ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

SCHOOL OF INFORMATION SCIENCE

Discovering Words Association to Enhance the Effectiveness of Amharic Information Retrieval System

MERHAWI TSEGAY ABEBE

February, 2017

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

SCHOOL OF INFORMATION SCIENCE

Discovering Words Association to Enhance the Effectiveness of Amharic Information Retrieval System

A thesis submitted to the School of Information Science of Addis Ababa University in partial fulfillment of the requirement for the Degree of Master of Science in Information Science

MERHAWI TSEGAY ABEBE

February, 2017

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCE

SCHOOL OF INFORMATION SCIENCE

By:

MERHAWI TSEGAY ABEBE

Name and signiture of Members of the Examining Board

Name	**Signiture**	**Date**
_____ Chairman	_____	_____
Million Meshesha(PhD) Advisor	_____	_____
Martha Yifiru (PhD) Examiner	_____	_____
Dereje Teferi(PhD) Examiner	_____	_____

Contents

Acknowledgment

First I would like to express my sincere thanks to my Parents Tsegay Abebe(I call him “አባባ”)Brhan haylu(I call my mom “አደዋይ”).

I would like to express my deep gratitude for my advisor Dr. Million M., who has always been making things simple to understand. And for his accommodating advise, support, understanding and patience. He has been remarkable in his attempt to keep me motivated in this thesis and has always tried to improve me with perfect feedback.

I would like to thank my brother Danyom Tsegay and Yonas Tadesse, for their constant feedback and motivation. I would like to thank my friend Ybeyn Tadesse for his help in checking for grammatical errors.I would like to thank each and every one who helped me throughout my work.

Abstract

Given the tremendous growth rate of the World Wide Web, it is increasingly necessary to develop appropriate tools that can assist users looking for information. The techniques used by search engines are based on statistical and syntactic analysis, and do not consider the semantics contained in the user's request and the available documents. However, solving user problem is often not interested in information similar to the query, but rather information that bears an interesting relation to the query.

The study attempted to understand the user query with the help of Semantic association expand to a user's query. Using the design science research methodology we developed a prototype. Amharic local news articles available in electronic form are used as a source of data. The system extracts the keywords using "term frequency inverse document frequency (TF-IDF) weighting scheme" and then determines the associations among terms using "apriori algorithm".

We have evaluated our system by comparing its results with the results before query expansion. Search results are evaluated with precision, recall and F-measure. The experiment results show that the performance of the system improved by 17% in terms of recall and 7.5% F-measure system improvement, while precision was declined 2.8%.

Search engines can perform quite remarkably result by retrieving what users need directly. In the analysis, it is found that the semantic association based model intend to retrieve more relevant document enhancing recall of the Amharic information retrieval system. The terms retrieved may not be relevant to user queries, but they have strong associations. This was the main challenge for word association based retrieval system. We recommended integrating mechanisms of controlling similarity terms in the association-based retrieval model to enhance precision.

CHAPTER ONE

INTRODUCTION

1.1 Background

Information Retrieval (IR) systems aim at providing the most relevant documents to a user's query. Early IR systems were primarily used by retrieval experts; hence the initial IR methodology was based on keywords manually assigned to documents, and on complicated Boolean queries. As an automatic indexing and natural language queries is gaining popularity in the 1970's, IR systems became increasingly more accessible to non-expert users. Documents were indexed by automatically considering all terms with them as independent keywords, in what is known as the Bag-of-Words (BOW) representation, and query formatting was simplified to a short natural language formulation [**1**].

However, even as the keywords became "noisier," the basic methodology for indexing them remained unchanged. Thus, these non-expert users were increasingly faced with what was described as "the vocabulary problem" [**1**]. The keywords chosen by users were often different from those used by the authors of the relevant documents, lowering the systems' recall rates. In other cases, the contextual differences between ambiguous keywords were overlooked by the BOW approach, reducing the precision of the results. These two problems are commonly referred to as synonymy and polysemy, respectively [**2**].

To handle such problems, new query expansion methods that rely on corpus-based evidence were suggested. For example, [**3**] suggested identifying terms that co-occur with query keywords in the top-ranked documents for the query, to be used as expansion terms that are more broadly related to the query. Such approaches showed significant improvement, but require manual

tuning in order not to adversely affect performance: too few expansion terms may have no impact, and too many will cause a query drift [**4**].

To tackle polysemy, the main proposed method was to apply automatic word sense disambiguation algorithms to documents and query. Disambiguation methods use resources such as the WordNet thesaurus or co-occurrence data to find the possible senses of a word and map word occurrences to the correct sense. These disambiguated senses are then used in indexing and in query processing, so that only documents that match the correct sense are retrieved. The inaccuracy of automatic disambiguation is the main obstacle in achieving significant improvement using these methods, as incorrect disambiguation is likely to harm performance rather than merely not improve it [**5**] [**6**].

Concept-based information retrieval is an alternative IR approach that aims to tackle these problems differently. Concept-based IR represents both documents and queries using semantic concepts, instead of (or in addition to) keywords, and performs retrieval in that concept space. This approach holds the promise that represents documents and queries (or augmenting their BOW representation) using high-level concepts will result in a retrieval model that is less dependent on the specific terms use [**7**]. Such a model could yield matches even when the same notion is described by different terms in the query and target documents, thus alleviating the synonymy problem and increasing recall. Similarly, if the correct concepts are chosen for ambiguous words appearing in the query and in the documents. non-relevant documents that were retrieved with the BOW approach could be eliminated from the results, thus alleviating the polysemy problem and increasing precision.

Explicit Semantic Analysis, or ESA [**8**], is a recently proposed method for semantic representation of general-domain natural language texts. ESA represents meaning in a high-dimensional space of concepts, automatically derived from large-scale human-built repositories such as Wikipedia. Since it was first proposed, ESA has been successfully applied to

textcategorization, cross-language information retrieval, and concept-based information retrieval [**8**] [**9**]

Enormous research work is carried out to incorporate intelligence in an information retrieval system. Semantic web views WWW as a collection of data instead of a set of documents by incorporating intelligence in the existing web. Semantic web technologies retrieve more relevant information from web by understanding the user query with the help of Semantic association. Semantic association represents a collection of terms with their relations (association rule) between them, their instances in machine understandable and human readable format.

1.2 Statement of the Problem and Justification

In Ethiopia there are more than 80 languages spoken [**10**]. Amharic is the most widely used and it is the working language of the Federal Government of Ethiopia. According to a census report ECSA (Ethiopian Central Statistics Authority), Amharic is the mother tongue for more than 25 million and second language for over 15 million people. The language is used in offices, schools and other public services. As a result, a large number of documents are increasingly published and distributed in both hard and soft copies. As the amount of information to be processed gets larger and larger, systematic representation of information is of paramount importance for facilitating retrieval of relevant information. Such representation might be difficult unless automatic information storage and retrieval is used.

There are several researches done in Amharic information retrieval to enhance the effectiveness of the system. Tewodros [**11**] developed latent semantic indexing system to solve the problem of exact term matching retrieval system. It helps to retrieve documents which have the same concept. However, in the experiment there was a challenge that system affected by stemmer and part of speech tagger used in developing Amharic information retrieval system.

Tessema et al. [**12**] designed and implemented Amharic search engine. The vector space model has been used to compute similarity scored documents with user queries. In the experiment there was a challenge that the system is affected by part of speech tagger and document similarity.

Abay [**13**] used statistical co-occurrence analysis, face problem of polysemy and synonymy that exists in Amharic. Iman [**14**]applies word sense disambiguation to choose the right term that can be used to expand the query. The inaccuracy of automatic disambiguation was the main obstacle in achieving significant improvement using these methods, as incorrect disambiguation is likely to harm performance rather than merely not improve it. Welde [**15**] designed Ontology based query expansion model to improve the performance of the Amharic IR system using manual construction ontology. The manual construction of ontology is labor intensive and time-consuming task. In addition to those Tewodros [**11**], Amanuel [**16**], Samrawit [**17**]and Tessema [**12**] have tried to develop Amharic text retrieval systems. But neither of them has used association-based retrieval approach.

Traditional information retrieval (IR) systems provide techniques to retrieve documents that contain the same or similar keywords as in a user query. However, a problem solving user has been often not interested in information similar to the query, but rather information that bears an interesting relation to the query, e.g. explanation, elaboration, cause-effect and so on. Even though this information is likely to exist in the documents that match the query, finding parts of the document that capture such associative knowledge by manual inspection is both a time consuming task and is critically dependent on the degree of expertise of the user.

The association-based retrieval approach proposed in this paper allows retrieval of terms that are associated with a query rather than being similar to it. This approach is especially useful for search in domain-specific documents because they contain a lot of domain knowledge in the form of associative relations, e.g. symptom →disease → treatment in medical patient records,

team → game → score(result) in sport cases, qualification → human error → accident in industrial accident reports.

Hence the study attempts to explore and discover word association automatically for improving the performance of the Amharic information retrieval system. To this end the following research questions are investigated.

What are the confidence and support vale to discover words that are strong association?

How to design the integration of word association with user query?

To what extent the performance of IR system improved?

1.3. Objective of the study

1.3.1. General Objective

The general objective of the study is to discover Amharic words with strong association so as to enhance the effectiveness of the Amharic information retrieval system

1.3.2 Specific Objectives

With the aim of achieving the general objective of this study, the following specific objectives are formulated:-

- To review literature so as to understand concepts and related work.
- To prepare Amharic document corpus that is used in the experiment.
- To identify suitable algorithm for Amharic term-association learning.
- To develop a query expansion technique based on the association rule.
- To design an information retrieval system prototype that integrates term association.
- To analyze and evaluate the performance of the model.

1.4 Scope and limitation of the Study

The scope of this research is to develop a prototype IR system using term association. The IR system is designed to search within large size Amharic document corpus. The system is developed following the procedures of the IR system. For a given Amharic document corpus, text operations such as, tokenization, stop words removal and normalization done.

The scope of the thesis is limited to; design methods that take text query feed by user and taking advantage of the semantic association for expanding. Demonstrate the association based information retrieval by performing experimental analyses and evaluation of the effectiveness of the method.

Lack of standard large size document corpus and long processing time to construct relevance matrix based relevance judgment, considering the time constraint, the experiment was conducted only using 773 Amharic news documents.

1.5Significance of the Study

This study is expected to produce results in several ways. The main significance is designing an applicable Amharic information retrieval system that can help users search and retrieve relevant documents written in Amharic language. The association-based retrieval approach proposed in this paper allows retrieval of terms that are associated with a query rather than being similar to it. This approach is especially useful for search in domain-specific documents because they contain a lot of domain knowledge in the form of associative relations, e.g. symptom →disease → treatment in medical patient records, evidence → arguments → court decisionin lawsuit cases, qualification → human error → accident in industrial accident reports

There are lots of advantages in incorporating appropriate semantic web hidden knowledge in existing search feature specific contributions. First, this study produces results that can show the advantage of applying an association rule based information retrieval model for Amharic language. Second, it gives direction for researchers to select the best model for future works on Amharic text retrieval based on the performance achieved. Third, the model can easily extend it can be applied to any problem where there is a need to retrieve any documents written in local Semitic languages (such as, Guragegna, Tigrigna, Siltegnaetc) from large collection. Additionally, it is an academic exercise to fulfill the requirement of masters program the researcher is enrolled in. Students and researchers will benefit from the research. The user of Amharic languages and government, are also primary beneficiates of the research.

1.6 Methodology of the study

To achieve the objectives stated in Section 1.3, the researcher use of the following method:

Since this study relies on the experimental result that deals with quantitative data. And there is a need for design for the experiment. The research method is a design science research methodology.

The design science research paradigm is highly relevant for information systems (IS) research because it directly addresses two of the key issues of the discipline: the central, albeit controversial, role of the IT artifact in IS research and the perceived lack of professional relevance of IS research [**18**]. Design science, supports a pragmatic research paradigm that calls for the creation of innovative artifacts to solve real-world problems. Thus, design science research combines a focus on the IT artifact with a high priority on relevance in the application domain.

In the scientific method, an experiment is an empirical procedure that arbitrates between competing models or hypotheses. Researchers also use experimentation to test existing theories or new hypotheses to support or disprove them. An experiment usually tests a hypothesis, which is an expectation about how a particular process or phenomenon works. However, an experiment may also aim to answer a "what-if" question, without a specific expectation about what the experiment reveals, or to confirm prior results [**18**].

Therefore, quantitative research is essentially about collecting numerical data to explain a particular phenomenon, particular questions seem immediately suited to being answered using quantitative methods. For example,

How many relevant documents are retrieved by system X?
What percentage of correctly classified using system X?
What is the F-measure score value by the system?
For how many terms association rules generated using confidence n%?

So we used a mixed of design science research and quantitative research methodology.

1.6.1 Literature Review

Different literatures reviewed in order to understand more about the information retrieval system, concept of Semantic search and association rule. Investigate approaches used for constructing association rule from text corpus and other related issues, and how to use existing association rule learning methods. Additional literature review and document analyses are made to investigate and identify the feature of Amharic text, which is important to the research in the course of Amharic text operations.

1.6.2 Data Sources for the Experiment

Amharic local news articles available in electronic form are used as a source of data. These news articles were obtained from different Amharic news broadcast websites. the web site of Walta Information Center (http://www.waltainfo.com) 23 news articles, Reporter (https://www.ethiopianreporter.com) 82 news articles, Ethiopian Broadcast Corporation (http://www.**ebc**.et) 30,Ethiopian news agency(http://www.ena.gov.et)310 newsarticles, Fana Broadcast Corporation(http://www.**fana**bc.com)110 new articles.

The researcher inspired to use news articles as a test data because; news articles are easier to access, available in electronic form. Previous researches such as Tewodros [**11**] used 200,Abay [**13**] and Samrawit [**17**]used 300 collections of local news articles. We used bigger corps data than the previous researchers so we can genereat effective rules.

Several works have been done as to preprocessing the document collection/corpus used in this research. These tasks include; stemming, stop word removal and normalization.Using TF-IDF weighting scheme, we scored each terms in the given text corpus.

1.6.3Tools &Techniques

Owing to its good string manipulation features and user-friendliness the JAVA programming language was used to develop the prototype. In addition, Microsoft SQL database was used to store the indexing terms, their frequency information, and their document references. The prototype system was run on hplaptop machine with Windows 10 operating system, 2.5GHz speed, and 8GB RAM.Apriori algorithm is used to discover the association rule.

1.6.4 Evaluation techniques

The standard approach to information retrieval system evaluation revolves around the notion of relevant and non-relevant documents [43]. With respect to a user information need, a document in the test collection is given a binary classification as either relevant or non-relevant. Previous

researches such as Tewodros [**11**], Abay [**13**] and Samrawit [**17**] used ten test queries with word size of each query 2-4.We use ten test queries like previous works and the test queries word size used are 2- 7.

The most common and basic statistical measures recall, precision and F-measure are used. Precision is the fraction of the documents retrieved that relevant to the user's information need, and recall is the fraction of the documents that are relevant to the query that is successfully retrieved. F-measure is the weighted harmonic mean of precision and recall. Using these techniques, effectiveness of the retrieval system is measured and compared with previous works.

1.7. Organization of the Thesis

This paper starts by presenting background, statement of the problem, objective and methodology of the study, and other introductory parts. In chapter two a review of related literatures term association based information retrieval, and automatic association rule construction is presented. Chapter three details the methodology of the automatic association rule generation system. The algorithms and techniques used in the development of the system are discussed thoroughly. Chapter four explains the experimentation carried out in explaining the development of an IR system that integrates thesaurus in its structure. The research evaluates the performance of the system. Chapter five discusses the conclusion and suggests some recommendation for future study.

CHAPTER TWO

Literature review

2.1 Information Retrieval

Information Retrieval (IR) is concerned with the creation, storage, organization, and retrieval of information. The focus of IR is on unstructured information, like document collections or the web. Structured information retrieval, usually named Data Retrieval, consists of retrieving all items that satisfy a clearly defined expression (e.g. SQL). The goal of IR is to provide users with those documents that will satisfy their information need. The information need is typically expressed in natural language, not always well structured and often semantically ambiguous [**19**].

If we consider the information retrieval (IR) process, we might illustrate it as in Fig. 2.1. Here, the user issues a query q from the front-end application (accessible via, e.g., a Web browser); q is processed by a query interaction module that transforms it into a "machine-readable" query qto be fed into the core of the system, a search and query analysis module. This is the part of the IR system having access to the content management module directly linked with the back-end information source (e.g., a database). Once a set of results r is made ready by the search module, it is returned to the user via the result interaction module; optionally, the result is modified or updated until the user is completely satisfied. The most widespread applications of IR are the ones dealing with textual data. As textual IR deals with document sources and questions, both expressed in natural language, a number of textual operations take place "on top" of the classic retrieval steps. Figure 2.1 sketches the processing of textual queries typically performed by an IR engine:

1. The user need is specified via the user interface, in the form of a textual query q_U (typically made of keywords).

2. The query q_U is parsed and transformed by a set of textual operations; the same operations have been previously applied to the contents indexed by the IR system (see Sect. 2.1); this step yields a refined query q_U .

3. Query operations further transform the preprocessed query into a system-level representation, q_S.

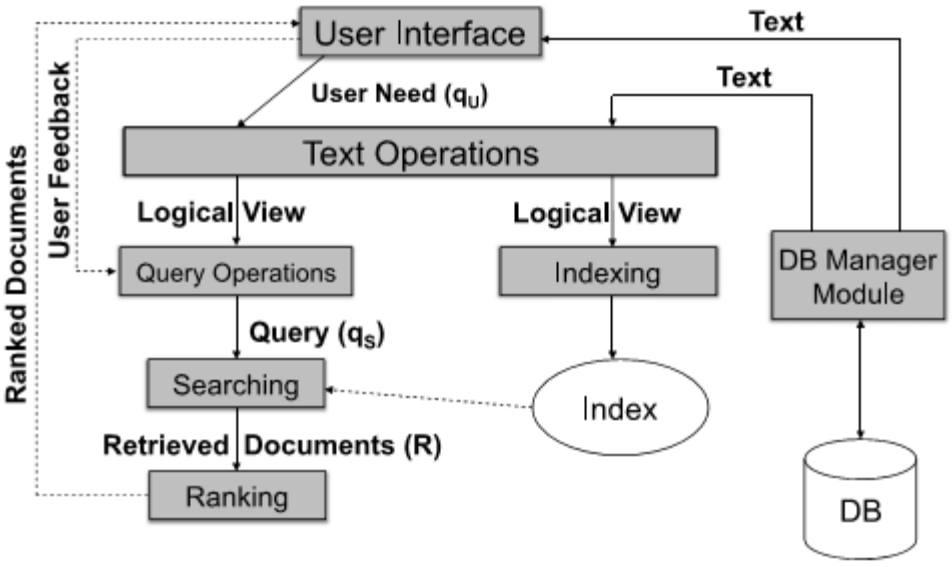

Figure 2.1 the information retrieval process

4. The query q_S is executed on top of a document source D (e.g., a text database) to retrieve a set of relevant documents, R. Fast query processing is made possible by the index structure previously built from the documents in the document source.

5. The set of retrieved documents R is then ordered: documents are ranked according to the estimated relevance with respect to the user's need.

6. The user then examines the set of ranked documents for useful information; he might pinpoint a subset of the documents as definitely of interest and thus provide feedback to the system.

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. Such a "logical", machine-readable representation of documents is discussed in the following section.

This section discussed on information retrieval process. Search for text is the most widespread information retrieval application; we devote particular emphasis to textual retrieval. The fundamental phases of document processing are illustrated along with the principles and data structures supporting indexing.

When we consider natural language text, it is easy to notice that not all words are equally effective for the representation of a document's semantics [**19**].That why, the IR system also preprocesses the text of the documents to determine the most "important" terms to be used as index terms; a subset of the words is therefore selected to represent the content of a document. The textual preprocessing phase typically performed by an IR engine, taking as input a document and yielding its index terms.

Document Parsing: Documents come in all sorts of languages, character sets, and formats; often, the same document may contain multiple languages or formats, e.g., a French email with Portuguese PDF attachments. Document parsing deals with the recognition and "breaking down" of the document structure into individual components. In this preprocessing phase, unit

documents are created; e.g., emails with attachments are split into one document representing the email and as many documents as there are attachments.

Lexical Analysis. After parsing, lexical analysis tokenizes a document, seen as an input stream, into words. Issues related to lexical analysis include the correct identification of accents, abbreviations, dates, and cases. The difficulty of this operation depends much on the language at hand: for example, the English language has neither diacritics nor cases, French has diacritics but no cases, German has both diacritics and cases. The recognition of abbreviations and, in particular, of time expressions would deserve a separate chapter due to its complexity and the extensive literature in the field [**2**].

Stop-Word Removal. A subsequent step optionally applied to the results of lexical analysis is stop-word removal, i.e., the removal of high-frequency words. For example, given the sentence "search engines are the most visible information retrieval applications" and a classic stop words set such as the one adopted by the Snowball stemmer,1 the effect of stop-word removal would be: "search engine most visible information retrieval applications".

However, as this process may decrease recall (prepositions are important to disambiguate queries), most search engines do not remove them [**19**]The subsequent phases take the full-text structure derived from the initial phases of parsing and lexical analysis and process it in order to identify relevant keywords to serve as index terms.

Stemming and Lemmatization. Following phrase extraction, stemming and lemmatization aim at stripping down word suffixes in order to normalize the word. In particular, stemming is a heuristic process that "chops off" the ends of words in the hope of achieving the goal correctly most of the time; a classic rule based algorithm for this was devised by Porter [26]. According to the Porter stemmer, our example sentence "Search engines are the most visible information

retrieval applications" would result in: "Search engine are the most visible inform retrieve application".

Lemmatization is a process that typically uses dictionaries and morphological analysis of words in order to return the base or dictionary form of a word, thereby collapsing its inflectional forms [26]. For example, our sentence would result in "Search engine are the most visible information retrieval application" when lemmatized according to a Word Net-based lemmatizer.

Weighting: The final phase of text preprocessing deals with term weighting. As previously mentioned, words in a text have different descriptive power; hence, index terms can be weighted differently to account for their significance within a document and/or a document collection [38]. Such a weighting can be binary, e.g., assigning 0 for term absence and 1 for presence.

Indexing: The indexing process of translating a document into a set of relevant terms or keywords. The first step requires defining the text data source. Then, a series of content operations transform each original document into its logical representation; an index of the text is built on such a logical representation to allow for fast searching over large volumes of data.

