ENHANCING EFFECTIVENESS OF AFAAN OROMO INFORMATION RETRIEVAL USING LATENT SEMANTIC INDEXING AND DOCUMENT CLUSTERING BASED SEARCHING

MSc. THESIS

BELETE BOGALE

SEPTEMBER 2020

HARAMAYA UNIVERSITY, HARAMAYA

Enhancing Effectiveness of Afaan Oromo Information Retrieval Using Latent Semantic Indexing and Document Clustering Based Searching

A Thesis Submitted to Department of Information Science,

Post Graduate Program Directorate

HARAMAYA UNIVERSITY

In Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE IN INFORMATION SCIENCE

Belete Bogale

September 2020

Haramaya University, Haramaya

HARAMAYA UNIVERSITY

POSTGRADUATE PROGRAM DIRECTORATE

I hereby certify that I have evaluated this Thesis entitled **Enhancing Effectiveness of Afaan Oromo Information Retrieval Using Latent Semantic Indexing and Document Clustering Based Searching** that prepared by Belete Bogale under my advice and I recommend that it to be submitted as fulfilling the thesis requirement.

Dr.Million Meshesha (PHD) ________________ ___________

Principal Advisor: signature date

As a members of the Board of Examiners of the MSc Thesis Open Defense Examination, I certify that I have evaluated the Thesis prepared by Belete Bogale and examined the candidate. I recommend that the thesis be accepted as fulfilling the Thesis requirements for the degree of Master of Science in Information Science.

________________________ ____________________ ________________

Chairperson Signature Date

________________________ ____________________ ________________

Internal Examiner Signature Date

________________________ ____________________ ________________

External Examiner Signature Date

Final approval and acceptance of the Thesis is contingent upon the submission of its final copy to the Council of Graduate Studies(CGS) through the candidate’s department or Postgraduate committee (DGC or SGC).

DEDICATION

This work is dedicated to my parents. For their endless love, support and encouragement throughout my life.

STATEMENT OF THE AUTHOR

By my signature below, I declare and affirm that this Thesis is my own work. I have followed all ethical and technical principles of scholarship in the preparation, data collection, data analysis and compilation of this Thesis. Any scholarly matter that is included in the Thesis has been given recognition through citation. This Thesis is submitted in partial fulfillment of the requirement for an MSc degree at Haramaya University. The Thesis is deposited in the Haramaya University Library and is made available to borrowers under the rules of the library. I solemnly declare that this Thesis has not been submitted to any other institution anywhere for the award of any academic degree, diploma or certificate. Brief quotations from this Thesis may be used without special permission provided that accurate and complete acknowledgement of the source is made. Requests for permission for extended quotations from, or reproduction of this Thesis in whole or in part may be granted by the Head of the School or Department when in his or her judgment the proposed use of the material is in the interest of scholarship. In all other instances, however, permission must be obtained from the author of the Thesis.

Name_________________________ Signature______________________

School/ Department_______________ Date__________________________

BIOGRAPHICAL SKETCH

The author was born in Arsi Zone, Oromia region on Dec 21, 1994 from Ato Bogale Gurmu and W/ro Kefene Guluma. He attended his elementary (1-8) at T/waji elementary school and his secondary (9-10) Chilalo Terara secondary school and preparatory at Assella preparatory school. He joined Haramaya University in 2012 and graduated with Bachelor degree in Information Science in July 2014. After graduation, he was employed at Haramaya University as Graduate Assistant and served there for about one year. Then in 2015, he has joined the Postgraduate Program at Haramaya University for regular program and continued his studies in Information Science.

ACKNOWLEDGEMENT

First of all, I would like to thank one above all GOD for every success in my life, for giving me wisdom and strength. Secondly, I would like to say as the great scientist Albert Einstein said "If I have seen further than others, It is because I have stood upon the shoulder of giants." Intended, I would like to thank my advisor Dr. Million Meshesha, for his critical comments on my work, from the beginning to the end of the study, and his patience in helping me to complete my work. Thirdly it is my pleasure to thanks all my family, department staff, and other people for helping me when I was working on this study. Finally, I offer my regards and blessings to all of those who supported me in any aspect during the completion of this thesis as well as expressing my apology that I could not mention all.

LIST OF ACRONYMS AND ABBREVIATIONS

AOIRS	Afaan Oromo Information Retrieval System
CLIR	Cross Language Information Retrieval
IR	Information Retrieval
IRS	Information Retrieval System
LSI	Latent Semantic Indexing
LC	Lexical Chain
MR	Relevance Model
NLP	Natural Language Processing
SVD	Single Value Decomposition
VSM	Vector Space Model

Table of Contents

LIST OF TABLES

LISTS OF FIGURES

ABSTRACT

This research work comes up with Latent Semantic Indexing and Document Clustering based searching for Afaan Oromo documents. It intends to apply LSI and K-means clustering to handle the semantic structure of words in documents. This mainly consists of three components; indexing, clustering, and searching. Latent Semantic Indexing (LSI) model is a concept based retrieval method that exploits the idea of a vector space model and singular value decomposition. On the other hand, document clustering was investigated for improving the performance of information retrieval system. Document clustering is an issue of measuring similarity between documents and grouping similar documents together. K-means clustering was used to cluster the document using the Singular Value Decomposition (SVD) matrix. Then, the retrieval process is further refined by making a similarity measure between the query vector and cluster centroid vectors. IR pre-processing for tokenization, normalization, stop word removal, and stems were used for selecting index and query terms. Finally, a comparison is made between the SVD model with K-means clustering, VSM and SVD model. The performance evaluation of the system was performed by using a selected set of documents and queries. The experimental result showed that the proposed prototype registered on average 70% recall, 80% precision, and 72% F-measure. Therefore, it indicated that the proposed method (SVD with Kmeans) achieved significant improvement compared to the VSM and SVD model. Nevertheless, the performance of the system is greatly affected by the statistical extraction of synonyms and polysemy, mis-clustering, standard corpus, and stemming which need further research.

1. INTRODUCTION

1.1. Background of the Study

Information Retrieval is defined as finding relevant documents that satisfy the information needs of users from an unstructured large collection (Christopher *et al.*, 2008). Information Retrieval is one of the major branches of the Information Science discipline (Moukdad, 2003). The trend in information storage and retrieval can be traced back to 2000 BC when people of Sumerians chosen a special place to store clay tablets with cuneiform inscriptions (Christopher *et al*., 2008). After they understand that their work is efficient in the use of information, they developed a special categorization system that identifies every tablet and its content.

One of the major evolutions in Information Retrieval is the invention of the print machine in 1450 A.D (Judit and Gutman, 2003). A German goldsmith Johannes Gutenberg invented the first movable printer, thousand years later after the Chinese invented paper which provides means for disseminating and storing knowledge. Gutenberg‗s aim was allowing direct access to mass information that was contained in the Bible and other scholarly works.

At the beginning of the 1990s, a single fact changed once and for all the perceptions towards Information Retrieval was the introduction of the World Wide Web (Baeza-Yates and Ribeiro-Neto, 2000). The Web is becoming a universal repository of human knowledge and culture which has allowed unprecedented sharing of ideas and information on a scale never seen before (Baeza-Yates and Ribeiro-Neto, 1999).

Information Retrieval has two main subsystems (Deerwester *et al*., 1989), Indexing, and Searching. Indexing is an offline process of representing and organizing large document collection using indexing structure such as Inverted file, sequential files, and signature file to save storage memory space and speed up searching time. Searching is on the other hand an online process of relating index terms to query terms and return relevant hits to the user's query.

Nowadays with the advent of digital databases and communication networks, huge repositories of textual data have become available to a large public (Thomas, 1999). Although the use of elaborate ergonomic elements like computer graphics and visualization has proven to be extremely fruitful to facilitate and enhance information access (Thomas, 1999). The need to store and retrieve written information became increasingly important over centuries, especially after the inventions of a scientific paper (Christopher *et al*., 2008).

General search engines on the Web such as Google, Yahoo, and Microsoft Network are among the popular tools to search for, locate, and retrieve information, and their use has been growing fast (Moukdad, 2003). These engines handle English queries more or less in the same way, but their handling of non-English queries is different from how these queries are handled by non-English search engines that were designed for specific languages. Most general search engines enable people to search using the English language. But people usually want to get the information they need in the language that they can understand. In Ethiopia more than 80 languages are there, from these languages Afaan Oromo is the most popular and widely spoken language by Oromo people (Ibrahim, 2015).

Today, in Ethiopia, Afaan Oromo is among the major languages that are widely spoken and it is considered to be one of the five most widely spoken languages from thousand languages of Africa (Abera, 2009). Afaan Oromo is also relatively distributed within Ethiopia and some neighboring countries like Kenya and Somalia (Ibrahim, 2015). Afaan Oromo is part of the Lowland East Cushitic group within the Cushitic family of the Afro-Asiatic phylum (Melvyn, 2009), unlike Amharic (an official working language of Ethiopia) which belongs to the Semitic language family. Afaan Oromo has a very rich morphological variation like other African and Ethiopian languages (Ibrahim, 2015). Therefore Afaan Oromo needs improved retrieval system.

In fact, there is no perfect Afaan Oromo Information Retrieval system that can take into account semantically related words during searching (Kula *et al*., 2007). There are two sides' issues regarding the semantic relation of keywords; as broadly classified into synonymy and polysemy (Thomas, 1999). Synonymy describes the fact that there are many ways to refer to the same object. Users in different contexts or with different needs, knowledge, or linguistic habits will describe the same information using different terms (Thomas, 1999). Polysemy refers to the general fact that most words have more than one distinct meaning (Thomas, 1999). This can be when the same term is used in different contexts or by different people.

There are various methodologies and techniques applied to come up with an effective and efficient way of retrieving documents from a very large and unstructured corpus (Recardo, 1999). Some of the scientific approaches to resolving the problems are ontology-based Information Retrieval System, latent semantic indexing, query expansion, and reformulation.

Latent Semantic Indexing (LSI) is one method that helps to overcome the problems of lexical matching (Christopher, 2009). It assumes that there is a latent structure in word usage that words are partially obscured by variability in word choice. On the other hand, most of the current document clustering algorithms are consider the semantic relationships which produce clustering results (Neepa and Sunita, 2012). This clustering plays an important role in providing intuitive navigation and browsing mechanisms by organizing such large amounts of information into a small number of meaningful clusters (Andreas *et al.*, 2003).

Here this study attempt has been done to design and develop an enhanced Afaan Oromo Information Retrieval System; mainly consisted of three components; latent semantic indexing, K-means clustering, and searching. Latent Semantic Indexing is an efficient vector space retrieval approach, uses the Singular Value Decomposition (SVD) technique to reduce the rank of the original term-document matrix TFIDF. The low-rank approximation yields a new representation for each document in a collection. on the other hand document clustering was investigated for improving the precision and/or recall in information retrieval systems by automatically grouping documents that belong to the same topic in order to provide user's browsing of retrieval results (Adrian Kuhn *et al.*, 2006). More recently, clustering has been proposed for use in browsing a collection of documents or in organizing the results returned by a search engine in response to a user's query (Ramampiaro and Paulsen, 2009). From flat clustering techniques, K-means is widely used in document clustering. K-means is based on the idea that a center point can represent a cluster.

1.2. Statement of the Problem

Afaan Oromo is among the major languages that are widely spoken in Ethiopia and it has its own writing system (Abera, 2009). With regard to the writing system, Qubee (Latin-based alphabet) has been adopted and became the official script of Afaan Oromo since 1991 (Abera, 2009). Currently, Afaan Oromo is an official language of Oromia Regional State and used as an instructional media for primary and secondary schools of the region (Ibrahim, 2015). Furthermore, literature, newspapers, magazines, educational resources, news, online education, books, entertainment Media, videos, pictures, official documents, and religious writings are available on the internet and offline. Therefore, these huge amounts of information available in an electronic format both on the internet and offline need a potentially powerful Information Retrieval System with good performance.

Accordingly, there were several research works done in Afaan Oromo Information Retrieval to enhance the effectiveness of the system, especially on the text retrieval system. From those, Wegari, (2017) conducted a rule-based root generation system for Afaan Oromo. In this investigation, a rule-based method was used to develop a root generation system for Afaan Oromo and is mainly used as a starting point to develop a complete morphological analysis and information retrieval for the target language. Berhan, (2019) Explored the potential applications of Latent Semantic indexing approach in Afaan Oromo text retrieval. The goal of the study was to examine the benefits of LSI text retrieval approach for Afaan Oromoo text retrieval. LSI was used to creat a concept space such that this concept space is much smaller than the word space (mapping of terms against documents). Melkamu, (2017) integrated query expansion for enhancing the effectiveness of the Afaan Oromo text retrieval system. Melkamu applied the designed query expansion for the Afaan Oromo information retrieval system using WordNet that was constructed as a reference for identifying the sense and meaning of the user's query based on word sense disambiguation by semantic similarity measure. Recently, Ashenafi, (2019) has made attempts to extend the application of word sense disambiguation based query expansion. In this study word sense disambiguation based query expansion approach was used to determine the senses of words in queries by using Afaan Oromoo lexical resource.

However, these research works were more or less not consider the semantic nature of the language. Having words in the information by itself does not meet the user's information needs. It happened because of different reasons but the major one is not properly addressed the semantic natures of the language. The systems simply represent documents and queries as a "bag-of-words" with its weight (Bo-Yeong, 2013), when users give some query the system tried to match those documents which have terms in the query and the documents will be automatically retrieved but not otherwise. Semantic and contextual understanding of the user information need and information in the collection has to be well-managed to return an effective result to the user. There are several problems tied to this simple "bag-of-words" representation of documents (Bo-Yeong, 2013). One of them is semantic indexing (polysemy and synonymy) and the other is document clustering (some documents are not retrieved because they do not share terms with the query).

Hence the aim of this study is to design enhanced Afaan Oromo Information Retrieval system using semantic indexing and document clustering for organizing the Afaan Oromo document corpus based on their similarity for effective searching to retrieve relevant documents.

To this end, this study attempts to explore and answer the following research questions:

- What is the suitable text preprocessing applied to document corpus for semantic indexing document clustering based searching?
- What is the best approach for semantic indexing and document clustering based searching to enhance Afaan Oromo Retrieval system?
- To what extent the proposed prototype improve the performance of the Afaan Oromo Information Retrieval system?

1.3. Scope and Limitation of the Study

Semantic indexing and document clustering based searching was includes; semantic indexing, document clustering, and searching Afaan Oromo textual document from the corpus. To do these, the study was carried out using Latent Semantic Indexing (LSI) and K-means clustering. LSI is a variant of the vector retrieval method or SVD that exploits dependencies or "semantic similarity" between terms. LSI became famous as one of the first IR techniques exhibiting effectiveness in dealing with the problems of synonymy, polysemy, and dimension reduction (Frakes and Baeza-Yates, 1992). It helps with multiple meanings because the meaning of a word can be conditioned not only by other words in the document but by appropriate words in the query not used by the author of a particular relevant document. K-means clustering which basically included (dataset preprocessing, labeling for clustering, and formation of clusters). The document that is related to each other in a particular cluster is placed under that cluster. Thus, based on the relatedness among the documents and the initial cluster of the clusters were formed.

The semantic document clustering based Afaan Oromo retrieval system has been developed and tested using Afaan Oromo news article documents that were collected from Oromia Broadcast Network (OBN). As a result of the time factor, it was difficult to prepare all relevant Afaan Oromo documents. Therefore limited corpus were selected to developing and evaluating the performance of this study. On the other hand, data types such as image, video, audio, and graphics are out of focus of this study.

1.4. Significance of the Study

The semantic document clustering based Afaan Oromo retrieval system helps Afaan Oromo information seekers to retrieve information needed and improve its performance. The system retrieves as per the information needs of the user by retrieving documents which have similarity to the query formulated by the user rather than exact terms match. The primary and target beneficiaries of this are Afaan Oromo native speakers who can read, understand, and fluent enough to produce queries to search for the information they need. Thus, users can able to search and retrieve Afaan Oromo documents. In addition, the computational cost of the retrieval system was also reduced. This happened during the system looking for a query result, it is not going through the whole documents instead it searches only from a specific cluster which depends on specific relation and dimension reduction. The system was also contributes to future researchers in the area of Information Retrieval, especially the ontology-based Information Retrieval system. Generally, the research outcome gave benefits to individuals, groups, and future researchers.

1.5. Objective of the Study

1.5.1. General Objective

The general objective of this study is to design enhanced Afaan Oromo information retrieval using semantic indexing and document clustering based searching.

1.5.2. Specific Objectives

In order to accomplish the general objective the following specific objectives were formulated:

- To perform text preprocessing and extract Afaan Oromo lexical texts.
- To identify and experiment better semantic indexing and document cluster based searching algorithm.
- To design the architecture and develop the proposed information retrieval system for Afaan Oromo.
- To evaluate the performance of the proposed system using Information Retrieval effectiveness measures (such as precision, recall, and F-measure).

2. LITERATURE REVIEW

2.1. Overview

This section presents an overview of Information Retrieval System, indexing, document clustering, searching and Afaan Oromo linguistic feature. This is broadly divided into three sections. The first section discusses concepts related to IR text preprocessing, indexing, components of the IR system, and document clustering. Concepts like extraction semantics of the term and term weighting are also included. In the second section, the features of the Afaan Oromo writing system related to Information Retrieval is briefly discussed. The Afaan Oromo alphabets, numbers, punctuation marks, morpheme, and parts of speech of Afaan Oromo are also introduced. The third section focuses on research works related to the Afaan Oromo Information Retrieval system.

2.2. History of Information Retrieval System

The need to store and retrieve written information became increasingly important over the centuries, especially inventions like scientific paper (Christopher *et al*., 2008). Soon after computers were invented, people realized that they could use the machine for storing and mechanically retrieving large amounts of information. In 1945 Vinegar Bush published a groundbreaking article titled “As We May Think” that gave birth to the idea of automatic access to large amounts of stored knowledge. In the 1950s, this idea materialized into more concrete descriptions of how to archives the text could be automatically searched (Christopher *et al*., 2009). Several key developments in the field happened in the 1960s. The other most notable was the works of Gerard Salton and his students; they were laid the foundation for the retrieval system. They developed a Retrieval System and formulated a technique to evaluate the IR system, which is using still today (Singhal, 2008).

On the other hand the perceptions towards Information Retrieval was greatly changed due to introduction of the World Wide Web at the beginning of the 1990s and Information Retrieval becoming the dominant form of information access, overtaking traditional database-style searching (Christopher *et al*., 2008). It is defined as the process of looking for material (usually documents) of an unstructured nature that satisfies an information need of the user from within large collections (usually stored on computers) (Christopher *et al*., 2009). IR can also cover other kinds of data and information beyond textual documents; it could be the image, multimedia, or other data.

An Information Retrieval system does not inform (i.e. change the knowledge of) the user on the subject of his inquiry (Rijsbergen, 1979). It merely informs on the existence (or non-existence) and about documents relating to the user request. However, the computer is not likely to have stored the complete text of each document in the natural language in which it was written (Rijsbergen, 1979). It will have, instead, a document representative that may have been produced from the documents either manually or automatically.

Information Retrieval applications concerning textual documents use automatically generated free text index terms (post-coordinated), which are weighted by the statistical frequency of terms in documents and collections (Marco and Kathleen, 2009; Baeza-Yates *et al*., 2004). For many authors, the purpose of an Information Retrieval (IR) system is to process files of records and requests for information and identify and retrieve from the files certain records in response to the information requests (Marco and Kathleen, 2009).

2.3. Information Retrieval System Models

Potsdam, (2007) noted that, a retrieval model specifies representations used for documents and queries, and how they are compared. An Information Retrieval model is a formalization of the way of thinking about Information Retrieval (Potsdam, 2007). Such formalism can be defined in form of algorithms, mathematical formulas, etc. A document model is a formal document representation used by an Information Retrieval model to obtain document similarities (Potsdam, 2007). The three most commonly known models are the vector space model, the probabilistic models, and the inference network model (Recardo, 1999) as discussed below one after the other.

2.3.1. Vector Space Model

The Vector Space Model (VSM) has been a standard model of representing documents in Information Retrieval system for almost three decades (Salton and McGill, 1983; BaezaYates and Ribeiro-Neto, 1999). In this model, the text is represented by a vector of terms from the documents and query (Recardo, 1999).

A document collection containing a total of ***n*** documents described by ***m*** terms is represented as a term by document matrix in table 2.1. Each row of this matrix is a term vector and each column of this matrix is a document vector. The element at row *i*, column ***j***, is the weight of term *j* in the document (Prabhakar *et al*., 2008; Baeza *et al*., 1982).

Table 2.1 term-document matrix (Prabhakar *et al*., 2008)

Term list	**Doc1**	**Doc2**	**Doc3**	**Doc4**	**…**	**Doc n**
Term1	w_{11}	w_{12}	w_{13}	w_{14}	…	w_{1n}
Term2	w_{21}	w_{22}	w_{23}	w_{24}	…	w_{2n}
Term3	w_{31}	w_{32}	w_{33}	w_{34}	…	w_{3n}
Term4	w_{41}	w_{42}	w_{43}	w_{44}	…	w_{4n}
Term5	w_{51}	w_{52}	w_{53}	w_{54}	…	w_{5n}
…	…	…	…	…	…	…
Term m	w_{m1}	w_{m2}	w_{m3}	w_{m4}		w_{nm}

Vector Space Model has been introducing a term-weight scheme known as ***tf-idf*** weighting. These weights have a term frequency (***tf$_{i,j}$***) factor measuring the frequency of occurrence of the terms in the document or query texts and an inverse document frequency (***idf$_i$***) factor measuring the inverse of the number of documents that contain a query or document term (Sani *et al*., 2014; Baeza *et al*., 1982);

$$Wi,j = \ tfi,j\ x\ idfi \qquad (2.1)$$

$$Where \qquad tfi,j\ = \frac{freqi,j}{\max\limits_{freqij}} \qquad (2.2)$$

$$And \qquad idfi\ \ = log\ (\frac{N}{ni}) \qquad (2.3)$$

freq$_{i,j}$ is the number of occurrences of terms ***t$_i$*** in document ***d$_j$***, *N* is the number of documents in a collection, and ***n$_i$*** is the document frequency for term ***t$_i$*** in the whole document collection.

To assign a numeric score to a document retrieved by a query, the model measures the similarity between the query and vector created by vocabulary in the index versus documents in the corpus (Singhal, 2008). Typically, the angle between the two vectors is used as a measure of divergence between the vectors. The similarity can be measured using, such as cosine similarity (dot product) (Singhal, 2008). When using a search engine the user wants to retrieve relevant documents (Rosell, 2006), he or she gives some keywords, a query, as input, this query, gets represented in the same (or a similar) way as the texts, i.e. we get a vector ***q*** in the vector space representing the query. The idea is that the relevant texts are those closest to the query in the vector space. The most common measure of closeness or similarity is the *cosine measure*, the cosine of the angle between the query and a text (Baeza *et al*., 1982):

$$\mathrm{sim}(\mathrm{dj},\mathrm{q}\,) \; = \frac{\mathrm{dj.q}}{||\;\mathrm{dj}||\;||\mathrm{q}||} \quad = \frac{\sum_{\mathrm{j=1}}^{\mathrm{t}} \mathrm{W_{qj}} * \mathrm{w_{dij}}}{\sqrt{\sum_{\mathrm{j=1}}^{\mathrm{t}} \left(\mathrm{w_{qj}}\right)^{2} * \sum_{\mathrm{j=1}}^{\mathrm{t}} \left(\mathrm{w}_{\,\mathrm{dij}}\right)^{2}}} \tag{2.4}$$

where documents and queries are represented as vectors: d_j = ($w_{1,j}$, w_{2j}, … … … $w_{t,j}$)

and q= ($w_{1,j}$, w_{2j}, … … … $w_{t,q}$)

2.3.2. Probabilistic Model

Probabilistic IR models estimate the probability of relevance of documents for a user queries (Bruce, 1995), this estimation is the essential part of the model, and this is a point where probabilistic models different from others. It is based on the general principle that documents in a collection should be ranked by decreasing the probability of their relevance to a query or user information needs (Bruce, 1995). The probabilistic models work based on basic probability notation, the probability of relevance for the document in the log (Fuhr and Buckley, 1991). The parameters of the models relate to elements of the underlying representations (Fuhr and Buckley, 1991). The probabilities are estimated on the frequency of terms in each document, and they make use of user-determined relevance judgments (Salinas, 2009). In probabilistic relevance, the index term weight variables are all binary that a user query *q* is a subset of index terms (Sani *et al*., 2014). Let *R* be the set of documents known to be relevant and Let *R*' be the complement of *R* (that is, the set of non-relevant documents) (Sani *et al*., 2014), P(R/d) be the probability that the document *d* is relevant to query *q* and *P (R'/d)* be the probability that d is non-relevant to *q*. then the similarity sim (*d,q*) of the documents *d* to the query *q* is defined by the ratio of relevant document to non-relevant for a given query (Sani *et al*., 2014; Baeza *et al*., 1982).

$$sim(d,q) \; = \frac{P\left(\frac{R}{d}\right)}{P\left(\frac{R'}{d}\right)} \tag{2.5}$$

By using Bayes' rule equation defined above can be expressed as follows;

$$sim(d,q) \; = \frac{P\left(\frac{d}{R}\right)P(R)}{P\left(\frac{R'}{d}\right)P(R')} \tag{2.6}$$

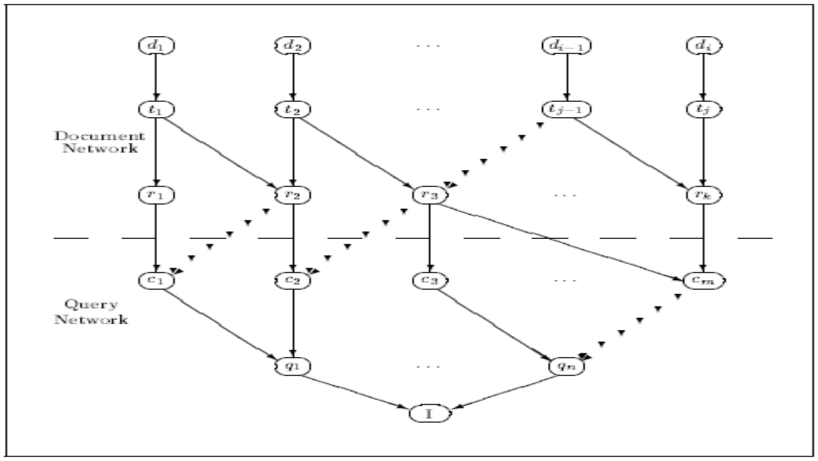
P(d/R) stands for the probability of randomly selecting the document *d* from the set *R* of relevant documents. *P(R)* is a probability document that is randomly selected from the complete set of documents that is relevant. The meaning attached to *P(d/R')* and *P(R')* is analogous and complementary.

