: BM25 vs. TF-IDF

S.No	Barbaacha (Query)	Dokumantii (Corpus)	TF-IDF	BM25
1	DaandiiWaaqayyoo (God's Way)	Doc16	0.0093	3.4266
2	SirnaGadaa (Gadaa System)	Doc32	0.0063	2.605
3	WaldaaIspoortii (Sport Team)	Doc41	0.0044	2.0806
4	RakkooNageenyaa (The Threat of Peace)	Doc47	0.1110	4.1192
5	BulchiinsaGaarii (Good Governance)	Doc94	0.0041	1.8833
6	BirmadummaaBiyaa (The Sovereignty Country)	Doc68	0.0088	3.2096
7	ManneenBarnootaa (Schools)	Doc56	0.0054	2.7446
8	QabeenyaUumamaa (Natural Resource)	Doc13	0.0071	3.0341
9	MasaraaMootummaa (Palace)	Doc74	0.0054	2.5789
10	QulqullinaBarnootaa	Doc37	0.0050	2.0459

As we can see from the above table BM25 registered better results, as well as the number of relevant documents retrieved, is also increased when compared to TFIDF because in BM25 longer documents have a chance to contain a longer query that improves retrieval performance and optimizes TFIDF since it uses an additional parameter to adjust document length and weight. So we based our experiment conclude that generally, BM25 performs better than TFIDF because of this our study is designed on this. The following chart depicts both algorithms that have higher scoring values when applied to the prepared corpus.

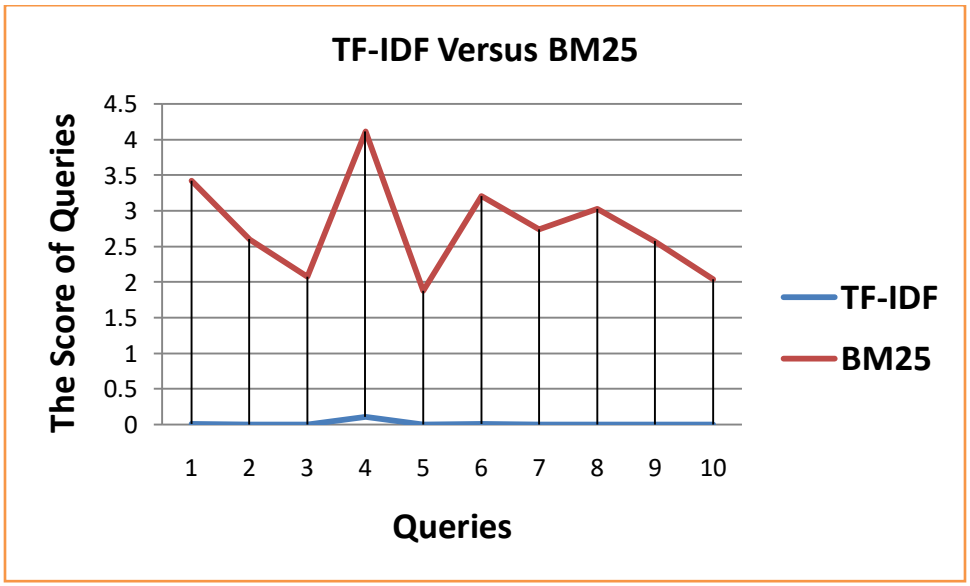


Figure 17: Comparison between TF-IDF and BM25

From the above figure, the score of BM25 is higher than TF-IDF. This indicates that searching documents for each query in each document is improved by optimizing TF-IDF by BM25.

6.3. Evaluation of *Afaan Oromoo* IR System

The effectiveness and efficiency of the IR System are evaluated. *Efficiency* measures the time taken for searching. In this study elastic search is used in indexing to make fast searching. *Effectiveness* is a technique to measure recall, precision, and F-Measure of search results of the documents.

6.3.1. Evaluation of Retrieval Performance Before Query Reformulation

we use the BM25 scoring function to score and retrieve documents, therefore while retrieving a ranked document from document corpus the designed performance and effectiveness should be measured to evaluate to what extent users are satisfied because of this we evaluate the result before query expansion and after query expansion by using different IR effectiveness measure such as accuracy by using precision and recall, and also F- measure. The below table presents the initial result without query expansion for the eight queries used to test the system.

Table 7: Document retrieved result before query expansion

S.N	Barbaacha (Query)	Relevant Documents in the corpus	No of Retrieved Documents	No of Relevant document	R	P	F
Q1	DaandiiWaaqayyoo (God's Way)	10	8	8	0.80	1.0	0.889
Q2	SirnaGadaa (Gadaa System)	9	10	9	1.0	0.90	0.947
Q3	WaldaaIspoortii (Sport Team)	8	9	7	0.875	0.778	0.824
Q4	RakkooNageenyaa (The Threat of Peace)	8	12	7	0.875	0.583	0.70
Q5	BulchiinsaGaarii (Good Governance)	6	8	6	1.0	0.75	0.857
Q6	BirmadummaaBiyaa (The Sovereignty Country)	7	6	6	0.857	1.0	0.923
Q7	ManneenBarnootaa (Schools)	11	8	8	0.727	1.0	0.842
Q8	QabeenyaUumamaa (Natural Resource)	7	7	6	0.857	0.857	0.857
Q9	MasaraaMootummaa (Palace)	7	8	5	0.714	0.625	0.667
Q10	QulqullinaBarnootaa	7	10	7	1.0	0.70	0.824

From the above table, the average result of the recall, precision, and F-score of the system using the initial retrieve made by the model for the relevance of documents are 87.1%, 82%, and 83.3% respectively.