There are various information retrieval methods to search information with enhanced semantics from the user query input to retrieve high relevant information. In the literature there methods and approaches are classified into vector space model, probabilistic information retrieval, context based information retrieval, semantic similarity based information retrieval and semantic association based information retrieval[38].

2.2 Vector Space Based Information Retrieval

In the vector space model text is represented by a vector of *terms*. [**20**] The definition of a term is not inherent in the model, but terms are typically words and phrases. If words are chosen as

terms, then every word in the vocabulary becomes an independent dimension in a very high dimensional vector space. Any text can then be represented by a vector in this high dimensional space. If a term belongs to a text, it gets a non-zero value in the text-vector along the dimension corresponding to the term. Since any text contains a limited set of terms (the vocabulary can be millions of terms), most text vectors are very sparse. Most vector based systems operate in the positive quadrant of the vector space, i.e., no term is assigned a negative value.

To assign a numeric score to a document for a query, the model measures the *similarity* between the query vector (since query is also just text and can be converted into a vector) and the document vector. The similarity between two vectors is once again not inherent in the model. Typically, the angle between two vectors is used as a measure of divergence between the vectors, and cosine of the angle is used as the numeric similarity (since cosine has the nice property that it is 1.0 for identical vectors and 0.0 for orthogonal vectors). As an alternative, the inner-product (or dot-product) between two vectors is often used as a similarity measure. If all the vectors are forced to be unit length, then the cosine of the angle between two vectors is same as their dot-product. If $\bar{D}$is the document vector and $\bar{Q}$is the query vector, then the similarity of document D to query Q (or score of for Q) can be represented as:

$$Sim(\bar{D},\bar{Q}) = \sum_{t\ i\ 2\ Q;D} W\ t\ iQ * W\ t\ iD$$

Where $W\ t_{iQ}$ is the value of the i^{th} component in the query vector $\bar{Q}$, and $W\ t_{iD}$ is the i^{th} component in the document vector $\bar{D}$. (Since any word not present in either the query or the document has a $W\ t_{iQ}$ or $W\ t_{iD}$ value of 0, respectively, we can do the summation only over the terms common in the query and the document.) How we arrive at $W\ t_{iQ}$ and $W\ t_{iD}$ is not defined by the model, but is quite critical to the search effectiveness of an IR system. $W\ t_{iD}$is often referred to as the *weight* of term- i in document D.

The vector space model represents the documents and concepts as column and row vector where the cosine angles between the documents are used to measure the similarity between the documents. The similarity between the documents is determined by the closeness of documents in the vector space [**21**].

According to [**19**], Vector space model (VSM) has several advantages. It improves retrieval performance using its term-weighting scheme. Other, it sort and rank the documents according to their degree of similarity to the query using cosine similarity measurement. In addition to this, it is simple to implement and fast.

However, the model has also several disadvantages. It considers terms as unrelated objects in the semantic space. This means, if no common words are shared between the query and documents in text collection, the similarity value will be zero and no document will be retrieved. This is usually happened when users and authors prefer to use different words which have the same meaning [**20**]. Vector space model is not attempt to define uncertainty in IR system and its ranked answers sets is difficult to improve upon without the integration of query expansion and reformulation modules to the model [**21**].

2.3 Probabilistic Information Retrieval

Probabilistic models are based on the general principle that documents in a collection should be ranked by decreasing probability of their relevance to a query. This is often called *the probabilistic ranking principle* (PRP). [**22**] Since true probabilities are not available to an IR system, probabilistic IR models *estimate* the probability of relevance of documents for a query. This estimation is the key part of the model, and this is where most probabilistic models differ from one another. The initial idea of probabilistic retrieval was proposed by Maron and Kuhnsin a paper published in 1960. [**23**]Since then, many probabilistic models have been proposed, each based on a different probability estimation technique.

Due to space limitations, it is not possible to discuss the details of these models here. However, the following description abstracts out the common basis for these models. We denote the probability of relevance for document D by P (R| D). Since this ranking criteria is monotonic under log-odds transformation, we can rank documents by $\log \frac{P(R|D)}{P(\bar{R}|D)}$, where P ($\bar{R}$|D) is the probability that the document is non-relevant. This, by simple bayes transform, becomes $\log \frac{P(R|D)*P(R)}{P(R|D)*P(\bar{R})}$. Assuming that the prior probability of relevance, i.e., P (R),is independent of the document under consideration and thus is constant across documents, P (R) and P ($\bar{R}$) arejust scaling factors for the final document scores and can be removed from the above formulation (for ranking purposes). This further simplifies the above formulation to: $\log \frac{P(D|R)}{P(D|\bar{R})}$.

Based on the assumptions behind estimation of P (D| R), different probabilistic models start diverging at this point. In the simplest form of this model, we assume that terms (typically words) are mutually independent (this is often called the *independence assumption*), and P (D| R) is re-written as a product of individual term probabilities, i.e., probability of presence/absence of a term in relevant/non-relevant documents:

$$P(D|R) = \prod_{t_i \in Q,D} P(t_i|R) \cdot \prod_{t_j \in Q,\bar{D}} (1 - P(t_j|R))$$

Probabilistic information retrieval uses the similarity measure to retrieve the documents that are relevant to the query. The document relevance of the given query using probability model is estimated by the ratio of probability of relevant terms present in the document to the probability of irrelevant terms present in the same document [**24**].

Literature suggested that one of the alternative modeling paradigms, which is capable of solving the above mentioned problem of vector space model is probabilistic IR model [**2**]. However, the

probabilistic model has got its own drawbacks. First, the probability theory analysis takes much more time and efforts, and it offer unnecessary theoretical burden on the researcher. Second, probabilistic model need to guess the initial separation of documents into relevant and non-relevant sets. Third, the model does not take into account the frequency with which an index term occurs inside a document. In other word, all weights are binary [**21**].

2.4 Context Based Information Retrieval

Contextual parameters like location, time, date, and user's details contribute a lot in retrieving more relevant information [**9**].For example, the user has given the search term in context based model as 'dress shop', based on gender information from the user's profile and current location, the search suggest some nearby shops to buy cloths.

Various authors have taken various contextual parameters into account for retrieving relevant information. Liaqaut et al. [**25**]developed context aware information retrieval using role Ontology. This model considers various parameters like the user's profile details, time and date, device used and user's history to retrieve relevant information. This also uses words expander to expand the user terms based on thesaurus, domain Ontology, WorldNet and hyponyms. Multi-agent information retrieval framework based on Ontology developed by Wang and Zhu [**26**]also addresses the contextual problem in the existing information retrieval system. The user's input is converted into command parameter. The user can set his own customized Ontology or domain Ontology to use for searching. The custom Ontology consists of personalized user's requirements and regularly updated based on user's feedback.

Ontology based query expansion was developed by Tuominen et al. [**27**], which uses queries that are expanded with spatio-temporal Ontology. Spatio-temporal Ontology contains concepts related to user's geographical location and time. Gao et al. [**28**] developed an Ontology based enterprise information retrieval. The user query is processed by sentence splitting and extraction

of related enterprise information from Ontology. Semantic query is triggered with the extracted resource to retrieve the relevant information. Semantic based information retrieval for blog is developed by Chen et al. [**29**]. Triggering Semantic query on blog and domain Ontology that are specific to the blog context semantically expands the user's query request. Yu et al. [**30**] redesigned sliding window algorithm for context based information seeking. The semantic relations between the concepts are retrieved using sliding window algorithm. The concepts and the window size are given as input and the number of co-occurrence of concept is retrieved as result.

2.5 Semantic Based Information Retrieval

The semantic based information retrieval intends to search with concepts rather than with terms, which retrieves more relevant information. Various semantic based search techniques have been adopted since the evolution of semantic web. Xiao and Cruz [**31**]designed semantic query processing technique to retrieve personal information. Semantic query is constructed from the user's query input by using one or more domain Ontologies thereby retrieving related information. Ontology driven information retrieval developed by Guan et al. [**32**] allows user to type in or draw the query input. This input is semantically enhanced using query-processing module by triggering a RQL query to the Ontology. The semantically enhanced query is used to extract the information from the document set and retrieve relevant documents. User query input is sent to query builder to semantically enhance it using legal Ontologies in EgoIR [**33**] model. The concepts and instances are associated with the documents in the legal Ontologies thereby helping to enhance the user query input.

Chien et al. [**34**]developed fuzzy Ontology based information retrieval. The initial query is pre-processed to find keyword and eliminate stop words. Different Ontologies like WordNet, domain Ontology and automatically constructed Ontologies are used to semantically extend the user's input. A 3-layer indexing technique was created by Gu and Nian Yu [**35**]to index Ontology with

URL using keyword. The user query input is processed to obtain the semantic query by retrieving the related words from WordNet and fetch the concepts from Ontology for those terms.

Ontology query model was designed by Antonio et al. [**36**], which generates the set of concepts that are related to the user query. The user can visually select the concepts that are more relevant. Lexicon cleansing is used to remove the terms that are not available in the documents. Meta thesaurus based controlled vocabularies are used in Ontology based query expansion by Dong et al. [**37**]for biomedical information retrieval. Meta thesaurus consists of concepts and inter-concept relations. The query expansion uses weight for semantically related vocabularies. Sarkar [**38**]generated Ontology using Fuzzy-valued Variable. To retrieve the relevant document, first the keywords from the query are searched in Ontological forest, followed by the synonym of the keywords. The sub trees of the corresponding trees form the query set.

A Non-metallic pipe information retrieval system using domain Ontology was designed by Rong and Jun [**39**]. The user's natural language query undergoes its initial transformation into concepts. These concepts make use of knowledge base to transform itself to a knowledge base query. Kim and McLeod [**40**]developed "3-Tuple query interface" which uses statistical Ontology to find either subject, linking word or object from two known entities. Known-Net was adopted by Zhoua et al. [**41**] for improvement of the search query using Ontology label. The keyword is analysed for concepts, individuals, relations and attributes. The improvement of the user's query is done through merging, editing and natural language treatment.

2.6 Semantic Similarity Based Information Retrieval

Simple semantic based information retrieval is not complete in most of the scenarios. Semantic similarity is used to measure the similarity between the concepts of the documents to that of the user query. A model of concept-based information retrieval using Ontologies was developed by

Ozcan and Aslandogan [**42**]. The user query is analysed in word space for extraction of concepts. These extracted concepts are matched with knowledge repository retrieved from the word space. The similarity between the concepts within a domain is identified using latent semantic analysis. Nagypál [**43**]developed a model for effective information retrieval using semantic metadata generation and Ontology based query expansion. This method generates the transformed query from the user query by combining query processing from different Ontology heuristics and then perform search independently using traditional search method. Customers' complaint related information retrieval technique using Ontology was formulated byHui-nan et al. [**44**]. The user query is initially expanded using synonyms from thesaurus. Expanded query is compared with the complaint lexicon in order to retrieve feature word. With the help of feature words the search is performed in the entire Ontology model tree.

Qiuyan and Guisheng [**45**]developed semantic similarity based retrieval method. The compound words to the input terms are retrieved from WordNet. Further the argument sets are used for searching the relevant documents. Once the system receives the user query, it is sent to the query-handling module. The extraction of meaningful concepts from the user input and semantic query expansion is done in the query-handling module. The semantic query expansion is done in two stages. First, it expands with the equivalent words using WordNet, followed by a mathematical model to evaluate the weightage of concepts thereby providing higher percentage of relevancy in the retrieved document.

2.7 Semantic Association Based Information Retrieval

Associations are kind of direct or indirect relationship between the concepts that plays a major role in information retrieval. There are various approaches designed for association analysis and discovery. Song et al. [**46**] developed an association based query expansion, which can be employed in information retrieval. The user's query is initially used to retrieve relevant document using traditional keyword based approach. These documents are reweighted based on

association rules, feature selection and Ontologies. A hybrid approach for association search was developed by Jun et al. [**47**]. This method searches using keyword and association. An approximation space of property set is constructed using association search method. Finally the initial document and approximation space are compared and similarity is computed using approximation space and then final document is retrieved. As reported by Jun et al. [**47**]the evaluation F-measure for various query set concludes that this Ontology based retrieval has outperformed both Ontology based and keyword based retrieval.

Chen [**48**]proposed a framework for use of relations in Ontology for information retrieval. The user query is expanded based on candidate relation and the corresponding semantic family is used to expand the query based on hypernym, hyponym and synonym. Semantic association based information retrieval using Ontologies was developed by Barnaghi et al. [**49**]. The user query is semantically analyzed to find the association-using domain Ontology thereby triggering semantic query to retrieve relevant documents. A method to expand queries in the domain of Gene Ontology was proposed by Alejandra et al. [**50**]. The expansion of terms is done with the relationship in the Ontology. New terms from Ontology with lexical distance of one are extracted. If the terms are not in Ontology, then the closest terms to those words are taken to find the resources. Liu and Li [**51**]developed semantic based information retrieval for Enterprise search. The Query-document relevance is computed from the user query with semantically indexed documents. The "query concept recognition" identifies the candidate concepts in the domain Ontology and evaluates query-document relevance to retrieve the relevant documents.

2.7.1Association Rule Mining

Association is a data mining function that discovers the probability of the co-occurrence of items in a collection. The relationships between co-occurring items are expressed as association rules. Association rules were first introduced by Agrawal et al. [**52**] for the purpose of discovering interesting relations between items in retail transaction "baskets" where each basket is

represented as a record in a database. It can be formally described as follows. Let I = {i_1, i_2, . . . ,i_m} be a set of m literals, called items. Formally, let D = {t_1, t_2, . . . ,t_n} be a database of n transactions where each transaction is a subset of I. An itemset is a subset of items. The support of an itemset S, denoted by sup(X), is the percentage of transactions in the database D that contain S. An itemset is called frequent if its support is greater than or equal to a user specified threshold value.

An association rule r is a rule of the form X $\Rightarrow$ Y where both X and Y are non-empty subsets of I. X is the antecedent of r and Y is called its consequent. The support and confidence of the association rule r: X $\Rightarrow$ Y are denoted by sup(r) and conf(r).These measures can be interpreted in terms of estimated probabilities as sup(r) = P(X $\subseteq$ Y) and conf(r) = P(Y_jX) respectively. Another interesting measure is the lift or interest factor of a rule, denoted by lift(r), which computes the ratio between the rule probability and the joint probability of X and Y assuming they are independent, that is, P(X $\subseteq$ Y)/(P(X) P(Y)). If this value is 1, then X and Y are independent. The higher this value, the more likely that the existence of X and Y together in a transaction is not just a random occurrence, but because of some relationship between them.

The basic task of association data mining is to find all association rules with support and confidence greater than user specified minimum support and minimum confidence threshold values [2].

Mining association rules basically consists of two phases: (1) compute the frequent itemsets. The minimum support, and (2)generate rules from the frequent itemsets the minimum confidence. As the number of frequent sets can be very large, closed and maximal itemsets can be generated instead.

Association rules are often used to analyze sales transactions. For example, it might be noted that customers who buy cereal at the grocery store often buy milk at the same time. In fact, association analysis might find that 85% of the checkout sessions that include cereal also include milk. This relationship could be formulated as the following rule.

Cereal → milk with 85% confidence

This application of association modeling is called **market-basket analysis**. It is valuable for direct marketing, sales promotions, and for discovering business trends. Market-basket analysis can also be used effectively for store layout, catalog design, and cross-sell.

Association modeling has important applications in other domains as well. For example, in e-commerce applications, association rules may be used for Web page personalization. An association model might find that a user who visits pages A and B is 70% likely to also visit page C in the same session. Based on this rule, a dynamic link could be created for users who are likely to be interested in page C. The association rule could be expressed as follows.

A and B → C with 70% confidence

Support vs. Confidence

Support (sometimes called *frequency*) is simply a probability that a randomly chosen transaction *t* contains both itemsets*A* and *B*. Mathematically,

Support $(A \Rightarrow B)_t = P(A \subset t \wedge B \subset t)$ = the number of transactions containing both *A* and *B* divided by total number of transactions[43]

$$support(A \Rightarrow B)_t = P(A \subset t \wedge B \subset t) = \frac{\#\text{ of transactions containing both } A \text{ and } B}{\text{total }\#\text{ of transactions}}$$

In simplified notation used for support is

$$support(A \Rightarrow B) = P(A \wedge B)$$

Confidence (sometimes called *accuracy*) is simply a probability that an itemset *B* is purchased in a randomly chosen transaction *t* given that the itemset *A* is purchased. Mathematically [43],

Confidence $(A \Rightarrow B)_t = P(B \subset t \mid A \subset t)$ the number of transactions containing both *A* and *B* divided by total number of transactions containing A.

$$confidence(A \Rightarrow B)_t = P(B \subset t \mid A \subset t) \frac{\#\text{ of transactions containing both } A \text{ and } B}{\text{total } \#\text{ of transactions containing A}}$$

Simplified notation used for confidence is

$$confidence(A \Rightarrow B) = P(B \mid A)$$

It should be mapped that to find interesting rule, there is a need to detect those with high support and high confidence. To this end, one has to select appropriate thresholds for both measures and then look for all subsets that fulfill given support and confidence criteria.

2.7.2 Term association graph model

The term association graph model is an enhanced model of the VSM. The traditional weighting schemes such as Boolean weighting, term frequency (*tf*) weighting and term frequency-inverse document frequency (*TF-IDF*) weighting assign weight value to a term according to its importance without considering the term associations. The Boolean weights, *tf* weights and *TF-IDF* weights determine the weight of each term in a document independently. Thus, rich

information such as relationships existing among the terms in a document is not considered by traditional term weighting schemes [**53**].

A graph is a pair $G=(V,E)$. The elements of V are the vertices (or nodes) of the graph G and the elements of E are its edges. Usually a graph is represented by drawing a dot for each node and joining two of these dots by a line if they are connected by an edge and labelling of nodes and edges are marked if necessary. Figure 2.1 shows the process involved in construction of Term Association Graph model.

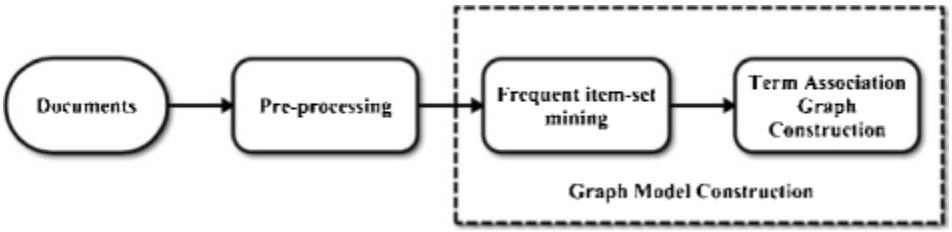

Figure 2.2 Process involved in term association graph model.

Preprocessing

Extract all the terms from the collection of documents. Each text document in the corpus is treated as a transaction in which each word is an item similar to that of a transaction in Association Rule Mining (ARM) approach described in Han & Kamber [**54**]. However, not all the terms in document are important to be retained in the collection. Typically, there are two-step processes to preprocess natural language text documents. Firstly, stop words removal (e.g., articles, prepositions, etc.). The terms that appear frequently in the document but have no significant meanings may be removed. Secondly, stemming of words is done to keep only the root form of words. The most widely used stemming algorithms are porter and Lancaster algorithm.

Frequent item-set mining: The idea here is to capture the relationships among the terms in document using the frequent item-set mining algorithm which is based on Apriori algorithm

in Agrawal & Srikant [**55**]. In this context, an item-set is a set of words that occur together. Each resulting frequent item-set is a document description which is a possible terms or concepts. An item-set can define as frequent if it appears in more than two documents [**52**].

Term association graph construction: The goal is to explore and discover the relationships among terms of the text document in the corpus and to devise a strategy to make use of Semantic ranking schemes for information retrieval these semantic associations in the re-ranking task [**55**].In order to construct the graph from the set of frequent item-sets mined from the text collections, firstly create a node for each unique term that appear at least once in the frequent item-sets. Secondly, create edges between two node 'a' and 'b' if and only if they are both contained in one frequent item-set. Besides this, the weight of the edge between a and b is the largest support value among all the frequent item-sets that contain both terms i.e., a and b

2.8 Query Expansion

Query expansion (**QE**) is the process of reformulating a seed query to improve retrieval performance in information retrieval operations [**2**]In the context of search engines, query expansion involves evaluating a user's input (what words were typed into the search query area and sometimes other types of data) and expanding the search query to match additional documents. Query expansion involves techniques such as [**56**]

- Finding synonyms of words, and searching for the synonyms as well
- Finding all the various morphological forms of words by stemming each word in the search query
- Fixing spelling errors and automatically searching for the corrected form or suggesting it in the results
- Re-weighting the terms in the original query

Search engines invoke query expansion to increase the quality of user search results. It is assumed that users do not always formulate search queries using the best terms. Best in this case may be because the database does not contain the user entered terms.

By stemming a user-entered term, more documents are matched, as the alternate word forms for a user entered term are matched as well, increasing the total recall. This comes at the expense of reducing the precision. By expanding a search query to search for the synonyms of a user entered term, the recall is also increased at the expense of precision. This is due to the nature of the equation of how precision is calculated, in that a larger recall implicitly causes a decrease in precision, given that factors of recall are part of the denominator. It is also inferred that a larger recall negatively impacts overall search result quality, given that many users do not want more results to comb through, regardless of the precision.

The goal of query expansion in this regard is by increasing recall, precision can potentially increase (rather than decrease as mathematically equated), by including in the result set pages which are more relevant (of higher quality), or at least equally relevant. Pages which would not be included in the result set, which have the potential to be more relevant to the user's desired query, are included, and without query expansion would not have, regardless of relevance. At the same time, many of the current commercial search engines use word frequency (TF-IDF) to assist in ranking. By ranking the occurrences of both the user entered words and synonyms and alternate morphological forms, documents with a higher density (high frequency and close proximity) tend to migrate higher up in the search results, leading to a higher quality of the search results near the top of the results, despite the larger recall [**57**].

There are two types of query expansion strategies [**58**]: global analysis and local analysis. Global analysis strategy examines all documents in the collection so as to expand query. Local analysis examine only documents retrieved automatically for a given query q to determine query expansion. Query expansion techniques can be classified into the following main groups according to survey [**56**].

External Resource Based Query Expansion: In these approaches, the query is expanded using some external resource like WorldNet, lexical dictionaries or thesaurus. These dictionaries are built manually which contain a piping's of the terms to their relevant terms. There techniques involve look up in such resources and adding the related terms to query.