2.3.3. Inference Network Model

Inference Network Model attempts to model document retrieval as an inference process (Singhal, 2008). In the simplest implementation of this model, a document instantiates a term with a certain strength, and the credit from multiple terms is accumulated given a query to compute the equivalent of a numeric score for the document (Recardo, 1999). From an operational perspective, the strength of the instantiation of a term for a document can be considered as the weight of the term in the document, and document ranking based on the term strength of the documents. The basic network model for information retrieval consists of two-component networks (Robert, 1991); a document network, and a query network. The document network represents the document collection using a variety of document representation schemes. The document network is built once for a given collection and its structure does not change during query processing. The query network consists of a single node that represents the user's information need and one or more query representations that express that information need. A query network is built for each information need and is modified during query processing as existing queries are refined or new queries are added in an attempt to better characterize the information need.

Document and query networks are joined by links between representation concepts and query concepts (Robert, 1991), as shown in figure 2.1. All nodes in the inference network are binary valued and take on values from the {*false, true*} (Bonald and Bruce, 2004)

Fig. 2.1 Basic inference network model (Robert, 1991)



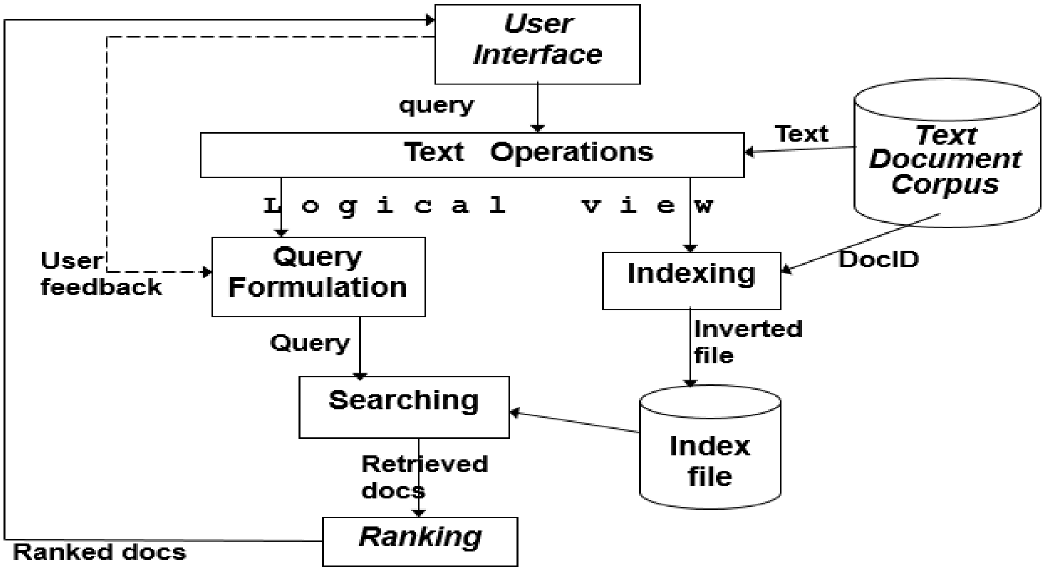
The document network consists of document nodes (d_i*'s*), text representation nodes (t_i*'s*), and concept representation nodes (r_k*'s*). Let D be the set of documents, T be the set of text representations, and R be the set of representation concepts, where the cardinality of these sets in n_d, n_t, and $\mathrm{n_r}$, respectively, then the event space represented by document network is $E_d = DxTxR$ (Donald and Bruce, 2004). Since all propositions are binary-valued, the size of the event space is *2nc. 2nt.2nr* and each representation node contains a specific of the conditional probability associated with the node given its set of parent text nodes (Robert, 1991). This specification incorporates the effect of any indexing weights (for example, term frequency for each parent text) or term weights (for example, inverse document frequency) associated with the representation concepts.

The query network is an "inverted" directed acyclic graph with a single leaf corresponds to the event that an information need is met and multiple roots that correspond to the concepts to the concepts that express the information need (Singhal, 2008). A set of intermediate query nodes may also be used in cases where multiple query representations are used to express the information need. These nodes are a representation of convenience; it is always possible to eliminate them by increasing the complexity of the distribution specification at the node representing the information need (Singhal, 2008). If we let C represent the set of query concepts and Q represents the set of queries where n_c, and n_q are the cardinalities of these sets, then the event space represented by the query network is $E_q = CxQxI$. Since we can always eliminate query nodes $|E_q| \leq 2n_c+1$. The event space represented by the entire inference network is then E_dxE_q.

2.4. Information Retrieval Process

Information Retrieval can be subdivided in many ways; it seems that there are three main areas of research which between them make up a considerable portion of the subject (Rijsbergen, 1979). They are content analysis, information structures, and evaluation. Briefly the first is concerned with describing the contents of documents in a form suitable for computer processing; the second is exploiting relationships between documents to improve the efficiency and effectiveness of retrieval strategies; the third with the measurement of the effectiveness of retrieval. The term information structure covers specifically a logical organization of information, such as document representatives, for Information Retrieval (Rijsbergen, 1979). Figure 2.2 presents the generic architecture of an information retrieval system with indexing and searching phases.

Fig. 2.2. Architecture of a textual IR system (Ceri *et al.*, 2013).



2.4.1. Text operations

Textual IR exploits a sequence of text operations that translate the user's need and the original content of textual documents into a logical representation more amenable to indexing and querying (Ceri *et al.*, 2013). Textual operations translate the user's need into a logical query and create a logical view of documents (Ceri *et al.*, 2013). Text operations are the task document preprocessing which is a stage before a document model construction when data is analyzed, transformed, and filtered from the document content (Potsdam, 2007).

During the preprocessing phase, some important text operations can be performed (Rijsbergen, 1979; Christopher *et al.*, 2009). It is performed to control the size of the vocabulary. Some of the major text operations are the following.

Lexical analysis: is a process of collecting tokenized terms that needs some other operation (Singhal, 2008; Christopher *et al.*, 2009). It is the process of converting a stream of characters (the text of the documents) into a stream of words (the candidate words to adopt as index terms) (Christopher *et al.*, 2009). Thus, one of the major objectives of the lexical analysis phase is the identification of the words in the text. It also incorporates a sort of text cleaning process, such as the conversion of abbreviations and acronyms into their full text, conversion of cases, removal of numbers and symbols (Matsumura *et al.*, 2002), (example $, @, %, #, etc) which do not make up good index terms.

Elimination of stop words: is a process of avoiding a word which has fewer relevancies to determine the content of the document (Potsdam, 2007). Sometimes, some extremely common words and non-content bearing words have little value in selecting documents matching a user need are excluded from the vocabulary entirely. In English language, these words are most of the time articles, prepositions, pronouns, conjunctions, and others. The elimination of stopping words is a very critical task to control the size of the vocabulary (Christopher *et al*., 2009; Singhal, 2008). Stop words usually refers to the most common words in a language, there is no single universal list of stop words used by all-natural language processing tools, and indeed not all tools even used by such a list (Singhal, 2008). However, the most common stop words in English language includes; articles (a, an, the), prepositions (in, on of …), conjunctions (and, or but, if…), pronouns (I, you, them, it) and possibly some verbs, adverbs, adjectives. Since stop words have no discriminatory power, they can be removed from the list of index and query terms.

Stemming: It is the process of finding the root of the words given word variants (Christopher *et al*., 2009). In any language, they observe the occurrence of variants of the same word into little modification in its morpheme, such as prefix, suffix and circumfix (Unubi *et al*., 2017). Prefix is attached to the beginning of a root word, i.e., in a word untouch, un is a prefix. Suffix is on the other hand attached to the end of the root word i.e., in word touchable, able is the suffix. Also, infix occurs when an affix is inserted within a root word, it is used rarely i.e., spoonful can be as spoonsful and s is infix and circumfix is an affix attach to both the beginning and the end of a word, i.e., il-legal-ity, in which il as prefix and ily as a suffix.

The goal of stemming is to reduce inflectional forms and derivational forms of a word to a common base form or it is referring to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes (Cambridge, 2009) For instance:

- The word '*nation*' can be written as *nationalized, nationality, national,* and others by adding suffixes.
- am , are, is => be
- car, cars, car's, cars' => car, so this different form of the same word comes to its root term by applying stemming algorithms.

The most common algorithm for stemming English words, and one that has repeatedly been shown to be empirically very effective, is porter's algorithm (Cambridge, 2009). Porter's algorithm consists of 5 phases of word reductions, applied sequentially (Cambridge, 2009). Within each phase, there are various conventions to select rules, such as selecting the rule from each ruling group that applies to the longest suffix. For instance (Cambridge, 2009):

Rule		Example	
SSES	SS	caresses	caress
IES	I	ponies	poni
SS	SS	caress	caress
S		cats	cat

2.4.2. Term Weighting

Term weighting for the IR model is handled by statistics (Potsdam, 2007). There are three main factors of term weighting: term frequency factor, term collection frequency factor, and document vector length normalization factor. The end term weight might be constructed from all or a subset of mentioned factors. For example, the inverse document frequency assumes that the importance of a term is proportional with the number of documents the term appears in (Salton, 1983), a document that mentions a query term more often has more to do with that query and therefore should receive a higher score. Thereof, assign to each term in a document weight for term that depends on the number of occurrences of the term in the document (Christopher *et al*., 2009). This concept can be applied through assigning the weight; it is equal to the number of occurrences of term ***t*** in document ***d***. This weighting scheme is referred to as term frequency, denoted by ***tf***, and the document in its order (Christopher *et al*., 2009). The exact ordering of the terms in a document is ignored but the number of occurrences of each term is material (in contrast to Boolean retrieval). We only tried to capture information concerning the number of occurrences of each term (Christopher *et al*., 2009).

Inverse Document Frequency: It will create a problem when we make all terms are equally important; the impact would be explicitly viewed when it comes to assessing relevancy against a query (Christopher *et al*., 2009). In fact certain terms have little or no discriminating power in determining relevance. An immediate idea is to scale down the term weights with high collection frequency, defined to be the total number of occurrences of a term in the collection. The idea would be to reduce the *tf* weight of a term by a factor that grows with its collection frequency (Christopher *et al*., 2009). Document frequency of the term, *dft*, defined to be the number of documents in the collection that contain a term *t* aimed to obtain document level statistics, which deals with the number of documents containing a term (Christopher et al., 2009). Using *dft* is difficult to measure the discrimination power of the term among the documents. To overcome this problem, it is better to use inverse document frequency; this is the logarithmic function of the total number of document *N* over *dft* (Christopher et al., 2009). It is defined as **(**Baeza *et al*., 1982);

$$idf = \log N/df \tag{2.5}$$

TFIDF weighting: Combining the definitions of term frequency and inverse document frequency helps to produce a composite weight for each term in each document. The *TFIDF* weighting scheme assigns to term *t* in document *d* given by (Baeza *et al*., 1982):

$$TFIDF = TF * IDF \tag{2.6}$$

TFIDF $_{\mathbf{t,\,d}}$ assigns to term *t* is a weight in document *d* that is;

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

2.4.3. Indexing Approaches

An indexing language is a language used to describe documents and requests (Rijsbergen, 1979), the elements of the index language are index terms, which may be derived from the text of the document to be described, or maybe arrived at independently. The transformation from a document text into a representation of text is known as indexing the documents (Sajendra and Ram-Kumar, 2012). Indexing can be defined as a process that automatically collects, parse, and stores data to facilitate fast and accurate Information Retrieval (Clarck and Cormack, 1995).

Indexing techniques have been developed in order to make possible the identification of the information content of documents (be they text documents, hypermedia, or multimedia ones) (Lei Shi, 2005). It simply means pointing to or indicating the content, meaning, purpose, and features of messages, texts, and documents (Xiao, 2010).

Automatic indexing is the ability of a computer to scan large volumes of documents against a controlled vocabulary, taxonomy, thesaurus, or ontology and use those controlled terms for quickly and effectively index large document collections (Rijsbergen, 1979). Typically the indexing of a textual document is obtained through the identification of a set of terms or keywords which characterize the document content that describe the topics dealt with in the document (LeiShi, 2005). The terms included in this set have not only to be representative of the topics covered in the documents, but they also need to be distinguishing the possible way to discriminate one document against the other documents in the collection covering the same or similar topic.

There are two major approaches for the automatic indexing of text documents (Lei Shi, 2005); linguistic and statistical approaches.

2.4.3.1. Linguistic Approach

The linguistic approach involves syntactical analysis (Aurélie *et al*., 2008). It is based on the knowledge base of the language it is working on (Latifur *et al*., 2013). Therefore, it needs ontology of the language to determine the terms which describe a document. It can use semantics by extracting content bearing words from the document that is going to be indexed (Latifur *et al*., 2013). The use of ontology enables one to define concepts and relations representing knowledge about a particular document in domain-specific terms (Paralic and Kostial, 1999). In order to express the contents of a document explicitly, it is necessary to create links (associations) between the document and relevant parts of a domain model, i.e. links to those elements of the domain model, which are relevant to the contents of the document.

Ontology is a collection of concepts and their interrelationships that can collectively provide an abstract view of an application domain (Latifur *et al*., 2013). The ontology (WordNet) lexical database is now quite large and offers broad coverage of general lexical relations in the English language (David *et al*., 2013). It describes the relationship between the words in different situations. WordNet has been employed as a resource for many applications in natural language processing (NLP) and Information Retrieval (IR). Word relationships in the database are useful for NLP and IR applications, are not necessarily appropriate for a general, sometimes the WordNet can be a domain-independent lexical database (Marti, 2008; Singhal, 2008). For information retrieval purpose ontology enables the concept semantic indexing which is based language knowledge base. Semantic indexing is used improve the performance of a traditional keyword-based search, documents should be represented with their concept rather than bag-of-words'(Bo-Yeong, 2013). It assumes that there's some underlying or context structure in word usage that's partially obscured by variability in word selection (Ding, 1999). However; most of the previous works on indexing and Information Retrieval depend on lexical analysis and statistical methods. Using these techniques is difficult to abstract the semantics of the documents, since it needs the language expert (Bo-Yeong, 2013). A better approach would allow users to retrieve information on the basis of a conceptual topic or meaning of a document (Barbara, 2000). One way of expressing their commonality is to think of a searcher as having in mind a certain meaning, which he or she expresses in words, and the system as trying to find a text with the same meaning (Deerwester *et al*., 1989). Then, depends on the system representing query and text meaning in a manner that correctly reflects their similarity for the human.

2.4.3.2. Statistical Approach

The statistical approach relies on various word counting techniques and vector space model. The aim of statistical indexing is to capture content bearing words that have a good discriminating ability and a good characterizing ability for the content of a document (Cambridge, 2010).

Latent Semantic Indexing (LSI) was one of an efficient vector space model retrieval approach, uses the Singular Value Decomposition (SVD) technique to reduce the rank of the original term-document matrix (Hofmann, 1999). Theoretically, SVD, a dimensionality reduction technique, performs a term-to-concept mapping, and therefore, conceptual indexing and retrieval are made possible. The main objective is not to describe the concepts verbally, but to be able to represent the documents and terms in a unified way for exposing document-document, document-term, and term-term similarities or semantic relationship which are otherwise hidden (Hofmann, 1999). It tries to beat the problems of lexical matching by using statistically derived conceptual indices instead of individual words for retrieval.

Latent semantic indexing tries to overcome the problems of diversity in the words people use to describe the same object or concept (synonymy), and the same word often has more than one meaning (polysemy) by automatically organizing objects into a semantic structure more appropriate for information retrieval (Dumais, 1992). This is done by modeling the implicit higher-order structure in the association of terms with objects. Deerwester *et al*., (1990) tried to overcome the problems of lexical matching by using statistically derived conceptual indices instead of individual words for retrieval. LSI assumes that there is some underlying or latent structure in word usage that is partially obscured by variability in word choice (Rosario, 2000).

In the latent semantic space, a query and a document can have high cosine similarity even if they do not share any terms - as long as their terms are semantically similar in a sense to be described later (Rosario, 2000). This latent semantic space has fewer dimensions than the original space and a method for dimensionality reduction. A dimensionality reduction technique takes a set of objects that exist in a high-dimensional space and represents them in a low dimensional space, often in a two-dimensional or three-dimensional space for the purpose of visualization (Rosario, 2000).

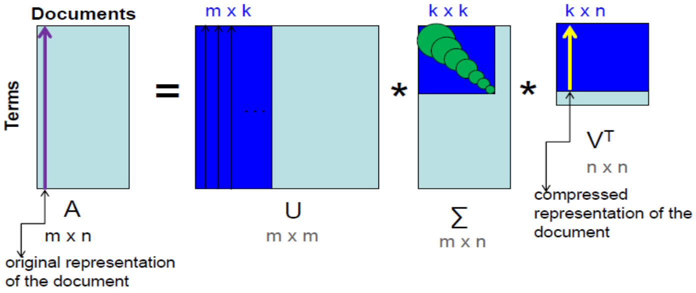
The goal of semantic indexing is to use semantic information (within the objects being indexed) to improve the quality of information retrieval. Compared to traditional indexing methods, based on keyword matching, the use of semantic indexes means that objects are indexed by the concepts they contain rather than just the terms used to represent them (Marco and Kathleen, 2009);

- A semantic index is inherently multidimensional since any combination of properties cast into a document concept can serve as an indexing element.
- As a structured concept, the indexing elements are not just attributed values but can be based on complex descriptions of related objects.
- A semantic index as a whole is highly adaptable to patterns of usage. Indexing concepts can be added or removed at will, making it very dense and precise with respect to interesting sets of individuals, or very sparse in other less interesting areas.
- Since the index is actually a set of partial descriptions for the indexed objects, lots of information can be drawn from the index alone without accessing individual descriptions at all.

Latent semantic indexing is the application of a particular mathematical technique, called Singular Value Decomposition (SVD), to a word-by-document matrix (Rosario, 2000). It is used to estimate the structure in word usage across documents (Rosario, 2000). The truncated SVD, in one sense, captures most of the important underlying structure in the association of terms and documents, yet at the same time removes the noise or variability in word usage that plagues word-based retrieval methods and it is represented as shown in figure 2.3 below.

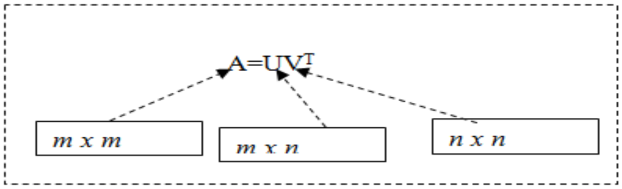
Fig. 2.3 Singular Value Decomposition based Latent Semantic Indexing

$$\mathbf{A = U\Sigma V^T}$$


Documents
Terms
=
m x k
*
k x k
*
k x n
A
m x n
original representation of the document
U
m x m
∑
m x n
V^T
n x n
compressed representation of the document

Where: U and V are orthogonal matrices of m x n and m x m size respectively, and $\sum$ is diagonal matrix of m x m. SVD finds the optimal projection to a low dimensional space is the key property for exploiting word co-occurrence patterns.

For an *m x n* matrix *A* of rank ***r*** there is exists a factorization (Singular Value Decomposition = **SVD**) as follows:



The columns of **U** are orthogonal eigenvectors of $\mathbf{AA^T}$

The columns of **V** are orthogonal eigenvectors of $\mathbf{A^TA}$

Eigenvalues $\lambda_1, \lambda_2 \ldots \lambda_n$ of $\mathbf{AA^T}$ are the eigenvalues of $\mathbf{A^TA}$

$\sum = \text{diag}\ (\sigma_1, \sigma_2 \ldots \sigma_n),\ \sigma_i \geq \sigma_{i+1}$

where, $\sigma_1, \sigma_2 \ldots \sigma_n$ are singular values.

2.4.4. Query Matching and Searching

A query is a keyword that is consulted during search (Christopher *et al*., 2009). It is formulated by users and compared to the index keywords to retrieve information needed. A comparison of query and document representations is a very important task of the IR system. It is performed in order to retrieve documents that are more similar to the specific query (Recardo, 1999). The ‘Search’ is a systematic examination of information in a database, aiming in view to identify the items or objects, which satisfy particular preset criteria (Juban and Falguni, 2007). In another way, searching means the operation of locating a specific object in a given sequence of ‘n’ objects.

All search strategies are based on a comparison between the query and the stored documents (Rijsbergen, 1979), sometimes this comparison is only achieved indirectly when the query is compared with clusters (or more precisely with the profiles representing the clusters). The distinctions made between different kinds of search strategies can sometimes be understood by looking at the query language that is the language in which the information need is expressed and the nature of the query language often dictates the nature of the search strategy. These strategies are based on the common notion that the more often terms are found in both the document and the query, the more "relevant" the document is deemed to be to the query (Grossman *et al*., 2004). A retrieval strategy is an algorithm that takes a query Q and a set of documents $D_1, D_2, ..., D_n$ and identifies the Similarity Coefficient (Q, D_i) for each of the documents $1 \leq i \leq n$ and the retrieval strategies identified are (Rijsbergen, 1979; Grossman *et al*., 2004):

Boolean Indexing: A score is assigned such that an initial Boolean query results in a ranking. This is done by associating a weight with each query term so that this weight is used to compute the similarity coefficient.

Vector Space Model: Both the query and each document are represented as vectors in the term space. A measure of the similarity between the two vectors is computed.

Probabilistic Retrieval: A probability based on the likelihood that a term will appear in a relevant document is computed for each term in the collection. For terms that match between a query and a document, the similarity measure is computed as the combination of the probabilities of each of the matching terms.

Language Models: A language model is built for each document, and the likelihood that the document will generate the query is computed.

Inference Networks: A Bayesian network is used to infer the relevance of a document to a query. This is based on the "evidence" in a document that allows an inference to be made about the relevance of the document. The strength of this inference is used as the similarity coefficient.

Latent Semantic Indexing: The occurrence of terms in documents is represented with a term-document matrix. The matrix is reduced via Singular Value Decomposition (SVD) to filter out the noise found in a document so that two documents that have the same semantics are located close to one another in a multidimensional space.

Neural Networks: A sequence of "neurons," or nodes in a network, that fire when activated by a query triggering links to documents. The strength of each link in the network is transmitted to the document and collected to form a similarity coefficient between the query and the document. Networks are "trained" by adjusting the weights on links in response to predetermined relevant and irrelevant documents.

Genetic Algorithms: An optimal query to find relevant documents can be generated by evolution. An initial query is used with either random or estimated term weights. New queries are generated by modifying these weights. A new query survives by being close to known relevant documents and queries with less "fitness" are removed from subsequent generations.