As we see from the table, all relevant documents from the corpus are not retrieved; this makes a low percentage for recall. And also from retrieved documents, some documents are not relevant to the queries; this low percentage for precision. The low percentage of recall and precision proceed for a low percentage of F- Score. Generally, the retrieval result before query reformulation is not satisfactory for the users. The reason for the low performance of *Afaan Oromoo* IR in this initial retrieval is the Synonymous and polysemous words problem. For example, for query ‘daandii waaqayyoo’ (God’s Way) two relevant documents are not retrieved. The synonym meaning of ‘daandii waaqayyo’ is equivalent to the query ‘karaa rabbii’ (God’s Way). Thus, the documents that contain the query ‘karaa rabbii’ are not retrieved. This is the problem of synonymous. In another, the query ‘rakkoo nageenyaa’ (the Threat of Peace) three irrelevant documents are retrieved. Another meaning of ‘rakkoo nageenyaa’ is ‘rakkoo fayyaa’(illness). Such a type of ambiguity is the problem of polysemous words.

6.4. Document retrieved result after query expansion

So to resolve the above-mentioned problem query expansion technique is a technique of modifying the query is used that is used to meet users' needs and improve retrieval performance. Table 6.4 shows the result after query expansion is used.

Table 8: Document retrieved result after query expansion

S.No	*Barbaacha (Query)*	*Relevant Documents in the corpus*	*Number of Retrieved Documents*	*Number of Relevant documents retrieved*	*Recall*	*Precision*	*F-Measure*
Q1	DaandiiWaaqayyoo (God's Way)	10	10	10	1.0	1.0	1.0
Q2	SirnaGadaa (Gadaa System)	9	10	9	1.0	0.90	0.947
Q3	WaldaaIspoortii (Sport Team)	8	9	7	0.875	0.778	0.824
Q4	RakkooNageenyaa (The Threat of Peace)	8	12	8	1.0	0.667	0.80
Q5	BulchiinsaGaarii (Good Governance)	6	8	6	1.0	0.75	0.857
Q6	BirmadummaaBiyaa (The Sovereignty Country)	7	7	7	1.0	1.0	1.0
Q7	ManneenBarnootaa (Schools)	11	11	11	1.0	1.0	1.0
Q8	QabeenyaUumamaa (Natural Resource)	7	7	7	1.0	1.0	1.0
Q9	MasaraaMootummaa (Palace)	7	8	5	0.714	0.625	0.667
Q10	QulqullinaBarnootaa	7	10	7	1.0	0.70	0.824

As it is shown from the above table, the average result of the recall, precision, and F-score of the system after query reformulation is applied for the relevance of documents are 95.9%, 84.2%, and 89.2% respectively. When the retrieval performance after query reformulation is compared with initial retrieval, recall, precision, and F-score are increased by 8.8%, 2.2%, and 5.99% respectively. On recall, it shows that relevant documents in the collection are almost retrieved except for some documents. Those documents that are not retrieved after query reformulation is

because of the sense of new queries are constructed on synsets to gloss and also because of the polysemous problem some relevant documents are not retrieved after query is reformulated. Finally, the increase of F-score shows as the IRs are improved after query expansion with BM25 is implemented. Therefore, the enhancement of *Afaan Oromoo* Information Retrieval after query expansion is satisfactory.

6.5. Human Evaluation Result

As we mentioned earlier, we prepared *Afaan Oromoo* interface to make it easy for a user to use as well as to evaluate the results, so any user that can understand and write *Afaan Oromoo* can use it to retrieve relevant documents that fit his/her requests through a prepared user interface. So we experimented to evaluate the result obtained by the user.

The following table describes the general performance evaluation by the user of the designed system for sampled query by summarizing a number of the relevant and not relevant documents that were retrieved and not retrieved.

Table 9: Human evaluation result

Barbaacha (Query)	***Corpus***	***Relevant***	***Not Relevant***	***Total***
DaandiiWaaqayyoo	Retrieved	8	-	8
	Not Retrieved	2	90	92
	Total	**10**	**90**	**100**
SirnaGadaa	Retrieved	9	1	10
	Not Retrieved	-	90	90
	Total	**9**	**91**	**100**
WaldaaIspoortii	Retrieved	8	2	10
	Not Retrieved	-	90	90
	Total	**8**	**92**	**100**
RakkooNageenyaa	Retrieved	7	4	11
	Not Retrieved	1	88	89
	Total	**8**	**92**	**100**

BulchiinsaGaarii	Retrieved	6	2	8
	Not Retrieved	-	92	92
	Total	**6**	**94**	**100**
BirmadummaaBiyyaa	Retrieved	6	3	9
	Not Retrieved	1	90	91
	Total	**7**	**93**	**100**
ManneenBarnootaa	Retrieved	8	2	10
	Not Retrieved	3	87	90
	Total	**11**	**89**	**100**
QabeenyaUumamaa	Retrieved	6	5	11
	Not Retrieved	2	87	89
	Total	**8**	**92**	**100**
MasaraaMootummaa	Retrieved	5	3	8
	Not Retrieved	2	90	92
	Total	**7**	**93**	**100**
QulqullinaBarnootaa	Retrieved	7	3	10
	Not Retrieved	-	90	90
	Total	**7**	**93**	**100**

6.7. Comparison with Existing *Afaan Oromoo* IR System

As we discussed under section 2.10 several researchers were tried to conduct different research on information retrieval for different languages. So, we tried to review those works to identify the major gaps that existed in the research and we design our best solution to improve the information retrieval system performance; therefore, under this section, we mainly considered and focused on those works done on *Afaan Oromoo* so that it is possible to evaluate most important contribution done in this study to improve those existing system.

First, we reviewed the works done by Solomon R.("SOLOMON REGASSA GUDA July, 2016 Adama, Ethiopia," 2016) in which he conducted *Afaan Oromoo* IR system by using one of probabilistic model, BIM and his results were unsatisfactory in that lesser number of relevant documents are retrieved as per his prepared corpus and also he didn't use a stemming algorithm that decreases system performance. Additionally, he didn't use any techniques to control the effect of word ambiguity which can be also taken as a major issue that has an impact on his result.

Second, we compared and checked the works done by Gutema Gezahegn(Eggi, 2012); he worked on how to retrieve *Afaan Oromoo* documents by using VSM, which is a model that is unable to control uncertainty that occurred in document retrieval and it is low performing model when we compared with the probabilistic model and he didn't use query expansion technique in his study that can increase and improve the retrieval system.

