ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL SCIENCE
SCHOOL OF INFORMATION SCIENCE

Word Sense Disambiguation using Semantic Similarity for Query Expansion in Amharic Information Retrieval

Samrawit Zewdneh

October 2014

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL SCIENCE
SCHOOL OF INFORMATION SCIENCE

Word Sense Disambiguation using Semantic Similarity for Query Expansion in Amharic Information Retrieval

Samrawit Zewdneh

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

October 2014

Addis Ababa, Ethiopia

DECLARATION

This thesis is my original work and has not been submitted as a partial requirement for a degree in any university.

_____

Samrawit Zewdneh

October 2014

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

_____

Million Meshesha (PhD)

October 2014

ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL SCIENCE
SCHOOL OF INFORMATION SCIENCE

Word Sense Disambiguation using Semantic Similarity for Query Expansion in Amharic Information Retrieval

By:

Samrawit Zewdneh

Name and signature of Members of the Examining Board

Name		Signature	Date
_____	Chairman	_____	_____
Million Meshesha (PhD)	Advisor	_____	_____
Dereje Teferi (PhD)	Examiner	_____	_____
Solomon Teferra (PhD)	Examiner	_____	_____

Table of Contents

Dedication

To the one and only, who lost his life while on scientific expedition in the Denakl Depression

Dr. Kurkura Kabato

Who has given me the support and encouragement I ever needed. I will always treasure the moments we had with you.

Acknowledgment

First and foremost, I would like to thank God, who makes everything possible. I would like to gratefully and sincerely thank my advisor Dr. Million Meshesha for his guidance, understanding, patience and excellent supervision throughout this study.

I am very thankful for the management and staff of MU and all MU-ICT for their encouragement, understanding, and patience and for giving me this great opportunity to pursue my master's degree in the institution.

This thesis would not have been possible without the help, support of my friend and colleague Tsegay Semere. I would like to thank my friend Selamawit for her kindness, friendship and support during my stay. I would also like to thank my friends Eyouel, Kalkidan, Adugna and Bruk for your understanding and help. I thank Solomon for his critical comments on my work.

My special thanks go to Hareg and the family for making all those hardships I had to go through easier. Hareg, you are the best mom and best friend, Rome I thank you for your love, Hirut I thank you for your prayers, Melaku for your encouragement and Tsehay for your care.

I would like to express my heartfelt gratefulness to Besufikad for your support and making each day count.

Last but not least, I would like to give my thanks to my Family for their support, endless love and encouragement. Especially Tsehaye, you are the reason for the person I become today. I thank you all.

List of Figures

List of Tables

List of Acronyms

IC	Information Content
IR	Information Retrieval
NLP	Natural Language Processing
QE	Query Expansion
TF-IDF	Term Frequency Inverse Document Frequency
WSD	Word Sense Disambiguation
WWW	World Wilde Web

Abstract

Query expansion is an effective technique to control the effect of polysmous and synonymous nature of words, thereby improving the performance of information retrieval system. The source of the expansion terms is an important issue in query expansion and determining the sense of each query term is essential for effective retrieval. This study attempts to extend the application of query expansion using semantic similarity measure towards designing an effective word sense disambiguation. Word sense disambiguation is one of the problems involved with context based query expansion. How to use sense information to expand the query is another issue when dealing with query expansion.

This study presents approaches to determine the senses of words in queries by using Amharic lexical resource. The lexical resource like WordNet is the first component that is used as knowledge base. Word Sense Disambiguation is second, which is used to identify the sense of the given query using semantic similarity measure from the knowledge base. Using the idea of lesk algorithm, word sense disambiguation is performed with two methods; *gloss to gloss* and *synset to gloss* by comparing information associated with its synonyms and gloss definition with reference to Amharic WordNet. The third one is Query reformulation which helped to expand the query with the identified sense using word sense disambiguation from the knowledge base. The combination of the two disambiguation methods formulates the modified query and used for expanding the original query. Finally, the query expansion module is integrated with Information Retrieval system to show the enhancement of Amharic IR system performance.