Fuzzy Set Retrieval: A document is mapped to a fuzzy set (a set that contains not only the elements but a number associated with each element that indicates the strength of membership). Boolean queries are mapped into a fuzzy set intersection, union, and complement operations that result in a strength of membership associated with each document that is relevant to the query. This strength is used as a similarity coefficient

2.5. Document Clustering

Clustering is the process of partitioning a set of data objects into subsets (Recardo, 1999). It is a sub-field in artificial intelligence and machine learning that refers to a group of algorithms that try to find a natural grouping of objects based on some objective metric (Christoper, 2010). Clustering is the most common form of unsupervised learning (Cambridge, 2009). No super-vision means that there is no human expert who has assigned documents to classes.

Clustering is a powerful technique for large-scale topic discovery from the text (Bjornar and Chinatsu, 1999). It involves two phases: feature extraction and applies clustering algorithms. Feature extraction maps each document or record to a point in high-dimensional space, and then clustering algorithms automatically group a set of documents based on their similarity and dissimilarity.

2.5.1. Approaches of Document Clustering

The two main types of cluster analysis approaches are hierarchical and flat clustering. Hierarchical clustering is another unsupervised learning algorithm that is used to group together the unlabeled data points having similar characteristics (Recardo, 1999). There are two basic methods of hierarchical clustering (Recardo, 1999): the first one is Agglomerative, which start with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters. This requires a definition of cluster similarity or distance. The other one is Divisive, which starts with one, all-inclusive cluster, and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide, at each step, which cluster to split and how to perform the split.

The other one is flat clustering. A flat/nonhierarchical clustering are creates a flat set of clusters without any explicit structure that would relate clusters to each other (Cambridge, 2009). From flat clustering K-means is perhaps the most widely used hard clustering algorithm due to its simplicity and efficiency (Cambridge, 2009). The term K-means was first used by James Macqueen in 1967 (Ramakrishna at el., 2014). K-means is partitioning based clustering algorithm, it groups the objects in continuous n-dimensional space, which uses centroid as mean of the group of objects (Bader *et al*., 2009).

K-means divide a data set of *N* items into *K* clusters, and the hierarchical, which produce a nested data set in which pairs of items or clusters are successively linked (Edie, 2015). Unlike hierarchical clustering, *K*-means (partitioned) clustering directly divides data objects into *K* clusters, without any corresponding hierarchical structure (Bader *et al*., 2009). Let P= {Pi}, i=1...n be the set of data points, to be clustered into a set of K number of clusters without any prior knowledge of the input objects. Predefined number of groups is indicated with K, where K is provided as an input parameter. Assigning of each object to a cluster is based on the objects proximity to the mean of the cluster. Then the mean of the cluster is in turn recomputed and the process of assigning objects to cluster resumes. It is done by minimizing the sum of squares of distances between objects and their corresponding cluster centroid. It groups the objects into *K* clusters and keeps iteratively moving the cluster centers and re-assigning objects into clusters, based on minimum distance to the closest cluster's centroid. The process terminates when cluster centers are not moved anymore and all objects have been assigned to their closest cluster center.

For *K*-means clustering, the cosine measure is used to compute which document centroid is closest to a given document (Michael *et al*., 2009). While a median is sometimes used as the centroid for *K*-means clustering. There are other distance measures (Ramakrishna *at el.*, 2014): Manhattan, Cosine similarity and Jaccard.Cosine and Jaccard measures are appropriated for document clustering. Cosine similarity is mostly appropriate for document similarity. The cosine similarity is measured in the following manner.

$$\mathrm{sim}(\mathrm{d1}, \mathrm{d2}\ ) = \frac{\mathrm{d1.d2}}{||\ \mathrm{d1}||\ ||\mathrm{d2}||} = \frac{\sum_{\mathrm{j}=1}^{\mathrm{t}} d1_{*\,d2}}{\sqrt{\sum_{\mathrm{j}=1}^{\mathrm{t}} \left(_{\mathrm{d1}}\ \right)^{2} * \sum_{\mathrm{j}=1}^{\mathrm{t}} \left(_{d1}\right)^{2}}} \tag{2.7}$$

Here d1 and d2 are two representative points contain n number of values. *K*-means clustering is very useful in exploratory data analysis and data mining in any field of research, and as the growth in a computer, power has been followed by a growth in the occurrence of large data sets (Chartier, 2013). Its ease of implementation, computational efficiency, and low memory consumption have kept the *K*-means clustering very popular, even compared to other clustering techniques.

In the Information Retrieval (IR) field, cluster analysis has been used to group similar documents together with the goal of improving the efficiency and effectiveness of retrieval or to determine the structure of the literature of a field (Edie, 2015). Cluster-based retrieval has as its foundation the cluster hypothesis, which states that closely associated documents tend to be relevant to the same requests (Rijsbergen, 1979). In other words, documents within a cluster should be as similar as possible; and documents in one cluster should be as dissimilar as possible from documents in other clusters (Cambridge, 2009).

2.6. Challenges of the Language in Information Retrieval System

Lemmatization and Stemming: Morphology is the study of the structure of a word and how it is built (Norrby, 2016). (Martin and Jurafsky, 2019) describe it as small building blocks of a word are called morphemes and they can be divided into two groups, stems, and affixes. A stem is the main part of a word and affixes are morphemes that are added to a stem to give different meanings to it. In the word dogs, for example, the dog is the stem and -s is the affix. Using affixes allows a word to occur in different forms, it can give it different inflections and derivations. Lemmatization and stemming are two methods that can be used to handle the challenges that the variations of words impose. Christopher *et al*., (2008) explain that this is done by reducing a word to a simpler form, and even though lemmatization and stemming aim for the same goal they achieve it in different ways. Stemming cuts of the ends of a word, to get rid of affixes and get the simple form of a word (Norbby, 2016). For example, the word writing would with a stemmer be reduced to "writ". This process is not flawless and can give incorrect results. According to Christopher *et al*. (2008), such like incorrect results are improve the recall while it lowers the precision. Lemmatization does it in a more proper way, by doing a morphological analysis using a vocabulary to get the dictionary form of the given word (Christopher *et al*., 2008). The word writing will in this case be normalized to “write” which is the dictionary word form.

Compound Splitting: split compound words seem particularly interesting when it comes to information retrieval of languages with that characteristic (Norrby, 2016). When splitting compound words there are different method. (Barbel, 2004) explain that a splitting method can be regarded as an aggressive one, which means that it will split the compound into its smallest parts. This means that some words such as “football” will be split into “foot”, “ball”. Since documents with the word foot or ball may have nothing to do with football, this may be a disadvantage that causes an IR to find irrelevant documents. Such like these also happened in Afaan Oromo (i.e., “abbaa gadaaa” will be split into “abbaa” and “gadaa”).

Irregular morphology: antennae are only a plural of the type of antenna that is associated with an insect, not with a television antenna (Krovetz, 2000). Similarly, media is the plural of the medium used in the sense of entertainment, not in the sense of a spiritualist.

Part-of-speech: intimation is derived from intimate (v), and intimately is derived from intimate (adj). Because suffixes only attach to roots with particular parts-of-speech, this information can be used to discriminate one homograph from another.

Run-ons: These are words that are morphologically related to the headword (the word that is being defined in the dictionary) (Krovetz, 2000). They generally appear at the end of a homograph. Run-ons have a predictable relationship to the root form (the headword) and are primarily a way for the lexicographer to include additional entries without taking up much space. For example, the dancer can appear as a run-on for dance, and craftsmanship as a run-on for the craftsman. Comparing run-on entries with main entries can help us focus on the information that is needed to disambiguate them (e.g., boxer as a type of dog versus a human).

Synonymy and polysemy: synonymy creates a problem when a document is indexed with one term and the query contains a different term, and the two terms share a common meaning (Savoy, 2000). Most of human knowledge is coded in natural language. However, it is difficult to use natural language as a knowledge representation language for computer systems. The current retrieval models are based on either keyword for search or author. This keyword representation creates a problem during retrieval due to polysemy, homonymy, and synonymy. Polysemy involves the phenomenon of a lexeme with multiple meanings. Keyword matching may not always include word sense matching (Savoy, 2000).

2.7. Overview of Afaan Oromo

Afaan Oromo is the second most widely spoken indigenous language in Africa next to Hausa in Nigeria (Feyisa, 1996). It is one of the highly developed languages of the Cushitic languages spoken in Ethiopian, Somalia, Sudan, Tanzania, and Kenya (Ibrahim, 2015). From forty different Cushitic linguistic and cultural groups in Africa, the Afaan Oromo speakers are considered as one of the largest in terms of population and their language (Feyisa, 1996). In Oromia state, in Ethiopia, it is the official language used in courts, schools, and administration (Feyisa, 1996).

Currently, there are a growing number of publications in hard copies and a vast amount of information in electronic formats for Afaan Oromo (Tilahun, 2008). This was made for the convenience of the Latin script for the writing of Afaan Oromo from the linguistics, pedagogic, and practical reasons. It is believed that many fold more texts were written in Afaan Oromo since then than before. Afaan Oromo is a language that is used in a wide area in the country. According to Gragg, (1976), quoted in (Wakshum, 2000), four major categories can be identified. These are Western (Wellega, Iluababor, Kaffa, and parts of Gojjam), Eastern (Harar, Eastern showa, and parts of Arsi and Bale), Central (Central Showa, Western Showa, and possibly Wollo), and Southern (Parts of Arsi, Sidamo, and Borena).

2.7.1. Features of Afaan Oromo writing system

Afaan Oromo Writing Style: According to (Ladefoged, 1955), quoted in (Morka, 2001) some of the finer anatomical feature involved in speech production include the vocal cords, velum, tongue, teeth, palates, the alveolar ridge, the mouth, and lips. These anatomical components move to different positions to produce various sounds and are referred to in articulators. Most of the characters of a sound are determined by the position of these articulators in the oral tract.

The Oromo Alphabet and sounds (Qubeeleewwan fi Sagaleewwan): The alphabets of Afaan Oromo are often called "Qubee Afaan Oromo (Ibrahim, 2015). Qubee has 33 characters representing distinct sounds. It has both capital and small letters. Afaan Oromo has a considerable amount of glottal stops. The major representatives of sources of the sound in a language are the vowels and consonants. Afaan Oromo is a phonetic language, which means that is spoken in the way it is written (Wakshum, 2000). The Afaan Oromo vowels represented by letters (a, e, o, u, and i) are called "Dubbiftu/Dubbachiftu" in Afaan Oromo, and the consonants known as "dubbifamaa" in Afaan Oromo. In the Afaan Oromo alphabet, a letter consists either of a single symbol or a digraph (ch, dh, ny, ph, sh, ts zh). Gemination is not obligatorily marked for the digraphs (Encyclopedia, 2018).

Table 2.2. The qubee in international phonetic writing (IPA) (Ibrahim, 2015)

Qubee		IPA	Qubee		IPA	Qubee		IPA
A	a	/a/	L	l	/l/	W	w	/w/
B	b	/b/	M	m	/m/	X	x	/x/
C	c	/c/	N	n	/n/	Y	y	/y/
D	d	/d/	O	o	/o/	Z	z	/z/
E	e	/e/	P	p	/p/	CH	ch	/ch/
F	f	/f/	Q	q	/q/	DH	dh	/dh/
G	g	/g/	R	r	/r/	NY	ny	/ny/
H	h	/h/	S	s	/s/	PH	ph	/ph/
I	i	/i/	T	t	/t/	SH	sh	/sh/
J	j	/j/	U	u	/u/	TS	ts	/ts/
K	k	/k/	V	v	/v/	ZH	zh	/zh/

Consonants (Dubbifamaa) and Vowels (Dubbachiiftuu) phonemes: Afaan Oromo has the typical Southern Cushitic set of five short (a, e, i, o, u) and five long vowels, indicated in the orthography by doubling the five vowel letters (aa, ee, ii, oo, uu). The difference in length of vowels results in change of meaning (Mewis, 2001). The Afaan Oromo vowels always are pronounced in sharp and clear fashion which means each and every word is pronounced strongly, for example;

Table 2.3 Afaan Oromo Vowels

Vowels	**Afaan Oromo**	**English**
A	Hara	lake
	Haaraa	new
E	Eegee	Waiting
	ijoollee	Children
I	amajji	May
	Ilaali	look
O	oromoo	Oromo
	Haaloo	revenge
U	Utubuu, beekuu	Support, knowing

Gemination (doubling a consonant) is also significant in Afaan Oromo. That is, consonant length can distinguish words from one another (Tesfaye, 2010) Example:

Table 2.4 Gemination (doubling a consonant) in Afaan Oromo

Afaan Oromo	**English**
Badaa	bad
Baddaa	highland
Hatuu	cheat/steal
Hattuu	thief

Punctuation Marks: Words in Afaan Oromo sentences are separated by white spaces the same way as it is used in English. Different Afaan Oromo punctuation marks follow the same punctuation pattern used in English and other languages that follow the Latin writing system (Megersa, 2002). For example, comma (,) is used to separate listing of ideas, concepts, names, items, etc and the full stop (.) in the statement, the question mark (?) in interrogative and the exclamation mark (!) in command and exclamatory sentences mark the end of a sentence. All punctuation marks that are used in English are also used for the same purpose in Afaan Oromoo except the apostrophe. Unlike its use to show possession in English, it is used as a symbol to represent a hiccup (called hudhaa) sound in Afaan Oromo writing. An apostrophe, and less commonly a hyphen, is used "'" represent this sound in writing. Sometimes an H, which represents the closest glottal sound, is also used in place of an apostrophe (Ibrahim, 2015), for a reason to be apparent later, the apostrophe will be considered as a distinct symbol (say, as the 27th letter of the alphabet).

Personal pronoun: Oromo pronouns include personal pronouns (refer to the persons speaking, the persons spoken to, or the persons or things spoken about), indefinite pronouns, relative pronouns (connect parts of sentences) and reciprocal or reflexive pronouns (in which the object of a verb is being acted on by verb's subject).

Table 2.5 Afaan Oromo personal pronouns

English	**Base**	**Subject**	**Dative**	**Instrumental**	**Locative**	**Ablative**	**Possessive adjectives**
I	ana, na	ani, an	naa, naaf natti	naan	natti	narraa	koo, kiyya
you (sg.)	si	Ati	sii, siif, sitti	siin	sitti	sirraa	kee
he	isa	Inni	isaa, isaa(itti)f ,isaatti	isaattin	isatti	isarraa	isaa
she	isii, ishii, ishee	isiin, etc	ishii, ishiif, ishiitti	ishiin	ishiitti,etc	ishiirraa	isii ishii
we	nu	nuti, nu'i, nuy, nu	nuu, nuuf, nutti	nuun	nutti	nurraa	keenna keenya
you (pl.)	isin	Isini	isinii, isiniif, isinitti	isiniin	isinitti	isinirraa	keessan
they	isaan	Isaani	isaanii, isaaniif,	isaaniitiin	isaanitti	isaanirraa	isaani

Adjectives: Adjectives are very important in Afaan Oromo because its structure is used in every day conversation (Gezehagn, 2012). Afaan Oromo Adjectives are words that describe or modify another person or thing in the sentence.

Adverbs (Ibsa Xumuraa): Afaan Oromo adverbs are part of speech or words that modify any part of language other than a noun. Adverbs can modify verbs, adjectives (including numbers), clauses, sentences and other adverbs (Mylanguage.org, 2011). The four categories of adverbs in Afaan Oromo: Adverb of time, adverbs of place, adverbs of manner and adverbs of frequency. Lists of adverbs are given in table 2.6. The lists on left side are in English while those on the right side equivalent meaning of the terms in Afaan Oromo.

Table 2.6 Adverbs in Afaan Oromo

Adverb of time		**Adverb of manner**	
English	**Afaan Oromo**	**English**	**Afaan Oromo**
Yesterday	Kaleessa	Very	Baayyee
Today	Harr’a	Quite	Baayyee
Tomorrow	Bor	Really	Dhugumaan
Now	Amma	Fast	Dafee
Then	Gaafas	Well	Gaarii
Later	Ager	Hard	Cimaa
Tonight	Edana	Quickly	Dafee
Right now	Amma is amma	Slowly	Suuta
Last night	Eda	Carefully	Qalbiidhan
This morning	Ganama kana	Absolutely	Matuma
Next week	Turban dhufu	Together	Walii wajjin
Recently	Dhiyeenya kana	Alone	qophaa
Soon	Dhiyootti		
Immediately	hatattamaan		
Adverbs of place		**Adverbs of frequency**	
Here	As	Always	Yeroo hunda
There	Achi	Sometimes	Gaaffii gaaf
Over there	Gara sana	Occasionally	Gaaffii gaaf
Everywhere	Iddoo hunda	Seldom	Darbee darbee
Nowhere	Eessayyu	Rarely	Darbee darbee
Home	Mana	Never	yoomiyyuu
Away	Fagoo		
Out	Ala		

Prepositions: Prepositions in Afaan Oromo are links nouns, pronouns and phrases to other words in a sentence. The word or phrase that the preposition introduces is called the object of the preposition (Mylanguage.org, 2011). Here are list of some prepositions:

Table 2.7 Prepositions in Afaan Oromo.

English prepositions	**Oromo prepositions**	**English prepositions**	**Oromo prepositions**
about	waa’ee	since	ergii
above	gubbaa/gararraa	than	manna
across	gama	through	gidduu
after	booddee/booda	till	hamma
against	faallaa	to	tti
among	jara giddu	towards	garas
around	naannoo	under	jala/gajjallaa
As	akka	unlike	faallaa
At	Itti	until	hamma
about	waa’ee	since	ergii
above	gubbaa/gararraa	than	manna
across	gama	through	gidduu
after	booddee/booda	till	hamma
against	faallaa	to	tti
among	jara giddu	towards	garas
around	naannoo	under	jala/gajjallaa
As	akka	unlike	faallaa
At	Itti	until	hamma
before	Dura	up	gubbaa
behind	dudduuba/dugda duuba	via	karaa
below	jala/gajjallaa	with	wajjin
beneath	gajjallaa	within	keessatti
beside	Bira	without	malee
between	gidduu	two words	jechoota lama
beyond	garas	according to	akka kanaatti
But	garuu	because of	kanaaf
English prepositions	**Oromo prepositions**	**English prepositions**	**Oromo prepositions**
By	..dhaan	close to	bira
despite	ta’uyyuu	due to	kanaaf
Down	Lafa	except for	kana malee
during	Utuu	far froom	Irraa, siqee/iraa fagaatee
except	malee	inside of	keessa isaa
For	F	instead of	kanaa manna
Table 2.7, Prepositions in Afaan Oromo		**Continued** …	

From	Irraa	near to	itti aanee
In	keessa	next to	itti aanee
inside	keessa	outside of	kanaa alatti
into	keessatti	prior to	kanaan dura
near	Bira	three words	jechoota sadii
next	ittaanee	as far as	hamma
Of	Kan	as well as	fi
On	Irra	in addition to	datalatees
opposite	fuullee/ fallaa	in front of	fullee isaa
out	Ala	in spite of	ha ta’uyyu malee
outside	Alla	on behalf	maqaa …
over	irraan	on top of	kana irraan
per	..tti	demonstrative prepositions	agarsiisoo
plus	Fi	this	kana/tana
round	naannoo	that	sana
		these	warra kana/ jara kana

2.7.2. Morphology of Afaan Oromo:

Morphology is a way of studying the language word structure (Algeo, 2010). It is about the way words are put together, their internal structure. It is a level at which the structure of language is analyzed. It deals with the analysis and examination of meaningful units of forms that make up sentences. The smallest meaningful units of forms are called "morpheme", a meaningful form that cannot be divided into smaller meaningful parts (Algeo, 2010; Ibrahim, 2015). Morphemes are either "free" or "bound" (Richard, 2003). A free morpheme can occur on its own whereas bound morphemes do not occur alone. Bound morphemes are of three types, these are, prefix attached to the initial positions, infix inserted in the middle, and suffix attached to the final position of the word. All three types of bound morphemes are called "affixes".

Every language has its own morphological structure that defines rules used for combining the different components the language may have (Wakshum, 2000). The Afaan Oromo for instance is different in its morphological structure from French, English, and other languages that use Latin characters.

Morphologically there are many word-formation processes in Afaan Oromo (Wakshum, 2000), affixation and compounding are among these word-formation processes. Affixation is generally described as the addition of affixes at the beginning, in between, and/or at the end of a root/stem depending on whether the affix is a prefix, infix, or suffix (Wakshum, 2000). In agglutinative languages like Afaan Oromo, most of the grammatical information is conveyed through affixes, (that is, prefixes and suffixes) attached to the root or stem of words.

Afaan Oromo word is composed of two parts (Ibrahim, 2015): (1) the root (base morpheme), which generally consists of basic sound and provides the basic lexical meaning of the word, and (2) the pattern, which consists of prefixes and/or suffixes and gives grammatical meaning to the word. Thus, the root /Bar/ combined with the pattern /-e/ gives Bare ‘learned,’ whereas the same root combined with the pattern /-te/gives barte ‘she learns’.

There are more than ten major and very common plural markers in Afaan Oromo, including -oota, -oolii, -wwan, -lee, -an, een, -eeyyii, -oo, etc.). As an example, the Afaan Oromo singular noun mana (house) can take the following different plural forms: manoota (mana + oota), manneen (mana + een), and manawwan (mana + wwan). The construction and usages of such alternative affixes and attachments are governed by the morphological and syntactic rules of the language (Kekeba *et al.*, 2007).

Compounding is the joining together of two linguistic forms, which functions independently suffix (Wakshum, 2000). Examples compound nouns include; abbaa-buddenaa ‘step father’ from abba-‘father’ and buddena ‘food’.

The extensive inflectional and derivational features of Afaan Oromo are presenting various challenges for text processing and information retrieval tasks in the language (Tesfaye, 2010). In information retrieval, the abundance of different word forms and lexical variability may result in a greater likelihood of mismatch between the forms of a keyword in a query and its variant forms found in the document index databases. Stemming, a technique that is used to bring morphological variants of a word into the root or stem word, plays an important role in this regard.

2.8. Related Works

2.8.1. Foreign Related Works

Xiaoyong, (2010) proposed two new models for cluster-based retrieval and evaluate them. The study used document clustering model, is viewed as a mixture model of three sources: the document, the cluster/topic the document belongs to, and the collection. A relevant document assumed being generated by this mixture model. Both partitioning and hierarchical agglomerative clustering algorithms have been studied in the context of IR. They used a three-pass K-means algorithm as an example of partitioning methods in their static clustering experiments. Two sets of experiments are performed in this study. The first set of experiments investigates whether a simple language model of clusters can be used to rank clusters. And the second set of experiments examines the effectiveness of cluster-based retrieval in the context of query likelihood retrieval and the relevance model (RM), for both static clustering and query-specific clustering. Both experiments have given promising result to design and develop clustering-based retrieval which were more effective than the traditional, cluster free search engine.

bharathi and venkatesan, (2012) investigated on improving information retrieval using document clusters and semantic synonym extraction. Initially, document clustering was investigated for improving the precision or recall in information retrieval systems and as an efficient way of finding the nearest neighbors of a document. They proposed as clustering is important in browsing a collection of documents or in organizing the results returned by a search engine in response to a user's query. This paper presents a new semantic synonym based correlation indexing method in which documents are clustered based on nearest neighbors from the document collection and then further refined by semantically relating the query term with the retrieved documents by making use of a thesaurus or ontology model to improve the performance of Information Retrieval System (IRS) by increasing the number of relevant documents retrieved. Results show that the proposed method achieves significant improvement than the existing methods and may generate the more relevant document in the top rank.

Maruf and Yuji, (2015) were conducted research on document clustering: Before and After the Singular Value Decomposition. In this paper, they discussed the clustering of documents by calculating pair-wise similarity between documents using the original term-document matrix and the decomposed term-document matrix. They also reported an evaluation method based on the clustering hypothesis and analyze the clustering results. They necessarily address issues involving representation of documents and computation of similarities between a set of documents.

Zhang et al., (2016) proposed clustering by using SVD on clusters to improve precision of inter-document similarity. They discussed as LSI (Latent Semantic Indexing) based on SVD (Singular Value Decomposition) was proposed to overcome the problems of polysemy and homonym in traditional lexical matching. However, it is usually criticized as with low discriminative power for representing documents although it has been validated as with good representative quality. In this paper, SVD on clusters is proposed to improve the discriminative power of LSI. The contribution of this paper is three manifolds. Firstly, they made a survey of existing linear algebra methods for LSI, including both SVD based methods and non-SVD based methods. Secondly, they proposed SVD on clusters for LSI and theoretically explain that dimension expansion of document vectors and dimension projection using SVD are the two manipulations involved in SVD on clusters. Moreover, they developed updating processes to fold in new documents and terms in a decomposed matrix by SVD on clusters. Thirdly, two corpora, a Chinese corpus and an English corpus, are used to evaluate the performances of the proposed methods. Experiments demonstrate that, to some extent, SVD on clusters can improve the precision of inter-document similarity measure in comparison with other SVD based LSI methods.