Thirdly, we checked research done on "*Afaan Oromoo* text retrieval using probabilistic approach" by Isayas Wakgari(Wakgari et al., 2016). He used a probabilistic model to retrieve documents even though he didn't work clearly on the effect of adding synonymous words since he used it manually by himself without any predesigned algorithm and techniques used for query expansion.

Therefore, we tried to identify the major gaps that existed on the previously developed *Afaan Oromoo* IR system, and we designed a better solution so that it is possible to overcome those

problems as discussed in the following chapter 6.8

6.8. Contribution

In this study, we used the probabilistic model which is a model that can control document uncertainty and is used to retrieve a better number of documents, and also BM25 is used for better similarity measurement. Second, since Afaan has many ambiguity words, we used query expansion technique that improves system performance by adding semantically related words that can be used to clarify the original user query, so that it is possible to retrieve other missed relevant documents and finally we used the stemming algorithm that can improve system performance. Therefore, our system improves *Afaan Oromoo* IR system performance with better effectiveness and efficiency level than previous research work.

In short, the development of the *Afaan Oromoo* IR system under this study contributes many additional useful things besides document retrieval. Some of them are: -

- ✓ retrieving *Afaan Oromoo* documents using the BM25 approach; while searching relevant documents for a specified query, each document is ranked based on probabilistic way and better documents are returned that better fits users interest.
- ✓ Address the impact of word ambiguity on system performance and disambiguate those words by adding other synonymous words on the result experimentally.
- ✓ We used a stemming algorithm on our prepared corpus for better document retrieval is also one of the major contributions of the study.

Generally, this study contributed higher to the development of *Afaan Oromoo* and helps speakers of this language to search documents by using their language with better performance.

6.8. Discussion

The objective of this study is mainly to design an *Afaan Oromoo* information retrieval system, to achieve this goal the research question mentioned under section 1.5 is answered as the following:

RQ1:How to adopt a probabilistic model with query expansion?

To answer this question we constructed a probabilistic correlation between document and query terms through BM25 approaches which helped us to select better expansion terms that are used for the new coming queries. In addition, to evaluate the impact of the probabilistic model, we selected the two most widely used probabilistic BM25 model and TF-IDF of vector space model and we made an experiment based on the above two models respectively to identify which model ranks document in a better way. Therefore BM25 has a higher scoring value and also can control document length and saturation than TF-IDF which indicates that a better number of relevant documents are retrieved in BM25 and the system has good retrieval performance.

RQ2: How query expansion technique is applied to reformulate a new query?

As mentioned in chapter 2 in the process of expanding a query different techniques are there which is used to reformulate new queries from the original query. We used a knowledge-based technique that needs other external resources as a resource for purpose of generating a new query. Therefore, we prepared a sample of *Afaan Oromoo* WordNet as a dictionary which

contains words with their meanings and is used to identify the meaning of each ambiguous word found in the documents and we used synsets to synsets method to formulate new queries that solve the problem of synonymous words (semantically related words).

RQ3: *To what extent does the proposed approach enhance the effective performance of the Afaan Oromoo IR system?*

This question is answered by performing and evaluating query expansion on the prepared corpus, thus we prepared an experiment to identify before and after query expansion, so as per our experiment we identify that the result registered before expansion is unsatisfactory because some relevant documents are not retrieved as well as some of the retrieved documents are not relevant because of word ambiguity, therefore low recall is registered, while after the query is expanded the original queries were modified to improve retrieval performance. The following table depicts the evaluation result before and in terms of recall, precision, and F-measure.

Table 10: Result before and after query expansion

Measurement Techniques	***Original query***	***Modified query***
Recall	87.1%	95.9%
Precision	82%	84.2%
F-Measure	83.3%	89.2%

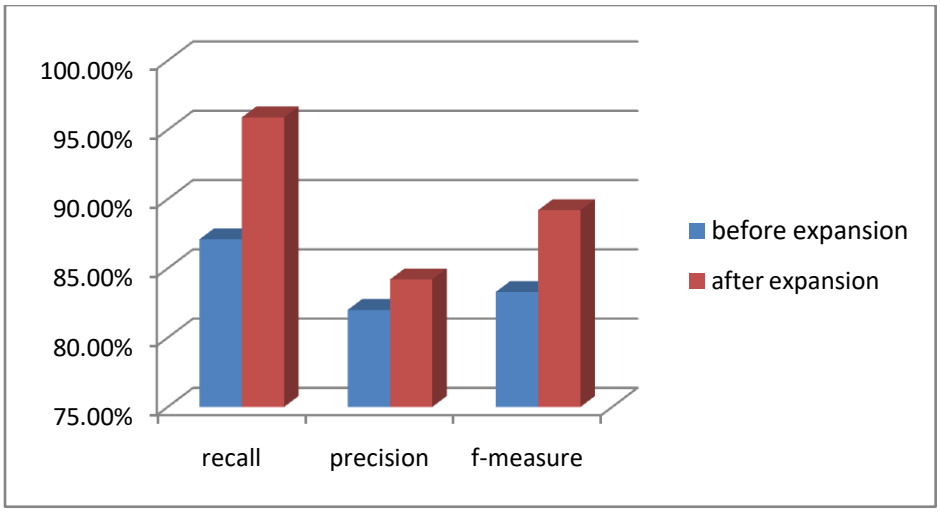


Figure 18: Evaluation result of before and after query expansion

CONCLUSION, RECOMMENDATION, AND FUTURE WORK

Conclusion

Information retrieval is a technique used to store, search and retrieve relevant documents that fit users' needs from the document collection. Some researchers developed *Afaan Oromoo*IR systems using different IR models even though most of them could not solve the problem of IR uncertainty and word ambiguity that is difficult for users to get relevant documents. Therefore, this system is designed based on a probabilistic model that can control the uncertainty, and also the words are disambiguated after the query is reformulated to improve the performance of the *Afaan Oromoo* retrieval system.