This study shows an effective use of WSD using semantic similarity for identifying the sense and to form the new query. As the experimental result show, the method using synset for query expansion register performance of 59% F-measure. This method registered an improvement of 6% from original query. The number of information associated to each terms is limited because of the lack of resource. Therefore, the use of similarity measure and the use of query expansion terms are limited based on the information available on the lexical resource.

Keywords: *Word Sense Disambiguation, Semantic similarity, Information Retrieval*

CHAPTER ONE

Introduction

1.1 Background

In the past years the growth of the World Wide Web (WWW) both in content and users as well as the vast improvement in search engine technology has radically changed the way knowledge and information is collected and shared [1]. Information Retrieval (IR) is the science of searching for information or documents based on users information need from a huge set of documents [2]. According to Bhogal [3], Information Retrieval is the process of translating a set of information needs into queries and searching for a set of relevant documents that satisfy user's information needs. The goal of retrieval system is therefore to retrieve all relevant documents to the user query while retrieving as few non-relevant documents as possible [4].

A perfect retrieval system would retrieve only the relevant documents and no irrelevant documents [5]. However, perfect retrieval systems do not exist and will not exist, because search statements are necessarily incomplete and relevance depends on the subjective opinion of users. Different users may use the same query to an information retrieval system and judge the relevance of the retrieved documents differently; some users may like the result while others may not. Ordinary web users in many cases simply do not know how to create efficient queries and even the more experienced users usually cannot create good queries when moving on to an unknown domain. A fundamental problem in information retrieval is word mismatch, which refers to the phenomenon that the users of IR systems often use different words to describe the concepts in their queries than the authors use to describe the same concepts in their documents [6][7]. The other main challenges of searching in information retrieval are the fact that the users mostly submit very short and ambiguous queries; and they do not know how to create an efficient query for the exact information they need. This may lead to the retrieval of irrelevant documents depending on the query formulated.

Current practice of information retrieval process with most search engines works at a lexical level, which is retrieving only documents containing the words from the query and this, is called targeted information/document retrieval. The alternative situation in which the words from the query do not exist in the relevant documents, is called imprecise retrieval [8][9]. Numerous information retrieval techniques have been developed based on keywords. These techniques use keyword list to describe the content of the information without addressing anything about the semantic relationships of the keywords. This leads to the difficulty of understanding the meaning of the keywords. Synonym and polysemy are two prominent issues. A synonym is a word which means the same as another word. A polysemy is a word with multiple, related meanings. Query expansion (QE) addresses imprecise retrieval by modifying the query, adding important words related to the original query words [9].

Query Expansion is a way to manipulate user's query in order to retrieve more relevant documents, thereby improving overall performance. It is needed due to the ambiguity of natural language and also the difficulty in using a single term to represent an information concept [10]. The first query is usually tentative and/or an inadequate representation of information need, either in itself or in relation to the representation of ideas in documents [11]. QE attempts to increase the possibility of a match between the query and relevant documents by adding semantically related terms (called expansion terms) to a user's query [12]. The source of the expansion terms is an important issue in query expansion. Then determining the sense of each query term is essential for effective retrieval.

In query expansion, users give additional input on query words or phrases, possibly suggesting additional query terms [13]. However, since the users might be reluctant to provide feedback, researchers started focusing on contextual IR [3]. Contextual IR integrates the user context into the retrieval process. Various approaches exist for conducting query expansion. These approaches can be categorized as either global or local [14]. While global techniques rely on analysis of a whole collection to discover word relationships, the local techniques emphasize analysis of the top-ranked documents retrieved for a query. Global techniques use a thesaurus as a source for query term expansion. This has the advantage of not requiring any user input. Local techniques analyze each document in the result set to find word co-occurrence. More recently,

ontologies have been used in an interactive manner for supporting search queries [3]. Ontological methods suggest an alternative approach which uses semantic relations drawn from the selected terms. According to Fensel [15], Ontologies provide a structured way of describing knowledge. The basic building blocks of ontologies are concepts and relationships. The concepts in the ontology can be used for word sense disambiguation and subsequent query expansion [3]. As Buckland (2003), cited in [10] an ontological model can effectively disambiguate meanings of words from free text sentences. Because of this ontologies have been used to aid query expansion since the early nineties with mixed success [1].