2.8.2. Local Research Works

There were several research works done in Afaan Oromo Information Retrieval to enhance the retrieval performance of the system. These research works are discussed below based on their chronological order.

Tewodros, (2003) was developed Amharic text retrieval using latent semantic indexing with singular value decomposition. In this thesis, the potential of LSI approach in Amharic text retrieval was investigated and 206 Amharic documents and 25 queries were used to test the approach. Automatic indexing of the documents resulted in 9256 unique terms which are not in the stop-word list were used for the research. Finally, the performance of this study was compared with the standard vector space and (0.80) precision of the LSI approach was indicated above that of the standard vector space.

Tesfaye, (2010) was conducted research on IR, designed, and developed a search engine for Afaan Oromo the performance evaluation using a selected set of documents and queries. Experiment on some specific features of the language against the design requirements is also made. In this case, the average precision for the top 10 results is 76%. This indicates that displaying only the first few top results, the top 10 in this case, entails better user satisfaction as compared to displaying all the results. When a query is posed to the search engine, the search was taking place using the default OR operator of Lucene's search on the terms in the query. This causes more number of documents to be retrieved. In effect, the precision was being negatively affected. The future directions stated in the paper are: developing standard relevance judgment metrics, testing efficiency and effectiveness by comparing other algorithms, the work in the paper supposed to be dynamic whereas it is not; so it needs further work to make it dynamic, and the work was used a limited number of documents since it better if it used large documents.

Gezehagn, (2012) further intended to make possible retrieval of Afaan Oromo text documents by applying techniques of the modern Information Retrieval system. Information Retrieval is a mechanism that enables finding relevant information material of unstructured nature that satisfies the information need of users from the large collection. Afaan Oromo text retrieval developed in this study has indexing and searching parts. The Vector Space Model of Information Retrieval system was used to guide searching for the relevant document from the Afaan Oromo text corpus. The model is selected since the Vector space model is the widely used classic model of the Information Retrieval system. The index file structure used is an inverted index file structure. In this study, the experimental result shows that the performance is on the average (57.5%) precision and (62.64%) recall. The challenging tasks in the study are the absence of standard corpus, handling synonymy and polysemy, the inability of the stemmer algorithm to all word variants, and ambiguity of words in the language. The future direction stated by the author includes: stemmed index terms might improve performance of the system and recommended to be used in the further work, and the algorithm LSI may record better performance if it is used. In this study, VSM is used and the indexing mechanism is used in here inverted index.

Eyob, (2013) tried to design and develop a corpus-based Afaan Oromo–Amharic cross-lingual Information Retrieval system that able to retrieve Amharic information using Afaan Oromo queries. This approach selected to be followed in the study is a corpus-based, particularly parallel corpus. For this study parallel documents including news articles, bible, legal documents, and proclamations from customs authority were used. The system is tested with 50 queries and 50 randomly selected documents. Two experiments were conducted, the first one by allowing only

one possible translation to each Afaan Oromo query term and the second by allowing all possible translations. The retrieval effectiveness of the system is measured using recall and precision for both monolingual and bilingual runs. Accordingly, the first experiment returned a maximum average precision of 0.81 and 0.45 for monolingual (Afaan Oromo queries) and bilingual (translated Amharic queries) run. The result of the second experiment showed a better result of recall and precision than the first experiment. The result obtained in the second experiment is a maximum average precision of 0.60 for the bilingual run and the result for the monolingual run remained the same. From these results, he was concluded that cross-lingual Information Retrieval for two local languages namely Afaan Oromo and Amharic could be developed and the performance of the retrieval system could be increased with the use of larger and clean corpora.
Mulualem, (2013) explored semantic indexing and document clustering for Amharic information retrieval using semantic indexing and document clustering. In order to solve the issues, integrating of semantic indexing and document clustering techniques with generic IR system were improved retrieval performance. Here the system comprises all processes exist in generic IR plus to the C-value technique multi word term extraction, K-means algorithm document clustering base searching strategy used. The system was tested using tagged Amharic news documents size of 650Kb and it registered F-measure of 66% accuracy. Nevertheless, the performance of the system is greatly affected by synonyms and polysemy, incorrect clustering, cluster representative problems, Amharic knowledge base.

Tewordros, (2014) Amharic text retrieval developed using thesaurus based semantic compression for improving the performance. For this study Amharic text document corpus were prepared by the researcher encompassing different news articles, books and Amharic websites. Also various techniques of text preprocessing including tokenization, normalization, stop word removal and stemming were used to identify content-bearing words. Once content bearing terms are identified, then semantic compression technique is applied to identify semantic representation that is more precise descriptor for each term. Tewordros was used thesaurus based inverted indexing scheme for semantic compression of terms in indexing. In the same way, in the searching side also thesaurus-based query expansion technique was employed. Experimental result on the average 70% precision and 77% recall respectively. The major challenges of this study were, the IR system include lack of standard thesaurus for Amharic language and ineffectiveness of Amharic stemmer to conflate Amharic inflectional words into their stem. Therefore, in order to improve the performance of the system there is a need to develop an effective Amharic thesaurus as well as Amharic stemmer.

Daniel, Ramesh, and Dereje, (2015) attempted to develop the Afaan Oromo-English CLIR system which enables Afaan Oromo native speakers to access and retrieve the vast online information sources that are available in English by writing queries using their own (native) language. Evaluation of the system was conducted by both monolingual and bilingual retrievals. The performance of the system was measured by recall and precision. As they were mentioned in their study larger documents were retrieved for the monolingual run (i.e. for the retrieval of documents by using baseline queries of Afaan Oromo) than for the bilingual run (i.e. for the retrieval of English documents using Afaan Oromo queries after being translated into English). The performance of the system propose was highly affected by the size, reliability, and correctness of the corpus used for the study. Maximum average precision of 0.468 and 0.316 for Afaan Oromo and English were obtained respectively. In this study, Afaan Oromo- English CLIR that is based on a corpus-based approach was developed for Afaan Oromo users to specify their information need in their native language and to retrieve documents in English. The performance of CLIR systems using a corpus-based approach is highly affected by the size, reliability, and correctness of the corpus used for the study. However, the size of the documents used for this research was limited in size and not quite reliable and clear which affected the level of performance to be achieved. Moreover, the domains of parallel documents used to carry out the research were limited. The paper indicated future directions, encouraging developing Afaan Oromo-English CLIR by using this approach. However, the low performance achieved and also affect the accuracy of English documents are retrieved.

Workineh and Debela, (2017) was developed sense disambiguation which finds the sense of Afaan Oromo words based on surrounding contexts. The idea behind this approach was to overcome the problem of scarcity of training data by using unsupervised approach that exploits sense in a corpus which is not labelled. The context of a given word is captured using term co-occurrences within a defined window size of words. The similar contexts of target words are computed using vector space model and then clustered by identifying semantically related Afaan Oromo words. In order to develop semantic model for Afaan Oromo they followed three steps process which involve corpus preprocessing which tokenize and remove stop words and perform normalization. Extract context terms providing clue about the senses of the target word, and then clustering to group similar context terms of the given target words, the number of clusters representing the number of senses encoded by the target word. The result argued that the system yields an accuracy of 81% which was encouraging result. Furthermore, the study has been pointed out how NLP plays a significant role in enhancing the computer's capability to process texts. To the end, they were pointed out as a semantic is one component of NLP contributing a lot to the effort of solving the problem of Information Retrieval Systems in answering users' requests by introducing semantics of a query term and index terms.

Wegari, (2017) conducted a rule-based root generation system for Afaan Oromo. In their investigation; they have been shown that a rule-based method can be used to develop a root generation system for Afaan Oromo. The system is mainly used as a starting point to develop a complete morphological analysis and information retrieval for the target language. The experimental results show that the methodology proposed is effective in identifying root boundaries. The main challenge in this paper is that, since the majority of the proposed rules are based on suffixes, the most errors of the system are due to strings that are found both as parts of roots and suffixes in the language. For instance, for the word rukuta, the system indicated ruk as its root instead of rukut. The future directions of this paper show that the rule-based method can be used to develop a root generation system for Afaan Oromo. It might be used as a starting point to develop a complete morphological analysis and information retrieval for the target language.

Melkamu, (2017) integrate query expansion for enhancing the effectiveness of the Afaan Oromo text retrieval system. this study was applied the designed query expansion for the Afaan Oromo information retrieval system involves lexical resource WordNet that was constructed as a reference for identifying the sense and meaning of the user's query using word sense disambiguation by semantic similarity measure. The experimental result of this paper showed that the integration of query expansion registers 95% recall, 41% precision, and 56% F-measure. The challenges indicated were the absence of standard well-crafted WordNet, an effective stemmer algorithm, and an organized corpus. The future direction of this paper indicates applying query expansion module with Afaan Oromo IR to see the extent to which it improves the performance of the Afaan Oromo retrieval system. However precision of the system needs more improvement because of the polysemy nature of Afaan Oromo.

Table 2.8 local related work approach

Author (year)	**Problem**	**Approach**	**Result**	**Gap**
Tesfaye, (2010)	Searching and ranking technique Afaan Oromo documents	Boolean search and phrasal search approaches	average precision for the top 10 results is 76%	Better stemming algorithm, extracting semantic future. only the first few top results, the top 10 limited number of document are retrieved
Gezehagn, (2012)	accessing information that satisfies information needs of Afaan Oromo users	statistical retrieval approach(VSM)	average (57.5%) precision and (62.64%) recall	stemming algorithm is not working well handling synonym and polysemy
(Berhanu, 2019)	LSI in Afaan Oromo text retrieval	LSI	0.67(67%) precision and 0.63(63%) recall	LSI is used statistically related term extraction, while it is better to use language knowledge base to extract it LSI is more concern to recall but n prizes, but clustering approach with LSI is better to come up with good IR performance
Melkamu, (2017)	enhancing the performance of Afaan Oromo text retrieval system	Query expansion involves lexical resource WordNet	95% recall, 41% precision, and 56% F-measure	absence of standard well-crafted WordNet, effective stemmer algorithm and organized corpus, extracting semantically related words
Ashenafi (2019)	Word sense disambiguation based query expansion so as to improve the performance of Afaan Oromoo text retrieval system	word sense disambiguation based query expansion	66% F-measure	Used limited number of lexical resource because of the lack of resource, since similarity measure and the use of query expansion terms are limited based on the information available in the lexical resource.

Berhanu, (2019) Explored the potential application of Latent Semantic indexing approach in Afaan Oromo text retrieval. The goal of the study was to examine the benefits of LSI text retrieval approach for Afaan Oromoo text retrieval. LSI attempts to use this as a basis for creating a concept space such that this concept space is much smaller than the word space (mapping of terms against documents). Singular value decomposition (SVD) of the term by document matrix was used for indexing and retrieval system. The performance of the system after User Relevance Feedback is measured using recall and precision, the experiment shows that the performance of the prototype is on the average 0.67(67%) precision and 0.63(63%) recall registered.

Recently, Ashenafi, (2019) has attempts to extend the application of word sense disambiguation based query expansion. In this study word sense disambiguation based query expansion approach was used to determine the senses of words in queries by using Afaan Oromoo lexical resource. WordNet which is constructed by translating Princeton University's WordNet to Afaan Oromoo WordNet. WordNet is consists of all parts of speech such as nouns, verbs, adjectives and adverbs is the first component that is used as knowledge base in this study and Word Sense Disambiguation is second, which is used to identify the sense of the given query using semantic similarity measure from the knowledge base. Then Query reformulation by adding terms with highest overlap with each term in original query with reference to Afaan Oromoo WordNet. Finally, the query expansion module is integrated with Information Retrieval system to show the enhancement of Afaan Oromoo IR system performance. The experimental result showed an effective use of WSD using semantic similarity for identifying the sense with 66% F-measure performance. The indicated are lack of lexical resource which affect the similarity measure, since query expansion terms are limited based on lexical resource available.

Table 2.8 Local Related Research Works Approach

Generally, the above researchers are enabling the Afaan Oromo retrieval system using different methods, but they are still not considered document clustering and latent semantic indexing (synonym and polysemy). As pointed out by, Melkamu, (2017) and Gezehagn, (2012) the existence of synonym and polysemy Afaan Oromo words greatly affect the retrieval performance. Berhanu, (2019) was used LSI to extract Afaan Oromo semantically related words. However, when semantically related terms are used it increases recall and less precision. This is suppress the results to query during searching. Therefore developing enhanced Afaan Oromo information retrieval is takes into account; Latent Semantic indexing and clustering based searching to improve the performance of the Afaan Oromo information retrieval system.

3. METHODS AND TECHNIQUES

3.1. Overview

This chapter discusses the design and development of Semantic Indexing with the Document Clustering based searching for the Afaan Oromo information retrieval system. In the next section the architecture of the proposed IR system for Afaan Oromo is designed and presented. Further, methodology, research design, implementation tools, data preparation tasks such as tokenization, normalization, stop words removal and stemming of Afaan Oromo documents and query, selecting index terms, weighting TFIDF, Latent Semantic Indexing, K-means clustering, cluster-based searching, and evaluation procedures are discussed in detail.

3.2. Methodology of the study

The methodology describes actions to be taken to investigate a research problem and the rationale for the application of specific procedures or techniques used to identify, select, process, and analyze information applied to understand the problem, thereby, allowing the reader to critically evaluate a study's overall validity and reliability (Hevner and Chatterjee, 2015). A methodology is "a system of principles, practices, and procedures applied to a specific branch of knowledge" (Robertson, 2012). Such a methodology helps researchers to produce and present high-quality scientific research.

3.3. Research Design

Research design is considered as the structure of research or the arrangement of conditions for collection and analysis of data in a manner that aims to combine relevance to the research purpose with the procedure (Klinger and Lee, 2000). It is the plan, structure, strategy, and investigation conceived to obtain and ensure a logical basis for the decision and the blueprint for the collection, preprocessing, and experimenting (Tanner, 2018). This study follows experimental research design; thus, it is used to test theories or construct the theoretical explanations that require careful attention to all details from setting up the developing procedures for implementing the experiment.

To conduct an experimental research, the researcher followed a three step approach: document corpus preparation, implementation of a prototype and evaluation. Tasks performed and methods used during each step are stated as follows.

3.3.1. Document Corpus Preparation

The source of documents corpus for this study was Afaan Oromo news articles that were collected from Oromia Broadcast Network (OBN), www.obnoromia.com. Since it is difficult to prepare all relevant documents, a limited document corpus was selected to develop a prototype and evaluate the performance of the prototype. For experimenting and measuring the effectiveness of the proposed approach 243 Afaan Oromo news articles were randomly selected from this corpus, the articles are taken from agriculture, politics, agri-business, business, culture, education, health, religious, sport, and social aspects. All these articles are in txt format and reorganized based on their contents. The documents were preprocessed and prepared in an appropriate format before experimentation. For the preprocessing Afaan Oromo collected news articles, text operation tasks such as tokenization, normalization, stemming, and stop word removal techniques were applied for both indexing and searching. The analysis begins with a weighted term-document matrix and this weighted matrix was used for indexing and retrieval.

3.3.2. Implementation Tools

Python programming language is used to develop the intended prototype. Python was developed by Guido Van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands (Halterman, 2011) and it is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small-Talk, UNIX shell, and other scripting languages. Higher-level programming languages like Python allow are programmers to express solutions to programming problems in terms that are much closer to a natural language (Halterman, 2011). The reason to use Python programming language is; on one hand, the exposure of the researcher to the language and on the other hand, since python is used for developing clustering, indexing, and searching in IRS. Python is also a dynamic programming language that is used in a wide variety of scientific application domains (Python Software Foundation, 2012). It is simple, strong, involves natural expression of procedural code, modular, dynamic data types, and embeddable within applications as a scripting interface.

3.3.3. Evaluation procedure

For evaluation purposes; Precision, Recall, and F-measure are used to measure the effectiveness of IR systems. Based on the concept of relevance (i.e. the correspondence of documents with a given query or information need), there are several techniques for measuring the effectiveness of

IR, such as precision and recall, *F*-measure, *E*-measure, MAP (Mean average precision), *R*-measure (Christopher, 2011). To do this, appropriate queries, were systematically and subjectively selected after reviewing the content of each article manually to test the performance of the system either relevant or irrelevant to make relevance evaluation.

To show these metrics, assume the document collection be *D*. Let R_t is all retrieved documents from the collection *D* and R_l a number of relevant documents in *D*. The joint of R_t and R_l is a set of documents retrieved and relevant. Then, the recall and precision can be calculated using the following equation. The recall is the percentage of relevant documents retrieved from the corpus in response to users query and precision is the percentage of retrieved documents that are relevant to the query (Christopher *et al.*, 2009), thus;

$$\mathrm{Precision(P)} \ = \frac{|\{\mathrm{Rl} \cap \mathrm{Rt}\}|}{|\{\mathrm{Rt}\}|} \qquad (3.1)$$

$$\mathrm{Recall(R)} = \frac{|\{\mathrm{Rl} \cap \mathrm{Rt}\}|}{|\{\mathrm{Rl}\}|} \qquad (3.2)$$

A measure that combines precision and recall is *F*-measure or balanced *F*-score (Christopher *et al.*, 2009). *F*-measure is the harmonic mean of precision and recall, which is calculated as follows.

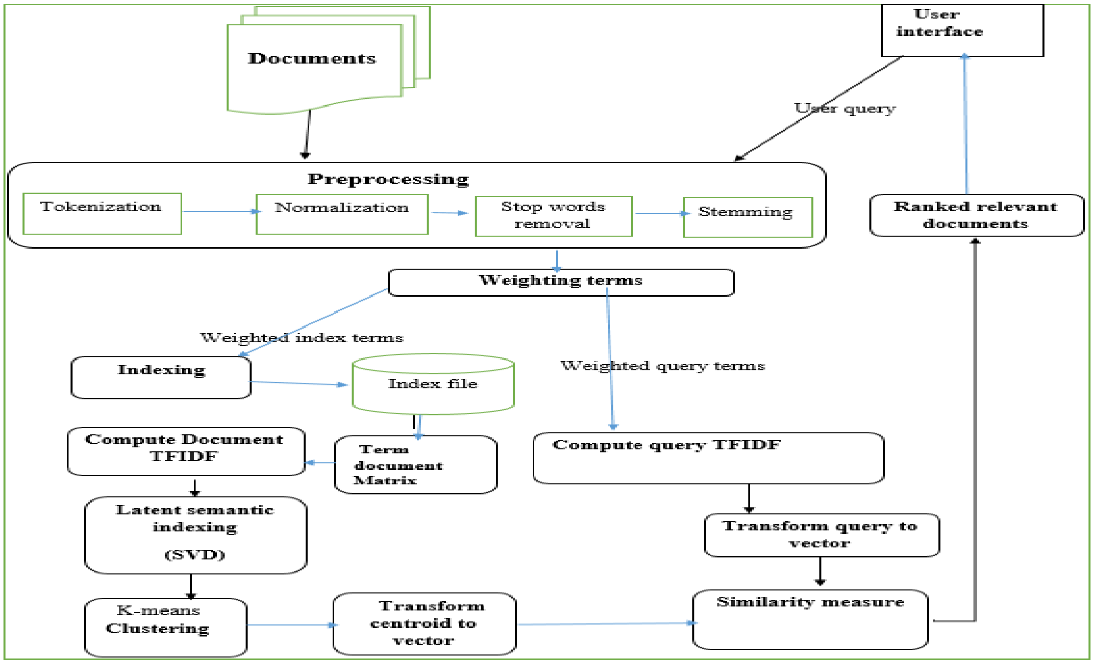
$$\mathrm{F - measure} \ = \frac{2(\mathrm{RP})}{\mathrm{R+P}} \qquad (3.3)$$

Where, *R* and *P* are recall and precision respectively.

3.4. The Architecture of AOIRs Using Semantic Indexing and Document Clustering Based Searching.

The architecture shows the different modules integrated with generic information retrieval model to design Semantic Indexing and Document Clustering based Searching for Afaan Oromo information retrieval. Here below figure 3.1 depicted the system architecture of the Afaan Oromo information retrieval process implemented in this study.

Fig 3.1 Architecture for AOIRs



This Architecture was adopted from generic information retrieval system architecture (Ceri *et al*., 2013) and modified by integrating SVD with K-means clustering methods. Below Tasks performed and methods used during each step in the architecture are stated in detail.

3.4.1. Dataset preprocessing

In this phase the different data preprocessing and preparation tasks are performed. These tasks include tokenization, normalization, stop word removal, and stemming of Afaan Oromo documents and queries.

Tokenization: In a given character sequence and a defined document unit, tokenization is the task of chopping it up into tokens, perhaps at the same time throwing away certain characters, such as punctuation marks (Singhal, 2008). A token is an instance of a sequence of characters in some particular documents that are grouped together as a useful unit for processing the document. In this study tokenization algorithm, 3.1 is used for splitting documents into a sequence of words based on word separators.

Algorithm 3.1: Tokenization algorithm (Gezehagn, 2012)

```
 FUNCTION Tokenization ( )
     //INPUT: Afaan Oromo documents
     //OUTPUT: sequence of words
       FOR file in corpus DO
             DEFINE whitespace OR punctuation as word delimiter
             READ each file in corpus
             FOR file in read DO
                   IF there is whitespace OR punctuation THEN
                         Put each term as separate token
                   END IF
             END FOR
       END FOR
       RETURN tokens
END Function
```

For tokenization purpose, whitespace and punctuation were used as a delimiter. Punctuations marks in Afaan Oromo are almost similar to that of English except for apostrophe (" ' ") which is considered as part of the letter in Afaan Oromo. In Afaan Oromo apostrophe is used as a part of latter and punctuation mark. Some of the punctuation marks are; ? ፡ " ! | / ? @ # * ~ $ % ^ & ( ) { } < > [ ] _ + = - , ." ...\; - _ + and removal of punctuation marks helps to consider similar words with different punctuations mark, in a similar way with no distinction of punctuation mark linked to it. For example 'mootummaa?', 'mootummaa!', 'mootummaa.', 'mootummaa, and etc. all are represented as 'mootummaa.

Normalization: after processing files into separated tokens, it should come to uniform writing format, so that matches occur despite superficial, hyphen, abbreviation, and equivalent character differences in the character sequences of the tokens (Mulualem, 2013). Primarily every term in the document should be converted into lowercase. For instance 'MOOTUMMAA', 'Mootummaa', 'mootummaa' are all normalized to lowercase 'mootummaa' (government). Numbers are also removed with punctuation marks.

Algorithm 3.2: Normalization algorithm (Gezehagn, 2012)

```
Function normalization ()
  //INPUT: sequence of words/tokens
  //OUTPUT: normalized sequence of words/tokens
  READ each tokens in file
  FOR word w in tokens DO
     IF word w is not lowercase THEN
         Convert to lowercase
          IF word w in tokens have any punctuation marks THEN
               Remove punctuation marks
               IF word w in tokens is not digit THEN
                  RETURN tokens
               END IF
          END IF
      END IF
   END FOR
END Function
```

Removal of Stop Word: Some extremely common words which would appear to be of little value or relevance in discrimination documents are removed from the token set (Zipf, 1932). These words are called stop words. The general strategy for determining a stop list is to sort the terms by collection frequency (the total number of times each term appears in the document collection), and then to take the most frequent term to remove. In addition, the other method is using a stop word list for the language itself (Singhal, 2008). Some of the design of modern Information Retrieval Systems has focused precisely on how we can exploit the statistics of language so as to be able to cope with common words in better ways (Tewdros, 2003). In Afaan Oromo; conjunctions (fi, yokaan, moo…), pronouns (ani, naaf, narraa, koo, isaan…), prepositions (waa'ee, ergii, hamma, garas, faallaa…), verbs (ture, jira, jedhamaa), adverbs (kaleessa, amma, dafee, cimaa, baayyee…) and adjectives (bareedduu, dheeraa, midhagduu …) are treated as stop words and still extracted manually from other words. In this study, the researcher was taken stopwords organized by (Gezehagn, 2012) and also added missing stop-words from lists of documents.

Algorithm 3.3: Stop word removal algorithm (Gezehagn, 2012)

```
FUNCTION Stop word-removal ()
    //INPUT: list of tokens, Stop words list
    //OUTPUT: List of non-stop words
    OPEN Stop words list file
    READ tokens list
     FOR each word w in tokens list
        IF a word w is in stop words list THEN
                REMOVE a word w
        END IF
      END FOR
      RETURN non-stop word tokens
END Function
```

Stemming: For grammatical reasons, documents are written using different forms of a word, though such morphological variant words have similar semantic interpretations (Nega and Peter, 2004). Morphological variation of words is also there in Afaan Oromo, for instance, ‘mootummaa’, ‘mootummaan’, ‘mootummotta’, ‘mootummollen’. As a result, such words have to be reduced to their root, ‘mootummaa’ using stemming. In addition to that stemming is also used to reduce the dictionary size (i.e. the number of distinct terms used in representing a set of documents). The smaller the dictionary size results in the smaller storage space and reduces the processing time required. Generally, the goal of stemming is to reduce inflectional forms and derivationally related forms of a word to a common base form, while it is language-dependent. Therefore, every language is using its own specific stemming technique. In the Afaan Oromo text, there are many word variants/affixes. To conflate them into stem words HornMorpho 2.5 was used for this work.