QUERY LOGS BASED EXPANSION: With the increase in usage of Web search engines, it is easy to collect and use user query logs. Query logs are maintained by each search engine in order to analyze the behavior of the user while interacting with search engine. These kinds of approaches use these query logs to analyze the user's preference and adds corresponding terms to query. This method can fail when user wants to search something which is not at all related to earlier searches.

RELEVANCE FEEDBACK BASED EXPANSION: Relevance feedback based methods execute the initial query on collection and extract top k documents. Then the ranked documents are used to improve the performance of retrieval. It is assumed that the initial retrieved documents are relevant and thus can be used to extract expansion terms. These models fail when the initial retrieval algorithm of search engine is poor.

QUERY EXPANSION USING WIKIPEDIA: Wikipedia is the biggest encyclopedia available freely on the web. Wikipedia content is well structured and correct Li et al [**59**]. Many works suggest use of Wikipedia as source for query expansion [**8**] [**59**] and [**60**]. [**59**]Proposed query expansion using Wikipedia by utilizing the category assignments of its articles. The base query is run against a Wikipedia collection and each category is assigned a weight proportional to the number of top-ranked articles assigned to it. Articles are then re-ranked based on the sum of the weights of the categories to which each belongs.

2.9 Related works

Song et al. [**46**] developed semantic query expansion using association rules. This model first fetches set of documents with the user query. Based on the user selection, the query is reformulated by analyzing the association in the pages with the user input terms. A method of rough Ontology based information retrieval was proposed by Jun et al. [**47**] where the initial set of documents are retrieved using regular keyword based approach and an association search in Ontology is also made in parallel. Degree of similarity of elements is used in order to rank the documents.

Gantayat [**61**] designed automatically constructs the domain ontology from the given lecture notes. He shows how this ontology can be used to identify the pre-requisites and follow-up modules for a given query (lecture topic). And also provide the user with a dependency graph which gives a conceptual view of the domain. The system extracts the concepts using "term frequency inverse document frequency (TF-IDF) weighting scheme" and then determines the associations among concepts using "apriori algorithm".

Chen [**48**]explored the use of relations in Ontology to retrieve relevant information where the user query terms are identified in Ontology and further their relations are used to expand the query. Barnaghi et al. [**62**]reformulated semantic association discovery algorithm, where complex relations between entities is found by recursively finding all the paths between the Ontologies and finally ranking based on domain Ontology. Alejandra et al. [**50**] designed Ontology based query expansion to search for learning resource. Common Ontological relations, domain specific relations, traditional terminological relations are used in query expansion process with a lexical distance of one If the terms don't match, then closest term is used in retrieval of documents.

On August 2013 Google updated to Hummingbird [**63**]algorithm which makes the search more semantic way, where it understands the meaning behind the user query to retrieve the documents. Google incorporated this semantic search to analyze the user query and linking it to knowledge graph. Bing also has introduced adaptive search [**64**]where the system knows the user and the context of search to determine the search results. So, different user with the same keyword will get different results based on their previous activities.

There are lots of advantages in incorporating appropriate semantic in existing search feature. The search engine understands the user query by incorporating semantics of the given keyword in the search. Search engines can perform quite remarkable by getting the user directly what they need. In the analysis, it is found that the semantic association based model intend to retrieve more relevant documents. This work utilizes the association rules extensively from the corpus to retrieve more relevant documents.

Local Related works

Explosion in electronic text written in Amharic language caused an increasing need of designing effective and efficient IR system for Amharic texts [**11**]. A number of IR systems developed so far for retrieving Amharic text and attempts to optimize search result using query expansion.

Abey [**13**]used Co-occurrence analysis method in his model so as to improve the power of the query in representing the user information need. He attempted a semantic based query expansion to minimize the problem of a polysemous Amharic word. In his model he tested statistical Co-occurrence analysis, bi-gram and thesaurus three different methods for selecting the terms which are relevant to the expansion.

The statistical Co-occurrence analysis technique, which select expansion terms from the pseudo relevance feedback by the terms only analyzing terms Co-occurrence information with query terms. The Co-occurrence of the terms only analyzed from the top n relevant documents returned from the first searching attempt. The experimented result with this method exhibited that it increases precision by 14% with a decrease in recall by 10%.

The bi-gram technique which also uses the pseudo relevance feedback for expansion, selects those terms which appear to the right or left of a query term from the top n relevant document returned from the first search attempt. The experiment result with this method exhibited that it increases the precision by 18% with decrease in recall by 16%.

The 3ndmethodhe used was bi-gram thesaurus technique. Unlike the other two this method uses a pre-built thesaurus to select expansion terms. This method used the whole corpus when the bi-gram thesaurus is constructed unlike the other methods. The experiment result with this method showed that it increased the precision by 18% and decreasing of recall by 16%.

Among the three methods he concluded that the statistical Co-occurrence analysis is better one. However, because some expansion terms selected to expand the original query becomes polysemous themselves and affect the performance of the expansion and effectiveness of Amharic IR system. Such approaches showed significant improvement, but require manual tuning in order not to adversely affect performance: too few expansion terms may have no impact, and too many will cause a query drift [2]

The research done by Iman [**14**] on query expansion Amharic IR system, followed the recommendation of Abay [**13**] that is query expansion using ontology. External information source WordNet is used to expand the query. And she used the word sense disambiguation technique to overcome the polysemous problem experienced in Abye [**13**]research.

WordNet is used to find out additional terms for expansion process. Once the additional terms are identified from the WordNet, before the terms are used to reformulate the original query, their sense is checked using word sense disambiguation method. Using this technique in her proposed model, terms which have a correct sense are used to expand the original query.

The experiment result showed that the system overall performance improved by 31% and promisingly avoid polysemous word problems due to the ability of the system in identifying the correct sense using the word sense disambiguation technique. Using Word Net as a source of expansion terms may improve the result. However, WordNet is too broad; it may include useless terms for the given query and can bring noises in the retrieved result.

Samrawit [**17**] study word sense disambiguation using semantic similarity for query expansion. The word sense disambiguation and query expansion was based on lexical resource form Amharic dictionaries. The number of information associated to each terms was limited do to lock of resource. The inaccuracy disambiguation is the main obstacle in achieving significant improvement using these methods, as incorrect disambiguation is likely to harm performance rather than merely not improve it.

Ontology based query expansion for enhancing the performance of Amharic information retrieval is further attempted by Welde [**15**].The research was to explore the impact of the use of ontology for query expansion to improve information retrieval results. In the research, an attempt was made to integrate an ontology base query expansion technique to enhance the effectiveness of the system.

The ontology domain used was manually constructed on the tourism sector. It covers few in the tourism. The ontology contains 7 top level class, 20 subclass(concept) and several instances per classes.

Experiments were done to explore the extent to which ontology based query expansion techniques improve performance of the system. The first experiment was carried out by

concatenating user query and the terms found from the ontology. The experiment results shows that the performance of the system improved by 42% in terms of recall and 7% overall system improvement, while precision was declined. Using terms found from the ontology helps more to retrieve more documents. However, to increase the precession of the retrieval result he suggests further research. And automatically constriction of the ontology because manually of ontology labor intensive and time-consuming task

Berhanu [**65**] present context aware semantic search engines for mobile phones. He developed the architecture of context aware semantic search engines on the top of semantic web. Ontology based context modeling is used to represent user contexts. The user's query history and submitted query are also analyzed to identify the concept of the search. Prototype of the system was also developed to demonstrate and the test applicability and effectiveness of the proposed approach.

Summary

Various types of information techniques like keyword based search, vector space model, probabilistic model, context based model and semantic based model have been discussed. The conventional information retrieval system retrieves documents based on pattern matching. This method does not fetch relevant documents unless the user knows the exact keyword as specified in the documents. To overcome this issue, many researchers have been performed to incorporate the semantic intelligence in search engine, which helps to fetch more relevant documents.

In vector space model [**21**], the documents are retrieved based on the cosine similarity between the documents and the keywords. In probabilistic model [**22**], the documents are retrieved based on the ratio of relevant portion of the document to the irrelevant portion of the documents. Context based model [**25**]had some impact over other models as it considers the user context of search such as user profile, location, time, etc.

Semantic search is performed based on semantic similarity, semantic association and semantic annotation. Semantic similarity is the method of adding similar terms from any lexical data source, which provided alternate words for a given word [**51**]. Semantic association based information retrieval analyses the relation between the user entered concepts to understand the user context of query to retrieve the relevant documents. In semantic annotation based information retrieval, the developer will annotate the webpage with all possible keywords through which the user will get into the webpage. It makes the implementation of search engine much simpler as it uses the tags as given by developers to index the documents [**46**].

Even if several IR systems have been developed in the last decade, most of works are attempt to design an Amharic IR system using vector space model that face problem with control uncertainty nature of IR system. Probabilistic model make the initial guess based on Boolean expression, which inhibit to know important words to represent a document and, accordingly may not retrieve relevant documents that contain large number of terms found in a given query. Co-occurrence analysis from the top n relevant documents returned from the first searching attempt. Flails, because some expansion terms selected to expand the original query becomes polysemous themselves and affect the performance of the expansion and effectiveness of Amharic IR system.

External information source WordNet is used to expand the query. Word Net was too broad; it may include useless terms for the given query and can bring noises in the retrieved result. Manually constructed ontology based query expansion have need expert and takes time. The study attempts to explore and discover word association automatically to enhance the performance of Amharic information retrieval system.

CHAPTER THREE

Methods and Procedures

The aim of this chapter is to discuss on methods and procedures used in designing word association rule based information retrieval. The two major components which are indexing and searching are used [**21**]. Indexing is an offline process used to facilitate access to preferred information from document corpus accurately and efficiently as per users query. Searching is an online process used to scan document collection so as to find relevant documents that satisfy users need .

3.1 The Architecture

This part describes the architecture used in our association rule based information retrieval. In the architecture design described below, we did not consider the type of relation for calculation of similarity between terms. We assumed that the set of relevant concepts in the query is known. But this condition can be achieved with any technique for assigning concepts from ontology to a query, e.g. based on manual assignment or based on synonyms to query terms, making use of WordNet or other.

Figure 3.1 depicts the word association rule based IR system architecture designed and implemented in this research. It indicates that, the IR system has online (Searching) and offline (Indexing) processes. During the offline process, Amharic documents are organized using inverted indexing structure. To ease the indexing task, text operations are applied to the text of the documents in order to transform them into their logical representation. The documents are indexed and the index is used to generate association rule and execute the search. The searching task is an online process that accepts users query. The query is processed to identify terms using which searching is done to identify and retrieve ranked relevant documents.

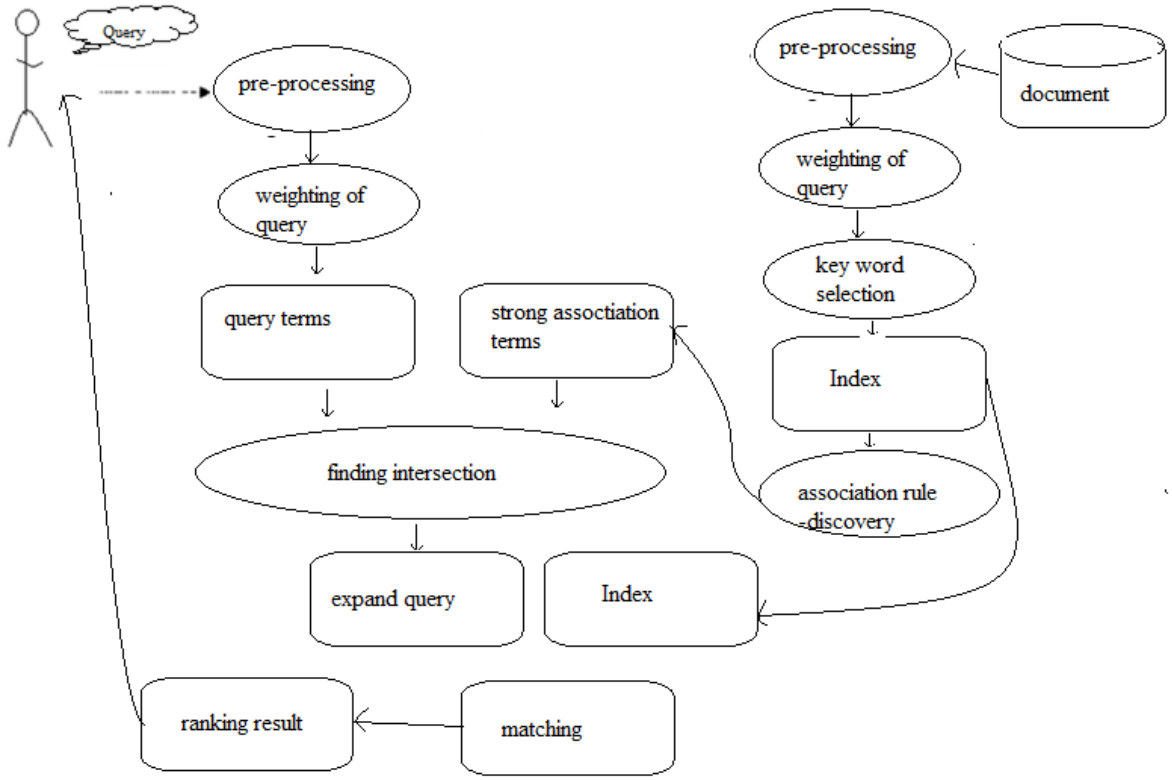

Figure 3.1 The system architecture of association rule based information retrieval

The way in which a query is processed by this approach is shown in the Figure 3.1.First text operation pre-process including (i.e., tokenization, stop word elimination stemming, and normalization) are done. After preparing the corpus, we select terms (keywords extractions) using TFIDF. In this process, we extract all the terms (one or more words) used to denote concepts in the domain, using the method of "repeated segments" (frequent item set) and the TFIDF score. For a given query term appropriated concepts are retrieved from the association rule. Then the set of concepts associated with each document is retrieved from database. As next, these two sets are compared using simple metric, which expresses the similarity between a document D_i and given query Q vector similarity.

$\text{Sim}_{\text{assoc}}(\bar{Q},\bar{D}_i)=\begin{cases}(Qcon \cup Dcon) if (\text{Qcon} \cup Di,con) \\ k\end{cases}$

Where Q_{con} is a set of concepts assigned to query Q and D_{con} is a set of concepts assigned to document Di, and k is small constant. Resulted number represents association-based matching similarity measure. The final similarity is then computed as multiplication e.g.

$$Sim(\bar{Q},\bar{D}_i) = Sim_{assoc}(\bar{Q},\bar{D}_i) * sim_{\text{TF-IDF}}(\bar{Q},\bar{D}_i)$$

The system architecture for association rule based information retrieval system is depicted in figure 3.1 the system architecture has text operation pre-processing including (i.e., tokenization, stop word elimination stemming, and normalization).

3.2 processing and text operation

Several works have been done as to index document collection/corpus. These tasks include; tokenization, stemming, stop word removal and normalization.

3.2.1Tokenization

Tokenization is the process of forming words from the sequence of characters in a document [**57**]. In English text, this appears to be simple. In many early systems, a "word" was defined as any sequence of alphanumeric characters, terminated by a space or other special character.

To be able to incorporate the range of language processing required makes matching effective, the tokenizing process should be both simple and flexible. One approach to doing this is for the first pass of tokenizing to focus entirely on identifying markup or tags in the document. This could be done using a tokenizer and parser designed for the specific markup language used (e.g., HTML), but it should accommodate syntax errors in the structure, as mentioned previously. A second pass of tokenizing can then be done on the appropriate parts of the document structure.

Some parts that are not used for searching, such as those containing HTML code, will be ignored in this pass [**2**].

The most general tokenizing process will involve first identifying the document structure and then identifying words in text as any sequence of alphanumeric characters, terminated by a space or special character. This is not much more complicated than the simple process. We described, but it relies on information extraction and query transformation to handle the difficult issues. In many cases, additional rules are added to the tokenizer to handle some of the special characters, to ensure that query and document tokens will match.

In this research, such kind of problems is solved

• Special characters: include any special sequence of characters delimited by letters or spaces.

• Words in Latin characters: The non-Amharic words, mainly in Latin characters are simply detected by their *graphic*.

• Compound words: we assumed compound words are appeared in bigram form.

3.2.2Compilation of the Stop word List

Stop words can be compiled from the sample text using rank-frequency distribution or by checking for list of known stop word lists (dictionary). The rank-frequency distributions of words, especially in a language which have high morphological complex words are not convenient to determine stop words. But, the frequency may guide in selecting stop words [**21**]. Some words such as prepositions, conjunction and articles in Amharic can exist affixed to words.

The second method is using stop word list for the language [**2**] [**11**]. In this research the second method was used to apply stop word removal. The stop word list is adopted [**16**]. Removing stop words were applied after stemming is implemented. This is because, Amharic language is rich in word variant, thus applying stemming prior to stop word removal reduces the problem of considering morphological variant words as different in meaning from IR system..

The common stop word list includes lists such as connecters, articles, infinitives and so on. It can also includes verb to be words such as **አነ** /’ ‘äni/ ‘am, was’, **አንተ** /‘äntä/ ‘are, were’ **አንቺ** /‘änchi/‘are, were’, **እናንተ**/w‘anante/ ‘are,were’, **እሱ**/’asu ’/ ‘is, was’, **እኛ** /ana/ ‘are, were’, **እነሱ** /änesu/‘are, were’, **አንትን** /äntn/ ‘are, were’, **እስዋ**/’aswa/ ‘is, was’.

Different words including stop words can exist in various forms due to affixes. Hence, it is better to use stop word list to remove stop words. For this work, lists of stop words dictionary was adopted from [**11**], [**16**], and Amharic language stop words. In addition, we used document frequency and TFIDF to huddle document specific stop words. Then automatic checking was done before stemming as well as during each stemming process. The complete list of the stop word lists compiled from the sample text is given in Appendix III. Table 3.1 presents a sample list of stop words in Amharic writing system.

Stop Word Lost									
ነው	እኔ	እኛ	እነሱ	እሱ	እሷ	አንተ	እናንተ	እና	ወደ
ነይ	ወይ	ከ	ናቸው	ትናት	ጥቂት	በርካታ	ብቻ	ሁሉም	ሌላ
ሌሎች	ሁሉ	እያንዳንዱ	እያንዳንዳቸው	ስለ	እንዲሁም	እንጂ	ደግሞ	መካከልከ	ሰሞኑን
ከሰሞኑ	በሰሞኑ	የሰሞኑ	ትናንት	ትናንትና	ጋራ	የጋራ	ከጋራ	ተለያዩ	ተለያየ
ድረስ	እስከ	በጣም	ግን	ሲሆን	ሲል	ዉስጥ	ላይ	ናት	ነበሩ
ነበረች	ያ	ወይዘሮ	ወይዘሪት	ነገሮች	ከፊት	ከላይ	ታች	ከታች	በታች
የታች	በውስጥ	ከውስጥ	ጋር	ናቸው	ይህ	በላይ	ወደ	ወዘተ	እና
ወይም	እንደ	አቶ	ፊት	ወደፊት	ነገር	በፊት	በሆላ	በኩል	

Table 3.1 List of stop word (adopted from Amanuel [16])

Stop word elimination algorithm

Step 1: open database and select table containing the tokenized term
Step 2: get term
Step 3 for each term do -> If not it in stop word list go to step 4
Step 4: remove term; return 0;
Step 5: return term;

3.2.3 Stemming

Stemming [**19**] is the process of reducing morphological variants of a word into a common form. It is used to decrease the number of words to be indexed and also to increase the performance of keyword identification. In Amharic writing system, words are morphological variants. Morphological variant words have similar semantic interpretations. In IR system, those words considered as equivalent words. Therefore, words have to be reduced to their root using stemming technique. On the other hand, stemming 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 the smaller storage space and processing time required.

Amharic stemming algorithms have been developed by different parties of researchers. Alemayehu and Willet [**66**]developed an iterative and context sensitive stemmer that removes both prefixes and suffixes; Alemu and Asker [**67**]developed a rule-based stemmer that uses a machine-readable dictionary to reduce terms to their root forms.

By stemming all Amharic words, then we get only stemmed words that decrease the number of unique words in the given corpus, resulting efficient keyword extraction. As morphological variant words are grouped together into a single term, it reduces the number of words in the system and hence increases the performance of the system. There are many methods for stemming some uses linguistic dictionaries for stemming whereas others use algorithmic rules based on suffixes list to get the stemmed word.

According to Alemayehu and Willet [**66**], a steamer that stems words without consideration of remaining stem, which removes words that are similar to prefix and suffix list but that are not actually affixes is called context-free stemmer. For instance, the English word “regular” and “metal” will be “gular” and “met” respectively if context-free steamer removes “re” and “al”. Similarly, weak result will appear if such technique is applied for Amharic language. For example, for the Amharic word “**ክለከል**” which means forbid, if we remove **ክ**”ke” which is

found in prefix list, the result is**ለክል**which does not have any relation with the correct stem. Therefore, in this study context-sensitive approach is used with affix removal being controlled. We used a list of prefix and suffix which was used by Amanuel [**16**]. Some of the prefix and suffix used for Amharic stemming are presented in table 3.2.