From the probabilistic approach, we selected BM25 to design this system by incorporating with elastic search to increase searching speed and additionally we performed an experiment to compare BM25 with the most widely used ranking function TF-IDF and we conclude BM25 improve the search result by optimizing TF-IDF. However, a better number of documents are retrieved in BM25 because of some word ambiguity some relevant documents are left which decreases system performance; therefore, query expansion is used in this study to overcome this problem. New query was reformulated by combining previously retrieved documents with prepared lexical resources in which *Afaan Oromoo* WordNet is prepared manually which is used as lexical resources.Finally,the experimental result indicates that the system registers better recall and precision after the query is expanded, which means the system improved by 5.99% F-measure from the original query which improves system performance. Generally, this result indicates that developing query expansion using the BM25 approach records better results and improves the *Afaan Oromoo* retrieval system.

.

Recommendation and Future Work

Afaan Oromoo Information Retrieval System is tried to develop by some researchers. We also developed and improved the performance of the retrieval system using query expansion techniques. However, the performance of this retrieval system is improved; there are many works to be done for the future. For example, to retrieve all relevant documents as per the users' need, word sense disambiguation will be developed based on all semantic relations such as antonymy, hyponymy, meronymy, troponymy, and entailment. *Afaan Oromoo*standard WordNet which includes all semantic relations should be constructed. To do so, other IR models and techniques should be applied to develop effective and efficient *Afaan Oromoo*IR. Thus, we recommend to the future researchers as they include the above research gaps in the future study.

There are many algorithms and techniques that we did not use in *Afaan Oromoo* IR because of the constraint of time. For example, query expansion using global analysis, local analysis techniques, pseudo feedback, and Rocchio algorithm. Therefore, to increase the user need and effective performance of retrieval system, these techniques and algorithms should be included in future work.

REFERENCES

Alemayehu, N. (2003). Analysis of performance variation using query expansion. *Journal of the American Society for Information Science and Technology*, *54*(5), 379–391. https://doi.org/10.1002/asi.10217

Aliwy, A. H., & Taher, H. A. (2019). Word Sense Disambiguation: Survey study. *Journal of Computer Science*, *15*(7), 1004–1011. https://doi.org/10.3844/jcssp.2019.1004.1011

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

Arukgoda, J., Bandara, V., Bashani, S., Gamage, V., & Wimalasuriya, D. (2015). A Word Sense Disambiguation Technique for Sinhala. *Proceedings - 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology, ICAIET 2014*, 207–211. https://doi.org/10.1109/ICAIET.2014.42

Ayetiran, E. F., & Agbele, K. (2018). An Optimized Lesk-Based Algorithm for Word Sense Disambiguation. *Open Computer Science*, *8*(1), 165–172. https://doi.org/10.1515/comp-2018-0015

Azad, H. K., & Deepak, A. (2019). Query expansion techniques for information retrieval: A survey. *Information Processing and Management*, *56*(5), 1698–1735. https://doi.org/10.1016/j.ipm.2019.05.009

Baeza-Yates, R., & Riberio-Neto, B. (1999). Baeza-Yates99Modern, *9*.

Bakala, N. (2019). Information Retrieval System By Using Vector Space Model, *8*(10), 1562–1568.

Banerjee, S., & Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. *IJCAI International Joint Conference on Artificial Intelligence*, 805–810.

Choudhary, L. (2012). Role of Ranking Algorithms for Information Retrieval. *International Journal of Artificial Intelligence & Applications*, *3*(4), 203–220. https://doi.org/10.5121/ijaia.2012.3415

Eggi, G. G. (2012). Gezehagn Gutema Eggi Advisor : Million Meshesha (PhD), (June).

Fuhr, N. (1992). Probabilistic models in information retrieval. *Computer Journal*, *35*(3), 243–255. https://doi.org/10.1093/comjnl/35.3.243

Gemechu, D. T., & Abebe, E. (2010). Designing a Rule Based Stemmer for Afaan Oromo Text. *International Journal of Computational Linguistics (IJCL)*, *1*(2), 1. Retrieved from http://www.cscjournals.org/csc/manuscriptinfo.php?ManuscriptCode=69.70.63.72.41.50.102

Haigh, T. (2011). The history of information technology. *Annual Review of Information Science and Technology*, *45*, 431–487. https://doi.org/10.1002/aris.2011.1440450116

Hamid, A. (2017). Relevance Feedback in Information Retrieval. *The SMART Retrieval System*, (October), 313–323.

Ibrahim Bedane. (2015). The Origin of Afaan Oromo : Mother Language. *Double Blind Peer Reviewed International Research Journal*, *15*(12).

Imran, H., & Sharan, A. (2009). Nd query. *Journal of Computer Science*, *1*(2), 89–97.

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

Johar, A. (2020). Information retrieval system for silte language using BM25 weighting. *ArXiv Preprint ArXiv:2012.08907*, 1–3.

Jones, K. S. (1979). Experiments in relevance weighting of search terms. *Information Processing and Management*, *15*(3), 133–144. https://doi.org/10.1016/0306-4573(79)90060-8

Junwei, L. U. O. (2010). Research on Information Retrieval System Based on Semantic Web and Multi- Agent, (3), 3–5. https://doi.org/10.1109/ICICCI.2010.35

K, A. M., & Anto, B. (2007). Ambiguities in Natural Language Processing. *International Journal of Innovative Research in Computer and Communication Engineering (An ISO Certified Organization)*, *32972*(5), 392–394. Retrieved from www.ijircce.com

Kadhim, A. I. (2019). Term Weighting for Feature Extraction on Twitter: A Comparison between BM25 and TF-IDF. *2019 International Conference on Advanced Science and Engineering, ICOASE 2019*, 124–128. https://doi.org/10.1109/ICOASE.2019.8723825

Khan, J. A. (2014). Comparative study of information retrieval models used in search engine.