Word Sense Disambiguation is essential in natural language processing. In the field of computational linguistics word sense disambiguation is defined as the problem of computationally determining which "sense" of a word is activated by the use of the word in a particular context. Lexical disambiguation in its broadest definition is nothing less than determining the meaning of every word in context, which appears to be a largely unconscious process in people. Word sense similarity is a generic issue for many applications of computational linguistics [15]. Semantic networks are considered as better choices for estimating semantic similarity than other lexical resources. Among lexical resource like WordNet has been commonly used to measure semantic similarity among words since it has the inherent advantages of being structured in the way of simulating human recognition behaviors

The coverage of concepts contained in document collection determines the effectiveness of WordNet to determine semantic similarity to be used for query expansion. Otherwise, a vocabulary mismatch may happen between the query terms and the concepts in the WordNet. Query expansion has some inherent dangers [1]. The main once are related to a phenomenon named query drift that is moving the query in a direction away from the user's intention. This happens frequently when the query is ambiguous. If the user enters the query with multiple senses the system might choose an interpretation different than the user's intention and augment the query with terms related to the wrong interpretation. This kind of query drift is quite common in ontological methods and stresses the importance of disambiguation of query terms and the query in general. Because of these most ontological methods include a disambiguation preprocessing step.

1.2 Statement of the problem

Searchers naturally prefer to post queries in their native languages. Many languages are spoken in Ethiopia, Amharic is dominant in that it is spoken as a mother tongue by a substantial segment of the population and it is the most commonly learned second language next to English throughout the country [16]. Mindaye et al. [16] stated that, according to Internet World Statistics, Ethiopia took 0.4 % of Internet users out of Africa's share in 2009. The statistics also shows that there was an increase of users of Internet in Ethiopia by 3500% during the years 2000-2009. Africa 2014 population and internet users' statistics for 2013 shows, Ethiopia Internet users increases to 0.8% [17]. Due to this, there is an increase in Internet population within the country and large number of population that speaks the language in Diaspora. At the same time, the number of web documents that are written in Amharic language and Ethiopic script is increasing. In order to search these documents there is a need of search engine that can handle Amharic queries, written in Ethiopic script, well.

Several works have been done in the last decade on Amharic information retrieval system, such as: N-Gram based automatic indexing for Amharic text [18], Amharic text retrieval using latent semantic indexing with singular value decomposition [19], design and implementation of Amharic search engine [20], and the application of probabilistic model for Amharic information retrieval [21]. However, the researchers suggested the need for improving the performance of the information retrieval system by controlling the effects of synonymous and polysemous terms in Amharic language. Most words in natural language are known to have ambiguity. This leads to a difficult task for IR system when users formulate queries that fully represent their information needs.

Synonymous and polysemous are the two major issues when dealing with query expansion. A synonym is a word which means the same as another word. For example, in phrases "አበበ አፋሸገ" and "አበበ ኣዛጋ" words "አፋሸገ" and "ኣዛጋ" have the same meanings. A polysemy is a word with multiple, related meanings [22]. The Amharic language has an abundant set of polysemous along with synonymous words. An example of polysemous word is the word "ጠላ". This word has two completely different meanings, traditional drink and hating. [23]. There are also different researches that were conducted to deal with ambiguities in Amharic Language. Teshome

conducts the first research that attempts in word sense disambiguation for Amharic which tries to resolve lexical ambiguity by demonstrating WSD based semantic vector analysis to improve the effectiveness of Amharic Information retrieval system.