Prefix lists					Suffix lists				
የ	ስለ	የሚ	እየ	ስለሚ	ች	ኝ	ችን	ቸው	ዊት
እያ	እንደ	አል	አለ	በ	ና	ዎች	ኛ	ዎቻቸው ም	ው ም
ለ	ከ	ይ	እንዳ ይ	ሲ	ው	ዎች ም	ው ያን	ዎቹ	ናቸው
እንዲ	እስክ	ከነ	እን	እነ	ባቸው	ዊያን	ነት	ያዊ	ን
					ት	ሉ	ችው	ዊ	ዊቷ
					ቹ ን	ዬ	ዎ	ህ	ሽ
					ዋ	ሁ	ለት	ላት	ላቸው
					ላችሁ	በት	ባት	ባቸው	ባችሁ
					ቱ	ሽ	ይቱ	የው	ኞች

Table 3.2 List of Amharic prefix and suffix (adopted from Amanuel [16])

Suffix algorithm

Step 1: open database and select table containing the tokenized term

Step 2 for each term do -> if term character size greater than 3

Step 3: THEN assign the last length (WORD) -2 Characters to a temporary suffix, TEMPP ELSE assign the first MAXPL characters to TEMPP (where MAXPL is the Length of the longest suffix in the list)

Step 4: Find TEMPP in the suffix list

Step 5 if

Step 5: IF found THEN check for context-sensitive rules If it is sensitive do noting go to step 7

Step 6: Remove suffix

Prefix algorithm

Step 1: open database and select table containing the tokenized term
Step 2 for each term do -> if term character size greater than 3
Step 3: THEN assign the first length (WORD) -2 Characters to a temporary prefix, TEMPP ELSE assign the first MAXPL characters to TEMPP (where MAXPL is the Length of the longest prefix in the list)
Step 4: Find TEMPP in the prefix list
Step 5: IF found THEN IF context sensitive THEN check the context-sensitive rules and GOTO step 7 ELSE GOTO step 6 ELSE IF length (TEMPP) >1 THEN assign length (TEMPP) -1 Characters to TEMPP and GOTO step 4 ELSE return WORD
Step 6: Remove prefix
7: return term

3.2.4 Normalizing Characters and Words

Text normalization is a process by which text is transformed in some way to make it consistent in a way which it might not have been before[37].Normalizing text before storing or processing it allows for separation of concerns, since input is guaranteed to be consistent before operations are performed on it. Text normalization requires being aware of what type of text is to be normalized and how it is to be processed afterwards; there is no all-purpose normalization procedure. It is rather language dependent.

The Amharic writing system contains characters or symbols, which have the same sound during reading. For example, the characters "ሀ, ሐ, ኀ, አ and ዐ" produces the same sound during reading with characters "ሃ, ሓ, ኃ, ኣ and ዓ" respectively. On the other hand, there are also alphabets that share the same sound, such as, "ሀ,ሐ, and ኀ", "ሰ and ሠ", "አ and ዓ", and "ጸ and ፀ". As a result,

words which have the same meaning may have a different spelling structure. In this research, such kind of problems is solved by changing the characters into their common standard form.

Normalization algorithm

Step 1: open database and select table containing the tokenized term
Step 2 for each term do -> check each characters of term
Step 4: IF term character Find in the normalization list
Step 5: changingthe characters in to their common standard form
Step 6: return term

3.3Term weighting

For weighting term and define their importance TF-IDF weighting technique is used. Term frequency inverse document frequency (TF-IDF) is defined as the number of occurrences of terms in a given document multiplied by the inverse of the number of documents where the term appears [**68**]. Given a document collection *D*, a word *w*, and an individual document $d \in D$TF-IDF *of* word *w* is defend as following.

$$\text{TF-IDF}wd = f_{w,d} * log(\frac{|D|}{fw,D})$$

Where

• $f_{w,d}$ equals the number of times *w* appears in *d*, |*D*| is the size of the corpus (number of documents), and $f_{w,D}$ equals the number of documents in which *w* appears in *D*.

The TF-IDF value is

• High when a term occurs many times within a small number of documents,

• Low when the term occurs fewer times in a document, or occurs in many documents,

• Lower when the term occurs in virtually all documents.

Using TF-IDF weighting scheme, we scored each word in the given text corpus. Then we found out the top frequent words, according to the TF-IDF weights. In this way, we extracted top keywords for the given text. All the keywords extracted are stored in a MySQL Database table.

3.4 Indexing:

Inverted indexes are unrivaled in terms of retrieval efficiency: indeed, as the same term generally occurs in a number of documents, they reduce the storage requirements. The principle behind the inverted index is very simple. First, a dictionary of terms (also called a vocabulary or lexicon), V, is created to represent all the unique occurrences of terms in the document collection C. Optionally, the frequency of appearance of each term ti∈ V in C is also stored. Then, for each term $t_i \in$ V, called the posting, a list L_i , the posting list or inverted list, is created containing a reference to each document $d_j \in$ C where t_i occurs. In addition, L_i may contain the frequency and position of t_i within d_j . The set of postings together with their posting lists is called the inverted index or inverted file or postings file [**2**].

3.5 Finding word association

In order to perform predictive analysis, it is useful to discover interesting patterns in the given data set that serve as the basis for estimating future trends. Association Analysis or Association Rule Mining is helpful here. This refers to the discovery of attribute-value associations that occur frequently together within a given data set [**51**] [**52**].

An association rule is a directed relation between two sets of items, from the antecedent to the consequent. Formally,: Let I = { i_1, i_2, ...,i_k} be set of items, D be task relevant data of transactions, T be each transaction, a set of items, such that T ⊂I where ⊂denotes proper subset and TID be the Transaction Identifier. An Association Rule is defines as an implication of type A => B, where A⊂I, B ⊂I and A ∩ B = Φ. The Rule holds in D with confidence C and support S,

where C: Confidence (A=>B) = P (A U B), S: Support (A=>B) = P (B | A) where P is probability.

To identify the relation between different keywords and to construct the association rule, the research use apriori algorithm. The Apriori Algorithm proposed by Agrawalet.et al. [**55**], finds frequent items in a given data set using the anti-monotone constraint. This algorithm embodies the following.

• Given a data set, the problem of association rule mining is to generate all rules that have support and confidence greater than a user-specified minimum support and minimum confidence respectively.
• Candidate sets having k items can be generated by joining large sets having k-1 items, and deleting those that contain a subset that is not large (where large refers to support above minimum support).
• Frequent sets of items with minimum support form the basis for deriving association rules with minimum confidence. For A=>B to hold with confidence C, C% of the transactions having A must also have B.
The Apriori Algorithm that fined the association among words is presented below.
• Frequent Item sets: The sets of item which has minimum support (denoted by L_i for i^{th} -Item set).
• Apriori Property: Any subset of frequent itemset must be frequent.
• Join Operation: To find L_k , a set of candidate k-itemsets is generated by joining L_{k-1} with itself

Find the *frequent item sets*: the sets of items that have minimum support

– A subset of a frequent item set must also be a frequent item set
• i.e., if {*AB*} is a frequent item set, both {*A*} and {*B*} should be a frequent itemset
– Iteratively find frequent item sets with cardinality from 1 to *k (k*-item set*)*

• Use the frequent item sets to generate association rules.

• Join Step: C_k is generated by joining L_{k-1} with itself

• Prune Step: Any (k-1)-item set that is not frequent cannot be a subset of a frequent k-item set

Apriori Algorithm

1. For each Candidate item set of size k
2. If Candidate item set greater minimum support
3. for each select frequent item set of size k that appeared with Candidate item set
4. compute the confidence
5. if confidence greater than minimum confidence go to step 6
6. return frequent items have strong association with Candidate item set
7. next frequent item set of size k step 3
8. next Candidate item set of size k step 1

3.4. Evaluation methods

Retrieving relevant document from the collection that satisfies users need is the heart of IR system evaluation in determining its effectiveness. Two strategies are identified in measuring the effectiveness performance of IR systems. The first is the user-centered strategy, which uses relevant judgment so as to evaluate the performance of the system and the second is system centered strategies which work based on reference judgment available prior to testing process [**19**].Based on the concept of relevance (i.e. to a given query or information need), there are several techniques of measures of IR performance available, such as, precision and recall, F-measure, E-measure, MAP (Mean average precision), R-measure, etc [**57**]. In this study, the

three widely used techniques precision, recall, and F-measure are used to measure the effectiveness of the IR system designed.

It is well accepted that a good IR system should retrieve as many relevant documents as possible (i.e., havea high recall), and it should retrieve very few non-relevant documents (i.e., have high precision). Unfortunately, these two goals have proven to be quite contradictory over the years. Techniques that tend to improve recall end to hurt precision and vice-versa. Both recall and precision are set oriented measures and have no notion of ranked retrieval.

Researchers have used several variants of recall and precision to evaluate ranked retrieval. For example, if system designers feel that precision is more important to their users, they can use precision in top ten or twenty documents as the evaluation metric. On the other hand if recall is more important to users, one could measure precision at (say) 50% recall, which would indicate how many non-relevant documents a user would have to read in order to find half the relevant ones.

One measure that deserves special mention is average precision, a single valued measure most commonly used by the IR research community to evaluate ranked retrieval. Average precision is computed by measuring precision at different recall points (say 10%, 20%,and so on) and averaging Relevant retrieved [**57**].Recall is the percentage of relevant documents retrieved from the database in response to users query, whereas precision is percentage of retrieved documents that are relevant to the query [**21**].

$$\text{Precision} = \frac{\text{number of Relevant item retrieved}}{\text{number of Retrieved}} \quad (3.1)$$

$$\text{Recall} = \frac{\text{number of Relevant item Retrieved}}{\text{Total Number Of Relevant}} \quad (3.2)$$

The F measure is an effectiveness measure based on recall and precision that is used for evaluating classification performance and also in some search applications. It has the advantage of summarizing effectiveness in a single number. It is defined as the harmonic mean of recall and precision, which is precision and recall (or positive predictive value and sensitivity) [**21**].

$$F=\frac{2\cdot precision*recall}{precision+recall} \quad (3.3)$$

The harmonic mean emphasizes the importance of small values, whereas the arithmetic mean is affected more by values that are unusually large (outliers). A search result that returned nearly the entire document collection, for example, would have a recall of 1.0 and a precision near 0. The arithmetic mean of these values is 0.5, but the harmonic mean will be close to 0. The harmonic mean is clearly a better summary of the effectiveness of this retrieved set [**57**].

CHAPTER FOUR

Experimentation

This research attempts to design an association rule-based information retrieval for Amharic language. The Inverted file indexing structure is used to organize documents so as to speed up searching. The term association rule attempts to simulate the concept relation to guide searching for relevant document that satisfy user query.

4.1 Sample Text

For this work, sample texts were prepared from different sources: health news from (web site of Ministry of Health news articles, the web site of Walta Information Center Health news articles, Hawass Online, Ethiopian Broadcast Corporation, and Admass news). And sport news articles from the web site of (Walta Information Center, Ethiopian Broadcast Corporation, Ethiopian news agency, Fana Broadcast Corporation, Ethiopian news agency and Reporter news). These documents are selected because they are available publicly on the internet. Summary of the document corpus collected from different sources is presented in table 4.1.

Source	Ministry of Health	Walta	Hawass Online	Ethiopian Broadcast Corporation	Admass news
Domain Title	News	□□(Health articles)	Health and Fitness	ጤናናውበት(Health news articles)	ጤና(Health articles)
Document no.	150	36	10	18	5
Source	Ethiopian news agency	Walta	Reporter	EBC	Fana
Domain Title	Sport News Articles	Sport News Articles	Sport News Articles	Sport News Articles	Sport News Articles
Document no.	310	23	82	30	111

Table 4.1Statistics of the testing dataset news article used.

4.2 Tokenization

In the corpus, we encounter elements that do not carry information and increase the processing time. This is mostly special characters, numbers, non- Amharic words, abbreviations and single letters. These should be deleted:

• Special characters: include any special sequence of characters delimited by letters or spaces.
• Numbers: We regroup all the character sequences located between two spaces containing numbers in a single occurrence. This method also has the advantage to combine the dates, the actual numbers and percentages.
• Words in Latin characters: The non-Amharic words, mainly in Latin characters are simply detected by their *graphic*.
The Java code used to select only Amharic characters:

```
File filename= new File(filename);
Scanner words = new Scanner (filename);
Words.useDelimiter ("[^ሀ-ፐ]+");
```

4.3. Stop Word Removal

Not all terms found in the document are equally important to represent documents they exist in. Some terms are common in most documents. Therefore, removing those terms, which are not used to identify some portions of the document collection, is important. Such terms are removed based on two methods. For example, the adverb ስለ‘sle’ may exist as ’ስለሚኖራቸው’,’ስለሚኖራቸውም’,’ስለሚኖር’,’ስለሚኖርባቸው’,’ስለሚኖርብን’,’ስለሚወስደው’,’ስለሚያስፈልገው’, ’ስለሚያስፈልጋቸው’,’ስለሚያበረታታ’ or ‘ስለሚገኙት’which makes identification problematic. We have made this list of stop words from the corpus on two principles: their frequency and their information content. We have sorted the most used words in the corpus according to their

frequency, and then we manually selected among them the words that do not have information related to the domain. Figure 4.1 presents java code that removed stop word. In total we used 129list of stop words.

```
Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("SELECT termdistrbtion.fn,termdistrbtion.word,termdistrbtion.tf FROM termdistrbtion,termfriquens where "
        + "termfriquens.word=termdistrbtion.word and termfriquens.wf>=5 and "
        + "CHAR_LENGTH(termfriquens.word) > 1 and NOT EXISTS (SELECT * FROM stopword where termfriquens.word=stopword.word )");
int n = 0;
while (rs.next()) {
            System.out.println(rs.getString(1) + "  " + rs.getInt(3));
            no = rs.getInt(3);
            String x = rs.getString(2);
            org = x;
            fn = rs.getNString(1);
            n++;

            String sql = "insert into selecte_5 (fn,word,tf)VALUES('" + fn + "','" + org + "'," + no + ")  on duplicate key update tf= VALUES(tf)+ " + no;
            Statement stmt1 = con.createStatement();
            stmt1.executeUpdate(sql);
}
```

Figure 4.1 Java code that removed stop word.

4.4 Normalization

The development of text technologies for Amharic languages often faces a resource bottleneck, in that the text that does exist is written by transcribers of very different skill levels, using a variety of orthographic conventions. This results in very heterogeneous text corpora, and normalization into an orthographically homogeneous form is made challenging by this very lack. Many communities have collections of texts in heterogeneous orthographies, and writers have often been trained in different orthographies (and trained to varying degrees), so the possibility of normalizing texts (both old and new) to a consistent format can solve many practical

problems. Result: from23017 words are selected into 18274 Normalization words are common sense (21%).

The Amharic writing system contains characters or symbols, which have the same sound during reading. For example, the characters "ሀ, ሐ, ኀ, ኣ and ዐ" produces the same sound during reading with characters "ሃ, ሓ, ኃ, ኣ and ዓ" respectively. On the other hand, there are also alphabets that share the same sound, such as, "ሀ,ሐ, and ኀ", "ሰ and ሠ", "ኧ and ዓ", and "ጸ and ፀ". As a result, words which have the same meaning may have a different spelling structure. In this research, such kind of problems is solved by changing the characters into their common standard form.

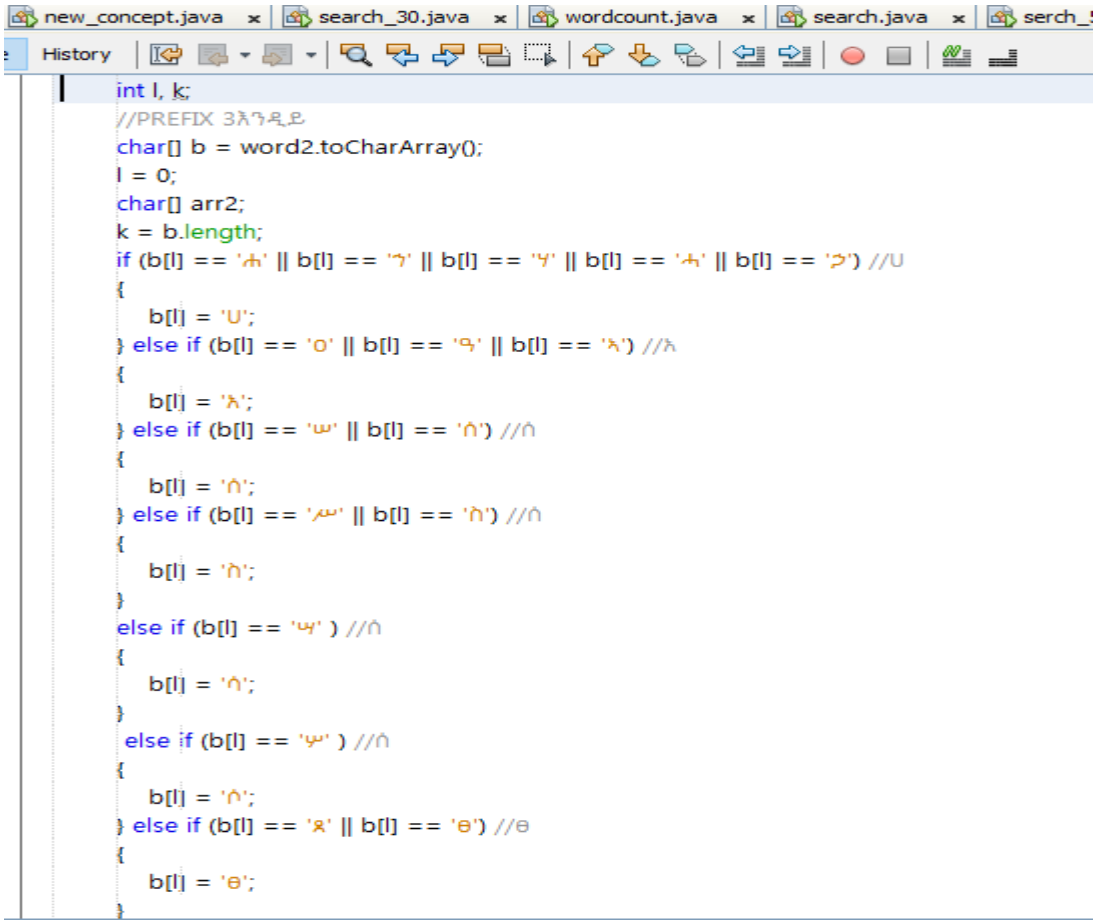

Figure . 4.2 Java code that removed stop word

4.5Stemming

Stemming is the process of reducing related words to their stem, base or root form through affix removal. Its aim is to adapt different derivational or inflectional variants of the same word to a single indexing form [**67**].

The steaming process is done whenever a word has more than two radicals, or length of a Word is greater than three depending upon the length of the affixes. The minimum numbers of radical to Amharic words are two, and the most common words contain three radicals and above [**66**]. The stemmed word also checked with stop word list and if it is part of stop words, it will be removed from the sample text.

In Amharic words are morphologically rich. Since morphological variant words have similar semantic interpretations, there is a need to stem words to their root. In this research, the stemming implementation has two modules, the prefix removal module and the suffix removal module. Figure 4.2 depicts java code implementation of prefix and suffix removal of Amharic words.

The algorithm is adopted from Amanuel [**16**]the Problem faced implementing stemming on Amharic word is sometimes over-stem words. This result creates new word that is not known (available) in Amharic language. For instance, "**ማግባት**" meaning to marry are over stemmed to "**ማግ**"; as a result of which both words are indexed as the same word located in different document.

Such misrepresentation of different words as similar affects the retrieval process. To overcome such problem we research check stemmed words for their availability on the corps dictionary. The corps as dictionary helps to find out whiter the stemmed word is an Amharic word or not.

The research uses corps as Amharic dictionary because it contains all words used to develop the system. Figure 4.2 shows java code for stemming Amharic word.

```
ResultSet rs = stmt.executeQuery("SELECT * FROM termdistrbtion");
int n = 0;
while (rs.next()) {
    System.out.println(rs.getString(2) + "  " + rs.getInt(3));
    no = rs.getInt(3);
    String term = rs.getString(2);
    String fn = rs.getString(1);
    org = term;     //orginal word
    prfix = s.step3(org);
    safix = s.stim(prfix);
    String check = "select word term where= word='" + safix + "'";;
    ResultSet check_term = stmt.executeQuery(check);
    if (!check_term.first()) {
        safix = org;
    }
```

Figure 4.3 java code for Amharic stemming

The code takes index terms and check if the length is greater that two or not. If the length of the word is less than or equal two, the word is returned without further processing and check for stop word. If the length of the word is greater than two, the code iterates on word and check characters if they matched with one of the prefix/ found in prefix list. If the character is matched, processing continues for checking context sensitivity of the word and finally the stemmed word is returned. After the prefix is removed from index terms, the next step is removing suffix. When removing suffixes, a similar procedure is followed like a prefix. The stemmer conflated derivational and inflectional affixes. It is not conflate irregular and compound words. These words are very complex and follow different (non-constant structure) patterns. Commonly used methods of stemmers such as using stop word lists and context sensitive rules are employed.

4.6Automatic extraction of terms & association rule discovery

After preparing the corpus, we move to the term extraction step. The processing is done in two steps. First; we extract all the terms (one or more words) used to denote concepts in the domain, using the method of repeated segments frequent item set: A significant term is used several times in a corpus.

Using TFIDF weighting scheme with cooperation to document frequency for term selection, we scored TFIDF for each word (unigram and bigram) in the given text corpus. Then we found out the top unigram and bigram according to the TF-IDF weights. In addition, TF-IDF we used to score the document frequency for each word (unigram and bigram). In this way, we extracted top keywords for the given text.

This step term selection result returned381unigram and 145 bigram total of 526terms selected using repeated support of 16-32 document frequency with TF-IDF value greater than 0.5.

In the second passage; we will seek the pairs of terms that co-occur more frequently in the corpus. The result of this processing provides us with a list of pairs of terms that will be used to discover the association rule that used in query expansion. Therefore, the objective of the first pass is to identify the terms that denote the concepts related to the domain, however the second pass is to identify among these terms, couples who have links with elements of the selected

terms.

```
        while (rss.next()) {
            term = rss.getString(1);
            int sport = rss.getInt(2);
            double confidence = 0;
            confidence = (double) sport / (double) dc;
            if (confidence >= 0.3 && !term.matches(org)) {
                top[k] = term;
                k++;
                System.out.println(k + "  ----------" + term + "----------" + confidence);

            }

        }
        Statement stmt2 = con1.createStatement();
        String sql3 = "insert into tfidf.`concept_30%` (word,t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14)VALUES('" + org + "','" + top[0] + "','" + top[1] + "','" + top[2] +
        stmt2.executeUpdate(sql3);
        top = null;
        con1.close();
    }

    con.close();

} catch (Exception eee) {
    System.out.println(eee);
}
```

Figure 4.4 java code for association rule discovery

The code takes keyword terms and check for document of that term. And then extract the document id. Using the document id we can get terms that paired on the selected documents. After all, we keep track term occurrence with then selected document so as it helps to analysis association tender.