2014 International Conference on Advances in Engineering and Technology Research, ICAETR 2014, 1–5. https://doi.org/10.1109/ICAETR.2014.7012832

Klink, S., Hust, A., Junker, M., & Dengel, A. (2002). Collaborative learning of term-based concepts for automatic query expansion. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2430*(December 2012), 195–207. https://doi.org/10.1007/3-540-36755-1_17

Kokane, C. D., & Babar, S. D. (2019). Supervised Word Sense Disambiguation with Recurrent Neural Network Model. *International Journal of Engineering and Advanced Technology*, *9*(2), 1447–1453. https://doi.org/10.35940/ijeat.b3391.129219

Manning, C. D., Raghavan, P., & Schutze, H. (2012). Index construction. *Introduction to Information Retrieval*, (c), 61–77. https://doi.org/10.1017/cbo9780511809071.005

Munteanu, D. (2007). Vector space model for document representation in information retrieval, 43–46.

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., & Ioma, C. (2006). Terrier: A High Performance and Scalable Information Retrieval Platform. *Proceedings of the Open Source Information Retrieval Workshop*, 18–25. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Terrier+:+A+High+Performance+and+Scalable+Information+Retrieval+Platform#0

Pal, D., Mitra, M., & Datta, K. (2014). Improving query expansion using WordNet. *Journal of the Association for Information Science and Technology*, *65*(12), 2469–2478. https://doi.org/10.1002/asi.23143

Rivas, A. R., Iglesias, E. L., & Borrajo, L. (2014). Study of query expansion techniques and their application in the biomedical information retrieval. *The Scientific World Journal*, *2014*. https://doi.org/10.1155/2014/132158

Salton, G., & McGill, M. J. (1983). Introduction to Modem Information, 375–384. Retrieved from http://portal.acm.org/citation.cfm?id=1893971.1894017

Shaalan, K., Al-Sheikh, S., & Oroumchian, F. (2012). Query expansion based-on similarity of terms for improving Arabic information retrieval. *IFIP Advances in Information and*

Communication Technology, *385 AICT*, 167–176. https://doi.org/10.1007/978-3-642-32891-6_22

Sharma, P., & Joshi, N. (2019). Knowledge-Based Method for Word Sense Disambiguation by Using Hindi WordNet. *Engineering, Technology & Applied Science Research*, *9*(2), 3985–3989. https://doi.org/10.48084/etasr.2596

Shiri, A. (2004). Introduction to Modern Information Retrieval (2nd edition). *Library Review*, *53*(9), 462–463. https://doi.org/10.1108/00242530410565256

Singhal, A. (n.d.). Modern Information Retrieval : A Brief Overview, 1–9.

SOLOMON REGASSA GUDA July, 2016 Adama, Ethiopia. (2016).

Svore, K. M., & Burges, C. J. C. (2009). A machine learning approach for improved BM25 retrieval. *International Conference on Information and Knowledge Management, Proceedings*, 1811–1814. https://doi.org/10.1145/1645953.1646237

Tamrakar, A., & Vishwakarma, S. K. (2016). Analysis of Probabilistic Model for Document Retrieval in Information Retrieval. *Proceedings - 2015 International Conference on Computational Intelligence and Communication Networks, CICN 2015*, 760–765. https://doi.org/10.1109/CICN.2015.155

Thacker, U., Pandey, M., & Rautaray, S. S. (2016). Performance of elasticsearch in cloud environment with nGram and non-nGram indexing. *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, 3624–3628. https://doi.org/10.1109/ICEEOT.2016.7755381

Trotman, A., Clarke, C. L. A., & Culpepper, S. (2012). *Open Source Information Retrieval : a Report on the SIGIR 2012 Workshop* (Vol. 46).

Vasiloaia, M. (2020). Semantic Ambiguity in Advertising Language, *23*(2), 95–99.

Wakgari, I., Advisor, K., & Assefa, S. (2016). BY: ISAYAS WAKGARI KELBESSA ADVISOR: SHIMELIS ASSEFA (PhD).

White, R., Jose, J., & Ruthven, I. (2002). Comparing explicit and implicit feedback techniques for web retrieval: Trec-10 interactive track report. *Proceedings of the Tenth Text Retrieval*

Conference (TREC-10), 534–538. Retrieved from http://strathprints.strath.ac.uk/2466/

Yue, W., Chen, Z., Lu, X., Lin, F., & Liu, J. (2005). Using query expansion and classification for information retrieval. *Proceedings - First International Conference on Semantics, Knowledge and Grid, SKG 2005*, (Skg 2005), 31. https://doi.org/10.1109/SKG.2005.139

Zheng, Yicheng1. Zheng Y, Deng F, Zhu Q, D. Y. C. storage and search for mass spatio-temporal data through P. V. and E. cluster. C. 2014-P. 2014 I. 3rd I. C. C. C. I. S. 2014;470–4., Deng, F., Zhu, Q., & Deng, Y. (2014). Cloud storage and search for mass spatio-temporal data through Proxmox VE and Elasticsearch cluster. *CCIS 2014 - Proceedings of 2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems*, 470–474. https://doi.org/10.1109/CCIS.2014.7175781

APPENDIXES

Appendix –I. *Afaan Oromoo* letters with their sound pronunciation (Eggi, 2012)**.**