Different researchers have done in the area of Amharic query expansion to deal with the problem of synonymous and polysemous words. One work is by Alemayehu [24], with the aim of applying query expansion to control synonymous words using thesaurus. But, because of the tradeoff between recall and precision and polysemous query terms existence; his proposed system decreased the overall precision of the IR system. As a continuation of Alemayehu's work, Abay [25], implemented query expansion using statistical co-occurrence analysis, bi-gram analysis and bi-gram thesaurus methods; He attempted to differentiate meanings of a polysemous query term using other query by a user. However, the expansion terms used are found to be polysemous themselves. This needs to consider the extent to which a term can be an expansion term in order to divert a query's meaning, which requires integrating ontology based query expansion.

To this end, Iman [23] undertook research for semantic query expansion in order to enhance the retrieval performance of Amharic IR system. After exploring synset and gloss definition for word sense disambiguation, Iman recommended the need for further work to construct a well-defined lexical resource such as Amharic WordNet for word sense disambiguation and sense similarity measurement.

Hence, the current research attempts to extend the application of query expansion using semantic similarity towards designing an effective word sense disambiguation, so as to control the effect of synonymous and polysomous Amharic words during searching. To this end the following research questions are explored and answered in this study.

- What are the suitable techniques for sense similarity measurement based on the constructed lexical resource?
- Does the sematic similarity technique help word sense disambiguation for correct sense identification?
- To what extent the designed word sense disambiguation based query expansion system improve the effectiveness of Amharic IR system?

1.3 Objective of the study

1.3.1 General Objective

The general objective of this research is to investigate the effectiveness of using semantic similarity measure in word sense disambiguation for query expansion so as to enhance the performance of Amharic information retrieval system.

1.3.2 Specific Objectives

In order to achieve the general objective, the study deals with the following specific objectives

- To review previous researches on related works so as to understand the state of the art in Amharic lexical resource, word sense disambiguation and query expansion
- To setup the experiment by organizing Amharic corpus, queries with ambiguity words and relevant judgments for the corpus and word sense disambiguation.
- To prepare lexical resource as a source for disambiguation and expansion
- To build a system and identify suitable approaches for Word Sense Disambiguation
- To design a prototype query expansion for Amharic IR system that can search for relevant documents from Amharic corpus.
- To evaluate the performance of the system using IR effectiveness measures such as recall, precision and F-measure.
- Forward conclusion and recommendations

1.4 Scope and Limitation of the Study

The scope of this research is limited to the development of Word sense disambiguation using semantic similarity for Amharic query expansion to enhance the effectiveness of the IR system. This study focuses on the use of semantic similarity measures towards effective word sense disambiguation for query expansion. The system is designed to search within large size Amharic document corpus. The corpus encompasses documents discussing issues such as, politics, sport, economic, social, accident, health, education, tourism and justice.

Word sense disambiguation based query expansion is designed in the following manner. Lexical resource like WordNet is prepared as a source for disambiguation and expansion. There is no standard WordNet, however, for this study the idea of pieces of information associated with each word WordNet taken to construct the lexical resource. Since WordNet is new to local researches, it only contains synonymous and gloss definition unlike the standard WordNet for English that includes synonyms, hyponyms, hypernyms, definitions of its synonyms and hyponyms, and its domains. Word Sense disambiguation using semantic similarity measure is applied for query expansion in three ways, using synset, gloss definition and the combination of both synset and gloss defination. The query expansion module is finally integrated with probabilistic IR system.

Due to the lack of resources to construct lexical resource such as WordNet and its difficulty and time consuming, the experiment was conducted only in two ways of semantic similarity measure for word sense disambiguation process.