HornMorpho was developed by a research group for human language technology and the democratization of information (Gasser, 2012), it is part of the *L3* project at Indiana University, which USA dedicated to developing computational tools for under-resourced languages such as Amharic, Afaan Oromo, and Tigrinya words into their constituent morphemes (meaningful parts) and generate words, given root or stem and a representation of the word’s grammatical structure and lists of tokens are returned by using the following algorithm 3.4.

Algorithm 3.4: stemming algorithm (Nega and Willett, 2002)

```
FUNCTION stemmer ()
    //INPUT: non-stop word tokens
    //OUTPUT: stemmed words
    IF word in a tokens matches with one of the rules
              REMOVE the suffix and do the necessary adjustments
              RETURN the word and RECORD it in stem dictionary
       ELSE
              IF there is no applicable condition and action exist
                     RETURN word as it is and RECORD it in stem dictionary
              END IF
    END IF
END Function
```

3.4.2. Indexing

The function of any Information Retrieval system is to process a user request for information needed and retrieve documents that could potentially satisfy the information need of the user (Salton and Mcgill, 1986). Indexing is the process of analyzing text and deriving such short-form descriptions for a document that together sum up the message of the document (Baeza *et al*., 1982). Inverted index, or sometimes inverted file, has become the standard term in information retrieval for organizing documents (Christopher, 2011).

An inverted index always maps back from terms to the parts document where they occur and the basic idea of an inverted index is a dictionary of terms (sometimes also referred to as a vocabulary or lexicon). Each item in the list appeared in a document is conventionally called a posting (Christopher, 2011), the list is then called a posting list. Among the different representations of index terms, an inverted file index term representation technique is used in this work. An inverted file consists of a list of tokens where each token is followed by the identifier of every document that contains the word along with their number of occurrences in the document is represented. Using this information inverted file allows an information retrieval system to quickly determine which documents contain a given set of words, and how often each word appears in the document term (Baeza-Yates and Ribeiro-Neto, 1999). Inverted indexes associate each word in the text with a list of pointers to the positions where the word appears in the document (Kumar *et al*., 2012). For each term, the list of all documents which contained that term and additional information about that term like frequency of the word is stored. For instance, suppose we have the following Afaan Oromo documents to construct an inverted index.

TF_{ij}, number of occurrences of term t_j in document $\mathbf{d}_i$

DF_j, number of documents containing t_j

CF_{j},, collection frequency of t_j in n_j then index based on sorted list of terms, with each term having links to the documents is described below in figure 3.2

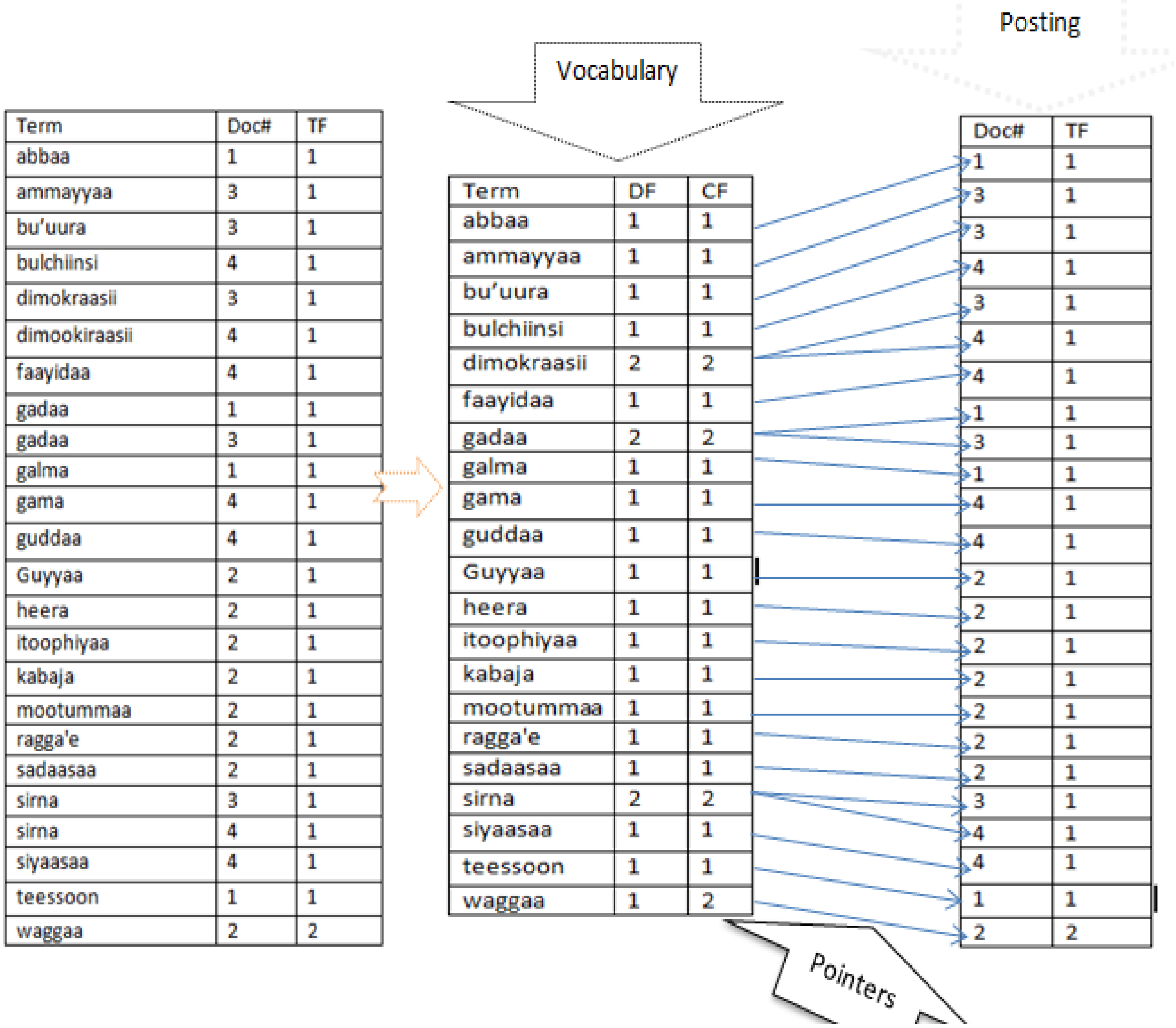
Doc1: Teessoon galma abbaa gadaa.

Doc2. guyyaan itti heerri mootummaa itoophiyaa itti ragga'e sadaasni 29 waggaa waggaan kabajamaa.

Doc3. sirni gadaa dimookiraasii ammayyaaf bu’uura kan ta’e dha.

Doc4. bulchiinsi sirna dimokraatawaa gama siyaasaatiin faayidaa guddaa qaba

Fig. 3.2. Inverted indexing for Afaan Oromo document



Term	Doc#	TF
abbaa	1	1
ammayyaa	3	1
bu'uura	3	1
bulchiinsi	4	1
dimokraasii	3	1
dimookiraasii	4	1
faayidaa	4	1
gadaa	1	1
gadaa	3	1
galma	1	1
gama	4	1
guddaa	4	1
Guyyaa	2	1
heera	2	1
itoophiyaa	2	1
kabaja	2	1
mootummaa	2	1
ragga'e	2	1
sadaasaa	2	1
sirna	3	1
sirna	4	1
siyaasaa	4	1
teessoon	1	1
waggaa	2	2

Term	DF	CF
abbaa	1	1
ammayyaa	1	1
bu'uura	1	1
bulchiinsi	1	1
dimokraasii	2	2
faayidaa	1	1
gadaa	2	2
galma	1	1
gama	1	1
guddaa	1	1
Guyyaa	1	1
heera	1	1
itoophiyaa	1	1
kabaja	1	1
mootummaa	1	1
ragga'e	1	1
sadaasaa	1	1
sirna	2	2
siyaasaa	1	1
teessoon	1	1
waggaa	1	2

Doc#	TF
1	1
3	1
3	1
4	1
3	1
4	1
4	1
1	1
3	1
1	1
4	1
4	1
2	1
2	1
2	1
2	1
2	1
2	1
2	1
3	1
4	1
4	1
1	1
2	2

The word 'abbaa' is in document 1 and its frequency is only once, the word 'gadaa' is in documents 1 and 3, and its term frequency once in both documents. Also, the word 'waggaa' is in document 2 and its frequency is twice. Just like these, all words in the collections contain the positions of each word within document or documents and its frequency.

3.4.3. TFIDF weighting

To compute index term weight, from the collection of documents, a term-document matrix was created and *TFIDF* was calculated for each index. The *TFIDF* weighting mechanism is based on term frequency, the number of occurrence of a term in a document, and the inverted document frequency of the term. The final weight of a term which was made of the term frequency and the inverse document frequency was calculated as follows.

Let $freq_{i,\,j}$ be the raw frequency of term k_i in document d_j (i.e the number of times the term k_i is mentioned in text of document d_j). Then, the normalized frequency $f_{i,j}$ of term k_i in document $\mathbf{d_j}$ is given by (Baeza *et al*., 1982);

$$f_{i,j} = \frac{freqij}{maxfreqij} \qquad (3.4)$$

where the maximum was computed over all terms which are mentioned in the text of the document d_j. Further, let *df* (document frequency) represent the number of documents in which the term k_i appears. This could assign a higher value to terms that appear in many documents and lower value for terms that appear only in few documents. However, terms that appear in many different documents are less indicative of the overall topic. Then, we can use idf_i**,** (inversed document frequency) to calculate the weight of a term and use the logarithm to dampen the effect of the *idf* as follows (Baeza *et al*., 1982);

$$idf = log(\frac{N}{df}) \qquad (3.5)$$

Hence the *idf* of a rare term is high, whereas the *idf* of a frequent term is likely to be low. Now combine the above definitions of term frequency and inverse document frequency equation, to produce a composite weight for each term in each document, the term weighting equation (Baeza *et al*., 1982);

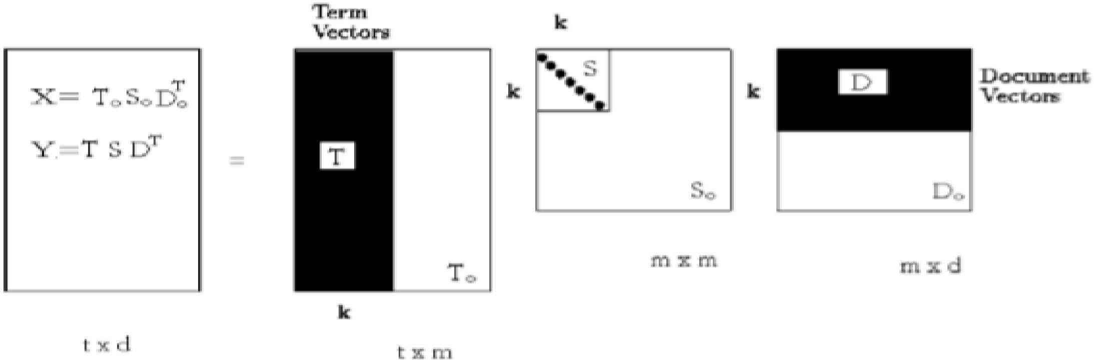
$$W_{i,j} = fi_j * idf \qquad (3.6)$$

These term weights are used to compute the degree of similarity between each document stored in the system and the user query. Moreover, the consideration of term weights can assist in ranking documents by sorting retrieved documents in decreasing order of their degree of similarity. However, in this study, the researcher used *TFIDF* for further process to apply LSI and clustering, since the objective of this study is to design Enhanced Afaan Oromo Information Retrieval using Semantic Indexing and Document based Clustering searching to improve performance.

3.4.4. Signle Value Decomposition (SVD)

Latent Semantic Indexing (LSI) is a method for discovering hidden concepts in document data (Dumais, 1992). Each document and term (word) is then expressed as a vector with elements corresponding to these concepts. Each element in a vector gives the degree of participation of the document or term in the corresponding concept. The goal is not to describe the concepts verbally, but to be able to represent the documents and terms in a unified way for exposing document-document, document-term, and term-term similarities or semantic relationship which are otherwise hidden. Latent Semantic Indexing, an efficient vector space retrieval approach, uses the Singular Value Decomposition (SVD) technique to reduce the rank of the original term-document matrix. Theoretically, SVD, a dimensionality reduction technique, performs a term-to-concept mapping, and therefore, conceptual indexing and retrieval are made possible. In this study, the researcher used SVD for the approximation of a term-document matrix by one of low rank using singular value decomposition. The low-rank approximation yields a new representation for each document in a collection. Here the original term-document matrix (TFIDF), X was therefore approximated as Y (see fig. 3.3 below). From the collection of documents, a term-document matrix is computed in which each entry consists of a weight corresponding to a specific term in a specific document.

Fig. 3.3: Singular Values Decomposition (Hasan and Matsumoto, 2012)



The SVD, $X = T_o\ S_o\ D_o^T$ results, where *t x m* matrix, T_o, the orthonormal columns of which are called left singular vectors, an *m x m* diagonal matrix, S_o of positive "singular values" sorted in decreasing order, and an *m x d* matrix, D_o^T, the orthonormal columns of which are called right singular vectors. The value *m* is the rank of the matrix, X. With the T_o, S_o, and D_o matrices, X can be reconstructed precisely.

The key advantage of LSI in this study is to retain only the *r* (reduced-rank) largest singular values in the S_o matrix and set the others to zero. The value of *r* is a design parameter and small values are typically chosen. The resulting singular vector and singular value matrices were used to map term-based vectors for documents into a subspace in which semantic relationships from the term-document matrix are preserved and the original matrix, X is approximated by:

$$Y = T\ S\ D^T \qquad (3.7)$$

where T is a *t x r* matrix with orthonormal columns, S is a positive definite *r x r* diagonal matrix, and D^T is a *d x r* matrix with orthonormal columns.

Eigenvectors and eigenvalues are numbers and vectors associated to square matrices, and together they provide the eigen-decomposition of a matrix that analyzes the structure of this SVD matrix. Suppose that there exists a number $\lambda \in R$ and a non-zero vector $v \in R_n$ such $Av = \lambda v.$ Then we say that λ is an eigenvalue of the matrix A, and that *v* is an eigenvector corresponding to the eigenvalue λ. Then $Av = \lambda v = \lambda Iv$, where I is the *n x n* identity matrix, so that $(A - \lambda Iv = 0)$. Since $v \in R_n$ is non-zero, it follows that we must have $det(A\text{-}\lambda I)=0.$

However, one of the limitations of LSI is determining the number of dimensions to be removed (Hansaem and Kyunglag, 2015). There is no exact method to find the right dimensions. Deerwester *et al*., (1989) suggest that this choice captures the underlying semantic structure (i.e., the concepts) in the term-document matrix while rejecting the “noise” that results from term usage variations. In other words, the elimination of the small singular values reduces the documents feature space into a “document concept space”. Here in this study organized and decomposed term-document matrix after singular value decomposition and cluster generation procedure operates on reduced vectors or points of a *t*-dimensional space and the optimum dimension (10 for this research) for the collection was chosen by the criteria of ‘best retrieval performance’.

The following pseudo-code describes how the Latent Semantic Indexing algorithm works (Phadnis and Gadge, 2014);

1. Convert each document from test collection to a list of words (terms) and then into a vector of words.
2. Scale each vector of terms so that every term reflects the frequency of its occurrence in the document.
3. Combine these column vectors into a large term-document matrix. Rows represent terms and columns represent documents. The cells in the matrix are filled with *TFIDF* weights.
4. Perform SVD on the term-document matrix. This will result in three matrices commonly called $T_{o,}$ S_o and D_o^T
5. Set all singular values to 0 except the r highest singular values.
6. Recombine the terms T, an eigenvector of matrix S and document (D^T), to form reduced matrix, Y $=TSD^T$.

3.4.5. K-means Clustering

Document clustering has been investigated in different areas of text mining and information retrieval. Initially, document clustering was investigated for improving the precision and/or recall in information retrieval systems by automatically grouping documents that belong to the same topic in order to provide user's browsing of retrieval results (Adrian Kuhn *et al.*, 2006). It has always been used as a tool to improve the performance of retrieval and navigation of large data (Chartier, 2013). More recently, clustering has been proposed for use in browsing a collection of documents or in organizing the results returned by a search engine in response to a user's query (Ramampiaro and Paulsen, 2009).

From partition clustering techniques, K-means is widely used in document clustering. K-means is based on the idea that a center point can represent a cluster. A proximity distance measure is needed to assign data objects to the closest centroid. In the K-means algorithm, many different distance measures are possible, the most widely used one is cosine similarity. This is defined as;

$$\mathrm{sim}(\mathrm{d1},\mathrm{d2}\ ) \ = \frac{\mathrm{d1.d2}}{||\ \mathrm{d1}||\ ||\mathrm{d2}||} \quad = \frac{\sum_{\mathrm{j}=1}^{\mathrm{t}} d1 * d2}{\sqrt{\sum_{\mathrm{j}=1}^{\mathrm{t}} \left(\mathrm{d1}\right)^{2} * \sum_{\mathrm{j}=1}^{\mathrm{t}} \left(d1\right)^{2}}} \tag{3.8}$$

d1 and 2 are two representative points contain n number of values respectively.

Since LSI increases recall and hurt precision, the researcher was applied clustering to improve the precision of the information retrieval system. To do this, a reduced matrix Y= TSD^{T}, which was computed based on dimension reduction using *r* singular value was used. A cluster integration with reduced SVD matrix procedure operates on vectors or points of a *t*-dimensional space, where *t* is the number of terms (or concepts) with document indexed to applying K-means clustering. The standard K-means algorithm works as, given a set of document *D* and a pre-specified number clusters *k*, *k* data objects are randomly selected to initialize *k* clusters, each one being the centroid of clusters (Adrian Kuhn *et al.*, 2006). The remaining objects are then assigned to the cluster represented by the nearest or most similar centroid. Next, new centroids are recomputed for each cluster and run all documents are re-assigned based on the new centroids.

Algorithm: 3.5 K-means clustering algorithm (Ramampiaro and Paulsen , 2009).

```
FUNCTION kmeans ( )
    // INPUT: The initial number of k clusters; reduced document collection matrix,
    //OUTPUT: k clusters of documents
    Repeat
        For dj = 1 to n do //for each document in the collection
            $min_{sim}$ := sim(dj,$c_0$)
            t := 0
            For Ci = 1 to k do //for each cluster $C_i$
                sim (dj, ci); //similarity of document $d_j$ and centroid $c_i$
                If sim ($d_j$, $c_i$) < $min_{sim}$ then
                    $min_{sim}$ := sim($d_j$,$c_i$)
                    t := i
                End If
            End For
            $C_t$ = $d_j$ //Assign $d_j$ to the cluster $C_t$ with the closest centroid
        End For
        For i = 1 to k do
            Recalculate the positions of the centroid of $C_i$
        End For
    Until the global centroids is no longer move
        Return k clusters of documents
End function
```

To determine the optimal number of clusters, the researcher has selected the value of *k* at the "elbow", i.e. the point after which the distortion/inertia start decreasing linearly. Thus for the given data, to conclude that the optimal number of clusters the Sum Square Error elbow method was used. In the elbow method, the variance (within-cluster sum of squared errors) is plotted against the number of clusters (Wulandari, 2019). The first few clusters will introduce a lot of variance and information, but at some point, the information gain will become low, thus imparting an angular structure to the graph. The graph is available in Appendix I. The optimal number of clusters was found out from this point; therefore, this is known as the "elbow criterion." But this point cannot always be determined without any sense of ambiguity and the drawback of the elbow method is this ambiguity. The elbow method is easy to implement by looking at the ideal *k* value graph with the position on the elbow along with the SSE (Sum of Square Error) (Syakur *et al.*, 2018).

Performance indicators use a number of squared errors (SSE) and clusters are said to be convergent when obtaining a smaller value compared to others (Yuan and Yang, 2019). The Elbow algorithm method in K-Means as shown below (El-Mandouh *et al.*, 2019).

Algorithm 3.6 elbow algorithm (Syakur, *et al.*, 2018)

```
FUNCTION ELBOW ()
        //INPUT: range of cluster, document matrix
        //OUTPUT: optimal k
        Initialize k=1
         REPEAT
                Increment the value of k
                Measure the cost of the optimal quality solution (using sum of
                squares method)
                IF at the same cost the solution drops dramatically then
                        RETURN k //that's the true k
        UNTIL the solution drops
End function
```

3.4.6. Searching

In Information Retrieval (IR) task, searching involves issues of comparing queries with a collection of documents to locate a set of documents relevant to a particular query. In most cases, a query is treated as a small document consisting of a few terms, and the similarities between the query vectors and the document vectors are compared to find relevant documents. For searching purpose similarity measure between the query vector and documents vector are computed, then documents can be ranked (retrieved) in order of decreasing similarity to a query by computing normalized inner products (cosine similarity) on the term-based vectors (Baeza-Yates and Ribeiro-Neto, 1999);

$$\mathrm{sim}(\mathrm{dj}, \mathrm{q}) = \frac{\mathrm{dj.q}}{||\ \mathrm{dj}||\ ||\mathrm{q}||} = \frac{\sum_{j=1}^{t} W_{qj} * w_{dij}}{\sqrt{\sum_{j=1}^{t} \left(w_{qj}\right)^{2} * \sum_{j=1}^{t} \left(w_{dij}\right)^{2}}} \qquad (3.9)$$

Where, *d* is document, *q* is query from user, w_{qij} is the weight of the $_{j}th$ term of the $_{i}th$ query and w_{dij} is the weight of the $_{j}th$ term of the $_{i}th$ documents.

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them (Singhal, 2008). Cosine similarity then gives a useful measure of how similar two documents are likely to be in terms of their subject matter. In the case of information retrieval, the cosine similarity of two documents will range from 0 to 1, since the term frequencies (*tfidf* weights) cannot be negative. Vector space model (*TFIDF*, cosine similarity) that help to index and rank the document according to query matching and as a result gives the documents which most similar to the keyword.

3.4.6.1. Cluster based searching

Cluster-based retrieval uses the cluster hypothesis to retrieve a set of documents relevant to a query (Liu and Croft, 2004). In the Cluster hypothesis documents in the same cluster behave similarly concerning relevance to information needs (Voorhees, 2005). If a document in a cluster is relevant to a query, then the rest of the documents in that cluster are potentially relevant to the same query. According to Voorhees (2005), there are two approaches in cluster-based retrieval. The first approach retrieves one or more clusters relevant to a query instead of retrieving documents relevant to a query. In other words, this approach retrieves and ranks the relevant clusters instead of the relevant documents. The second approach uses the generated clusters as a reference for improving the retrieved relevant documents. In this approach, the given document collection is clustered (static clustering) beforehand. When a set of documents is retrieved for a query, the generated clusters (static clusters) of the collection are used as a reference to update the retrieved relevant document list.

On the other hand Jardine and Rijsbergen (1979) suggested that document clustering could be used to improve the effectiveness as well as the efficiency of retrieval, if the retrieval system were able to find good clusters; retrieval performance can be improved over document-based retrieval.

In this study document clustering has been performed, on the reduced low rank from SVD of LSI, independent of the user's query and the task for the retrieval system is to match the query against clusters centroid of documents instead of individual documents, and rank documents based on the similarity of clusters centroid to the query. After the first ranked cluster is known the documents inside the first ranked cluster are ranked based similarity to query and relevant to users as relevant documents. These show that the clusters rather than first-ranked clusters are not relevant to the user query.

4. RESULTS AND DISCUSSION

4.1. Overview

In this section, the source of document collections, the prototype of documents preprocessing, latent semantic indexing, clustering-based searching, relevance judgment, an experimental result (using VSM, SVD, SVD with K-means), performance comparison of proposed work and challenges were discussed in detail.

4.2. Document Corpus

To test the proposed system 243 Afaan Oromo news articles collected from OBN were used. The articles are taken from agriculture, politics, business, culture, education, health, religious, agri-business sport, and social aspects (see table 3.1 below). All articles are in *.txt* format and before applying experimentation data was prepared in an appropriate format.