4.7Use of word association in query expansion

Discovering association rules with antecedents contained in the corpus can be accomplished by generating all rules with support and confidence above predefined thresholds first and then selecting rules with user query itemset. For a given query first appropriated terms with strong association are retrieved from the association rules. Then the set of concepts associated with each document is retrieved from database. As next, these two sets (user query and association rule) are

compared using simple metric, which expresses the similarity between a document D_i and given query Q.

```
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM `concept_40%` where word='" + s + "'");
int n = 0;
while (rs.next()) {
   System.out.println( rs.getString(1) + " ----------" + rs.getString(2) + "  " + rs.getString(3) + "  " + rs.getString(4) + "  " + rs.getString(5) + "  " + rs.getString(
   for (int r = 1; r < 10; r++) {
      s = rs.getString(r);
      if (words_fre.containsKey(s)) {
         int a = words_fre.get(s);
         words_fre.put(s, a + 1);
      } else {
         words_fre.put(s, 1);
         uwords++; // unique words count
      }
   }
}
con.close();
```

Figure 4.4 java code for association rule based information retrieval system

For instance in user query "የኢትዮጵያፕሪሚዬርሊግየውድድርመርሐግብር": with query terms "የኢትዮጵያፕሪሚዬር" and "ፕሪሚዬርሊግ" for this query terms the system retrieved appropriated terms with strong association from the association. For "ፕሪሚዬር"→(ፕሪሚዬርሊግ,ፋሲል, ቅዱስጊዮርጊስ , ደደቢትመከላከያ)and " ፕሪሚዬርሊግ"→(ፕሪሚዬር, ቅዱስጊዮርጊስ, ሊጉን, መከላከያደደቢትፋሲል). Then system finds the intersection between query terms and the association terms once we find the intersection based it used for expanding query rank search result on the top with high intersection value. The terms retrieved may not be relevant, but they have strong association.

4.8 Evaluation

In this section we describe the evaluation of document retrieval experiments. Precision, Recall and F-measure are the most frequent and basic statistical measures which are widely used measures to assess the effectiveness of IR system (i.e., the quality of the search results) [43]. These three parameters are used in this research so as to measure the effectiveness of the designed Association rule based IR system.

Research on Amharic IR system has been used ten test query [**16**] [**15**] and [**13**]. Previous researches such as Tewodros [**11**], Abay [**13**] and Samrawit [**17**] used ten test queries with a word size of each query 2-4. We use ten test queries and the test queries a word size used are 2-7 like previous researcher of IR systems. The most common and basic statistical measures recall, precision and F-measure was used [15] [64] and [13]. Precision is the fraction of the documents retrieved that relevant to the user's information need, and recall is the fraction of the documents that are relevant to the query that is successfully retrieved. F-measure is the weighted harmonic mean of precision and recall. Using these techniques, effectiveness of the retrieval system is measured and compared to previous works.

The test queries were formulated by experts for testing the effectiveness of IR system. Relevance judgment was prepared by language experts and sport fun to construct document query matrix that shows all relevant documents for each test query. An evaluation was done by measuring the precision and recall of each test query and the average of each query result is obtained as the initial performance of the system.

As shown in the tables bellow keyword terms selected using term weighting techniques that have high document frequency and TFIDF. With the support of high TFIDF value we generate the association rule. Table 4.2 presents the sum of candidate terms association Rules generated in percentage that have been done in different experiments. Depending on the experimental result, we can't find more association when we use confidence greater than 60%. Hence, we have to go for low confidence to get big enough association rules for our query expansion. Using 50%

confidence in the rule mining process and the system generates association rule for 241 keywords. This one shows it cannot generate association rule for 50% candidate terms. It has only generated for 45.8% candidate terms which is below 50%. Using 40% confidence to the rule mining process system generates association rule for 364 keywords. This shows it can generate association rule for 69% candidate terms. When we use 30% confidence and the system generates association rule for 484 keywords. This shows it can generate association rule for 92% candidate terms.

Number of candidate	526 candidate terms selected with TFIDF greater than 0.5		
confidence used in rule mining process	50% candidate	40% candidate	30% candidate
number of candidate terms association Rules generated	241	364	484
candidate terms association Rules generated in percentage	45.8%	69%	92%

Table 4.2Association Rules generated

Evaluation task

The purpose of the evaluation task is to estimate how useful rules retrieved. For our evaluation we assume the incident analysis. Given some information about an incident, the system retrieves association rules which may contribute to the explanation of the underlying cause or point to some critical information lacking in the initially available information. Domain expert determine to what extent the retrieved rules are useful. Although we have manual evaluation in our plans, we used automatic procedure that evaluates the antecedent terms of the retrieved rules.

For evaluation, the dataset is divided into three parts: training set with 70% of the documents and testing set 30%. Our automatic evaluation task is to restore terms that were artificially removed from a documents in the testing set. Terms are restored by retrieving them from documents in the training set based on association with the remaining terms. The evaluation scores are based on the association between the restored and the removed terms.

confidence used in rule mining process	50% candidate	40% candidate	30% candidate
number of candidate terms association Rules generated	241	364	484
candidate terms association Rules generated in percentage	45.8%	69%	92%
Domain expert useful %	87%	75%	60%
automatic evaluation	85%	77%	53%

Table 4.3Association Rules evaluation

Systems Evaluation

The first query expansion experiment is done using 50% confidence generated association rule. The system scoures6.3% recall, 84%F-measure and81.8%precision.This increase the recall in 15% and 7%F-measure, but decrease precisionby2.2% when we compared to the result before expansion. Our second experiment is done with 40% confidence generated association rule. The system scours88.2% recall, 84.5% F-measure and81.2%precision. This shows improvement of recall 17% and, 7.5%F-measure, but decrease precision 2.8% when compared to result before query expansion. The third query expansion experiment use 30% of confidence generated association rule. The system scour 95.3% recalls, 68%F-measure and 53% precision which shows a significant improvement to the recall by 24%, but badly affect by 9%F-measure and decrease precision 31 %compared to result before query expansion. Table 4.2 shows evaluation is done by measuring the recall and precision respectively for each test query and the average of each query result on recall-Precision**.** Appendix III Test query, relevant document top 10 ranked lists result for each query.

		Before				association rule with 50% confidence				association rule with 40% confidence				association rule with 30% confidence			
	Relevant (Judgment)	Retrieved	Relevant	Recall	Precision	Retrieved	Relevant - Retrieved	Recall	Precision	Retrieved	Relevant - Retrieved	Recall	Precision	Retrieved	Relevant	Recall	Precision
የኢትዮጵያ ፕሪሚዬር ሊግ ኢትዮጵያ ቡና ከ ቅዱስ ጊዮርጊስ	32	28	23	0.71	0.82	39	32	1	0.821	39	32	1	0.82	67	32	1	0.48
ዋልያዎቹ የአፍሪካ ዋንጫ ማጣሪያ	37	39	31	0.84	0.79	39	31	0.84	0.795	39	31	0.8	0.79	39	34	0.92	0.87
አበረታች መድሃኒት	21	17	16	0.76	0.94	40	21	1	0.525	31	21	1	0.68	91	21	1	0.23
የስፖርታዊ ጨዋነት ጉድለት	33	25	23	0.69	0.92	41	33	1	0.805	41	33	1	0.8	80	33	1	0.41
የኢትዮጵያ ፕሪሚዬር ሊግ የውድድር መርሐግብር	29	37	19	0.65	0.51	29	20	0.69	0.69	34	26	0.9	0.76	75	29	1	0.39
ኢትዮጵያ በማራቶን	47	27	27	0.57	1	27	27	0.57	1	27	27	0.6	1	50	40	0.85	0.8
የዳይመንድ ሊግ ውድድር	22	22	22	1	1	33	22	1	0.667	35	22	1	0.63	53	22	1	0.42
ዓመታዊ የስፖርት ፌስቲቫል	21	17	17	0.8	1	17	17	0.81	1	17	17	0.8	1	31	17	0.81	0.55
ጥሩነሽ ዲባባ ና እህቱ	49	32	22	0.44	0.69	45	45	0.92	1	57	45	0.9	0.79	64	49	1	0.77
አትሌቲክስ ፌዴሬሽን	32	31	31	0.97	1	31	31	0.97	1	31	31	1	1	31	31	0.97	1
Average	323	275	231	0.72	0.84	341	279	0.86	0.818	351	285	0.88	0.81	581	308	0.95	0.53

Table 4.4 the initial performance evaluation of the system for each query

Table 4.5 shows evaluation done by measuring the recall, precision and F-measure respectively on average for each experiment average result.

		Relevant (judgment)	Retrieved	Relevant-retrieved	Recall	Precision	F- measure
Before association rule apply		322	275	231	0.7173913	0.84	0.773869
After association rule apply	50% Confidence	323	341	279	0.8637771	0.818182	0.840361
	40% Confidence	323	351	285	0.8823529	0.811966	0.845697
	30% Confidence	323	581	308	0.9535604	0.53012	0.681416

Table 4.5Recall, Precision and F-measure average 2

F-measure curve for all of the approaches described above are presented in Figure 4.5 Our experiments showed that the F- measure scored better with query expanded form association rule that generated with 40% Confidence. The approach based on an association rule whenever recall increases precision decrease. Even if the result is very promising, Precision was affected negatively retrieval efficiency than the result before we use the association rule approach. This gives because of the strong association give less non- relevant document in strong association precision. At weak strength of association the recall increase and at the same time non-relevant document retrieved more than strong association rule. Fig 4.5 shows before & after association rule applied.

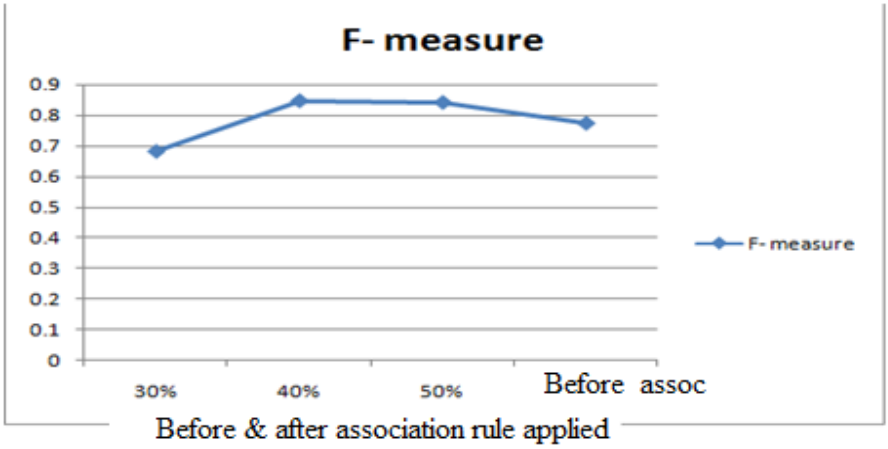

Figure 4.5 F-measure curve analyzed retrieval approaches in different confidence.

4.9 Finding

The experiment result shows, the average precision and recall of 88.23% and 81.2% respectively. As the experimental result show, association rule based Amharic IR system register a good performance and score on the average 84.5% F-measure. The System evaluation shows a promising result to design an applicable IR system. The optimum value of Recall was obtained40% confidence with good precision.

However, more experiments on various domain and large corps size are required before these results can be generalized. Moreover, the goodness metric can be improved by considering a higher number of keywords, at the cost of increase in execution time.

After conducting different experiments, we observed that Recall was maximized when we use low confidence association rule. But, Precision decrease at low confidence because of the weak association strength leads to irrelevant information retrieval.

The stemmer conflated derivational and inflectional affixes. Commonly used methods of stemmers such as using stop word lists and context sensitive rules are employed. But it was not an excellent stemmer the preprocess task was facing such problem. Stemmer conflated derivational and inflectional affixes over stems word variants greatly affecting the performance of the system

Finding a large size and standard corpus for Amharic language is considered as one of the challenge faced in this research. The state of the art in the area of text processing indicates that, there is no any developed standard corpus for Amharic language. Thus, in this research the researcher uses small size corpus extracted from internet. To generate association rule the system need big corpus size. We have challenge finding association between terms with generic news articles. The news are unique and always come with new agenda.

Hence we can't find large data set for the experiment the test data set was single sport domain. The problem large size and standard corpus for Amharic language was only affect performance of the system. It also makes difficult to compare the result obtained with several researches since there is different in test queries, document content and size used for testing. Furthermore, from the above analysis shows sampling to work well for larger databases.

The terms retrieved may not be relevant, but they have strong association. This was the main challenge for word association based retrieval system. Irrelevant terms to users query, but because they have strong association system used in query expansion and this leads to retrieve irrelevant documents. This affects the performance of the information retrieval system. For example, the Amharic query "ጥሩነሽ ዲባባ ና እህቷ" the system retrieved terms ገንዘቤዲባባ and ታምራት that have strong association with key word term "ጥሩነሽ ዲባባ". The term ታምራት have a strong association with term "ጥሩነሽ ዲባባ" but it's not related to the user query.

Another thing we observe on the experiment is a same term with the same meaning used in different contexts which give different meaning synonyms and polysemy. For example the term "ዞን" which means "zone" it's used for zonal sport computation Ethiopian and "□□□□□ □□" which means premier league it's used may for Ethiopian premier league or England premier league. Such terms are contextual defendant location or users profile based information. Another example for polysemy word is a term "ለጋ" which means "ታዳጊ ወጣት" or it may used for completely different context "ኩስለጋ". Such issues need to be handled to satisfy the user information need.

CHAPTER FIVE

Conclusion& Recommendation

5.1 Conclusion

The association-based retrieval approach described in this paper opens new perspectives for effective use of knowledge contained in domain-specific documents. Instead of searching for information similar to a user query it allows retrieval of different, but somehow related pieces of information. We see the potential use of this method for such tasks as medical diagnosis, analysis of transportation and production accidents, mining of scientific papers and analysis of legal cases. Therefore, there is always a need towards the development of a specialized system to generate association rule automatically.

This research has been made an attempt to discover word association automatically for improving the performance of the Amharic information retrieval system. Design and develop an association rule based information retrieval system for Amharic language.

First text operation pre-process, including (i.e., tokenization, stop word elimination stemming, and normalization) are done. After preparing the corpus, and arrange them in term weighting, we select terms (keywords extractions). To detect relations among keywords, we used a set of text-mining techniques and generate the association rule automatically.

When the search component initiated the system generates terms ranked list for each query term relation (to counteract query concept). For a given query first appropriated terms with strong association are retrieved. Then the set of concepts associated with each document is retrieved from database. As next, these two sets are compared using simple metric, which expresses the

similarity between a document D_i and given query Q. According the system generates ranked list of relevant documents.

At the end, we have presented results of the experiments done in order to evaluate retrieval effectiveness of an association rule -based approach. System evaluation has been done to discover the extent to which the designed system enhances the performance of the Amharic IR system. Experiment result shows information retrieval register 88.23% Recall, 81.19%precision and 84.5% F-measure. This shows improvement of recall 17%, 7.5%F-measure and affected negatively precision by 2.8% when compared to result before query expansion. Association rule based information retrieval system gives a promising result in recall, precision and F-measure.

The researcher has challenged finding corpus. A very small sample size may generate many false rules, and thus degrade the performance. With that caution, we claim that for practical purposes we can use a single domain with minimum confidence for data mining.

The terms used in the discovery, associations are unigrams and bigram but the terms can be complexes, which are composed of several words used individually. Complex terms are constructed using a finite number of sequences of words.

The terms retrieved may not be relevant, but they have strong associations. Irrelevant terms to the user's query, but because they have a strong association system used in query expansion and this leads to retrieve irrelevant documents.

5.2 Recommendation

Future work should be focused on further enhancement of association-based retrieval mechanism using the more sophisticated inference mechanism for finding similar concepts to the given query using WordNet.

Location based information retrieval is an additional feature needs work. To solve the challenges we mansion in experiment, part with terms like" □□□□□□□". Use like Google maps for this purpose by means of which the system gets enhanced with this location dependent feature.

One of the problems in enhancing the performance of the Amharic IR system is the existence of synonyms and polysemy terms in Amharic text. To solve challenge we mansion in experiment part with terms like "ለጋ".We recommended integrating mechanisms of controlling synonym and polysemy terms in the association-based retrieval model to enhance precision.

Finding a standard corpus and test queries with relevance judgment for testing the designed system is one of the challenges faced in this research. Therefore, future research need to consider the development of a standard Amharic corpus that can be used by researcher to evaluate progress made in designing Amharic IR system.

An extended way of indexing the documents semantically using association rules would improve the search level. Once the documents are crawled, it must be pre-processed for indexing. A new model to store the index semantically by knowing the association between the documents could enhance the search results.

References

[1] LANDAUER, T. K., GOMEZ, L. M., AND DUMAIS FURNAS G. W., "The vocabulary problem in human-system communication," *Comm. ACM*, pp. 964–971, November 1987.

[2] S. Ceri, "Web Information Retrieval, Data-Centric Systems and Applications," *Springer-Verlag Berlin Heidelberg*, 2013.

[3] J. AND C ROFT, W. B XU, "Improving the effectiveness of information retrieval with local context," *ACM Trans. Inf. Syst*, pp. 79–112, 2000.

[4] M., SINGHAL, A., AND B UCKLEY MITRA, "Improving automatic query expansion," *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* , pp. 206–214, 1998.

[5] E. M VOORHEES, "Using wordnet to disambiguate word senses for text retrieval," *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 171-180, 1993.

[6] H. AND PEDERSEN, J. O SCHUETZE, "Information retrieval based on word senses.," *Annual Symposium on Document Analysis and Information Retrieval*, pp. 161-175, 1995.

[7] H. B. STYLTSVIG, "Ontology-based information retrieval," *Ph.D. thesis, Department of Computer Science, Roskilde University, Denmark.*, 2006.

[8] E. AND MARKOVITCH GABRILOVICH, "Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. ," *21st National Conference on Artificial Intelligence (AAAI'06). AAAI Press*, pp. 1301–1306., 2006.

[9] GABRILOVICH, E., AND MARKOVITCH EGOZI O., "Concept-Based feature generation and selection for information retrieval," *23rd AAAI Conference on Artificial Intelligence*, pp. 1132-1137, 2008.

[10] Ethiopia: Central Statistical Authority(ECSA), "The population and HousingCensus of Ethiopia:Results at Country Level," *Ethiopia: Central Statistical*, 2008.

[11] Tewodros H., "Amharic text retrieval: An Experiment Using Latent Semantic Indexing," *Addis Ababa University, Addis Ababa*, no. MSc Thesis, 2003.

[12] Hassen Redwan, Solomon Atnafu Tessema Mindaye, "Design and Implementation of Amharic Search Engine," *MSc Thesis, Addis Ababa University, Addis Ababa*, 2007.

[13] Abey Bruck, "Semantic Based Query Expansion Technique for Amharic IR," *Addis Ababa University, Ethiopia*, no. MSc Thesis, 2011.

[14] Iman Abrahim, "word sense disambiguation for amharic IR," *MSc Thesis, ADDIS ABABA UNIVERSITY*, 2013.

[15] Welde Janfa, "ontology based query expansion for enhancing the performance of Amharic information retrival," *Addis Ababa University, Addis Ababa*, 2014.

[16] AMANUEL H., "PROBABILISTIC INFORMATION RETRIEVAL SYSTEM," *ADDIS ABABA UNIVERSITY, Addis Ababa*, no. MSc thesis, 2012.

[17] Samrawit Zewdneh, "Word Sense Disambiguation Using Semantic Similarity in Amharic Information Retrieval," *Addis Ababa University,Addis Ababa*, no. MSc Thesis, School of Information Science, 2014.

[18] Tuure Tuunanen, Marcus A. Rothenberger and Samir Chatterjee Ken Peffers, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45-78, 2007.

[19] Baeza Yates and Ribeiro Neto, *Modern Information Retrieval*, 2nd ed. New York: Addison-WesleyLongman Publishers, 1999.

[20] Amit Singhal, "Modern Information Retrieval: A Brief Overview," *Google, Inc.*, 2001.

[21] Prabhakar Raghavan and Hinrich Schütze Christopher D. Manning, *An Introduction to Information Retrieval*. Cambridge , England: Cambridge University Press, 2009.

[22] S. E. Robertson, "The probabilistic ranking principle in IR," *Journal of Documentation*, pp. 294–304, 1977.

[23] M. E. Maron and J. L. Kuhns, "On relevance, probabilistic indexing and information

retrieval," *Journal of the ACM*, pp. 216–244, 1960.

[24] Laurence A. F. Park and Kotagiri Ramamohanarao, "Efficient storage and retrieval of probabilistic latent semantic information for information retrieval," *The VLDB Journal - The International Journal on Very Large Data Bases archive*, vol. 18-1, pp. 141 -155, 2009.

[25] Nadeem Iftikhar and Muhammad Abdul Qadir Humaira Liaqaut, "Context Aware Information Retrieval using Role Ontology and Query Schemas," *10th IEEE International Multitopic Conference*, pp. 244-249, 2006.

[26] Jing-Yan Wang and Zhen Zhu, "Framework Of Multi-Agent Information Retrieval System Based On Ontology," *International Conference on Machine Learning and Cybernetics*, pp. 1615-1620, 2008.

[27] Tomi Kauppinen, Kim Viljanen, and Eero Hyv¨onen Jouni Tuominen, "Ontology-Based Query Expansion Widget for Information Retrieval," *Workshop on Scripting and Development for the Semantic Web*, 2009.

[28] Jinghua Zhao, Qiuju Yin and Jingxia Wang Huiying Gao, "Ontology-based Enterprise Information Retrieval Model," *IEEE International Conference on Grey Systems and Intelligent Services*, pp. 1326-1330, 2009.

[29] Wenping Guo and Xiaoming Zhao Ying Chen, "A semantic Based Information Retrieval Model for Blog," *International Symposium on Electronic Commerce and Security*, pp. 257-260, 2010.

[30] Jie Gong and Fangfang Liu Jie Yu, "Building Search Context with Sliding Window for Information Seeking," *International Conference on Computer Research and Development*, pp. 274-277, 2011.

[31] Huiyong Xiao and Isabel F. Cruz, "A Multi-Ontology Approach for Personal Information Management," *1st Workshop on Semantic Desktop*, pp. 19-33, 2005.

[32] Xiang Zhang, Jianming Deng and Yuzhong Qu Jian Guan, "An Ontology-Driven Information Retrieval Mechanism for Semantic Information Portals," *the First International*

Conference on Semantics,Knowledge, and Grid, pp. 63-66, 2005.

[33] Fernando Ortiz-Rodríuez and Boris Villazó -Terrazas Asunció Góez -Péez, "Ontology-Based Legal Information Retrieval to Improve the Information Access in e-Government," *international conference on World Wide Web*, pp. 1007-1008, 2006.

[34] Chih-Hung Hu and Ming-Yi Ju Been-Chian Chien, "Intelligent Information Retrieval Applying Automatic Constructed Fuzzy Ontology," *the Sixth International Conference on Machine Learning and Cybernetics*, pp. 2239-2244, 2007.