1.5 Methodology of the study

Methodology covers the entire approach of research. This study is an experimental research, which is characterized by much greater control over the research environment and in this case some variables are manipulated to observe their effect on other variables [26]. Experimental research provides a systematic and logical method for exploring the performance of Amharic Retrieval system when the query expansion is integrated with different features.

Experimental research involves dataset preparation, system implementation and evaluation of performance. Methods and tools used at each step are discussed as follows.

1.5.1 Literature Review

To accomplish the objectives of these research literatures from books, journals articles, conference proceedings and the Internet are reviewed concerning Information retrieval, word sense disambiguation, semantic similarity, lexical resource and query expansion in general and specifically for Amharic language. Related works with Query expansion have also been reviewed to understand the state of the art in the field. This helped to investigate the underlying principles and theories, methodologies, techniques and tools of the various methods that can be

adopted, manipulated and used for synthesizing and gaining a new perspective in to what has been done in previous researches in the area and what can be done in this research.

1.5.2 Dataset selection and preparation

For the purpose of testing the impact of the proposed word sense disambiguation based query expansion on the performance of Amharic IR system, the current study use Amharic text documents collected by previous researchers which is from Amharic local news articles available on website of Walta Information Center (*http://www.newspapersites.net/ newspaper/ walta-information-center*) [21, 23] Walta Information Center is a private organization that produces and disseminate news on television and radio as well as e-newspaper.

The researcher preferred to use this dataset so as to easily observe the improvements made in the study by integrating word sense disambiguation based query expansion. The documents contain news related to different topics like sport, politics, justice, accident and health care.

1.5.3 Implementation

Python programming is used for developing the system. Python is selected because it is an open source, interpreted, object-oriented and high level programming language. It enables the researcher to implement the functionality of the sought system without much difficulty and allows writing programs that are clear and readable. It has extensive built-in help functions, which make it possible to learn new things and minimize programming errors.

Experts and beginners can easily understand the code and everyone can become productive in python very quickly [27]. It is a suitable programming language for text processing which is extensively done in information retrieval and search engine development.

1.5.4 Testing Procedure

To test the system ten Amharic queries formulated from previous work with its relevance judgment is used. For the ambiguous words from the formulated query prepared further its relevance judgment that shows the correct sense for each word. Testing is done in two ways for Word Sense Disambiguation and three ways for query expansion. The word sense

disambiguation is evaluated by the user-centered strategy, which uses relevant judgment so as to evaluate the performance of the system. The impact of query expansion is evaluated by integrating with the IR system.

The experimentation for evaluating the effectiveness of the system is done by using selected test documents and queries. Precision, recall and F-measure are used to measure the performance of the prototype system. Precision is the fraction of the documents retrieved that are 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 information retrieval system is measured and compared with previous works.

1.6 Significance of the study

The major contribution of this study is to design word sense disambiguation by applying semantic similarity measure based on the lexical resources prepared. The result of disambiguated concept used for query expansion techniques to enhance Amharic Information Retrieval system with a better performance. There are also specific contributions of the study. First, this work will have a huge effect on future researches of Amharic information retrieval system in improving its performance by understanding the context of queries. Second, it helps to understand the possible use of information associated with each word in WordNet for query expansion and in general the advantage of lexical resource like ontology in information retrieval area. The result of this study has a great help for future researchers that aim to work on developing query expansion technique for sematic languages like Tigregna, Guragigna, Siltigna etc.

1.7 Organization of the Thesis

This thesis is organized in to five chapters. The first chapter presents the general overview of the thesis that comprises the Background, Statement of the problem, General and Specific Objectives, Scope and Limitation, Significance, and Methodology of the study. Chapter two reviews different literatures regarding to IR models, word sense disambiguation, lexical resource, semantic similarity and query expansion techniques. Various related researches done locally and internationally on query expansion systems are also described briefly in this chapter.