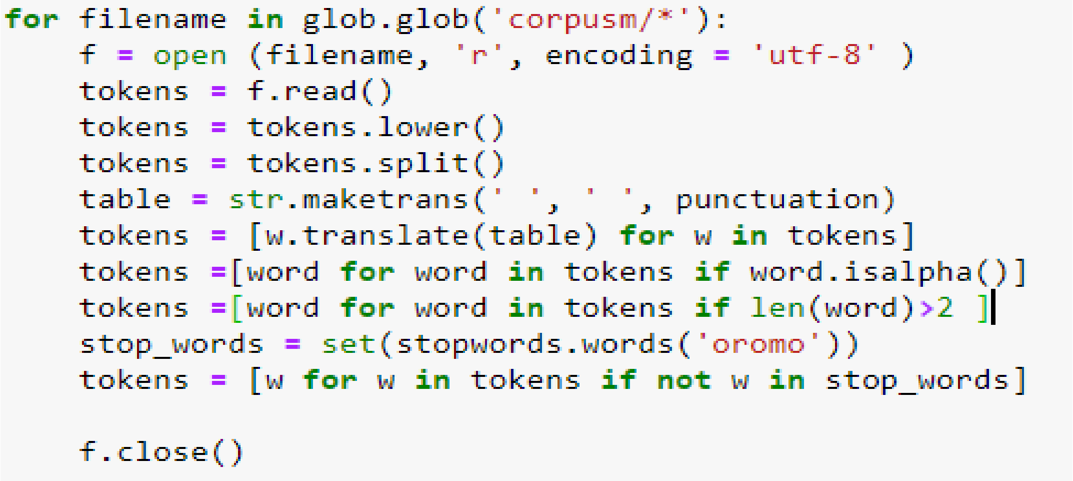
Table 4.1. List of documents collections

Document	**Frequency**	**Percentage (%)**
Agriculture	29	12.0
Agri-business	16	6.6
Business	8	3.2
Culture	15	6.1
Education	27	11.1
Health	9	3.7
Religious	18	7.4
Sport	12	4.9
Social	28	11.6
Politics	81	33.4
Total	243	100

4.3. Documents Preprocessing

The prototype system development begins with preprocessing documents and then, constructing an inverted file consisting of a vocabulary file and posting file, for indexing and organizing documents to speed up searching. Preprocessing tasks include tokenization, normalization, stop word removal, and stemming, then *TFIDF* term weighting has been employed. The system process the index terms and queries by using python 3.5. For this purpose, code 4.1 were used to preprocessing.

Code 4.1. A Program for Tokenization, Normalization and Stop Word Removal

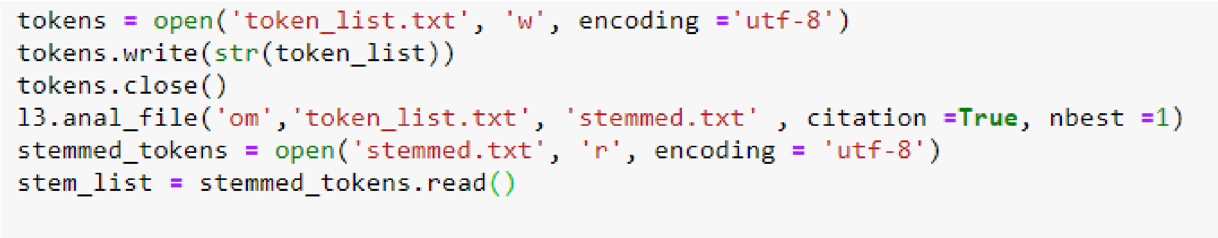


```
for filename in glob.glob('corpusm/*'):
    f = open (filename, 'r', encoding = 'utf-8' )
    tokens = f.read()
    tokens = tokens.lower()
    tokens = tokens.split()
    table = str.maketrans(' ', ' ', punctuation)
    tokens = [w.translate(table) for w in tokens]
    tokens =[word for word in tokens if word.isalpha()]
    tokens =[word for word in tokens if len(word)>2 ]
    stop_words = set(stopwords.words('oromo'))
    tokens = [w for w in tokens if not w in stop_words]

    f.close()
```

There were challenges encountered by the researcher during tokenization, normalization, and stop word removal. One of the challenges related to tokenization is differentiating single word from compound word, for instance, *AddisAbaba* versus *Addis Ababa*, *DirreeDawaa versus Dirree Dawaa*, *AbbaaGadaa versus Abbaa Gadaa*, *waljaalachuun versus wal-jaalachuun.* The other challenge is identifying numerical values like dates, phone numbers, and IP addresses, and apostrophes that are used in Afaan Oromo as a letter in the same words (i.*e.*, *ta'an si'a mo'achuudhaaf*, *waa'ee*); however, it is removed during tokenization. On the other hand, absence of a standard stop-word list was a problem faced during stop word removal. This was forced the researcher to use stop word used by previous research and added missing stop words in consultation with language experts.

Stemming is language-dependent that reduces tokens to their root form of words to recognize morphological variations. HornMorpho is stemmer based on context. Therefore the researcher was used this stemmer for conflating words in Afaan Oromo documents, as shown below in code, 4.2.

Code 4.2. Stemming using HornMorpho



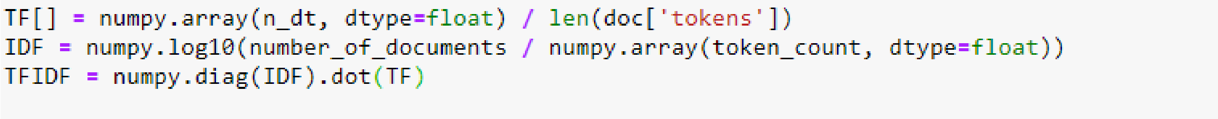
```
tokens = open('token_list.txt', 'w', encoding ='utf-8')
tokens.write(str(token_list))
tokens.close()
l3.anal_file('om','token_list.txt', 'stemmed.txt' , citation =True, nbest =1)
stemmed_tokens = open('stemmed.txt', 'r', encoding = 'utf-8')
stem_list = stemmed_tokens.read()
```

In the HornMorpho system for morphological processing linguistic consideration or irregular morphological features of the language, query expansion, and the problem of the compound word are not fully considered. There is limited number of available roots and stems, those when the root is not known the verb guesser analyzer produces different analyses when in many cases only one of these contains a root that actually exists in the language. However, the guesser analyzer itself is a useful tool for extending the lexicon; when an unfamiliar root is found in multiple word-forms and in multiple morphological environments, it can be safely added to the root lexicon. There is also the problem of handling ambiguity. Even when a root or stem is known, there are often multiple analyses, and the program provides no information about which analyses are more likely than others.

4.4. Single Value Decomposition with K-means Clustering

Term weighting, Terms that were in the documents or in the query needed to get hold of weight with respect to their documents. After inverted index file was created, the weight can be calculated using *TFIDF* term weighting methods and SVD was applied on this matrix. The term weighting was done by finding the frequency of terms and its inverse document frequency using code, 4.3 shown below.

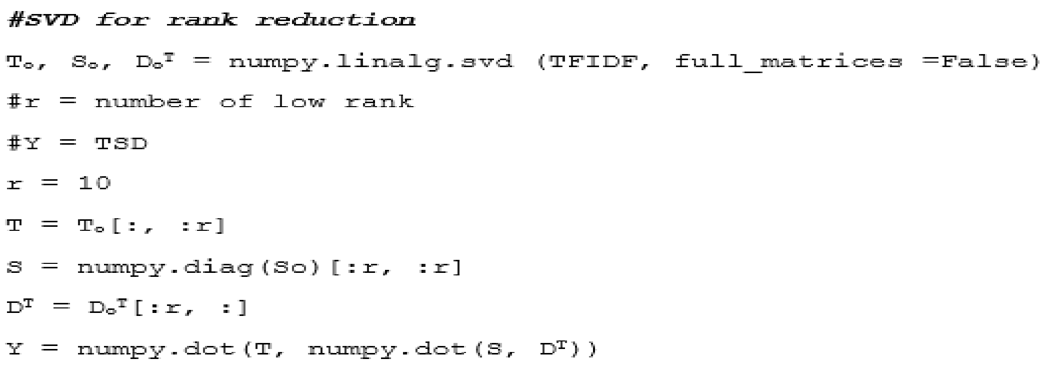
Code 4.3. TF, IDF and TFIDF using numpy built in algorithm.



```
TF[] = numpy.array(n_dt, dtype=float) / len(doc['tokens'])
IDF = numpy.log10(number_of_documents / numpy.array(token_count, dtype=float))
TFIDF = numpy.diag(IDF).dot(TF)
```

Singular Value Decomposition (SVD) was used to estimate the structure in word usage across documents, and statistically derived vectors are more robust indicators of meaning than individual terms. The weighted term-document matrix (*TFIDF*) is given to the SVD algorithm as an input and a reduced dimensional representation of the matrix is generated. The SVD of this term-document matrix was then computed and small singular values are eliminated from the singular value matrix. The resulting singular vector and singular value matrices are used to map term-based vectors for documents into a subspace in which semantic relationships from the term-document matrix are preserved. The SVD of the matrix was computed for a range of dimensions and the optimum dimension (10 for this research) for the collection was chosen by the criteria of 'best retrieval performance.'

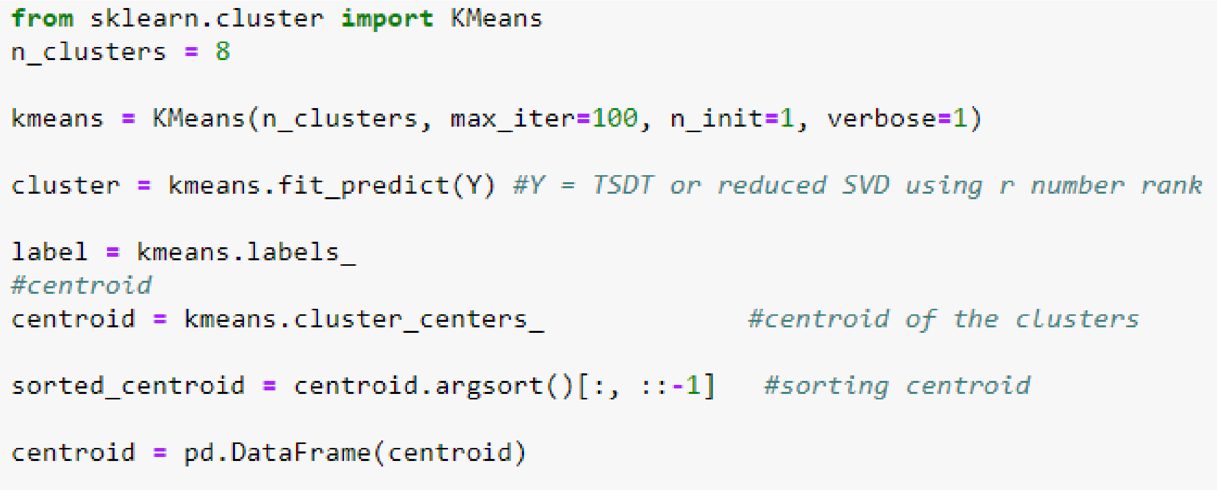
Code 4.4 Single Value decomposition



```
#SVD for rank reduction
To, So, Do^T = numpy.linalg.svd (TFIDF, full_matrices =False)
#r = number of low rank
#Y = TSD
r = 10
T = To[:, :r]
S = numpy.diag(So)[:r, :r]
D^T = Do^T[:r, :]
Y = numpy.dot(T, numpy.dot(S, D^T))
```

K-means clustering algorithm was used for document clustering. To determine the number of clusters, the researcher used the value of optimal-k found at elbow point after applied the method in K-means. Based on this elbow method the number of clusters for documents used in this study was 8 as indicated in Appendix I. The cluster was performed on reduced single value decomposition (after LSI). Thereby documents which have higher cosine similarity score with their centroid are under the same cluster. To perform this, K-means clustering algorithm from sklearn library was used as shown in code below.

Code 4.5 K-means clustering using python sklearn library



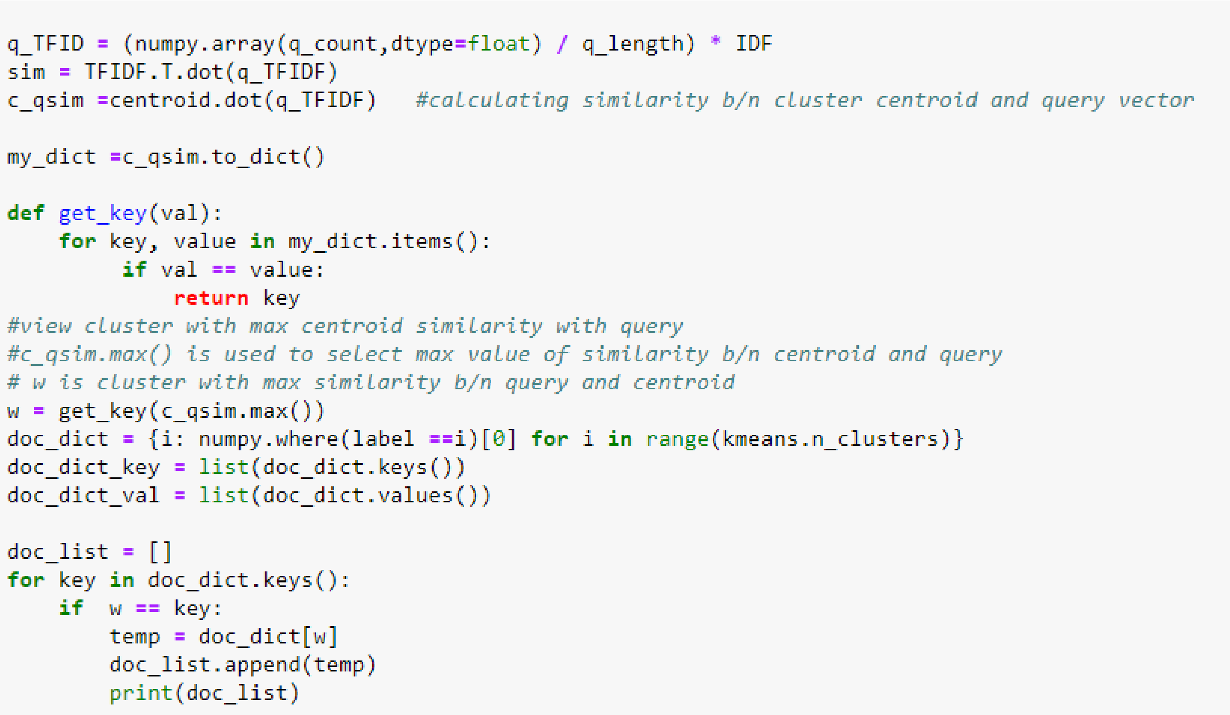
```
from sklearn.cluster import KMeans
n_clusters = 8

kmeans = KMeans(n_clusters, max_iter=100, n_init=1, verbose=1)

cluster = kmeans.fit_predict(Y) #Y = TSDT or reduced SVD using r number rank

label = kmeans.labels_
#centroid
centroid = kmeans.cluster_centers_              #centroid of the clusters

sorted_centroid = centroid.argsort()[:, ::-1]   #sorting centroid

centroid = pd.DataFrame(centroid)
```

Based on cosine similarity between query and clusters centroid relevant documents are retrieved. However, there is a major problem encountered during experimentation. Thus, Re-computing the SVD with larger data term-document matrix requires more computation time and, it may be impossible due to memory constraints as well as massive datasets are inefficient in K-means clustering. On the other hand, even if the researcher tried to use the elbow method to determine the optimal number of cluster, there also challenges to define exact optimum number of clusters. Here elbow is also needs to determine the range of clusters.

4.5. Clustering Based Searching

Searching compare query with a collection of documents representative to locate a set of relevant documents that satisfies information need of user. The point is if a cluster was relevant to a query, then the documents in that cluster were potentially relevant to the same query. Based on the cluster hypothesis, documents from the top ranked cluster are relevant to a query than the documents from the clusters with a least rank. For searching purpose similarity measure between the query vector and clusters centroid vector was computed, then documents were ranked in order of decreasing similarity to a query by computing normalized inner products (cosine similarity). After the similarity between the centroids vector and the query vector is performed, then clusters are ranked based on similarity to the query. Finally, documents in the first ranked cluster are relevant to users query and returned in response to users.

Code 4.6 Clustering based searching



```
q_TFID = (numpy.array(q_count,dtype=float) / q_length) * IDF
sim = TFIDF.T.dot(q_TFIDF)
c_qsim =centroid.dot(q_TFIDF)   #calculating similarity b/n cluster centroid and query vector

my_dict =c_qsim.to_dict()

def get_key(val):
    for key, value in my_dict.items():
         if val == value:
             return key
#view cluster with max centroid similarity with query
#c_qsim.max() is used to select max value of similarity b/n centroid and query
# w is cluster with max similarity b/n query and centroid
w = get_key(c_qsim.max())
doc_dict = {i: numpy.where(label ==i)[0] for i in range(kmeans.n_clusters)}
doc_dict_key = list(doc_dict.keys())
doc_dict_val = list(doc_dict.values())

doc_list = []
for key in doc_dict.keys():
    if  w == key:
        temp = doc_dict[w]
        doc_list.append(temp)
        print(doc_list)
```

4.6. Performance Measure

In this research, the researcher attempting to design and implement Semantic Indexing with and Document Clustering based searching for the Afaan Oromo information retrieval. As discussed in previous chapters, the system aims to overcome the semantic problems exhibited in indexing and similarity measurement. In this experiment 10 queries are used for evaluating the performance of the system and experimented at three-level using; VSM, SVD, and SVD with K-means clustering. Experimentation was done for identifying which documents were relevant for a given test query and a result of each query was presented in the form of precision, recall, and F-measure.

4.6.1. Query Selection and Relevance Judgment

Ten (10) appropriate queries, which able to describe the document are systematically and subjectively selected after reviewing the content of each article manually to test the performance of the system. These queries were used with different techniques to measure performance and queries are marked across each document as either relevant or irrelevant to make relevance evaluation. All queries and documents should be stored in a form that is used for the term-document matrix. *TFIDF* is assigned as a weight of the term-document matrix. Subjective relevance judgment of each document was identified as a relevant or non-relevant to each selected query for the experiment as listed below in table 4.2.

Table 4.2. Proposed system relevance judgment

No	Queries	Relevant	Non-relevant
1	Barnoota	32	211
2	Nageenya	38	205
3	sirna gadaa	21	222
4	hidha haaromsaa	130	113
5	balballomii ispoortii	12	231
6	rakkinoota hawaasa kessaa jiran	28	215
7	qonna bultooni beenya lafaa fudhatanii	30	213
8	hojii mootumman misooma, bulchinsa gaarii diriirsuu	23	220
9	Bulchiinsa Magaalaa kessaatti Qaamoleen hawaasaa gahee olaanaa qaba	35	208
10	labsiin yeroo ariifachisaa labsame nageenya biyyaa eegsisuu keessatti gahee olaanaa qaba jedhan	30	213

4.6.2. Experimental Result Using VSM

The researcher conduct an experiment to measure the effectiveness of vector space model (VSM) in retrieving relevant Afaan Oromo documents for the given 10 queries. Detailed experimentation result using VSM is depicted in table 4.3 below.

Table 4.3: Effectiveness of VSM based Afaan Oromo IR

No	Queries	R	P	F
1	Barnoota	0.40	0.30	0.30
2	Nageenya	0.40	0.20	0.26
3	sirna gadaa	0.43	0.30	0.20
4	hidha haaromsaa	0.7	0.32	0.20
5	balballomii ispoortii	0.20	0.13	0.15
6	rakkinoota hawaasa kessaa jiran	0.50	0.21	0.30
7	qonna bultooni beenya lafaa fudhatanii	0.23	0.24	0.20
8	hojii mootumman misooma, bulchinsa gaarii diriirsuu	0.70	0.40	0.20
9	Bulchiinsa Magaalaa kessaatti Qaamoleen hawaasaa gahee olaanaa qaba	0.80	0.20	0.30
10	labsiin yeroo ariifachisaa labsame nageenya biyyaa eegsisuu keessatti gahee olaanaa qaba jedhan	0.63	0.20	0.28
	Average	**0.50**	**0.25**	**0.24**

As shown in table 4.3 the average result obtained using VSM is 50%, 25%, 24% recall, precision, and F-measure respectively. The performance of the system was lowered for the reasons why VSM only retrieve relevant documents based on exact matching between the index and query terms. This means VSM has no mechanism to manage polysemy and synonymy. On the other side problem of morphological inflection nature of the Afaan Oromo writing system for the purpose of number, genders, possession, plural form, proposition, and conjunctions lowered VSM performance. For this experiment, search results sample using the first query 'barnoota' (see appendix III) was 46 documents.

4.6.3. Experimental Result Using SVD

The next experiment conducted in this study was SVD based latent semantic indexing. Accordingly, this was approach evaluated using same 10 queries used for in VSM retrieval. Table 4.4 presents retrieval result using SVD.

Table 4.4: Retrieved result to query using SVD

No	Queries	R	P	F
1	Barnoota	0.75	0.20	0.30
2	Nageenya	0.95	0.20	0.30
3	sirna gadaa	0.98	0.18	0.20
4	hidha haaromsaa	0.86	0.50	0.64
5	balballomii ispoortii	0.58	0.40	0.38
6	rakkinoota hawaasa kessaa jiran	0.96	0.20	0.30
7	qonna bultooni beenya lafaa fudhatanii	0.8	0.20	0.25
8	hojii mootumman misooma, bulchinsa gaarii diriirsuu	0.95	0.20	0.30
9	Bulchiinsa Magaalaa kessaatti Qaamoleen hawaasaa gahee olaanaa qaba	0.98	0.25	0.34
10	labsiin yeroo ariifachisaa labsame nageenya biyyaa eegsisuu keessatti gahee olaanaa qaba jedhan	0.96	0.20	0.25
	Average	0.88	0.25	0.34

As shown in Table 4.4 the proposed approach registers an average recall, precision, and F-measure of 88%, 25%, and 34% respectively. In this method highest recall is registered as compared to precision and F-measure. This result is registered because of the fact that the SVD method retrieves as many documents as possible that are considered relevant for the given query. For example the search result for sample query term ‘barnoota’ (see appendix III) was 158 documents. Obviously, as the number of documents increases recall increases at the expense of precision. This means some queries request an answer to specific issues and demand the returned documents to focus exclusively on that particular issue and this advantage of the LSI method can sometimes result in spurious answers. For instance; when the search for documents using ‘rakkinoota hawaasa kessaa jiran’, and ‘balballomii ispoortii’ the system using SVD has retrieved 236 and 162 documents respectively from 243 collections of documents used. This was provided the solution for the conceptual matching problem, while it indicates the number of recall increases and affects the precision.

4.6.4. Experimental Result Using Semantic Indexing and Document Clustering Based Searching

To handle the weakness of VSM and SVD, the researcher integrate SVD and document clustering. To do this document clustering has been applied, on the reduced low rank from SVD matrix of LSI, independent of the user’s query and the task for the retrieval system is to match the query against clusters centroid of documents instead of individual documents, and rank documents based on the similarity of clusters centroid to the query. On the other hand, searching is being processed by using a select query term. Finally, the cosine similarity measure identifies a relevant document and the result obtained in the experiments indicates the IR system build based on semantic indexing based document clustering registered better result than VSM and SVD (see appendix III). After the first ranked cluster is known the documents inside the first ranked cluster are ranked based on similarity to query and relevant to users as relevant documents. The performance of the proposed approach using SVD based K-means clustering is presented in table 4.5.

Table 4.5: Performance of SVD based K-means clustering for Afaan Oromo IR

No	**Query**	**R**	**P**	**F**
1	Barnoota	0.72	0.88	0.79
2	Nageenya	0.60	0.80	0.70
3	sirna gadaa	0.70	0.90	0.80
4	hidha haaromsaa	0.80	0.60	0.70
5	balballomii ispoortii	0.83	0.70	0.74
6	rakkinoota hawaasa kessaa jiran	0.40	0.73	0.51
7	qonna bultooni beenya lafaa fudhatanii	0.6	0.90	0.72
8	hojii mootumman misooma, bulchinsa gaarii diriirsuu	0.70	0.80	0.74
9	Bulchiinsa Magaalaa kessaatti Qaamoleen hawaasaa gahee olaanaa qaba	0.50	0.70	0.60
10	labsiin yeroo ariifachisaa labsame nageenya biyyaa eegsisuu keessatti gahee olaanaa qaba jedhan	0.83	0.92	0.87
	Average	0.70	0.80	0.72

Proposed prototype of Afaan Oromo information retrieval system was developed based on the vector space of information retrieval model, while to improve system performance the researcher was applied SVD and K-means clustering on the VSM matrix. The system has been recorded an average 70% recall, 80% precision, and 72% F-measure. However, their major problem encountered during experimentation were computation time for large collections, and it may be impossible due to memory constraints as well as massive datasets are inefficient in K-means clustering. The other problem of SVD was representing inherent semantics of the document, and to define a similarity measure based on the semantic representation such that it assigns higher numerical values to document pairs which have a higher semantic relationship.