[35] Zheng GU and Song-Nian Yu, "Ontology-Based Inverted Tables in Information Retrieval System," *International Conference on Semantics, Knowledge and Grid*, pp. 354-357, 2007.

[36] Rafael Berlanga-Llavori and Dietrich Rebholz-Schuhmann Antonio Jimeno-Yepes, "Ontology refinement for improved information retrieval," *Semantic Annotations in Information Retrieval*, vol. 46-4, pp. 426–435, 2010.

[37] Pradip K. Srimani and James Z. Wang Liang Dong, "Ontology Graph based Query Expansion for Biomedical Information Retrieval," *International Conference on Bioinformatics and Biomedicine*, pp. 488-493, 2011.

[38] Srabani Sarkar, "Fuzzy Information Retrieval based on Ontology generated by using concept of Fuzzy-valued Variable," *International Conference of Soft Computing and Pattern Recognition*, pp. 27-32, 2011.

[39] GUO Rong and WU Jun, "Design and Implementation of Domain Ontology-based Oilfield Non-metallic Pipe Information Retrieval System," pp. 813-816, 2012.

[40] Jinwoo Kim and Dennis McLeod, "A 3-Tuple Information Retrieval Query Interface with Ontology Based Ranking," *IEEE 13th International Conference on Information Reuse and Integration*, pp. 490-497, 2012.

[41] Bing-wu Liu and Jun Liu Hong Zhoua, "Research on Mechanism of the Information Retrieval Based on Ontology," *International Workshop on Information and Electronics Engineering*, vol. 29, pp. 4259–4266, 2012.

[42] Rifat Ozcan and Y. Alp Aslandogan, "Concept Based Information Access Using Ontologies and Latent Semantic Analysis," *University of Texas, Arlington, Tech*, 2008.

[43] Gábor Nagypál, "Improving information retrieval effectiveness by using domain knowledge stored in ontologies," *Springer Berlin Heidelberg, On the Move to Meaningful Internet Systems*, pp. 780-789, 2005.

[44] WU Jiang-ning and DANG Yan-zhong MA Hui-nan, "Ontology-Driven Information Retrieval System for Customers' Complaints of Mobile Communication Services," *Management Science and Engineering*, pp. 30-35, 2006.

[45] Sheng Qiuyan and Ying Guisheng, "Measuring Semantic Similarity in Ontology and Its Application in Information Retrieval," *International Conference of Soft Computing and Pattern Recognition*, pp. 27-32, 2011.

[46] Il-Yeol Song, Xiaohua Hu and Robert Allen Min Song, "Semantic Query Expansion Combining Association," *Data Warehousing and Knowledge Discovery Lecture Notes in Computer Science,* , pp. 326-335, 2005.

[47] LI Zhi-lu and GUAN Chun HU Jun, "A Method of Rough Ontology-based Information Retrieval," *IEEE International Conference on Granular Computing*, pp. 296-299, 2008.

[48] Miao Chen, "Exploring the Use of Ontological Relations in Information Retrieval," *Posters presented at the iConference*, 2009.

[49] Wei Wang and Jayan C Kurian Payam Barnaghi, "Semantic Association Analysis in Ontology-based Information Retrieval," *Handbook of Research on Digital Libraries: Design, Development and Impact, Chapter XIII*, pp. 131 -141, 2009.

[50] Salvador-Sáchez, Elena García-Barriocanal and Manuel Prieto Alejandra Segura N., "An empirical analysis of Ontology-based query expansion for learning resource searches using MERLOT and the Gene Ontology," *Knowledge-Based Systems*, vol. 24-1, pp. 119–133, 2011.

[51] Chunchen Liu and Jianqiang Li, "Semantic-based Composite Document Ranking," *Sixth*

International Conference on Semantic Computing, pp. 126-129, 2012.

[52] T. Imielinski, and A. Swami R. Agrawal, "Mining association rules between sets of items in large databases," *the ACM SIGMOD international conference on Management of data*, pp. 207–216, 1993.

[53] Do D B and Lin X Wang W, "Term Graph Model for Text Classification," *In: Proc. Lecture Note in Artificial Intelligence,Springer*, pp. 19–30, 2005.

[54] Han J and Kamber M, *Data mining concepts and techniques*, Second edition ed. San Francisco: Morgan Kaufmann publishers, Elsevier, 2006.

[55] Agrawal R and Srikant R, "Fast algorithm for mining association rules," *20th Intl. Conf. VLDB*, pp. 487–499, 1994.

[56] Ashish Kankaria, "Query Expansion techniques," *Indian Institute of Technology Bombay, Mumbai*, 2013.

[57] Ed Greengrass, "Information Retrieval : A Survey by Ed Greengrass," 2009. [Online]. www.csee.umbc.edu/cadip/readings/IR.report.120600.book.pdf

[58] Jinxi Xu and W. Bruce Croft, "Query Expansion Using Lo cal and Global Document Analysis," *Cen ter for In telligen t Information Retrieval University of Massachusetts, USA*, 1997.

[59] W. P. R. Luk, K. S. E. Ho, and F. L. K. Chung Y. Li, "Improving weak ad-hoc queries using wikipedia," *annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 797–798, 2007.

[60] S. Ganesh and V. Verma, "Exploiting structure and content of wikipedia for query expansion," *International Conference RANLP*, pp. 103–106, 2009.

[61] Neelamadhav Gantayat, "Automated Construction Of Domain Ontologies From Lecture Notes," *Indian Institute of Technology, Bombay*, June 2011.

[62] Wang W., & Kurian J. C. Barnaghi P., "Semantic association analysis in ontology-based information retrieval," 2009.

[63] Meet Hummingbird. (2016, Oct.) Google Just Revamped Search. [Online]. http://www.forbes.com/sites/roberthof/2013/09/26/googlejust-revamped-search-to-handle-your-long-questions

[64] (2016, Apr) Bing Gets More Personal With Adaptive Search. [Online]. http://searchengineland.com/bing-gets-more-personal-withadaptive-search-92858

[65] Berhanu Abebe, "Modeling Pervasive Context-aware Mobile Phone," *MSc Thesis, ADDIS ABABA UNIVERSITY*, 2015.

[66] Alemayehu and Willett, "Stemming of Amharic words for information retrieval," *Literary and Linguistic computing*, pp. 1-17, 2002.

[67] Alemu Aregaw and Lars Asker, "Amharic Stemmer: reducing words to their citation forms," *Association for computational linguistic proceeding of the 5th work shop on important,unresolved matters, Swiden, Czech Republic*, 2007.

[68] Juan Ramos, "Using tf-idf to determine word relevance in document queries," *First International Conference on. Machine Learning*, 2003.

[69] E. M. VOORHEES, "Query expansion using lexical-semantic relations," *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 61-69, 1994.

[70] ANDARGACHEW MEKONNEN, "AUTOMATIC THESAURUS CONSTRUCTION FOR," *Msc THESIS ADDIS ABABA UNIVERSITY*, no. ADDIS ABABA, 2009.

[71] ANDARGACHEW MEKONNEN, "AUTOMATIC THESAURUS CONSTRUCTION FOR," *Msc THESIS ADDIS ABABA UNIVERSITY*, no. ADDIS ABABA, 2009.

[72] N and Willett Alemayehu, "Stemming of Amharic words for information retrieval," *Literary and Linguistic computing*, pp. 1-17, 2002.

[73] M Tessema, "Design and Implementation of Amharic Search Engine," *MSc Thesis, Addis Ababa University, Addis Ababa*, 2007.

Appendix I: stop word

'ሁለ'ሁለም"ሁለተ"ሁለተኛ'ሁለተኛዉ"ሁለተኛዋ"ሁለተኛዋን"ሁለተኛው"ሁለተኝ"ሁለተኝነት"ሁለቱ"ሁለቱም"ሁለቱን"ሁለቱንም"ሁለት"ሁለትና"ሁለንተና"ሁለገብ"ሁሉ"ሁሉም"ሁሉቱም"ሁሉንም"ሁላችንም"ሆነ"ሆኑ"ሆኖ"ሊይ"ላሊ"ላልች"ላይ"ሌላ"መሆኑ"መካከሌከ"ማለት"ሰሞኑን"ሲሆን"ሲሌ"ሲሌ"ስሆ"በሆሊ"በሊይ"በርካታ"በሰሞኑ"በተሇይ"በታች"በኩሌ"በውስጥ"በዚህ"በጣም"በፉት"ብቻ"ተ□ያየ"ተ□ያዩ"ተገለፀ"ታች"ትናት"ትናንት"ትናንትና"ነበረች"ነበሩ"ነበር"ነው"ነይ"ነገር"ነገሮች"ነገሮችነው"ናት"ናቸው"አልነበረም"አሳስበዋል"አስታወቀ"አስገነዘቡ"አስገንዝበዋል"አብራርተዋል"አቶ"አንተ"አንድ"አይደለም"እሱ"እስከ"እሷ"እነሱ"እና"እናንተ"እኔ"እን□"እንደአስረዱት"እንዱሁም"እንዳለት"እንጂ"እኛ"እያንዲንደ"እያንዲንዲች"ከ"ከሊይ"ከሰሞኑ"ከታች"ከውስጥ"ከዚያ"ከጋራ"ከፉት"ወየተ"ወይ"ወይም"ወይየሪት"ወይየሮ"ወ□"ወ□ፉት"ወዲህ"ዉስጥ'የሰሞኑ"የታች"የጋራ"ያ"ያህል"ያለ"ያለው"ይህ"ይላል"ይገልጻል"□ግሞ"ዴረስ"ጋራ"ጋር"ግን"ጥቂት"ፉት'

Appendix II: Association rule based Amharic IR system java code

Indexing module

```
Classtermdistrbtuion {
Public static void main(String[] unused) {
Wordinsertww=new Word insert ();
Scanner in = new Scanner (System. in); // scanner object.
intawords=0,uwords=0;
try{
System.out.println("Enter your file name");
String fileName = in.nextLine();
String fn1;
for (int j=1;j<1300;j++)
{
fn1=""+j;
String fn =fileName+j+".txt";
File nf= new File(fn);
Scanner words = new Scanner(nf);
words.useDelimiter("[^□-□]+");
HashMap<String,Integer>words_fre = new HashMap<String,Integer>();
while(words.hasNext())
{
String s = words. next ().to String();
awords++; // words count
if(words_fre.containsKey(s))
{
int a = words_fre.get(s);
words_fre.put(s,a+1);
}
else {
words_fre.put(s,1);
uwords++; // unique words count
}
}
Object[] key =   words_fre.keySet().toArray();
Arrays.sort(key);
for (inti = 0; i<key.length; i++) {
System.out.println(key[i]+"= "+words_fre.get(key[i]));
String w = (String) key[i];

int c;
c = words_fre.get(key[i]);
int a=5;
try{
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/sport","root","P@$$w0rd");
Statement stmt=con.createStatement();
String sql ="insert into termdistrbtion(fn,word,tf)VALUES('" +fn+ "','" +w+ "','"+c+") " ;
```

```
stmt.executeUpdate(sql);
con.close();
}catch(Exception e){ System.out.println(e);}
}
System.out.println("Total Words = "+awords);
System.out.println("Unique Words = "+words_fre.size());
}
}catch(IOException e)
{
System.out.println(e);
}
}
}
```

Relation extracting

```
public class relation {
public static void main(String args[]) {
String org, fn;
int no;
try {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/sport_stim", "root", "P@$$w0rd");
//here sonoo is the database name, root is the username and root is the password
Statement stmt = con.createStatement();

ResultSetrs = stmt.executeQuery("SELECT * FROM sport_stim.stimed_5 group by word");
int n = 0;
while (rs.next()) {
System.out.println(rs.getString(1) );
String s, q, x = rs.getString(2);
s = null;
org = x;
String term = "word";
String nu = "tf";

n++;
System.out.println(n);
String query = "CREATE TABLE  IF NOT EXISTS  sport_stim."
+ org
+ " (word VARCHAR(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL, tf INTEGER, "
+ "PRIMARY KEY(word))";

Class.forName("com.mysql.jdbc.Driver");

Connection con1 = DriverManager.getConnection("jdbc:mysql://localhost:3306/sport_stim", "root", "P@$$w0rd");
//here sonoo is the database name, root is the username and root is the password
Statement stmt1 = con.createStatement();

stmt1.executeUpdate(query);
```

```
ResultSetrss = stmt1.executeQuery("SELECT * FROM sport_stim.stimed5 where sport_stim.stimed5.word= '" +
org + "'
and NOT EXISTS (SELECT * FROM sport.stopword where sport_stim.stimed5.word=sport.stopword.word )");
System.out.println("vbhjvjvhvj  b--");

int k = 0;
while (rss.next()) {
System.out.println(rss.getString(1) + "  " + rss.getNString(2));
fn = rss.getNString(1);
k++;
System.out.println(k);

Statement stmt2 = con.createStatement();

ResultSetrsss = stmt2.executeQuery("SELECT * FROM sport_stim.stimed_5 where fn= '" + fn + "'
and NOT EXISTS (SELECT * FROM sport.stopword where sport_stim.stimed_5.word=sport.stopword.word )");

HashMap<String, Integer>words_fre = new HashMap<String, Integer>();
while (rsss.next()) {
s = rsss.getString(2);
int b=rsss.getInt(3);
//                                          awords++; // words count
if (words_fre.containsKey(s)) {
int a = words_fre.get(s);
words_fre.put(s, a + b);
} else {
words_fre.put(s, b);
}
                                    }

int L = 0;

Object[] key = words_fre.keySet().toArray();
Arrays.sort(key);

for (inti = 0; i<key.length; i++) {
System.out.println(key[i] + "= " + words_fre.get(key[i]));
                                          String w = (String) key[i];

int c;
                                          c = words_fre.get(key[i]);
int a = 5;
                                          Statement stmt3 = con.createStatement();

                                          String sql3 = "insert into " + org + "(word,tf)VALUES('" + w + "'," + c
+ ")  on duplicate key update

                                          tf= VALUES(tf)+ " + c;
stmt3.executeUpdate(sql3);
                                    }
```

```
                                    }
            }

                        } catch (Exception eee) {
System.out.println(eee);
                        }
                }
}
```

Searching Module

```
public class search {
public static void main(String[] unused) {
Scanner in = new Scanner(System.in); // scanner object.
intUt = 0, awords = 0, uwords = 0, Qt;
System.out.println("Enter Search Term: ");
String fileName = "□□□□□□□□", fn;
System.out.println(fileName);
HashMap<String, Integer>words_fre = new HashMap<String, Integer>();
for (String retval : fileName.split(" ")) {
System.out.println(retval);
Ut += 1;
String s = retval;
awords++; // words count
if (words_fre.containsKey(s)) {
int a = words_fre.get(s);
words_fre.put(s, a + 2);
} else {
words_fre.put(s, 2);
uwords++; // unique words count
}
try {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/sport", "root", "P@$$w0rd");
Statement stmt = con.createStatement();
ResultSetrs = stmt.executeQuery("SELECT * FROM concept where word='" + s + "'");
int n = 0;
while (rs.next()) {
```

```
System.out.println(rs.getString(1) + "  " + rs.getString(2) + "  " + rs.getString(3) + "  " + rs.getString(4) + "  " +
rs.getString(5) + "  " + rs.getString(6) + "  " + rs.getString(7) + "  " + rs.getString(8));

for (int r = 1; r < 10; r++) {
s = rs.getString(r);
if (words_fre.containsKey(s)) {
int a = words_fre.get(s);
words_fre.put(s, a + 1);
} else {
words_fre.put(s, 1);
uwords++; // unique words count
}
}
}
con.close();
} catch (Exception e) {
System.out.println(e);
}
}
Qt = Ut + 3;
Set<Entry<String, Integer>> set = words_fre.entrySet();
List<Entry<String, Integer>> list = new ArrayList<Entry<String, Integer>>(set);
Collections.sort(list, new Comparator<HashMap.Entry<String, Integer>>() {
publicint compare(HashMap.Entry<String, Integer> o1, HashMap.Entry<String, Integer> o2) {
return (o2.getValue()).compareTo(o1.getValue());
}
});
int check = 0;
Map<String, Integer> map = new HashMap<String, Integer>();
String file;

for (HashMap.Entry<String, Integer> entry : list) {
System.out.println(entry.getKey() + " : " + entry.getValue());
```

```
if (check <Qt) {
fn = entry.getKey();

try {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/sport", "root", "P@$$w0rd");
Statement stmt = con.createStatement();

ResultSetrs = stmt.executeQuery("SELECT * FROM termdistrbtion where word='" + fn + "'");
int n = 0;
while (rs.next()) {
file = rs.getString(1);
if (map.containsKey(file)) {
int a = map.get(file);
map.put(file, a + 1);
} else {
map.put(file, 1);
uwords++; // unique words count}
}
con.close();
} catch (Exception e) {
System.out.println(e);   }
}
check++;
}
Set<Entry<String, Integer>> set1 = map.entrySet();
List<Entry<String, Integer>> list1 = new ArrayList<Entry<String, Integer>>(set1);
Collections.sort(list1, new Comparator<Map.Entry<String, Integer>>() {
publicint compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {
return (o2.getValue()).compareTo(o1.getValue());
}
});
int hash=0;
```

```
for (Map.Entry<String, Integer> entry : list1) {
hash+=1;
try {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/sport", "root", "P@$$w0rd");
System.out.println(hash+"  : "+entry.getKey() + " : " + entry.getValue());
Statement stmt3 = con.createStatement();
String sql3 = "insert into search_re(search_query,fn,reang)VALUES('" + fileName + "','" + entry.getKey() + "'," +
entry.getValue()+")" ;
stmt3.executeUpdate(sql3);
}
catch (Exception e) {
System.out.println(e);
}                    }
System.out.println("Total Words = " + awords);
System.out.println("Unique Words = " + words_fre.size());}
}
```

Appendix III

Relevant (judgment)	After		Before	
	Retrieved	Relevant	Retrieved	Relevant
የኢትዮጵያፕሪሚዬርሊግኢትዮጵያቡና ከ ቅዱስጊዮርጊስ				
1 : C: en199.txt : 8 2 : C: en21.txt : 8- 3 : C: ebc4.txt : 7-- 4 : C: walta5.txt : 7- 5 : C: fana7.txt : 7- 6 : C: sporten19.txt == 7r 7 : C: sporten192.txt == 7 8 : C:en184.txt : 7- 9 : C:en187.txt : 7- 10 : C:en7.txt : 6- 11 : C:en176.txt : 6 12 : C:fana100.txt : 5- 13 : C:fana111.txt : 5- 14 : C:en217.txt : 5 15 : C:en299.txt : 4 16 : C:en292.txt : 4- 17 : C:en9.txt : 4-r 18 : C:fana63.txt : 4 19 : C:en267.txt : 4 20 : C:en259.txt : 4 21 : C:en43.txt : 4 22 : C:en141.txt : 4 23 : C:fana1.txt : 4 24 : C:fana28.txt : 4 25 : C:fana8.txt : 4 26 : C:en283.txt : 4 27 : C:en209.txt : 4 28 : C:en242.txt : 4 29 : C:en238.txt : 4 30 : C:en288.txt : 4 31 : C:en20.txt : 4--r 32 : C:en57.txt : 4--r	1 : C:walta5.txt : 11-> yes 2 : C:en21.txt : 11->yes 3 : C:fana7.txt : 11->y 4 : C:en19.txt : 11->y 5 : C:en184.txt : 9->y 6 : C:en187.txt : 9-y 7 : C:ebc4.txt : 8->y 8 : C:en199.txt : 8->y 9 : C:en176.txt : 7->y 10 : C:en192.txt : 6 ->y 11 : C:en249.txt : 6->y 12 : C:en7.txt : 6->y 13 : C:fana28.txt : 6->y 14 : C:en43.txt : 6->y 15 : C:fana100.txt : 6->y 16 : C:en141.txt : 5->y 17 : C:fana111.txt : 5->y 18 : C:en57.txt : 5->y 19 : C:en108.txt : 4->y 20 : C:en292.txt : 4->y 21 : C:fana16.txt : 3 22 : C:fana89.txt : 3- 23 : C:fana94.txt : 3 24 : C:fana13.txt : 3 25 : C:en20.txt : 3->y 26 : C:en267.txt : 3->y 27 : C:en259.txt : 3->y 28 : C:fana109.txt : 3 29 : C:fana26.txt : 2 30 : C:en6.txt : 2 31 : C:en280.txt : 2->y 32 : C:en9.txt : 2-?y 33 : C:en209.txt : 2--y 34 : C:en80.txt : 2 35 : C:fana98.txt : 2 36 : C:fana63.txt : 2y 37 : C:fana1.txt : 2->r 38 : C:fana19.txt : 2?y 39 : C:en271.txt : 2	1 : C:walta5.txt 2 : C:en21.txt 3 : C:fana7.txt 4 : C:en19.txt 5 : C:en184.txt 6 : C:en187.txt 7 : C:ebc4.txt 8 : C:en199.txt 9 : C:en176.txt 10 : C:en192.txt 11 : C:en249.txt : 6->y 12 : C:en7.txt 13 : C:fana28.txt 14 : C:en43.txt 15 : C:fana100.txt 16 : C:en141.txt 17 : C:fana111.txt 18 : C:en57.txt : 5->y 19 : C:en108.txt : 4->y 20 : C:en292.txt : 4->y 25 : C:en20.txt : 3->y 26 : C:en267.txt : 3->y 27 : C:en259.txt : 3->y 25 : C:en20.txt : 3->y 26 : C:en267.txt : 3->y 27 : C:en259.txt : 3->y 31 : C:en280.txt : 2->y 32 : C:en9.txt : 2-?y 33 : C:en209.txt : 2--y 36 : C:fana63.txt : 2y 37 : C:fana1.txt : 2->r 38 : C:fana19.txt : 2?y	1 : C:en192.txt : 3- 2 : C:en176.txt : 3- 3 : C:en184.txt : 3-- 4 : C:fana7.txt : 3- 5 : C:en187.txt : 3- 6 : C:en19.txt : 3- 7 : C:en199.txt : 3- 8 : C:walta5.txt : 3- 9 : C:en21.txt : 3- 10 : C:en141.txt : 2- 11 : C:en249.txt : 2 12 : C:fana111.txt : 2- 13 : C:en7.txt : 2- 14 : C:ebc4.txt : 2- 15 : C:fana28.txt : 2- 16 : C:en57.txt : 2- 17 : C:en259.txt : 2- 18 : C:en43.txt : 2- 19 : C:fana16.txt : 2 20 : C:fana100.txt : 2- 21 : C:en108.txt : 1 22 : C:en20.txt : 1- 23 : C:fana97.txt : 1 24 : C:en157.txt : 1 25 : C:en292.txt : 1- 26 : C:en267.txt : 1- 27 : C:en9.txt : 1- 28 : C:en209.txt : 1-	1 : C:en192.txt : 3- 2 : C:en176.txt : 3- 3 : C:en184.txt : 3-- 4 : C:fana7.txt : 3- 5 : C:en187.txt : 3- 6 : C:en19.txt : 3- 7 : C:en199.txt : 3- 8 : C:walta5.txt : 3- 9 : C:en21.txt : 3- 10 : C:en141.txt : 2- 12 : C:fana111.txt : 2- 13 : C:en7.txt : 2- 14 : C:ebc4.txt : 2- 15 : C:fana28.txt : 2- 16 : C:en57.txt : 2- 17 : C:en259.txt : 2- 18 : C:en43.txt : 2- 20 : C:fana100.txt : 2- 22 : C:en20.txt : 1- 25 : C:en292.txt : 1- 26 : C:en267.txt : 1- 27 : C:en9.txt : 1- 28 : C:en209.txt : 1-
32	39	32	28	23
ዋልያዎቹየአፍሪካዋንጫማጣሪያ				
1 : C:en50.txt : 4- 2 : C:en280.txt : 4-- 3 : C:walta13.txt : 4r 4 : C:ebc27.txt : 4-r 5 : C:en189.txt : 4-?r 6 : C:fana30.txt : 4r 7 : C:en165.txt : 3r 8 : C:fana51.txt : 3-r 9 : C:en269.txt : 3-r 10 : C:ebc17.txt : 3--r 11 : C:ebc23.txt : 3-r 12 : C:fana100.txt : 3 13 : C:en141.txt : 3r 14 : C:en301.txt : 3-r 15 : C:en249.txt : 3-r 16 : C:en66.txt : 3r 17 : C:en48.txt : 3r 18 : C:en46.txt : 3-r 19 : C:en5.txt : 2 20 : C:en132.txt : 2 21 : C:en221.txt : 2-r 22 : C:en67.txt : 2 23 : C:en211.txt : 2 24 : C:en9.txt : 2-r	1 : C:en141.txt : 3->y 2 : C:en189.txt : 3->y 3 : C:en50.txt : 3->y 4 : C:en46.txt : 3y 5 : C:en301.txt : 3->y 6 : C:walta13.txt : 3->y 7 : C:fana30.txt : 3->y 8 : C:en269.txt : 2->y 9 : C:ebc17.txt : 2->y 10 : C:en280.txt : 2 11 : C:en66.txt : 2->y 12 : C:en165.txt : 20->y 13 : C:ebc23.txt : 2y 14 : C:en9.txt : 2-y 15 : C:fana51.txt : 2->y 16 : C:fana4.txt : 2->y 17 : C:en111.txt : 1 18 : C:fana72.txt : 1 19 : C:fana56.txt : 1 20 : C:en247.txt : 1-y 21 : C:en221.txt : 1y 22 : C:en249.txt : 1y 23 : C:en153.txt : 1 24 : C:en293.txt : 1->y	1 : C:en141.txt : 3->y 2 : C:en189.txt : 3->y 3 : C:en50.txt : 3->y 4 : C:en46.txt : 3y 5 : C:en301.txt : 3->y 6 : C:walta13.txt : 3->y 7 : C:fana30.txt : 3->y 8 : C:en269.txt : 2->y 9 : C:ebc17.txt : 2->y 10 : C:en280.txt : 2 11 : C:en66.txt : 2->y 12 : C:en165.txt : 20 13 : C:ebc23.txt : 2y 14 : C:en9.txt : 2-y 15 : C:fana51.txt : 2->y 20 : C:en247.txt : 1- 21 : C:en221.txt : 1y 22 : C:en249.txt : 1y 23 : C:en153.txt : 1 24 : C:en293.txt : 1->y 25 : C:fana2.txt : 1->y 26 : C:fana97.txt : 1--y 27 : C:en41.txt : 1-y 28 : C:en245.txt : 1y	1 :C:fana94.txt 2 : C:fana95.txt 3: C:en94.txt 4 : C:en104.txt : 1 5 : C:en104.txt : 1 6 : C:en111.txt : 1 7 : C:fana72.txt : 1 8 : C:fana56.txt : 1	3