The third chapter deals with the design of the proposed methods towards developing lexical resource based semantic similarity for word sense disambiguation to use it for query expansion system. The architecture of the proposed word sense disambiguation and the expansion system also explained in detail. In chapter four the experimentation and evaluation result of the proposed system are discussed. Findings of the results obtained from the experiment and challenges that affected the system's performance are also presented in this chapter. Finally, chapter five provides concluding remarks of the study and forwards recommendations for future work.

CHAPTER TWO

Literature Review

Information retrieval is the key technology behind search engines and a state of the art technology for many web users [21]. The basic concept of Information Retrieval (IR) is the effective retrieval of relevant information from large document corpus [4]. A fundamental problem in information retrieval (IR) is word mismatch, which refers to the phenomenon that the users of IR systems often use different words to describe the concepts in their queries than the authors use to describe the same concepts in their documents [6]. One way to overcome the problem of term mismatch is using query expansion. Query expansion is an effective technique to improve the performance of information retrieval systems [28].

2.1 Information Retrieval

The discipline of information retrieval is almost as old as the computer itself. An earlier, definition of information retrieval given by Mooers (1950), as it is cited on [29] is.

> *Information retrieval is the name of the process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in storage containing information useful to him.*

An information retrieval system is a software programme that stores and manages information on documents. The system assists users in finding the information they need. The system does not explicitly return the exact information; instead, it answers the existence and location of documents that might contain the information. If the suggested documents satisfy the user's information need, the documents are called relevant documents; on the other hand, if not accepted by the user's it is then called irrelevant documents. There is no perfect retrieval system, because the relevance of the document depends on the subjective opinion of the user.

Baeza-Yates and Ribeiro-Neto [4] described that IR is directly affected both by *the user task* and by *the logical view of the documents* adopted by the retrieval systems. In the user task, the user of a retrieval system has to translate his/her information need into a query in the language

provided by the system. With an information retrieval system, this implies specifying a set of words which covey the semantics of the information needed. With the logical view of the documents, the documents represented through a set of index terms or keywords. No matter whether these representative keywords are derived automatically or generated by specialists, it provides the logical view of the document.

According to Croft (1993), as cited in [30], there are three basic processes an information retrieval system has to support: the representation of the content of the documents (Indexing), the representation of the user's information need (query formulation), and the comparison of the two representations(matching process). Figure 2.1 depicts the step-by-step procedure in information retrieval.