4.6.5. Performance Comparison

After conducting a series of experiments using the selected 10 queries, the researcher selected the best performing approach based on their F-Measure that enables us to see to what extent both recall and precision of the retrieval result improved. Hence Table 4.6 below shows a comparison of the three approaches (VSM, SVD, and SVD with K-means) for designing a prototype Afaan Oromo information retrieval (AOIR).

Table 4.6: Summarized result of overall performance of Afaan Oromo IR system

IR effectiveness measures	VSM based AOIR	SVD based AOIR	SVD with K-means based AOIR
Recall	0.5	0.88	0.70
Precision	0.25	0.25	0.80
F-measure	0.24	0.34	0.72

The above table 4.6 showed that, while the proposed system score 72% F-measure, vector space model (VSM) and single value decomposition (SVD) model registers of 24% and 34% F-measure respectively. The proposed approach scores better performance since it attempts to measure semantic relation of words, rather than matching words based on the shape similarity of words, this further shows that the proposed approachable to control synonym and polysemy of words thereby improving the precision and recall of the retrieval system. As a result, the researcher proposed SVD with K-means clustering for designing the Afaan Oromo information retrieval system.

4.7. Discussion of Result

Previously, different efforts have been put to fill or mitigate the gap between the individual information needs and the provision of Afaan Oromo information retrieval. In addition to comparison between three level results of VSM, SVD and SVD with K-means, the researcher also tried compered this study with previous work done as follows.   This research records better performance with an improvement by 16% F-measure as compared to the work of Melkamu, (2017), who integrate WordNet based query expansion for enhancing the effectiveness of Afaan Oromo information retrieval. The approach followed by Melkamu may help to control synonym words of Afaan Oromo, but not polysemy words. This study also registered better performance than Berhanu, (2019) that explored the potential application of Latent Semantic indexing approach in Afaan Oromo text retrieval by 7% recall and 13% precision respectively. Furthermore this research scored better performance by 6% when compared to Ashenafi, (2019) work, attempts to extend the application of word sense disambiguation based query expansion and registered 66% of F-measure.

This research attempts further to integrate semantic indexing and cluster based searching to control the effect of polysemy and synonym Afaan Oromo words, as a result of which a promising result obtained for enhancing effectiveness of the IR system.

However, there were various challenges encountered during this study. The challenge started with the complexity of the Afaan Oromo linguistic feature itself. Even if the researcher tried to keep the semantics of the documents by extracting the statistical value of the term-document matrix using SVD. Afaan Oromo linguistic computational level of the language was not matured and getting the semantic meaning of the language was a bit difficult. If this problem is not addressed well, it is difficult to get correct semantics of the Afaan Oromo documents, since it needs well organized Afaan Oromo ontology to identify semantic relation (polysemy and/or synonymy) of words correctly. The other challenge was that there is no standard and organized documents for experimenting and measuring the effectiveness of the proposed approach in this study, here the researcher used documents collected from OBN daily news and re-organized based on their content, while it is not enough for better results. The third challenge was stemmer, there is no suitable stemmer that is available for Afaan Oromo; however, though the researcher used HornMorpho it does not still stem all words properly. There is also a problem related to the clustering algorithm. When the documents were not clustered properly, the system was limited to look for the relevant documents. Mis-clustering of documents occurred because a document might have more shared terms with one cluster but it was talking about the other cluster. For instance, if the document dealt with business and agriculture issues, agro-business has many terms of business or agriculture whereas the document has to cluster in business may concern about agriculture and visa verse.

5. CONCLUSION AND RECOMMENDATION

5.1. Conclusion

The purpose of an Information Retrieval (IR) system is to process a collection of documents as per requests for information, identify and retrieve from the corpus certain documents in response to the information requests. Since main objective of this research study was to address the semantics problem related to Afaan Oromo information retrieval and enhance the information retrieval system through the latent semantic indexing and document clustering approach. The previous research works in the Afaan Oromo information retrieval system were attempted to address different problems such as word-sense disambiguation, query expansion, latent semantic indexing, but not all are appropriately concerned with semantic relation and clustering to improving retrieval system performance. Therefore in this study, an attempt has been to design and develop an improved information retrieval system which is called enhanced Afaan Oromo Information retrieval. As it has been discussed in the previous sections, the designed and prototyped system mainly consisted of three components; latent semantic indexing, K-means clustering, and searching.

The indexing part of the work involves pre-processing, such as tokenization, normalization, stopwords removal, stemming, and weighting. The analysis begins with a weighted term-document matrix and this weighted matrix was used for indexing and retrieval. The weighted matrix was analyzed further by applying the SVD to derive the latent structure model which was used for indexing by using the LSI model. Document clustering is a sub-component of the system which aimed to cluster related documents into one category to facilitate searching. For documents clustering purposes K-means clustering was employed and total document provided to the system clustered in eight groups and these clusters centroids is used to retrieve documents based on a similarity measure. Searching mainly consists of query pre-processing, similarity measure, and retrieving the relevant document. Cosine similarity was measured between the query vector and cluster centroid vectors that were used to retrieve relevant documents to each query.

To evaluate the performance of the system 243 document collections are randomly selected from corpus collected from OBN with appropriate 10 queries. Based on experimentation the proposed approach for developing Afaan Oromo information retrieval registered 70% recall, 80% precision, and 72% F-measure. Finally, the experimentation indicated, the proposed system using Latent Semantic Indexing and Document Clustering based searching was improved the performance of the system in terms of precision, recall, F-measure, and reduce dimension when compared to VSM and SVD model. This is a promising result to design an applicable information retrieval system.

5.2. Recommendation and Future Direction

The proposed Afaan Oromo retrieval system was attempted to enhance the performance of the retrieval system by considering the semantic relation of words in documents. In this paper, the researcher addressed Semantic document clustering of a document collection using inter-document similarities derived from the term-document matrix. The need to investigate semantic document clustering is that documents that are semantically related to each other are grouped into the same cluster and documents that are semantically unrelated are grouped into another cluster.

Finally, based on the finding of the study, in order to obtain the optimum performance following recommendations are recommended.

The main objective of information retrieval system is a well-organized corpus and pre-processing based on the language knowledge. But there is no standard corpus, stemmer, tokenize, normalizer, and stop word detector developed for Afaan Oromo and spelling checker also being important for indexing and searching to write correct index terms or queries.

Advanced information retrieval system uses resources like a thesaurus, WordNet, machine-readable dictionaries, and a better latent semantic indexing system which is based on polysemy and synonyms of words rather than using statistically extracting semantic relation. Hence, the researcher recommends future research to implement context-aware IR system using a well-developed thesaurus or WordNet to aware of the context to which the terms in the text appear to enhance precision and recall of the system.

The other recommendation is extending this study using another clustering algorithm such as hierarchical and lexical chain clustering to handle the relationship between words used for explaining different domains.

On the other hand, still, Afaan Oromo retrieval systems were developed for textual, but there is also a need to further investigate the development of retrieval system for video, audio, and images.

6. REFERENCES

Abara Nefa. 1988. Long Vowels in Afaan Oromo: A Generative Approach. *MSc Thesis Addis Ababa University, Ethiopia*.

Abera Diriba. 2009. Automatic Classification of AFaan Oromo News Text:In Case of Radio Fana. *Msc Thesis Addis Ababa University, Ethiopia*.

Abey Bruck. 2011. Semantic Based Query Expansion Technique for Amharic IR. *MSc Thesis, Addis Ababa, Ethiopia*.

Adrian Kuhna, Stephane Ducasseb, Tudor Girba. 2006. Semantic Clustering: Identifying Topics in Source Code. *Journal of Information Systems and Technologie, France*.

Amira M. El-Mandouh Laila A. Abd-Elmegid, Hamdi A. Mahmoud, Mohamed H. Haggag. 2019. Optimized K-means clustering model based on gap statistic. *International Journal of Advanced Computer Science Application, Canada* , *10*(1): 183-188.

Andreas Hotho, Steffen Staab, Gerd Stumme. 2003. Wordnet improves Text Document Clustering. *In proc. of the SIGIR, Semantic Web Workshop, Germany*.

Artem Polyvyanyy, Dominik Kuropka. 2007. A Quantitative Evaluation of the Enhanced Topic-Based Vector Space Model. *Journal of Hasso planttner Institution, Potsdam,*.

Ashenafi Tulu. 2019. Word Sense Disambiguation Based Query Expansion for Improving the Performance of Afaan Oromoo Information Retrieval System, *Msc. Thesis, Haramaya University, Ethiopia*

Aurélie Névéol, James G. Mork, Alan R. Aronson. 2008. Automatic Indexing of Specialized Documents:Using Generic vs. Domain-Specific Document Representations. *National Library of Medicine,USA*.

Bader Aljaber, Nicola Stokes, James Bailey Jian Pei. 2009. Document clustering of scientific texts using citation contexts. *Springer Science Business Media, LLC, Canada*.

Baeza-Yates and Ribeiro-Neto. 1999. Modern Information Retrieval. *Journal of the Association for Computing Machinry. Cambridge University,, USA.*

Balwinder Saini, Vikram Singh,Satish Kumar. 2014. Information Retrieval Models and Searching Methodologies: Survey. *International Journal of Advance Foundation and Research in Science & Engineering ,India, 1*(2).

Barbara Rosario. 2000. Latent Semantic Indexing: An overview. *Springer, USA*.

Berhanu Anbase. 2019. Applications of Information Retrieval for Afaan Oromo text based on Semantic-based Indexing. Msc Thesis, Jimma University

Bethlehem Mengistu Hailemariam. 2002. N-gram-Based Automatic Indexing for Amharic Text. *Msc thesis school of information science Addis Ababa, Ethiopia*.

Bjornar Larsen and Chinatsu Aone. (1999). Fast and Effective Text Mining Using Linear-time Document Clustering. *Association for Computing Machinry*. *Cambridge University,, USA.*

Bo-Yeong Kang.(2013). A Novel Approach to Semantic Indexing Based on Concept, Department of Computer Engineering. *Kyungpook National University, Metropolitan*.

C. J. van Rijsbergen. 1979. Information Retrieval. *Elsevier Inc., USA*.

Cambridge . 2009. The term vocabulary and posting lists *Cambridge University, New-York*.

Cambridge. 2009. Flat clustering. *Cambridge University, New York*.

Cambridge. 2010. Evaluation of information retrieval system. *Cambridge University, New York*.

Cambridge. 2010. Performance of Two Statistical Indexing Methods, with and without Compound-word Analysis. *Cambridge University Press, New York*.

Catherine Griefenow-Mewis . 2001. A Grammatical Sketch of Written Oromo. *Koln Rudiger,* Switzerland

Charles L.A. Clarcke, Gordon V. Cormack. 1995. Dynamic Inverted Index for a Distributed Full text Retrieval System. *Journal of Tech. Rep MT, University of Waterloo, Canada*.

Choudhary and Bhattacharyya. 2002. Text clustering using semantics. *in Proceeding of the 11th International World Wide Web Conference, Bombay, India.*

Chris Buckley and Alan.F. Lewit. 1998. Optimizations of inverted vector searches In *Proceeding of the 8th International ACM-SIGIR Conference on Research and Development in information Retrieval USA: 97-110.*

Chris H.Q. Ding. 1999. A Similarity-based Probability Model for Latent Semantic Indexing. *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval, USA: 59-65.*

Christiane Fellbaum. 1998. WordNet: An Electronic Lexical Database. *Cambridge, MA. MIT Press, USA*

Christopher D. Manning, Prabhakar Raghavan, Hinrich Sch¨utze, D. 2011. an introduction to information retrieval. *Cambridge University press, New York*.

Christopher D. Manning, Prabhakar Raghavan, Hinrich Sch¨utze. 2009. Introduction to information retrieval.(second, Ed.) *Cambridge University press, New York,* .

Christopher D. Manning, Prabhakar Raghavan, Hinrich Sch¨utze. 2008. Introduction to Information Retrieval. (Third, Ed.) *Cambridge University Press, New York*.

Christopher Issal Magnus Ebbessoni. 2010. Document Clustering. *Master of Science Thesis, University of Gothenburg*, Sweden

ChunhuiYuan and Haito Yang. 2019. Research on K-Value Selection Method of KMeans Clustering Algorithm. *Journal of Multidisciplinay Science, China. 2*(2), (pp. 226-235).

Daniel Bekele Ramesh Babu, Dereje Teferi. 2015. A Cross Lingual Information Retrieval (CLIR) System for Afaan Oromo-English using a Corpus Based Approach. *International Journal of Engineering Research & Technology,Open Access, 4*(5).

Daniel Bekele Ayana. 2011. Afaan Oromo-English Cross Lanaguage Information Retrieval (CLIR): A Corpus Based Approach. *Msc Thesis, Addis Ababa, Ethiopia*.

Daphe Koller.1997. Hierarchically classifying documents using very few words. *Proceedings of the 14th International Conference on Machine Learning,ResearchGate*. 170-178.

Deerwester Scott, Susan T. Dumais, Richard Harshman. 1989. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science, Pennsylvania State University.*

Devika Deshmukh. Sandip Kamble Pranali Dandekar. 2013. Survey on Hierarchical Document Clustering Techniques. *International Journal of Advanced Research in Computer Science and Software Engineering, 3*(7).

Dinakar Jayarajan, Dipti Deodhare, B. Ravindran Sandipan Sarkar. 2000. Document Clustering using lexical chains. *Illinois Institute of Technology,India.*

Edie, Rasmussen. 2015. Clustering Algorithm. *Journal of American Society for Information Science, University of Pittsburgh USA 35*(1): 268-276).

Ellen M. Vdorhees. 2005. The cluster hypothesis revisited. *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, USA. 85*(1): 188-196.

Encyclopedia Britannica 2018. Oromo Language. *Retrieved September 24, 2010, from: DOI:https:// britannica.com/topic/Oromo*

Eyob Nigussie. 2013. Afaan Oromo-Amharic cross lingual information retrieval: corpus based approach. *Msc. thesis Addis Ababa University, Ethiopia*.

Feyisa Demie. 1996. Historical Challenges In The Development Of The Oromo Language. *Journal of oromo studies, Middle Tennessee State University, USA 3(2)*:18-27.

Fuhr and Buckley. 1991. A Probabilistic Learning Approach for Document Indexing. *ACM Transactions on Information Systems, 9*(3): 224-248.

Gasser, M. (2011). HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya*: Indiana University, India.*

Gerard Salton, Michael J. McGill. 1983,. Introduction to Modern Information Retrieval. Journa*l of McGraw-HillComputer Science Series, McGraw-Hill Book Company, New York*

Gerard Salton, Michael J. McGill. 1986. Introduction to modern information retrieval. *McGraw-Hill, Inc., USA*

Gezehagn Gutema. 2012. Afaan Oromo Text Retrieval System. *Msc Thesis, Addis ababa Universiity, Ethioppia* .

Giuliano Antoniol, Gerardo Canfora, Gerardo Casazza, Andrea De Lucia, Ettore Merlo. 2002. Recovering traceability links between code and documentation. *Transactions on Software Engineering, IEEE 28*(10): 970-983.

Greengrass, Ed. 2001. A survey of Information Retrieval. *Survey, RsearchGate*.

Grossman, David A., Frieder, Ophir. 2004. Information Retrieval Algorithms and Heuristics. *Springer Science+Business, Springer Netherlands.*

Halterman, Richard. 2011. Learning to Program with Python. *Southern Adventist University, Collegedale, Tennessee*

Hansaem Park, Kyunglag Kwon, Abdel-ilah Zakaria Khiati, Jeungmin Lee, and In-Jeong Chung. 2015. Agglomerative Hierarchical Clustering for Information Retrieval Using Latent Semantic Index. *International Conference on Smart City, IEEE*.

Howard R.Turtle and W. Bruce Croft. (1992). A comparison of text retrieval models. *Computer Journal, Elsevier, USA. 35*(3): 279-290.

Howard Robert Turtle, W. Bruce Croft. 2017. Inference Network For Document Retrieval. *ACM SIGIR*, 51(1)

Ibrahim Bedane. 2015. The Origin of Afaan Oromo: Mother Language. *Global Journal of Human Social Science: Linguistics and Education,Global Journals Inc., USA 15*(12).

James H. Martin, Daniel Jurafsky 2008. Discovery of WordNet Relations. *MIT Press Journals,USA.*

Juban and Falguni. 2007. Search Algorithms an Aid to Information Retreival in Digital Libraries. *5th International CALIBER, Indian Statistical Institute, India*: 401-414.

Judit Bar-Ilan and Tatyana Gutman. 2003. How do Search Engines Handle Non-English Queries? - A case study. *In Proceedings of the Alternate Papers Track of the 12th., Budapest, Hungary*.

K.Halliday Ma and Ruqaiya Hasan. 1976. Cohesion in English. *English Language Series, USA*.

kerlinger, F.N, and Lee, H.B. 2000. Foundation of Behavioral research. *harcourt college publisher Fort Worth Texas, USA:*

Kerry Tanner . 2018. Research Methods. (second, Ed.) *ScienceDirect*, Elsevier (pp. 159-192).

Khaled Hammouda and Mohamed Kamel. 2010. Collaborative Document Clustering. *Journal of Machine Intelligence Research Group, University of Waterloo, Canada* : 451-461.

Kothari, C. R. 2004. Research Methodology. (2nd Ed.). *New Age International Publishers, New Age International, New Delhi*

Kowalski, Gerald J. 1997. Information Retrieval Systems: Theory and Implementation. *springer*

Kula Kekeba Tune, Vasudeva Varma, Prasad Pingali 2007. Oromo-English Cross Language Information Retrieval. *Journal of IJCAI, Springer, Berlin, Germany*

Laurence Morissette and Sylvain Chartier. 2013. The k-means clustering technique: General considerations and implementation in Mathematica . *ResearchGate, Berlin, Germany. 9*(1): 15-24.

Lei Shi and Rada Mihalcea. 2005. Putting pieces together:Combining FrameNet, VerbNet and *WordNet for robust semantic parsing. In Computational linguistics and intelligent text processing, University of North Texas, Texas*.100-111.

Lewis, Melvyn. 2009. Ethnologue: Languaes of the World. 16th Ed. *SIL Internationl, ResearchGate*

Loulwah et al. 2007. Local Semantic Kernels for Text Document Clustering. *In Workshop on Text Mining, SIAM International Conference on Data Mining, George Mason University*.

M,.A Syakur, B K . KHotimah, E M S Rochiman, B D Statoto. 2018. Integration K-means clusteing Method and Elbow Method for Identification of the Best Customer Profile. *IOP Conference Seies Materials Science and Engineering, NASA, USA,*

Magnus Rosell. 2006. Introduction to Informtion Retrieval and Text Clustering . *Journal of KTH CSC, New York Texas*.

Marco Suárez Barón, Kathleen Salinas Valencia. 2009. An approach to semantic indexing and information retrieval. *Revista Facultad de Ingeniería Universidad de Antioquia, núm. Universidad de Antioquia Medellín, Colombia. 174-187*

Martin Brascheler and Barbel Ripplinger. 2004. . How Effective is Stemming and Decompounding for German Text Retrieval. *Proceedings of ACL-08: HLT, Short Papers , Columbus, Ohio, USA. pp 253–256*

Md Maruf Hasan Yuji Matsumoto. 2012. Document Clustering: Before and After the Singular Value Decomposition. *American Control Conference USA*.

Megersa, Diriba. 2002. An Automatic Sentence Parser for Oromo Language using Supervised Learning Technique, *Msc. thesis, Addis Ababa University, Ethiopia*.

Melkamu Abetu. 2017. Query expansion for Afaan oromo information retrieval based on wordnet. *Msc. Thesis, Haramaya University, Ethiopia*

Michael Steinbach George Karypis Vipin Kumar. 2009. A Comparison of Document Clustering Techniques. *University of Minnesota, Minnesota, USA.*

Morka Mekonnen. 2001. Text-to-Speech System for Afaan Oromo. *Msc. thesis at school of Information studies for Africa, Addis Ababa University, Ethiopia*.

Moukdad Haidar. 2003. Lost In Cyberspace: How Do Search Engines Handle Arabic Queries? *The 12th International World Wide Web Conference*.

Mulualem Wordofa. 2013. Semantic Indexing and Document Clustering for Ahmaric Information Retrieval. *Msc Thesis, Addis Ababa University, Ethiopia*.

M. Ramakrishna Murthy , J.V.R. Murthy , P.V.G.D. Prasad Reddy and Suresh C. Sapathy. 2014. A new approach for finding appropriate number of Clusters using SVD along with determining Best Initial Centroids for K-means Algorithm. International Journal of Enhanced Research in Science Technology & Engineering. 3(3) 430-437

Mylanguage.org. 2011. Oromo Numbers. *Retrieved September 21, 2020, from Doi: http://www.mylanguages.org/learn_oromo.php*

Norbby Saiyad, Prajapati and Dabhi. 2016 "A survey of document clustering using semantic approach," *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai*, 2016. 2555-2562,

Naik, M.P., Prajapati, H.B., & Dabhi, V.K. (2015). A survey on semantic document clustering. *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), India.* 1-10.

Naohiro Matsumura, Yikio Ohsawa, Mitsuru Ishizuka. 2015. A Survey Paper On Different Techniques Of Document Clustering. *International Journal of Current Engineering and Scientific Research (IJCESR), 2*(1): 2393-8374.

Naohiro Matsumura, Yukio Ohsawa, Mitsuru Ishizuka. 2002. Priming Activation Indexing: Automatic Indexing for Extracting Asserted Keywords from a Document. *ResearchGate, Berlin, German*.

Neelam Phadnis and Jayant Gadge. (2014). Framework for Document Retrieval using Latent Semantic Indexing. *International Journal of Computer Applications, Berlin, German*. 94(14)

Neepa Shah and Sunita Mahajan. 2012. Document Clustering: A Detailed Review. *International Journal of Applied Information Systems (IJAIS),Foundation of Computer Science, New York, USA , 52*(5): 42-52.

Nega Alemayehu and Peter Willett. 2002. Stemming of Amharic words for information retrieval, in Literary Linguistic Computing*, Literary and Linguistic Computing University of Sheffield, Sheffield, UK. 17*(1): 1-18.

Oren Zamir and Oren Etzioni. 1998. Web document clustering: a feasibility demonstration. *In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '98). Association for Computing Machinery, New York, NY, USA, 46–54*

Oren Zamir Oren Etzioni Omid Madani Richard M. Karp.1997. Fast and Intuitive Clustering of Web Documents. *In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, AAAI Press.* 287-290.

Paralic Jan and Ivan Kostial. 1999. Ontology-based Information Retrieval . *journal Web Technologies, ResearchGate, USA.*

Paul Mcnamee. Charles Kenneth. Nicholas James. Clifton Mayfield. 2009. Addressing Morphological Variation in Alphabetic Languages. *In Proceedings of the 32nd.,. international ACM SIGIR conference on Research and development in information retrieval, Cambridge University Press, USA.*

Ricardo Baeza-Yates, Carlos Castillo, Felipe Saint Jean. 2004. Web Dynamics, Structure and Page Quality. *Springer, Berlin*, 93-109.

Richard, H. 2003. *Encyclopedia of English*. Retrieved October 20, 2017, from An encyclopedia of English Grammar and Word Grammar: http://www.phon.ucl.ac.uk/home/dick/enc-gen.htm

Robert Krovetz. 2000. Viewing morphology as an inference process. *Elsevier Princeton, USA, 118*, 277–294.

Sajendra Kumar, Ram Kumar Rana , Pawan Singh, 2012. Ontology based Semantic Indexing Approach for Information Retrieval System. *International Journal of Computer Applications, Berlin, German, 49*(12).

Sara Norrby. 2016. Using Morphological Analysis in an Information Retrieval System for Résumés, *Stockholm, Sweden*.

Savoy Picard and Jacques Savoy. 2000. A logical information and retireval model based a combination propositional logic and probability theory. *ResearcGate, Berlin, German*.

Septian Wulandari. 2020, January). Analyze K-Value Selected Method of K-Means Clustering Algorithm to Clustering Province Based on Disease Case. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), Blue Eyes Intelligence Engineering & Sciences Publication, 9*(3).

Singhal, A. 2008. *Modern Information Retrieval*. Retrieved October, from A Brief Overview, *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, IEEE.*

Stefano Ceri, Alessandro Bozzon, Marco Brambilla Emanuele Della Valle,Piero Fraternali, Silvia Quarteroni. 2013. Web Information Retrieval, Data-Centric Systems and Applications, *Proceedings of the IEEE-PIEEE.*

Stephen E. Robertson. 2012. *The methodology of information retrieval experiment, springer, Berlin, German.*

Susan T. Dumais. 2007. Enhancing Performance in Latent Semantic Indexing (LSI) Retrieval. *40th Hawaii International Conference on Systems Science, Waikoloa, Big Island, HI, USA.* 1-19.