25 : C:en293.txt : 2--r 26 : C:walta11.txt : 2r 27 : C:en252.txt : 2 28 : C:en247.txt : 2--r 29 : C:fana2.txt : 2--r 30 : C:en41.txt : 2-r 31 : C:en245.txt : 2-r 32 : C:fana16.txt : 2-r 33 : C:en214.txt : 2 34 : C:fana4.txt : 2 35 : C:walta20.txt : 2 36 : C:fana97.txt : 2r 37 : C:en280.txt : 2	25 : C:fana2.txt : 1->y 26 : C:fana97.txt : 1--y 27 : C:en41.txt : 1-y 28 : C:en245.txt : 1y 29 : C:fana58.txt : 1 30 : C:en48.txt : 1y 31 : C:en104.txt : 1 32 : C:ebc27.txt : 1-y 33 : C:fana16.txt : 1-y 34 : C:walta11.txt : 1-y 35 : C:fana95.txt : 1 36 : C:en94.txt : 1 37 : C:en252.txt : 1-y 38 : C:fana94.txt : 1 39 : C:fana100.txt : 1--y	29 : C:fana58.txt : 1 30 : C:en48.txt : 1y 31 : C:en104.txt : 1 32 : C:ebc27.txt : 1-y 33 : C:fana16.txt : 1- 34 : C:walta11.txt : 1 39 : C:fana100.txt : 1		
37	39	31	8	
አበረታችመድሃኒት				
1 : C:en303.txt : 2-r 2 : C:fana107.txt : 2- 3 : C:fana110.txt : 2 4 : C:en239.txt : 2-r 5 : C:en232.txt : 1r 6 : C:en12.txt : 1 7 : C:en310.txt : 1-r 8 : C:en156.txt : 1 9 : C:en59.txt : 1-r 10 : C:en212.txt : 1-r 11 : C:en166.txt : 1--r 12 : C:en287.txt : 1--r 13 : C:en179.txt : 1--r 14 : C:en265.txt : 1r 15 : C:en1.txt : 1-r 16 : C:en75.txt : 1-r 18 : C:en196.txt : 1 19 : C:en255.txt : 1 20 : C:en18.txt : 1 21 : C:en305.txt : 2	1 : C:en310.txt : 2-y 2 : C:en212.txt : 2-y 3 : C:en303.txt : 2-y 4 : C:en232.txt : 2-y 5 : C:en305.txt : 2 6 : C:en287.txt : 2-y 7 : C:en265.txt : 2y 8 : C:en1.txt : 2--y 9 : C:en239.txt : 2-- 10 : C:en299.txt : 1 11 : C:en132.txt : 1 12 : C:en59.txt : 1y 13 : C:en115.txt : 1 14 : C:en188.txt : 1 15 : C:en197.txt : 1 16 : C:en158.txt : 1 17 : C:en166.txt : 1-y 18 : C:en196.txt : 1 19 : C:en255.txt : 1 20 : C:en18.txt : 1 21 : C:fana107.txt : 1 22 : C:en179.txt : 1y 23 : C:en97.txt : 1 24 : C:fana110.txt : 1 25 : C:en300.txt : 1 26 : C:en183.txt : 1 27 : C:en75.txt : 1-y 28 : C:fana23.txt : 1 29 : C:en44.txt : 1 30 : C:en65.txt : 1 31 : C:en156.txt : 1-y	1 : C:en303.txt : 2-r 2 : C:fana107.txt : 2- 3 : C:fana110.txt : 2 4 : C:en239.txt : 2-r 5 : C:en232.txt === 1--r 6 : C:en12.txt : 1 7 : C:en310.txt : 1-r 8 : C:en156.txt : 1 9 : C:en59.txt : 1-r 10 : C:en212.txt : 1-r 11 : C:en166.txt === 1-r 12 : C:en287.txt === 1-r 13 : C:en179.txt === 1--r 14 : C:en265.txt === 1--r 15 : C:en1.txt : 1-r 16 : C:en75.txt : 1-r 18 : C:en196.txt : 1 19 : C:en255.txt : 1 20 : C:en18.txt : 1	1 : C:en232.txt : 1- 2 : C:en310.txt : 1- 3 : C:en156.txt : 1- 4 : C:en305.txt : 1- 5 : C:en59.txt : 1- 6 : C:en212.txt : 1- 7 : C:en303.txt : 1- 8 : C:en166.txt : 1- 9 : C:en287.txt : 1- 10 : C:fana107.txt : 1- 11 : C:en179.txt : 1- 12 : C:en97.txt : 1 13 : C:fana110.txt : 1- 14 : C:en265.txt : 1- 15 : C:en1.txt : 1- 16 : C:en239.txt : 1- 17 : C:en75.txt : 1-	1 : C:en232.txt : 1- 2 : C:en310.txt : 1- 3 : C:en156.txt : 1- 4 : C:en305.txt : 1- 5 : C:en59.txt : 1- 6 : C:en212.txt : 1- 7 : C:en303.txt : 1- 8 : C:en166.txt : 1- 9 : C:en287.txt : 1- 10 : C:fana107.txt : 1- 11 : C:en179.txt : 1- 13 : C:fana110.txt : 1- 14 : C:en265.txt : 1- 15 : C:en1.txt : 1- 16 : C:en239.txt : 1- 17 : C:en75.txt : 1-
21	31	21	17	16
የስፖርታዊጨዋነትጉድለት				
1 : C:en256.txt : 2- 2 : C:en171.txt : 2- 3 : C:en20.txt : 2- 4 : C:en173.txt : 2- 5 : C:en270.txt : 2- 6 : C:en91.txt : 2- 7 : C:en209.txt : 2- 8 : C:en59.txt : 1- 9 : C:en4.txt : 1- 10 : C:en184.txt : 1- 11 : C:en191.txt : 1- 12 : C:en206.txt : 1- 13 : C:en242.txt : 1	1 : C:en59.txt : 3 2 : C:en4.txt : 3 3 : C:en191.txt : 3 4 : C:en206.txt : 3 5 : C:en242.txt : 3 6 : C:en263.txt : 3 7 : C:en91.txt : 3 8 : C:en208.txt : 3 9 : C:en182.txt : 3 10 : C:en217.txt : 3 11 : C:en244.txt : 3 12 : C:en209.txt : 3 13 : C:en21.txt : 3	1 : C:en256.txt : 2- 2 : C:en171.txt : 2- 3 : C:en20.txt : 2- 4 : C:en173.txt : 2- 5 : C:en270.txt : 2- 6 : C:en91.txt : 2- 7 : C:en209.txt : 2- 8 : C:en59.txt : 1- 9 : C:en4.txt : 1- 10 : C:en184.txt : 1- 11 : C:en191.txt : 1- 12 : C:en206.txt : 1- 13 : C:en242.txt : 1	1 : C:en256.txt : 1 2 : C:en59.txt : 1 3 : C:en171.txt : 1 4 : C:en4.txt : 1 5 : C:en184.txt : 1 6 : C:en20.txt : 1 7 : C:en191.txt : 1 8 : C:en206.txt : 1 9 : C:en242.txt : 1 10 : C:en263.txt : 1 11 : C:en173.txt : 1 12 : C:en270.txt : 1 13 : C:en57.txt : 1	1 : C:en256.txt : 2- 2 : C:en171.txt : 2- 3 : C:en20.txt : 2- 4 : C:en173.txt : 2- 5 : C:en270.txt : 2- 6 : C:en91.txt : 2- 7 : C:en209.txt : 2- 8 : C:en59.txt : 1- 9 : C:en4.txt : 1- 10 : C:en184.txt : 1- 11 : C:en191.txt : 1- 12 : C:en206.txt : 1- 15 : C:en263.txt : 1-

14 : C:en10.txt : 1 15 : C:en263.txt : 1- 16 : C:en57.txt : 1- 17 : C:en259.txt : 1- 18 : C:en2.txt : 1- 19 : C:en208.txt : 1- 20 : C:en182.txt : 1- 21 : C:en210.txt : 1- 22 : C:en217.txt : 1-- 23 : C:en204.txt : 1- 24 : C:en244.txt : 1- 25 : C:en21.txt : 1- 26 : C:en198.txt : 1- 27 : C:en8.txt : 1 28 : C:en210.txt : 1 29 : C:en153.txt : 1 30 : C:fana110.txt : 1 31 : C:en131.txt : 1 32 : C:en183.txt : 1 33 : C:ebc4.txt : 1	14 : C:en198.txt : 3 15 : C:en256.txt : 2 16 : C:en171.txt : 2 17 : C:en184.txt : 2 18 : C:en20.txt : 2 19 : C:en173.txt : 2 20 : C:en270.txt : 2 21 : C:en57.txt : 2 22 : C:en259.txt : 2 23 : C:en2.txt : 2 24 : C:en204.txt : 2 25 : C:en12.txt : 1 26 : C:en234.txt : 1 27 : C:en6.txt : 1 28 : C:en250.txt : 1 29 : C:en153.txt : 1 30 : C:fana110.txt : 1 31 : C:en131.txt : 1 32 : C:en183.txt : 1 33 : C:ebc4.txt : 1 34 : C:fana91.txt : 1 35 : C:en216.txt : 1 36 : C:en240.txt : 1 37 : C:en8.txt : 1 38 : C:en210.txt : 1 39 : C:en282.txt : 1 40 : C:en235.txt : 1 41 : C:fana90.txt : 1	14 : C:en10.txt : 1 15 : C:en263.txt : 1- 16 : C:en57.txt : 1- 17 : C:en259.txt : 1- 18 : C:en2.txt : 1- 19 : C:en208.txt : 1- 20 : C:en182.txt : 1- 21 : C:en210.txt : 1- 22 : C:en217.txt : 1-- 23 : C:en204.txt : 1- 24 : C:en244.txt : 1- 25 : C:en21.txt : 1- 26 : C:en198.txt : 1- 27 : C:en8.txt : 1 28 : C:en210.txt : 1 29 : C:en153.txt : 1 30 : C:fana110.txt : 1 31 : C:en131.txt : 1 32 : C:en183.txt : 1 33 : C:ebc4.txt : 1	14 : C:en91.txt : 1 15 : C:en259.txt : 1 16 : C:en2.txt : 1 17 : C:en208.txt : 1 18 : C:en182.txt : 1 19 : C:en210.txt : 1 20 : C:en217.txt : 1 21 : C:en204.txt : 1 22 : C:en244.txt : 1 23 : C:en209.txt : 1 24 : C:en21.txt : 1 25 : C:en198.txt : 1	16 : C:en57.txt : 1- 17 : C:en259.txt : 1- 18 : C:en2.txt : 1- 19 : C:en208.txt : 1- 20 : C:en182.txt : 1- 21 : C:en210.txt : 1- 22 : C:en217.txt : 1-- 23 : C:en204.txt : 1- 24 : C:en244.txt : 1- 25 : C:en21.txt : 1- 26 : C:en198.txt : 1-
33	41	18	25	24
የኢትዮጵያፐሪሚዬርሊግየውድድርመርሐግብር				
1 : C:en108.txt : 6r 2 : C:en21.txt : 6-r 3 : C:walta5.txt : 5r 4 : C:fana7.txt : 5-r 5 : C:en19.txt : 5r 6 : C:fana63.txt : 4r 7 : C:en184.txt : 4-r 8 : C:en20.txt : 4--r 9 : C:en187.txt : 4r 10 : C:fana35.txt : 3 11 : C:en292.txt : 3r 12 : C:en209.txt : 3 13 : C:en267.txt : 3r 14 : C:en199.txt : 3r 15 : C:en43.txt : 3-r 16 : C:fana12.txt : 3 17 : C:en141.txt : 3r 18 : C:en192.txt : 3 19 : C:fana26.txt : 3-r 20 : C:ebc15.txt : 3 21 : C:en7.txt : 3--r 22 : C:en211.txt : 3 23 : C:en265.txt : 3 24 : C:en176.txt : 3 25 : C:fana40.txt : 3 26 : C:fana28.txt : 3--r 27 : C:en57.txt : 3r 28 : C:ebc4.txt : 6 29 : C:en249.txt : 3	1 : C:walta5.txt : 9--y 2 : C:en21.txt : 9-y 3 : C:fana7.txt : 9--y 4 : C:en19.txt : 9--y 5 : C:en184.txt : 6y 6 : C:ebc4.txt : 6 7 : C:en187.txt : 6y 8 : C:en192.txt : 5y 9 : C:en176.txt : 5--y 10 : C:en108.txt : 5y 11 : C:fana28.txt : 5y 12 : C:en199.txt : 5y 13 : C:en43.txt : 5--y 14 : C:en141.txt : 4y 15 : C:en7.txt : 4--y 16 : C:en292.txt : 4-y 17 : C:en57.txt : 4--y 18 : C:fana26.txt : 3-y 19 : C:en249.txt : 3 20 : C:fana111.txt : 3--y 21 : C:fana89.txt : 3 22 : C:fana94.txt : 3y 23 : C:fana63.txt : 3 24 : C:en20.txt : 3 25 : C:en267.txt : 3y 26 : C:fana109.txt : 3 27 : C:fana100.txt : 3 28 : C:en280.txt : 2 29 : C:fana16.txt : 2-y 30 : C:en9.txt : 2 31 : C:en209.txt : 2-y 32 : C:en80.txt : 2 33 : C:fana13.txt : 2 34 : C:fana19.txt : 2-y	1 : C:en108.txt : 6--r 2 : C:en21.txt : 6r 3 : C:walta5.txt === 5r 4 : C:fana7.txt : 5r 5 : C:en19.txt : 5r 6 : C:fana63.txt : 4--r 7 : C:en184.txt === 4--r 8 : C:en20.txt : 4-r 9 : C:en187.txt : 4--r 10 : C:fana35.txt : 3 11 : C:en292.txt === 3--r 12 : C:en209.txt : 3 13 : C:en267.txt === 3--r 14 : C:en199.txt : 3-r 15 : C:en43.txt === 3r 16 : C:fana12.txt : 3 17 : C:en141.txt === 3--r 18 : C:en192.txt : 3 19 : C:fana26.txt === 3-r 20 : C:ebc15.txt : 3 21 : C:en7.txt : 3-- 23 : C:fana63.txt : 3 24 : C:en20.txt : 3 25 : C:ebc4.txt : 6 26 : C:en249.txt : 3	1 : C:en108.txt : 2- 2 : C:fana7.txt : 2- 3 : C:en19.txt : 2- 4 : C:walta5.txt : 2- 5 : C:en21.txt : 2- 6 : C:en286.txt : 1 7 : C:en141.txt : 1- 8 : C:en63.txt : 1 9 : C:en192.txt : 1- 10 : C:en176.txt : 1- 11 : C:fana63.txt : 1 12 : C:en72.txt : 1 13 : C:fana26.txt : 1- 14 : C:en184.txt : 1 15 : C:en20.txt : 1- 16 : C:en219.txt : 1 17 : C:fana57.txt : 1 18 : C:en187.txt : 1- 19 : C:en7.txt : 1 20 : C:en211.txt : 1 21 : C:fana10.txt : 1 22 : C:en292.txt : 1 23 : C:fana28.txt : 1- 24 : C:en267.txt : 1- 25 : C:en57.txt : 1- 26 : C:en95.txt : 1 27 : C:fana58.txt : 1 28 : C:en199.txt : 1- 29 : C:en210.txt : 1 30 : C:ebc5.txt : 1 31 : C:en279.txt : 1 32 : C:en43.txt : 1 33 : C:en177.txt : 1 34 : C:en265.txt : 1- 35 : C:en209.txt : 1 36 : C:fana25.txt : 1 37 : C:fana38.txt : 1	1 : C:en108.txt : 2- 2 : C:fana7.txt : 2- 3 : C:en19.txt : 2- 4 : C:walta5.txt : 2- 5 : C:en21.txt : 2- 6 : C:en286.txt : 1 7 : C:en141.txt : 1- 8 : C:en63.txt : 1 9 : C:en192.txt : 1- 10 : C:en176.txt : 1- 13 : C:fana26.txt : 1- 15 : C:en20.txt : 1- 18 : C:en187.txt : 1- 23 : C:fana28.txt : 1- 24 : C:en267.txt : 1- 25 : C:en57.txt : 1- 28 : C:en199.txt : 1- 28 : C:en199.txt : 1- 34 : C:en265.txt : 1-