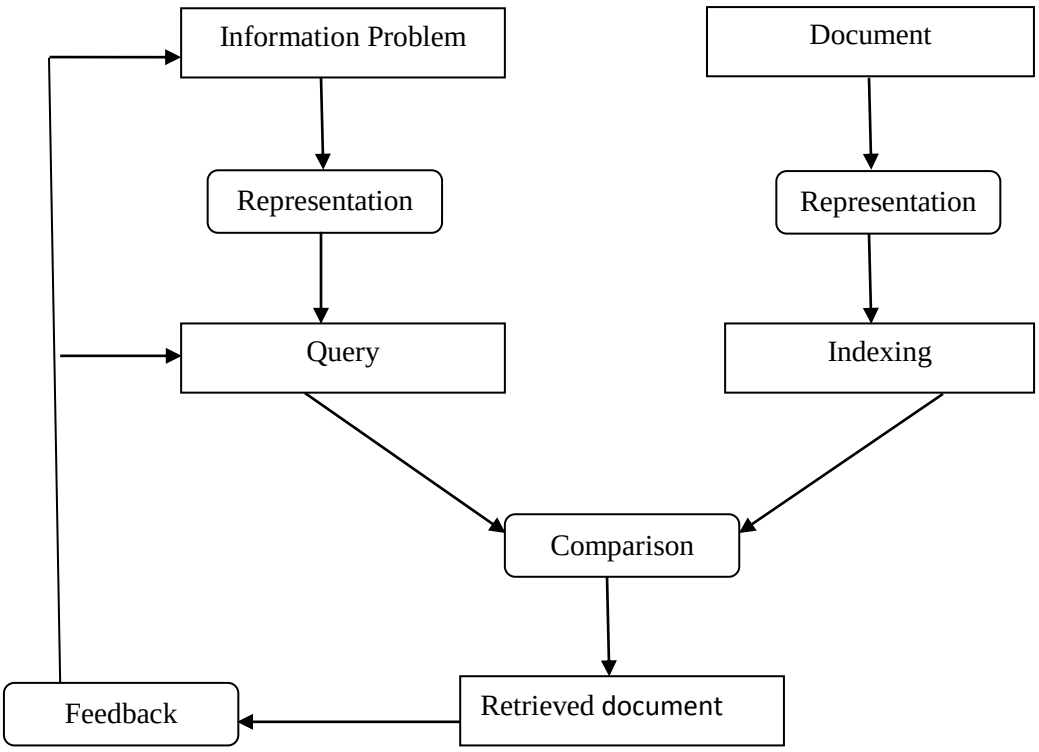

Figure 2.1 Information Retrieval Process

As depicted in figure 2.1 above, before the retrieval process the text database should be well defined and this is a need to specify the document to be used, the operation to be performed on

the text and the text model like the structure [4]. After the text operation performed such as tokenization, stop-words elimination, stemming and normalization, the process transforms the original document and generates a logical view of the document. Once the logical view of the document is defined, it slowly swings from full text searching to a set of index terms. This is called Indexing process. Indexing is a critical data structure because it allows fast searching over large volumes of data [4]. This indexing process takes place off-line, that is, the end user of IR system is not directly involved. These are different indexing structures the well-known and widely used one is inverted index file [21]. Inverted file is a mechanism for indexing a text collection so as to make the searching task fast. There are two elements involved in building the inverted file [3]: the vocabulary and the occurrence. The vocabulary file is the set of index terms in the text collection and it is organized by terms. The vocabulary file stores all of the keywords that appear in any of the documents in lexicographical order and for each word a pointer to posting file. The occurrence contains one record per term, listing all the text locations where the words occurs frequency of each term in a document [4].

Once the indexing process completed, retrieval process can be initiated and the information problem is requested by the user. The process of representing the information problem or information need is often referred to as the *query formulation* process. The resulting formal representation of the information need of users is the *query*. The query is then processed to retrieve relevant documents. This process also involves a serious of steps [31]. First, the user specifies his/her information need using the natural language (e.g. English, Amharic, etc.) supported by the IR system. Second, the system transforms the query into a logical format by applying text operation, which is also used when the document was indexed. Finally, the query is processed to retrieve relevant documents.

After documents are logically organized and the user query is processed, the next step is comparing the query against the document representations, which is called, the matching process [30]. Before sending searching result to the user, the retrieved documents are ranked according to their relevance. This process gives the result of ranked list of documents. Most of the time documents that are considered as relevant to users gets the biggest score and displayed at the top of the retrieved list. Thus, IR models guide the process of matching and ranking relevant

documents. The user then examines the ranked documents. At this point, when a user selects the document, the system then change the query formulation based on the selected document by the user. This can lead to a new cycle, which is modifying the query that helps to get a better representation of the real user need.

2.2 Query Reformulation

In Information retrieval one of the major difficulties is in the description and representation of information needs in terms of a query. Among the basic and essential feature of information retrieval is matching the text of the query to the text of the document. This matching process depends on the process of query reformulation [32]. Query reformulation is a process during which the original query issued by the user is transformed into a structured query representation that is consumed by the search engine. The process modifies the original keyword query submitted by the user to the search engine in order to better represent the underlying intent of the query. The formulated query is then used as an input to the search engine's ranking algorithm. Thus, the primary goal of query reformulation is to improve the overall quality of the ranking presented to the user in response to their query.