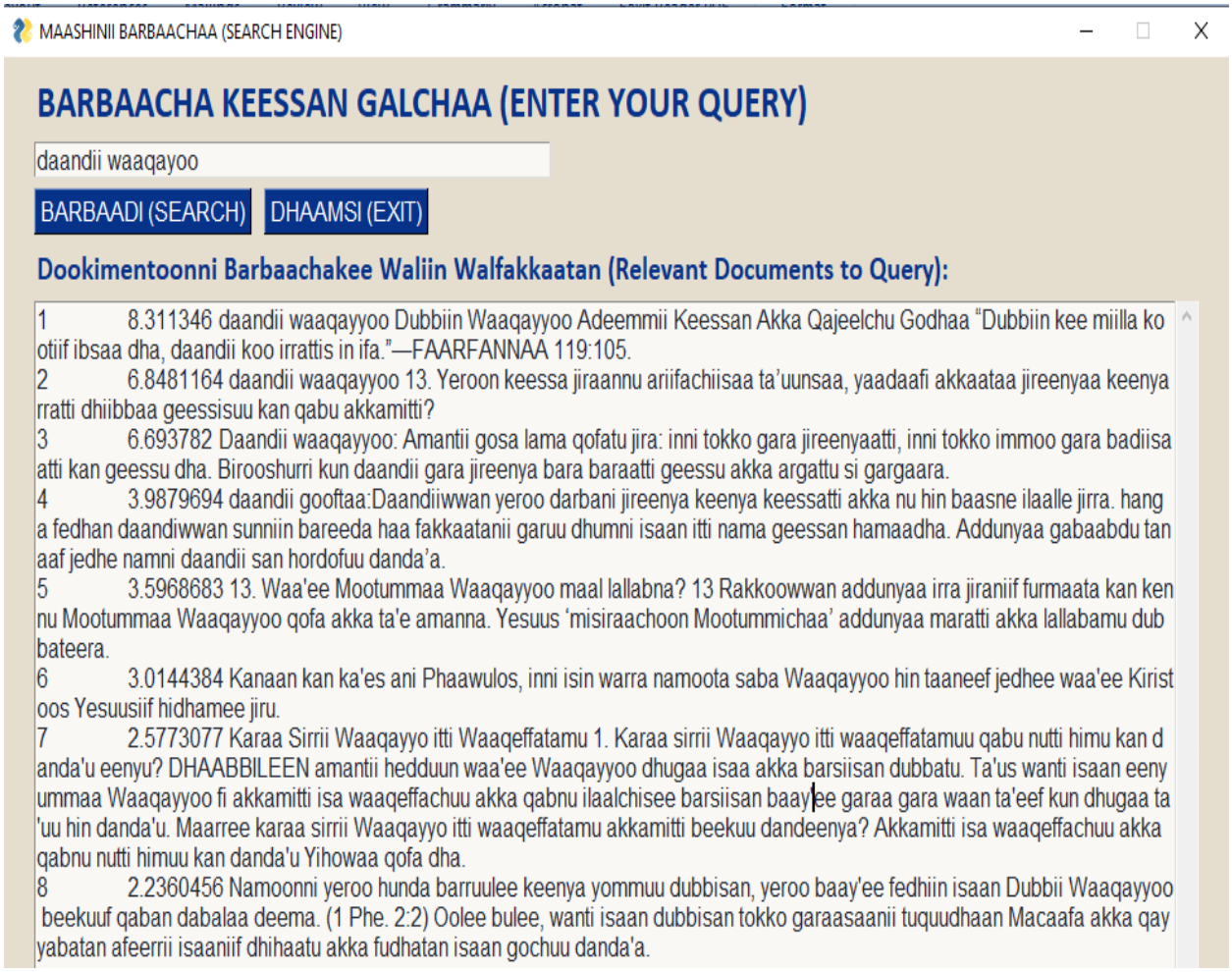
Alphabets	**Sounds**	**Alphabets**	**Sounds**	**Alphabets**	**Sounds**	**Alphabets**	**Sounds**	**Alphabets**	**Sounds**	**Alphabets**	**Sounds**
A a	*[aa]* like **ask**	**B b**	*[baa]* like **bi**rd	**C c**	*[Caa]* like **ca**t	**D d**	*[daa]* like **da**m	**E e**	*[ee]* like **a**te	**F f**	*[ef]* like **fu**ngi
G g	*[gaa]* like **gu**n	**H h**	*[haa]* like **ha**t	**I i**	*[ie]* like **I**ndia	**J j**	*[jaa]* like **Ju**st	**K k**	*[kaa]* like **Ca**st	**L l**	*[la]* like **li**fe
M m	*[ma]* like **ma**n	**N n**	*[naa]* like **na**sty	**O o**	*[oo]* like **o**ld	**P p**	*[pee]* like **pa**st	**Q q**	*[quu]* like **quit**	**R r**	*[ra]* like **ra**t
S s	*[saa]* like **sa**lad	**T t**	*[taa]* like to**ta**l	**U u**	*[uu]* like **u**rge	**V v**	*[vau]* like **va**ry	**W w**	*[wee]* like **wa**nt	**X x**	*[taa]* like ___
Y y	*[y]* like **you**th	**Z z**	*[Zay]* like **Tha**t	**CH ch**	*[chaa]* like **cha**t	**DH dh**	*[dhaa]* like ___	**SH sh**	*[shaa]* like **sh**y	**NY ny**	*[nyaa]* like _____
PH ph	*[phaa]* like										