29	34	26	37	19
ኢትዮጵያበሴቶችማራቶን				
1 : C:fana72.txt : 1 2 : C:walta4.txt : 1 3 : C:en17.txt : 1 4 : C:en162.txt : 1 5 : C:en243.txt : 1 6 : C:en89.txt : 1 7 : C:fana56.txt : 1 8 : C:en24.txt : 1 9 : C:en85.txt : 1 10 : C:fana50.txt : 1 11 : C:en255.txt : 1 12 : C:en161.txt : 1 13 : C:en233.txt : 1 14 : C:en15.txt : 1 15 : C:walta7.txt : 1 16 : C:walta22.txt : 1 17 : C:en289.txt : 1 18 : C:fana49.txt : 1 19 : C:ebc20.txt : 1 20 : C:en140.txt : 1 21 : C:en157.txt : 1 22 : C:en3.txt : 1 23 : C:en105.txt : 1 24 : C:fana53.txt : 1 25 : C:walta19.txt : 1 26 : C:en254.txt : 1 27 : C:en27.txt : 1 28 : C:en127.txt : 1 29 : C:fana6.txt : 1 30 : C:fana83.txt : 1 31 : C:fana87.txt : 1 32 : C:ebc3.txt : 1 33 : C:fana52.txt : 1 34 : C:fana101.txt : 1 35 : C:en128.txt : 1 36 : C:fana5.txt : 1 37 : C:fana46.txt : 1 38 : C:fana86.txt : 1 39 : C:en125.txt : 1 40 : C:en90.txt : 1 41 : C:fana9.txt : 1 42 : C:walta16.txt : 1 43 : C:en194.txt : 1 44 : C:fana68.txt : 1 45 : C:en103.txt : 1 46 : C:en302.txt : 1 47 : C:en159.txt : 1	1 : C:en93.txt : 1 2 : C:fana72.txt : 1 3 : C:walta21.txt : 1 4 : C:en136.txt : 1 5 : C:en219.txt : 1 6 : C:en15.txt : 1 7 : C:en289.txt : 1 8 : C:ebc20.txt : 1 9 : C:en10.txt : 1 10 : C:en118.txt : 1 11 : C:en105.txt : 1 12 : C:en3.txt : 1 13 : C:en237.txt : 1 14 : C:walta19.txt : 1 15 : C:en254.txt : 1 16 : C:en27.txt : 1 17 : C:en156.txt : 1 18 : C:ebc3.txt : 1 19 : C:en8.txt : 1 20 : C:fana101.txt : 1 21 : C:en223.txt : 1 22 : C:ebc22.txt : 1 23 : C:en128.txt : 1 24 : C:en306.txt : 1 25 : C:en194.txt : 1 26 : C:fana77.txt : 1 27 : C:en159.txt : 1	1 : C:en93.txt : 1 2 : C:fana72.txt : 1 3 : C:walta21.txt : 1 4 : C:en136.txt : 1 5 : C:en219.txt : 1 6 : C:en15.txt : 1 7 : C:en289.txt : 1 8 : C:ebc20.txt : 1 9 : C:en10.txt : 1 10 : C:en118.txt : 1 11 : C:en105.txt : 1 12 : C:en3.txt : 1 13 : C:en237.txt : 1 14 : C:walta19.txt : 1 15 : C:en254.txt : 1 16 : C:en27.txt : 1 17 : C:en156.txt : 1 18 : C:ebc3.txt : 1 19 : C:en8.txt : 1 20 : C:fana101.txt : 1 21 : C:en223.txt : 1 22 : C:ebc22.txt : 1 23 : C:en128.txt : 1 24 : C:en306.txt : 1 25 : C:en194.txt : 1 26 : C:fana77.txt : 1 27 : C:en159.txt : 1	1 : C:en93.txt : 1 2 : C:fana72.txt : 1 3 : C:walta21.txt : 1 4 : C:en136.txt : 1 5 : C:en219.txt : 1 6 : C:en15.txt : 1 7 : C:en289.txt : 1 8 : C:ebc20.txt : 1 9 : C:en10.txt : 1 10 : C:en118.txt : 1 11 : C:en105.txt : 1 12 : C:en3.txt : 1 13 : C:en237.txt : 1 14 : C:walta19.txt : 1 15 : C:en254.txt : 1 16 : C:en27.txt : 1 17 : C:en156.txt : 1 18 : C:ebc3.txt : 1 19 : C:en8.txt : 1 20 : C:fana101.txt : 1 21 : C:en223.txt : 1 22 : C:ebc22.txt : 1 23 : C:en128.txt : 1 24 : C:en306.txt : 1 25 : C:en194.txt : 1 26 : C:fana77.txt : 1 27 : C:en159.txt : 1	1 : C:en93.txt : 1 2 : C:fana72.txt : 1 3 : C:walta21.txt : 1 4 : C:en136.txt : 1 5 : C:en219.txt : 1 6 : C:en15.txt : 1 7 : C:en289.txt : 1 8 : C:ebc20.txt : 1 9 : C:en10.txt : 1 10 : C:en118.txt : 1 11 : C:en105.txt : 1 12 : C:en3.txt : 1 13 : C:en237.txt : 1 14 : C:walta19.txt : 1 15 : C:en254.txt : 1 16 : C:en27.txt : 1 17 : C:en156.txt : 1 18 : C:ebc3.txt : 1 19 : C:en8.txt : 1 20 : C:fana101.txt : 1 21 : C:en223.txt : 1 22 : C:ebc22.txt : 1 23 : C:en128.txt : 1 24 : C:en306.txt : 1 25 : C:en194.txt : 1 26 : C:fana77.txt : 1 27 : C:en159.txt : 1
47	27	27		
የዳይመንድሊግውድድር				
1 : C:en40.txt : 3-r 2 : C:fana15.txt : 3--r 3 : C:walta15.txt : 3--r 4 : C:en39.txt : 3r 5 : C:fana20.txt : 3r 6 : C:en295.txt : 3-r 7 : C:en51.txt : 3--r 8 : C:en55.txt : 3--r 9 : C:fana43.txt : 3-r 10 : C:en296.txt : 3-r 11 : C:fana17.txt : 3-r 12 : C:en218.txt : 3--r 13 : C:en37.txt : 3r 14 : C:en54.txt : 3--r 15 : C:en73.txt : 3-r 16 : C:en278.txt : 3--r 17 : C:en273.txt : 3--r	1 : C:fana17.txt : 3 2 : C:en230.txt : 3 3 : C:en55.txt : 3 4 : C:en39.txt : 3 5 : C:fana20.txt : 3 6 : C:en37.txt : 3 7 : C:en40.txt : 3 8 : C:fana15.txt : 3 9 : C:walta15.txt : 3 10 : C:en51.txt : 3 11 : C:fana43.txt : 3 12 : C:en273.txt : 2 13 : C:en294.txt : 2 14 : C:en218.txt : 2 15 : C:en54.txt : 2 16 : C:en283.txt : 2 17 : C:ebc29.txt : 2	1 : C:fana17.txt : 3 2 : C:en230.txt : 3 3 : C:en55.txt : 3 4 : C:en39.txt : 3 5 : C:fana20.txt : 3 6 : C:en37.txt : 3 7 : C:en40.txt : 3 8 : C:fana15.txt : 3 9 : C:walta15.txt : 3 10 : C:en51.txt : 3 11 : C:fana43.txt : 3 12 : C:en273.txt : 2 13 : C:en294.txt : 2 14 : C:en218.txt : 2 15 : C:en54.txt : 2 16 : C:en283.txt : 2 17 : C:ebc29.txt : 2	1 : C:fana17.txt : 1 2 : C:en273.txt : 1 3 : C:en230.txt : 1 4 : C:en294.txt : 1 5 : C:fana15.txt : 1 6 : C:en55.txt : 1 7 : C:en218.txt : 1 8 : C:en283.txt : 1 9 : C:walta15.txt : 1 10 : C:ebc29.txt : 1 11 : C:en39.txt : 1 12 : C:fana20.txt : 1 13 : C:en37.txt : 1 14 : C:en51.txt : 1 15 : C:ebc14.txt : 1 16 : C:fana43.txt : 1 17 : C:en296.txt : 1	1 : C:fana17.txt : 1 2 : C:en273.txt : 1 3 : C:en230.txt : 1 4 : C:en294.txt : 1 5 : C:fana15.txt : 1 6 : C:en55.txt : 1 7 : C:en218.txt : 1 8 : C:en283.txt : 1 9 : C:walta15.txt : 1 10 : C:ebc29.txt : 1 11 : C:en39.txt : 1 12 : C:fana20.txt : 1 13 : C:en37.txt : 1 14 : C:en51.txt : 1 15 : C:ebc14.txt : 1 16 : C:fana43.txt : 1 17 : C:en296.txt : 1

18 : C:en230.txt : 3--r 19 : C:en294.txt : 3--r 20 : C:en283.txt : 3-r 21 : C:ebc29.txt : 3--r 22 : C:ebc14.txt : 2	18 : C:ebc14.txt : 2 19 : C:en296.txt : 2 20 : C:en295.txt : 2 21 : C:en73.txt : 2 22 : C:en278.txt : 2 23 : C:en98.txt : 1 24 : C:en89.txt : 1 25 : C:fana56.txt : 1 26 : C:en247.txt : 1 27 : C:fana57.txt : 1 28 : C:en110.txt : 1 29 : C:en114.txt : 1 30 : C:fana66.txt : 1 31 : C:fana60.txt : 1 32 : C:fana73.txt : 1 33 : C:ebc6.txt : 1 34 : C:fana77.txt : 1 35 : C:fana104.txt : 1	18 : C:ebc14.txt : 2 19 : C:en296.txt : 2 20 : C:en295.txt : 2 21 : C:en73.txt : 2 22 : C:en278.txt : 2	18 : C:en40.txt : 1 19 : C:en54.txt : 1 20 : C:en295.txt : 1 21 : C:en278.txt : 1 22 : C:en73.txt : 1	18 : C:en40.txt : 1 19 : C:en54.txt : 1 20 : C:en295.txt : 1 21 : C:en278.txt : 1 22 : C:en73.txt : 1
22	35	22	22	22
ዓመታዊየስፖርትፌስቲቫል				
1 : C:en178.txt : 2--r 2 : C:en173.txt : 1--r 3 : C:en270.txt : 1 4 : C:en122.txt : 1r 5 : C:en202.txt : 1--r 6 : C:en80.txt : 1-r 7 : C:en16.txt : 1-r 8 : C:en216.txt : 1-r 9 : C:en227.txt : 1 10 : C:en171.txt : 1--r 11 : C:en197.txt : 1 12 : C:en264.txt : 1r 13 : C:en250.txt : 1--r 14 : C:en231.txt : 1r 15 : C:en279.txt : 1--r 16 : C:en244.txt : 1 17 : C:en300.txt : 1r 18 : C:fana81.txt : 1r 19 : C:en272.txt : 1-r 20 : C:en120.txt : 1r 21 : C:en157.txt : 1r	1 : C:en173.txt : 1--y 2 : C:en122.txt : 1y 3 : C:en202.txt : 1--y 4 : C:en80.txt : 1y 5 : C:en178.txt : 1y 6 : C:en16.txt : 1y 7 : C:en216.txt : 1--y 8 : C:en171.txt : 1y 9 : C:en264.txt : 1y 10 : C:en250.txt : 1-y 11 : C:en231.txt : 1-y 12 : C:en279.txt : 1y 13 : C:en300.txt : 1y 14 : C:fana81.txt : 1--y 15 : C:en272.txt : 1-y 16 : C:en120.txt : 1y 17 : C:en157.txt : 1y	1 : C:en173.txt : 1--y 2 : C:en122.txt : 1y 3 : C:en202.txt : 1--y 4 : C:en80.txt : 1y 5 : C:en178.txt : 1--y 6 : C:en16.txt : 1y 7 : C:en216.txt : 1--y 8 : C:en171.txt : 1y 9 : C:en264.txt : 1--y 10 : C:en250.txt : 1--y 11 : C:en231.txt : 1--y 12 : C:en279.txt : 1y 13 : C:en300.txt : 1y 14 : C:fana81.txt : 1--y 15 : C:en272.txt : 1y 16 : C:en120.txt : 1y 17 : C:en157.txt : 1y	1 : C:en173.txt : 1 2 : C:en122.txt : 1 3 : C:en202.txt : 1 4 : C:en80.txt : 1 5 : C:en178.txt : 1 6 : C:en16.txt : 1 7 : C:en216.txt : 1 8 : C:en171.txt : 1 9 : C:en264.txt : 1 10 : C:en250.txt : 1 11 : C:en231.txt : 1 12 : C:en279.txt : 1 13 : C:en300.txt : 1 14 : C:fana81.txt : 1 15 : C:en272.txt : 1 16 : C:en120.txt : 1 17 : C:en157.txt : 1	1 : C:en173.txt : 1 2 : C:en122.txt : 1 3 : C:en202.txt : 1 4 : C:en80.txt : 1 5 : C:en178.txt : 1 6 : C:en16.txt : 1 7 : C:en216.txt : 1 8 : C:en171.txt : 1 9 : C:en264.txt : 1 10 : C:en250.txt : 1 11 : C:en231.txt : 1 12 : C:en279.txt : 1 13 : C:en300.txt : 1 14 : C:fana81.txt : 1 15 : C:en272.txt : 1 16 : C:en120.txt : 1 17 : C:en157.txt : 1
21	17	17	0	
ጥሩነሽዲባባናአህቱ				
1 : C:walta21.txt : 3- 2 : C:fana50.txt : 3- 3 : C:fana53.txt : 3- 4 : C:fana83.txt : 3- 5 : C:fana46.txt : 3- 6 : C:walta16.txt : 3- 7 : C:fana77.txt : 3- 8 : C:walta20.txt : 3- 9 : C:en162.txt : 3- 10 : C:en10.txt : 3- 11 : C:fana101.txt : 3- 12 : C:fana60.txt : 3- 13 : C:ebc22.txt : 3- 14 : C:en125.txt : 3- 15 : C:ebc2.txt : 3- 16 : C:en116.txt : 2- 17 : C:en248.txt : 2- 18 : C:en230.txt : 2- 19 : C:en42.txt : 2- 20 : C:ebc12.txt : 2- 21 : C:fana76.txt : 2- 22 : C:en40.txt : 2 23 : C:en114.txt : 2- 24 : C:en246.txt : 2--	1 : C:walta21.txt : 3 2 : C:fana50.txt : 3 3 : C:fana53.txt : 3 4 : C:fana46.txt : 3 5 : C:walta16.txt : 3 6 : C:fana77.txt : 3 7 : C:en162.txt : 3 8 : C:en10.txt : 3 9 : C:fana101.txt : 3 10 : C:ebc22.txt : 3 11 : C:ebc2.txt : 3 12 : C:fana83.txt : 2 13 : C:walta20.txt : 2 14 : C:en111.txt : 2 15 : C:en138.txt : 2 16 : C:en125.txt : 2 17 : C:en116.txt : 1 18 : C:en248.txt : 1 19 : C:en230.txt : 1 20 : C:en42.txt : 1 21 : C:ebc12.txt : 1 22 : C:en290.txt : 1 23 : C:fana76.txt : 1 24 : C:en114.txt : 1	1 : C:walta21.txt : 3- 2 : C:fana50.txt : 3- 3 : C:fana53.txt : 3- 4 : C:fana83.txt : 3- 5 : C:fana46.txt : 3- 6 : C:walta16.txt : 3- 7 : C:fana77.txt : 3- 8 : C:walta20.txt : 3- 9 : C:en162.txt : 3- 10 : C:en10.txt : 3- 11 : C:fana101.txt : 3- 12 : C:fana60.txt : 3- 13 : C:ebc22.txt : 3- 14 : C:en125.txt : 3- 15 : C:ebc2.txt : 3- 16 : C:en116.txt : 2- 17 : C:en248.txt : 2- 18 : C:en230.txt : 2- 19 : C:en42.txt : 2- 20 : C:ebc12.txt : 2- 21 : C:fana76.txt : 2- 23 : C:en114.txt : 2- 24 : C:en246.txt : 2-- 25 : C:fana15.txt : 2	1 : C:walta21.txt : 3- 2 : C:fana50.txt : 3- 3 : C:fana53.txt : 3- 4 : C:fana83.txt : 3- 5 : C:fana46.txt : 3- 6 : C:walta16.txt : 3- 7 : C:fana77.txt : 3- 8 : C:walta20.txt : 3- 9 : C:en162.txt : 3- 10 : C:en10.txt : 3- 11 : C:fana101.txt : 3- 12 : C:fana60.txt : 3- 13 : C:ebc22.txt : 3- 14 : C:en125.txt : 3- 15 : C:ebc2.txt : 3- 18 : C:en248.txt : 1 19 : C:en230.txt : 1 20 : C:en42.txt : 1 21 : C:ebc12.txt : 1 22 : C:en290.txt : 1	1 : C:walta21.txt : 3- 2 : C:fana50.txt : 3- 3 : C:fana53.txt : 3- 4 : C:fana83.txt : 3- 5 : C:fana46.txt : 3- 6 : C:walta16.txt : 3- 7 : C:fana77.txt : 3- 8 : C:walta20.txt :3- 9 : C:en162.txt : 3- 10 : C:en10.txt : 3- 11 : C:fana101.txt : 3- 12 : C:fana60.txt : 3- 13 : C:ebc22.txt : 3- 14 : C:en125.txt : 3- 15 : C:ebc2.txt : 3- 18 : C:en248.txt : 1 19 : C:en230.txt : 1 20 : C:en42.txt : 1 21 : C:ebc12.txt : 1 22 : C:en290.txt : 1

25 : C:fana15.txt : 2 26 : C:fana70.txt : 2- 27 : C:walta15.txt : 2- 28 : C:en104.txt : 2- 29 : C:fana65.txt : 2- 30 : C:en128.txt : 2- 31 : C:ebc14.txt : 2- 32 : C:ebc26.txt : 2- 33 : C:en111.txt : 2- 34 : C:fana72.txt : 2- 35 : C:en55.txt : 2- 36 : C:en138.txt : 2- 37 : C:fana56.txt : 2- 38 : C:en260.txt : 2- 39 : C:en39.txt : 2 40 : C:fana20.txt : 2- 41 : C:en307.txt : 2- 42 : C:en37.txt : 2 43 : C:en136.txt : 2- 44 : C:en97.txt : 2- 45 : C:en110.txt : 2- 46 : C:fana62.txt : 2 47 : C:en283.txt : 2- 48 : C:fana43.txt : 2- 49 : C:en73.txt : 2-	25 : C:en246.txt : 1 26 : C:fana70.txt : 1 27 : C:walta15.txt : 1 28 : C:en104.txt : 1 29 : C:fana65.txt : 1 30 : C:en81.txt : 1 31 : C:en128.txt : 1 32 : C:ebc14.txt : 1 33 : C:ebc26.txt : 1 34 : C:fana72.txt : 1 35 : C:en63.txt : 1 36 : C:en55.txt : 1 37 : C:fana56.txt : 1 38 : C:en260.txt : 1 39 : C:fana20.txt : 1 40 : C:en307.txt : 1 41 : C:en180.txt : 1 42 : C:en136.txt : 1 43 : C:en97.txt : 1 44 : C:en219.txt : 1 45 : C:en110.txt : 1 46 : C:fana62.txt : 1 47 : C:fana23.txt : 1 48 : C:en283.txt : 1 49 : C:fana60.txt : 1 50 : C:fana73.txt : 1 51 : C:fana43.txt : 1 52 : C:en107.txt : 1 53 : C:en73.txt : 1	26 : C:fana70.txt : 2- 27 : C:walta15.txt : 2- 28 : C:en104.txt : 2- 29 : C:fana65.txt : 2- 30 : C:en128.txt : 2- 31 : C:ebc14.txt : 2- 32 : C:ebc26.txt : 2- 33 : C:en111.txt : 2- 34 : C:fana72.txt : 2- 35 : C:en55.txt : 2- 36 : C:en138.txt : 2- 37 : C:fana56.txt : 2- 38 : C:en260.txt : 2- 40 : C:fana20.txt : 2- 41 : C:en307.txt : 2- 43 : C:en136.txt : 2- 44 : C:en97.txt : 2- 45 : C:en110.txt : 2- 47 : C:en283.txt : 2- 48 : C:fana43.txt : 2- 49 : C:en73.txt : 2-		
49	53	45	0	
አትሌቲክስሬዴሽን				
1 : C:en299.txt : 2- 2 : C:en310.txt : 2- 3 : C:en207.txt : 2- 4 : C:en122.txt : 2- 5 : C:en152.txt : 2 6 : C:fana45.txt : 2- 7 : C:en274.txt : 2- 8 : C:en287.txt : 2- 9 : C:fana107.txt : 2- 10 : C:en233.txt : 2- 11 : C:en139.txt : 2- 12 : C:en254.txt : 2- 13 : C:en284.txt : 2- 14 : C:fana87.txt : 2- 15 : C:en69.txt : 2- 16 : C:fana25.txt : 2- 17 : C:en33.txt : 2- 18 : C:en212.txt : 2- 19 : C:walta22.txt : 2- 20 : C:en237.txt : 2- 21 : C:en127.txt : 2- 22 : C:fana83.txt : 2- 23 : C:en305.txt : 2- 24 : C:fana81.txt : 2- 25 : C:en161.txt : 2- 26 : C:en70.txt : 2-- 27 : C:en242.txt : 2- 28 : C:en53.txt : 2- 29 : C:en253.txt : 2- 30 : C:en223.txt : 2- 31 : C:fana86.txt : 2- 32 : C:en302.txt : 2-	1 : C:en299.txt : 1 2 : C:en310.txt : 1 3 : C:en33.txt : 1 4 : C:en212.txt : 1 5 : C:en161.txt : 1 6 : C:en70.txt : 1 7 : C:fana107.txt : 1 8 : C:en233.txt : 1 9 : C:en242.txt : 1 10 : C:en139.txt : 1 11 : C:en53.txt : 1 12 : C:walta22.txt : 1 13 : C:en237.txt : 1 14 : C:en207.txt : 1 15 : C:en254.txt : 1 16 : C:en122.txt : 1 17 : C:en127.txt : 1 18 : C:en284.txt : 1 19 : C:fana45.txt : 1 20 : C:fana83.txt : 1 21 : C:fana87.txt : 1 22 : C:en305.txt : 1 23 : C:en253.txt : 1 24 : C:en223.txt : 1 25 : C:en274.txt : 1 26 : C:en287.txt : 1 27 : C:fana86.txt : 1 28 : C:en69.txt : 1 29 : C:fana81.txt : 1 30 : C:en302.txt : 1 31 : C:fana25.txt : 1	1 : C:en299.txt : 2- 2 : C:en310.txt : 2- 3 : C:en207.txt : 2- 4 : C:en122.txt : 2- 6 : C:fana45.txt : 2- 7 : C:en274.txt : 2- 8 : C:en287.txt : 2- 9 : C:fana107.txt : 2- 10 : C:en233.txt : 2- 11 : C:en139.txt : 2- 12 : C:en254.txt : 2- 13 : C:en284.txt : 2- 14 : C:fana87.txt : 2- 15 : C:en69.txt : 2- 16 : C:fana25.txt : 2- 17 : C:en33.txt : 2- 18 : C:en212.txt : 2- 19 : C:walta22.txt : 2- 20 : C:en237.txt : 2- 21 : C:en127.txt : 2- 22 : C:fana83.txt : 2- 23 : C:en305.txt : 2- 24 : C:fana81.txt : 2- 25 : C:en161.txt : 2- 26 : C:en70.txt : 2-- 27 : C:en242.txt : 2- 28 : C:en53.txt : 2- 29 : C:en253.txt : 2- 30 : C:en223.txt : 2- 31 : C:fana86.txt : 2- 32 : C:en302.txt : 2-	5 : C:en152.txt : 2-(not ritravid)	
32	31	31	0	

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. This thesis is my original work and has not been submitted as a partial requirement for a degree in any university.

_____

Merhawi Tsegay

February, 2017

Thesis has been submitted for examination with our approval as University advisor.

_____

Million Meshesha (PhD)

February, 2017