Tesfaye Guta. 2010. Afaan Oromo Search Engine. *Msc Thesis, Addis Ababa University, Ethiopia*.

Tewodros Hailemeskel. 2003. Amharic text retrieval: an experiment using Latent semantic indexing (LSI) with singular value decomposing (SVD). *Msc thesis, Addis Ababa University, Ethiopia*.

Tewodros Abebaw. 2014. Applying Thesaurus Based Semantic Compression For Improving The Performance of Amharic Text Retrieval. *Msc thesis, Addis Ababa University, Ethiopia*.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. *In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '99). Association for Computing Machinery, New York, NY, USA, 50–57.*

Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. *Journal of ACM SIGIR, USA.* 51(2)

Tilahun Gamta. 2000. The Politicization of My Oromo-English Dictionary. *Journal of Oromoo Studies. Addis Ababa University, Ethiopia*, 7(2): 1-17.

Unubi, Abraham. 2017. Selected Derivational Morphological Processes in English, Hausa, Igala and Some other Languages of the *World. World Wide Journal of Multidisciplinary Research and Development, Nigeria*

Vallet David, Fernández Miriam, and Castells Pablo. 2013. An Ontology-Based Information Retrieval Model,.*The Semantic Web: Research and Applications, Endland.* 455-470.

W. Bruce Croft. 1995. What Do People Want from Information Retrieval*?, Doi: hdl://cnri.dlib/november95-croft ‘.*

Wakshum Mekonen. 2000. Development of a stemming algorithm for Afaan Oromo text, *MSc. thesis,: Addis Ababa University, Ethiopia.*

Wegari Getachew. 2017. OroRoots: Rule-Based Root Generation System for Afaan Oromo. *Addis Ababa University Ethiopia*.

Woldemariam, Assefa. 2005. Development of Morphological Analyzer for Afaan Oromo. *MSc. thesis*. *Addis Ababa University Ethiopia*

Workineh Tesema and Debela Tesfaye. 2017. Word Sense Disambiguation and Semantics for Afan Oromo Words using Vector Space Model. *International Journal of Research Studies in Science, Engineering and Technology* 4(6), 10-15

Xiaoyong Liu and W. Bruce Croft. 2004. Cluster-based retrieval using language models. *In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '04). Association for Computing Machinery, New York, NY, USA, 186–193*

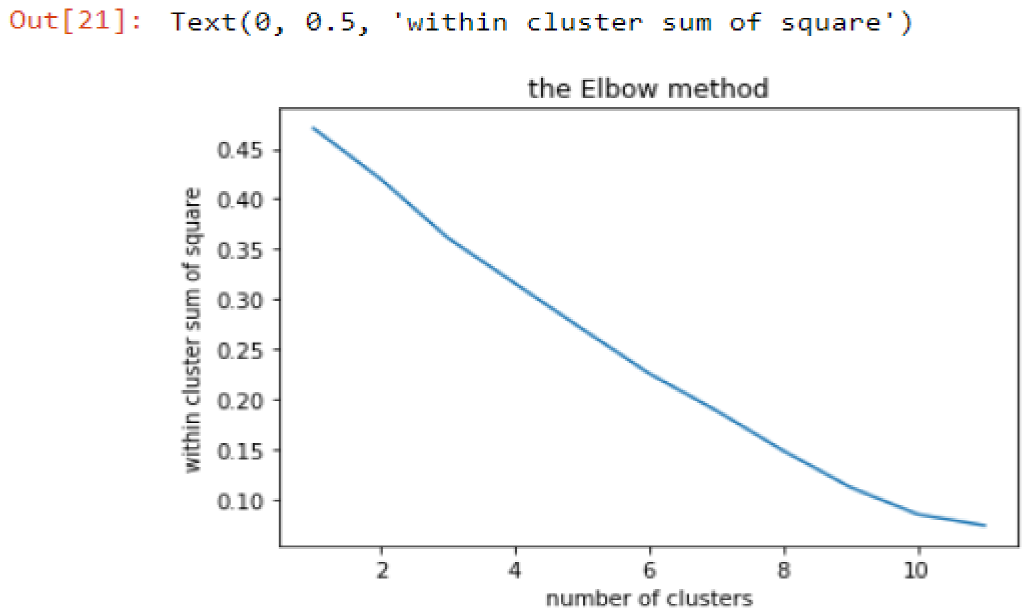
Yu Xiao. 2010. Survey of Document Clustering Techniques Comparison of LDA and moVMF.

Zipf, G. 1932. Selected Studies of the Principle of Relative Frequency in Language. *Cambridge: Harvard University press, USA.*

7. APPENDICES

Appendix I- Elbow Method Implementation Code and Graph

```
wcss =[]

for i in range(1,12):

    kmeans = KMeans(i)

    kmeans.fit(lowRankX)

    wcss_iter = kmeans.inertia_

    wcss.append(wcss_iter)

number_clusters = range(1,12)

plt.plot(number_clusters, wcss)

plt.title('the Elbow method')

plt.xlabel('number of clusters')

plt.ylabel('within cluster sum of square')
```



```
Out[21]:  Text(0, 0.5, 'within cluster sum of square')
```

Appendix II-Implementation Code in Python

Indexing Code

```
import sys

import glob

import numpy

import string

from matplotlib import pyplot

from nltk.corpus import stopwords

import pandas as pd

from string import punctuation

import matplotlib.pyplot as plt

import l3

import numpy

import pandas as pd


# Structures holding documents and terms

document_list = []

document_ids = {}

token_list = []

token_ids = {}

# Repeat for every document in processed_pages

for filename in glob.glob('corpus/*'):

    f = open (filename, 'r', encoding = 'utf-8')
```

```
  tokens = f.read()
  tokens = tokens.lower()
  tokens = tokens.split()
  table = str.maketrans(' ', ' ', punctuation)
  tokens = [w.translate(table) for w in tokens]
  tokens =[word for word in tokens if word.isalpha()]
  tokens =[word for word in tokens if len(word)>2 ]
  stop_words = set(stopwords.words('oromo'))
  tokens = [w for w in tokens if not w in stop_words]
  f.close()

  # Get the document name as last part of path
  article_name = filename[filename.rfind('/')+1:]
  doc_id = len(document_list)
  # Insert ID in inverse list
  document_ids[article_name] = doc_id
  # Populate token structure for all tokens in document
  for t in tokens:
      if t not in token_ids:
          token_ids[t] = len(token_list)
          token_list.append(t)
# Transform the document's token list into the corresponding ID list
  tids = [token_ids[t] for t in tokens]
```

```
# Store the document as both its token ID list and the corresponding set
    document_list.append({
        'name': article_name,
        'tokens': tids,
        'set': set(tids)
    })

tokens = open('token_list.txt', 'w', encoding ='utf-8')
tokens.write(str(token_list))
tokens.close()
#stemming using hornmorph stemmer
l3.anal_file('om','token_list.txt', 'stemmed.txt' , citation =True, nbest =1)
stemmed_tokens = open('stemmed.txt', 'r', encoding = 'utf-8')
stem_list = stemmed_tokens.read()
number_of_documents = len(document_list)
number_of_tokens = len(stem_list)
# Building the TF-IDF matrix
sys.stderr.write ('Building the TF matrix and counting term occurrencies\n')
token_count = [0] * number_of_tokens  # number of occurrences of word in document
TF = numpy.empty((number_of_tokens,number_of_documents), dtype=float)
# Scan the document list
for i,doc in enumerate(document_list):
            # Initialize with zeros
            n_dt = [0] * number_of_tokens
            # For all token IDs in document
            for tid in doc['tokens']:
                # if first occurrence, increase global count for IDF
                if n_dt[tid] == 0:
                    token_count[tid] += 1
                # increase local count
                n_dt[tid] += 1
            # Normalize local count by document length obtaining TF vector;
            # store it as the i-th column of the TFIDF matrix.
            TF[:,i] = numpy.array(n_dt, dtype=float) / len(doc['tokens'])
TF[] = numpy.array(n_dt, dtype=float) / len(doc['tokens'])
IDF = numpy.log10(number_of_documents / numpy.array(token_count, dtype=float))
TFIDF = numpy.diag(IDF).dot(TF)
TFIDF = pd.DataFrame(TFIDF)
```

#SVD for rank reduction

T_o, S_o, D_o^T = numpy.linalg.svd (TFIDF, full_matrices =False)

#r = number of low rank

#Y = TSD

r = 10

```
T = $T_o$[:, :r]

S = numpy.diag($S_o$)[:r, :r]

$D^T$ = $D_o^T$[:r, :]

Y = numpy.dot(T, numpy.dot(S, $D^T$))

#Kmeans clustering using reduced SVD

from sklearn.cluster import KMeans

n_clusters = 8

kmeans = KMeans(n_clusters, max_iter=100, n_init=1, verbose=1)

cluster = kmeans.fit_predict(Y) #Y=$TSD^T$ or reduced SVD using r number

label = kmeans.labels_

centroid = kmeans.cluster_centers_             #centroid of the clusters

sorted_centroid = centroid.argsort()[:, ::-1] #sorting centroid

centroid = pd.DataFrame(centroid)

centroid = kmeans.cluster_centers_        #centroid of the clusters

sorted_centroid = centroid.argsort()[:, ::-1]   #sorting centroid

centroid = pd.DataFrame(centroid)
```

searching code

```
query = input('please enter query/Jechaa keessan galchaa!! ')
query = query.lower()
q_split = query.split()
q_split = [w.translate(table) for w in q_split]
q_split =[word for word in q_split if word.isalpha()]
q_split =[word for word in q_split if not word.isdigit()]
q_split =[word for word in q_split if len(word)>2 ]
stop_words = set(stopwords.words('oromo'))
q_split = [w for w in q_split if not w in stop_words]
q_tokens = open('q_split.txt', 'w', encoding ='utf-8')
    q_tokens.write(str(q_split))
    q_tokens.close()
    l3.anal_file('om','q_split.txt', 'stemm_q.txt' , citation =True, nbest =1)
    fq = open('stemm_q.txt', 'r', encoding = 'utf-8')
    qr_list = fq.read()
q_tokens = set()
q_count = [0] * number_of_tokens
q_length = 0
for token in q_list:
        try:
                t_id = token_ids[token]
                q_count[t_id] += 1
```

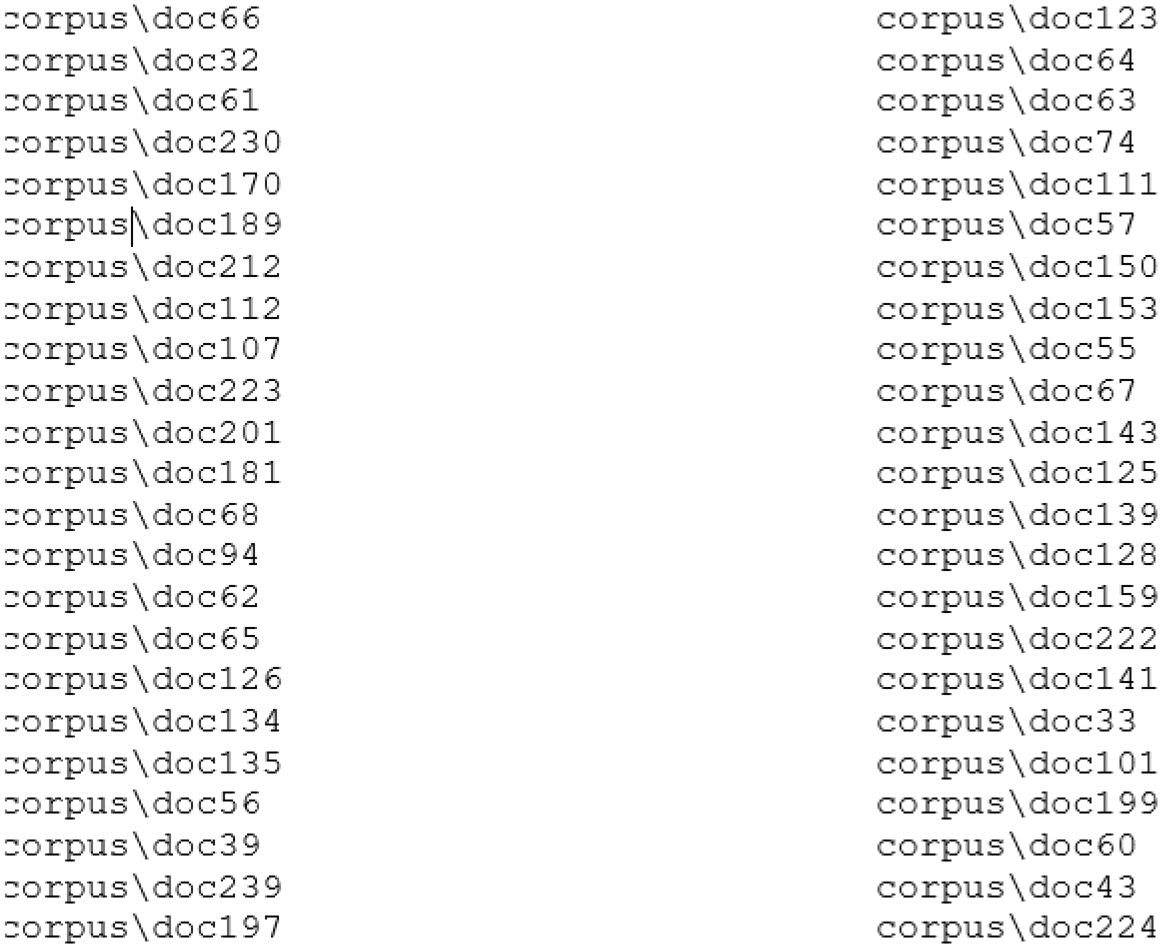
```
            q_length += 1

            q_tokens.add(t_id)

        except:

            Pass

q_TFIDF = (numpy.array(q_count,dtype=float) / q_length) * IDF

sim = TFIDFM.T.dot(q_TFIDF)

#retrieve query result using VSM

N= len(document_list)

def print_top(sim,N,smallest=False):

    sorted_sim = sorted(enumerate(sim),key=lambda t:t[1], reverse= not smallest)

    for i,s in sorted_sim[:N]:

        if s>0:

            print(  document_list[i]['name'])

#retrieve query result using SVD

# Compute the cosine similarity array given the SVD decomposition of the TFIDF matrix (computed above), the normalized TFIDF query vector q and the desired rank r


def reduced_similarity (r, T_o, S_o, D_o^T, q):

    T = T_o[:, :r]

    S = numpy.diag(S_o)[:r, :r]

    D^T = D_o^T[:r, :]


    q_r = S.dot(T.T).dot(q_TFIDF)
    return $D^T$.T.dot(q_r)

sim_r = reduced_similarity (r, $T_o$, $S_o$, $D_o^T$, q_TFIDF)

print_top (sim_r,N)
```

#retrieve query result using SVD with K-means

```
#calculating similarity b/n cluster centroid and query vector

c_qsim =centroid.dot(q_TFIDF)

my_dict =c_qsim.to_dict()

def get_key(val):

    for key, value in my_dict.items():

         if val == value:

             return key


# w is cluster with max similarity b/n query and centroid

w = get_key(c_qsim.max())

doc_dict = {i: numpy.where(label ==i)[0] for i in range(kmeans.n_clusters)}

doc_dict_key = list(doc_dict.keys())

doc_dict_val = list(doc_dict.values())

doc_list = []

for key in doc_dict.keys():

    if  w == key:

        print ( 'doc:', list(doc_dict[w]))
```

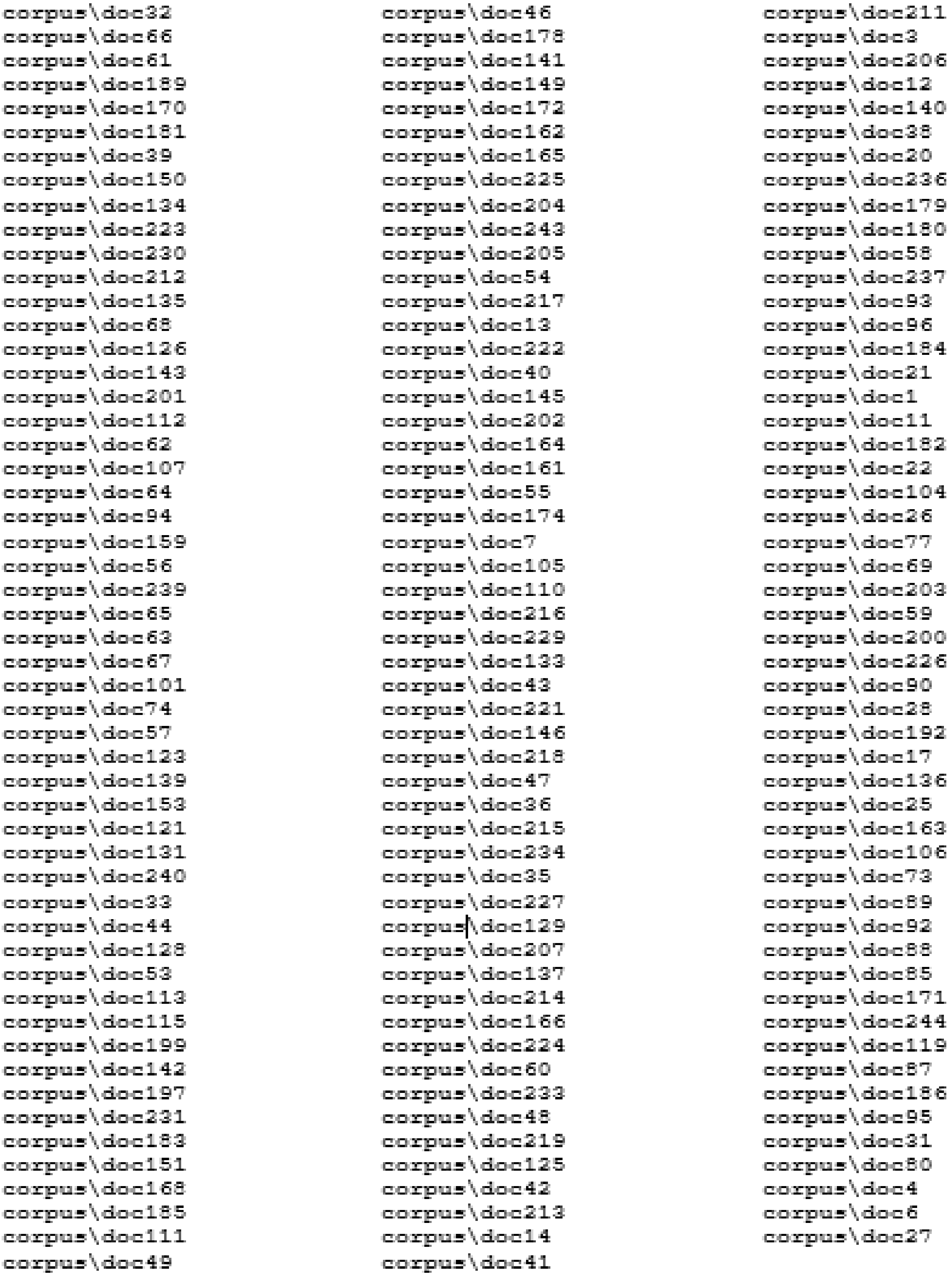
Appendix III- Sample of Proposed work Search Results

Vector Space Model (VSM) using ‘barnoota’ query word result



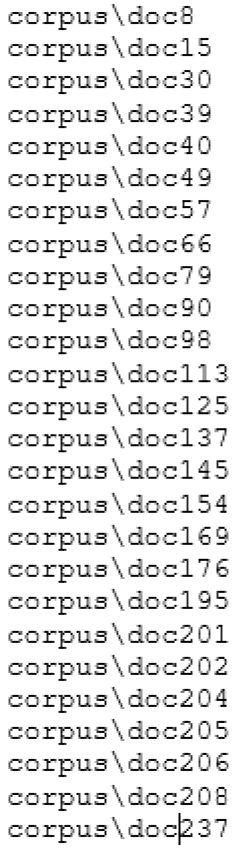
```
corpus\doc66                                          corpus\doc123
corpus\doc32                                          corpus\doc64
corpus\doc61                                          corpus\doc63
corpus\doc230                                         corpus\doc74
corpus\doc170                                         corpus\doc111
corpus\doc189                                         corpus\doc57
corpus\doc212                                         corpus\doc150
corpus\doc112                                         corpus\doc153
corpus\doc107                                         corpus\doc55
corpus\doc223                                         corpus\doc67
corpus\doc201                                         corpus\doc143
corpus\doc181                                         corpus\doc125
corpus\doc68                                          corpus\doc139
corpus\doc94                                          corpus\doc128
corpus\doc62                                          corpus\doc159
corpus\doc65                                          corpus\doc222
corpus\doc126                                         corpus\doc141
corpus\doc134                                         corpus\doc33
corpus\doc135                                         corpus\doc101
corpus\doc56                                          corpus\doc199
corpus\doc39                                          corpus\doc60
corpus\doc239                                         corpus\doc43
corpus\doc197                                         corpus\doc224
```

Single Value Decomposition (SVD) using ‘barnoota’ query word result



```
corpus\doc32
corpus\doc66
corpus\doc61
corpus\doc189
corpus\doc170
corpus\doc181
corpus\doc39
corpus\doc150
corpus\doc134
corpus\doc223
corpus\doc230
corpus\doc212
corpus\doc135
corpus\doc68
corpus\doc126
corpus\doc143
corpus\doc201
corpus\doc112
corpus\doc62
corpus\doc107
corpus\doc64
corpus\doc94
corpus\doc159
corpus\doc56
corpus\doc239
corpus\doc65
corpus\doc63
corpus\doc67
corpus\doc101
corpus\doc74
corpus\doc57
corpus\doc123
corpus\doc139
corpus\doc153
corpus\doc121
corpus\doc131
corpus\doc240
corpus\doc33
corpus\doc44
corpus\doc128
corpus\doc53
corpus\doc113
corpus\doc115
corpus\doc199
corpus\doc142
corpus\doc197
corpus\doc231
corpus\doc183
corpus\doc151
corpus\doc168
corpus\doc185
corpus\doc111
corpus\doc49
corpus\doc46
corpus\doc178
corpus\doc141
corpus\doc149
corpus\doc172
corpus\doc162
corpus\doc165
corpus\doc225
corpus\doc204
corpus\doc243
corpus\doc205
corpus\doc54
corpus\doc217
corpus\doc13
corpus\doc222
corpus\doc40
corpus\doc145
corpus\doc202
corpus\doc164
corpus\doc161
corpus\doc55
corpus\doc174
corpus\doc7
corpus\doc105
corpus\doc110
corpus\doc216
corpus\doc229
corpus\doc133
corpus\doc43
corpus\doc221
corpus\doc146
corpus\doc218
corpus\doc47
corpus\doc36
corpus\doc215
corpus\doc234
corpus\doc35
corpus\doc227
corpus\doc129
corpus\doc207
corpus\doc137
corpus\doc214
corpus\doc166
corpus\doc224
corpus\doc60
corpus\doc233
corpus\doc48
corpus\doc219
corpus\doc125
corpus\doc42
corpus\doc213
corpus\doc14
corpus\doc41
corpus\doc211
corpus\doc3
corpus\doc206
corpus\doc12
corpus\doc140
corpus\doc38
corpus\doc20
corpus\doc236
corpus\doc179
corpus\doc180
corpus\doc58
corpus\doc237
corpus\doc93
corpus\doc96
corpus\doc184
corpus\doc21
corpus\doc1
corpus\doc11
corpus\doc182
corpus\doc22
corpus\doc104
corpus\doc26
corpus\doc77
corpus\doc69
corpus\doc203
corpus\doc59
corpus\doc200
corpus\doc226
corpus\doc90
corpus\doc28
corpus\doc192
corpus\doc17
corpus\doc136
corpus\doc25
corpus\doc163
corpus\doc106
corpus\doc73
corpus\doc89
corpus\doc92
corpus\doc88
corpus\doc85
corpus\doc171
corpus\doc244
corpus\doc119
corpus\doc87
corpus\doc186
corpus\doc95
corpus\doc31
corpus\doc80
corpus\doc4
corpus\doc6
corpus\doc27
```

Single Value Decomposition with K-means Clustering using ‘barnoota’ query word result



```
corpus\doc8
corpus\doc15
corpus\doc30
corpus\doc39
corpus\doc40
corpus\doc49
corpus\doc57
corpus\doc66
corpus\doc79
corpus\doc90
corpus\doc98
corpus\doc113
corpus\doc125
corpus\doc137
corpus\doc145
corpus\doc154
corpus\doc169
corpus\doc176
corpus\doc195
corpus\doc201
corpus\doc202
corpus\doc204
corpus\doc205
corpus\doc206
corpus\doc208
corpus\doc237
```