Appendix – II. Sample of *Afaan Oromoo* Stop words

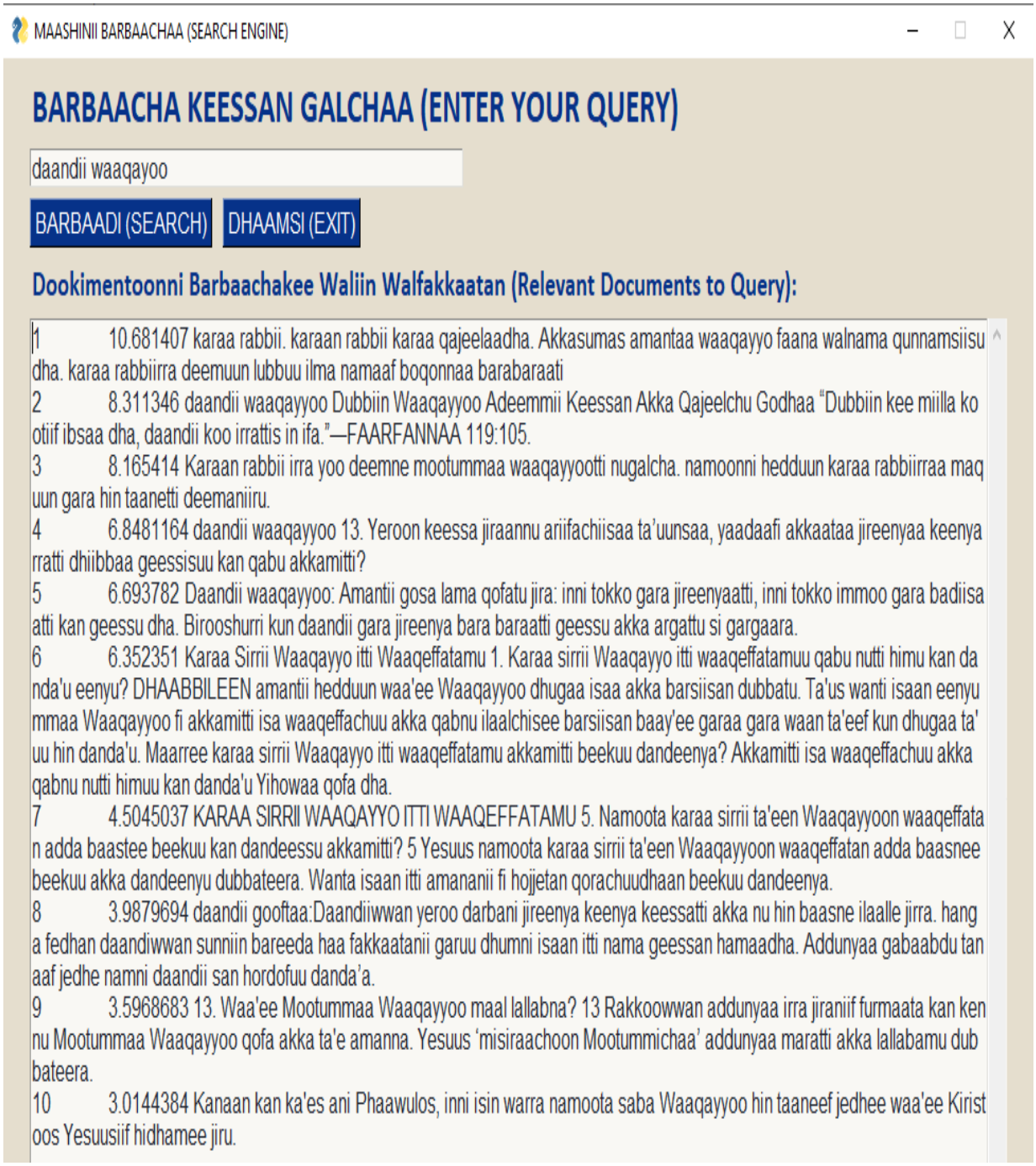
	Aanee	Gararraa	Ishiirraa	Koo	Siin
	agarsiisoo	garas	ishiitti	kun	silaa
	akka	garuu	ishiitti	lafa	silaa
	akkam	giddu	isii	lama	simmoo
	akkasumas	gidduu	isiin	malee	sinitti
	akkum	gubbaa	isin	manna	siqee
	akkuma	ha	isini	maqaa	sirraa
	ala	hamma	isinii	moo	sitti
	alatti	hanga	isiniif	na	sun
	alla	henna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

Appendix-III- *Afaan Oromoo* Information Retrieval Interface

a. *Afaan Oromoo* retrieval system before a query is reformulated



b. *Afaan Oromoo* retrieval system after a query is reformulated



Appendix –II.*Afaan Oromoo* WordNet Sample

aadaa@sagalee rakkinaa:aarii;akkkaataa namoonni biyya tokkoo keessaa jiraataan:ittiin bulmaata uummata oromoo;duudhaa:aadaa oromoo keeesssatti marqaan cuukkoon caccabsaan baay'ee jaallatamu

abdii@waan gara fuulduraatti argadha jedhanii eeggatani:ollaa abdatteeti dhirsa heexoo obafte

ayyaana@ guyyaa hojii kan hintaane:guyyoota akka Cuuphaa: Gubaa; Irreechaafi Iidaa kan hawaasti waliin kabajatu: guyyaa laguu akkuma aadaa hawaasaa keessatti ibsamu;afuura /ayyandaa keessa namaatti namatti dhagahamufi nama gajeelcha yn nama eega jedhamee amanamu

bara@ jabana: golga yeroo kan akka barkumee; barkurnee: geeddarama jeroo kan akka bara haraafi moofaa jidduu jiru;yeroo hiniraanfatamne kan akka bara torbaatamii torba bara waraan calanqoofi Waraana aanulee.

barbaaduu@ soquu: sakkatta'uu; iyyaafachuu: rukutuu kachachaluu; duukaa bu'uu

barnootaa@beekumsa barumsa irraa argatan:manni barnootaa bakka beekumsaati;barmoota:beekumsa daa'imni oggaa shan guunnaan mana barnootaa galchanii barumsa qulqullina qabu akka argatu gochuudha

beekamaa@beekamtii kan qabu:inni ogummaa harkaan beekamaadha;ulfaataa:duuressa kabajamaa

uummata@nama baay'ee:ilmaan namaa;hawaasa:Oromoon uummata addunyaa keessaa is tokko

boqonnaa@ kallattii jireenya haaraa, kutaa kitaabaa bakka yaanni ijoo gara yaada ijoo biraatti otoo hince'iin mul'atu, dalagaan erga raawwatanin booda haara galfii taasifamu, yeroo turtii samisteeaa manni barumsaa cufamee kaasee hanga samiteera itti aanutti

caffee@ lafa bichaan ciiisaa yeroo gannaa keessatti marguufi deedicha horiif hintaane: yaa'ii miseensota gadaa kan akka caffee Oromomiyaa; marga jiidhina qabu kan yeroo bara haarawaa ykn jila hawaasaa biro mana keessa hafan

cancala@ meeshaa fardaa keessaa isa tokko:rakkoo;hidhaa:dararama akkasumas gabrummaa: meeshaa fardaa sibiilairraa hojjate farad arkisuufi liiqamuuf gargaaru.

daakuu@midhaan hororame marqaa: buddeena ykn nyaata biraa hojachuuf qophaa'e;midhaan baaburiin ykn dhakaan hororuu ykn bulleessuu:gosa ispoortii bishaan keessaa daddaaquu;haleeluu ykn tumeen gargar facaasuu ykn haalaan loluu

dhaluu@ dahuu: oofkaluu; hiikkamuu

dhiiga@ biluu yeroo horiin gorra'aman dhangala'u: firummaa; garalaafummaa agarsiifamuu qaba.

dubbisuu@ nama nagaa gaafachuu: kitaaba qayyabachuu; du'a gahii deemuun jajjabeessu: bilbila keessaan haasawuuu

eeguu@ horiin ykn qabeenyaan akka nama jalaa hinbanne to:achuu:yeroo murtaa'aaf dhaabbatanii nama bakka tokko dhaqe.dhufe simachuu; mirkaneeffannaa yeroo tumaa gumaa /gadaa eebbisanii ittin cimsuu,

gaarii@ dansa: shaggaa; misha

gabbataa@ nama furdaa ykn ulfaatinaan madaala kaasu ykn borcii qabu: lafa biyyee isaa kosii qabuu irraa kan ka'e hoomisha gaarii kennu,

gadaa@golga yeroo waggaa saddeetitti qoodamu: sirna bulchiinsa hawaasaa dhimmoota siyaasaafi hawaas dinagdee hammmate; maqaa miseensota shaman ykn gogeessa gadaa kan akka michillee: roobalee; duuulo: horata; bitrmajii jedhamanii beekaman umrii 40-48 kan namni tokko aangoo gadaafudhatee ittiin bulchuu danda'u.

galaana@ qabeenyaa ykn addunyaa: qaama bishaanii kan dirrisee yaa'u ykn ciisu; tuuta namootaa kan ijji wal hingeenye

galchaa@ matadeebii: yeroo horiin gara qeyeetti dhufan; horii gara foonaatti oofuu ykn maallaqa baankiitti naquuf namoota affeeruu ykn kabajaan gaafachuu

Gogaa @muka jiidhina hinqabne: itilee ykn kalloo kan dhifame; qullaa kan akka buna gogaa-aannan malee: nama waa gadi hinhiixanne; otoo waa'e hinciniiniin kan akka afaan gogaa manaa bahuu-otoo waa hindhamdhamiin manaa bahuu.

haqa@ dugaa: mirkana; galfata ragaa sirriii

ilbiisa@ maqaa bineeyyii waliigalaa: wanta tuffatamu;wanta hoffaa kan akka galabaa kan madaala hingaafne

irreecha@ marga jiidhaa: maallaqa yeroo gumaa gumaan bahe akka mallattoo eebbatti itti fayyadaman; meeshaa ulfoo fakkoomii hormaata lubbuutiin bakka bu'u: ayyaaneffanaalee Waaqeffannaa sadarkaa maatii; gosaafi sadarkaa guutuu biyyaatti kabajamu.

jabaa@ muka yeroo gubaa kan ibissi isaa yeroo hedduuf ture: nama cimaa kan yoo dhaane malee hindhaanamne

kaawachuu@ meeshaalee kan akka huccuu: kofoofi gonfoo hufachuu; maallaqa ykn qarchii

mana baaniitti galfachuu: ambaa ykn qabeenyaa yeroo hamaafi umrii dullumaaf guduunfachuu; natti haa hafu jedhanii dubbii bulfachuu

karra@ horii bitimaa guutuu: cufaa foonaa horii ykn dallaa; idda dhaloootaa hanga balbala shaniitti jiru

kennuu@ hiixachuu: kormaykn horii biroo ayyaanaaf ykn afdaariik qaluu;gorra'uu: meeshaalee kan akka aalbeemii suuraafi kannee biraa gyyaa eebba nama badhaasuu; maallaqa hojii tolahoolmaaf buusii taasisii

lafa@dachee haadha margoo: garabaldhummaa ykn nama obsa qabu, bakka bobbaa ykn qonnaa ykn tikaa kan namoonni irraa hoomishatan ykn irraa deechifatan

lafee@qaama foon itti margu: oolmaa ykn harkadeebii; ittoo diimaa keessaa wanta qorkan

maddaa@ bishaan xuruuree yaa'u: bakka burkaa bishaan xuruuree yaa'uu; bu'uura idda dhaloota namaa fkn kan akka Kuushiifi walaabuu

naanniga@ rooba bifa harfixootiin bubbeen asiifi achii hoofu: bakka eela dhugaatii bishaanii hori; eebba warri ayyaantuu yeroo ayyaaneffannalee kora ayyaanaa: kudharfaniifi wadaajaa nama eebbisan

nagaa@ wanta guutuu kan cabaa hinqabne: haala tasgabbiifi naamusaa quubsaa; fayyaalessaa ooltee bultee namoota jidduu jiruu

Odaa@ gosa mukaa dameefi jirma heddu qabu: maqaa namaa fakkoommii idda dhalootaa heddummaachuu bakka bu'u; idda maqaalee wiirtuu tumaa gadaa waliigalaa kan akka Odaa Namee: Odaa Bultum

qaalii@ gatii haalaan ol ka'aa ykn mi'aawaa: wanta gita hinqabne kan akka lubbuu tokkittii ilaalamu

qaalluu@ waa beekaa kan moora ilaalu: ayyaantuu kan galma ayyaantummaa qabu/du; kan itti wareeggataniifi galchaa itti galfatan: kan ilmoo itti ammachifatan; raagaa/duu kan dhaha ilaalu/tu

qaama'uu@ wantoota bullaa'aa ta'an kannee akka sukkaraa afaanitti of darbanii xuxxachuu: baala jimaa kukkutanii afaan of kaa'anii dhuubbachuu

qaana'uu@ saalfachuun waan dubbatan wallaaluu: safuu waan hintaane addaan baasuun eeggachuu; yakka ykn safuu cabsuun kan qa'e itti gaabbuu