According to Bendersky et al. [32], Query reformulation is usually divided into two main processing stages. The first processing stage is query refinement that alters the query on the morphological level. The process involves tokenization, stemming, normalization, stop words removal etc. After the query refinement stage is completed the second processing stage alters the query on the structural level. This structural alteration includes among other actions, segmenting the query into atomic concepts, assigning weights to these concepts, or expanding the query with related weighted concepts.

Query reformulation uses two different methods called query expansion and term reweighting to enhance the performance of the retrieval system [4][21]. Term reweighting technique is a process of adjusting the weight of the term based on the users or system relevance judgment. One of the classical term reweighting techniques is Rocchio algorithm. It is proposed in 1971 for the Smart retrieval system [33]. It takes a set of documents as the feedback document set. Unique terms in this set are ranked in descending order of tf·idf weights. A number of top-ranked terms,

including a fixed number of non-original query terms, are then added to the query. It finds a query vector which increases similarity with relevant document while decreases similarity with non-relevant documents [32]. Other algorithms developed after decades, mostly derived from Rocchio's relevance feedback algorithm. Probabilistic reweighting is another technique of Term reweighting which is designed for probabilistic model. It attempts to predict the probability that a given document will be relevant to a given query [32].

The second method of query reformulation is query expansion the most widely used technique to bridge the vocabulary gaps by expanding original queries with related terms.

2.3 Query Expansion

Users of retrieval systems that use word matching as a basis for retrieval are faced with the challenge of phrasing their queries in the vocabularies of the documents they wish to retrieve. One method of easing the users' burden when selecting query words is for the retrieval system to expand the query by adding terms that are related to the words supplied by the user. In information retrieval, query expansion is referred to as the techniques, algorithms or methodologies that reformulate the original query by adding new terms into the query, in order to achieve better retrieval effectiveness [33].

Query expansion is needed due to the ambiguity of natural language and also the difficulty in using concept to represent an information concept. With query expansion, the users are guided to formulate their query which enables useful results to be obtained. The main aim of query expansion is to add new terms to the initial query. One theory stated that [34], behind query expansion is the problem of synonymy and polysemy of terms. By adding more terms into initial queries precision and recall of the IR system will increase because the expanded queries contains more terms, and accordingly the probability of matching them with terms in the relevant documents, to some extent, increases. Query expansion attaches new additional critical terms beyond the initial query terms (seed query) provided by the users to improve the precision and/or recall of the retrieval system [34].

Query expansion has received significant attention in IR research [11]. All query expansion approaches try to handle the process of reformulation of the original query as a way to improve retrieval performance. The process of adding terms can either be manual, automatic or user-assisted/Interactive [10] [11]. *Manual query expansion* relies on user expertise to make decisions on which terms to include in the new query. In the case of *automatic query expansion*, weightings are calculated for all terms and the terms which have the highest weighting are added to the initial query. Different weighting functions produce different results; therefore retrieval performance depends on the weighting technique used. With *user-assisted query expansion*, the system generates possible query expansion terms and the user selects which of these to include during query reformulation.

2.3.1 Query Expansion Approaches

In literature, Query expansion approaches are studied in different ways [9]. Manning et, al [13] and Wu et, al [35], categorized them into global and local methods, where *Global methods* are query-independent since all documents are examined for all queries. On the other hand, with *local method*, modify a query relative to the documents initially returned by the query. According to Billerbeck [36], query expansion using local methods involves selecting terms from the retrieved relevant documents and adding them to the original query. This new, expanded, query is then re-run on the collection.

Alternatively, Salton and McGill [37] characterized Query expansion approaches as extensional, intentional, or collaborative ones. The extensional approach materializes information need in terms of documents, for instance relevance feedback and local analysis methods. *Intentional* approaches, primarily thesauri/ontology-based, take advantage of the semantics of keywords. *Collaborative* ones exploit users' behavior, e.g., mining query logs, as a complement to previous approaches.

There are two main approaches to query expansion covered in this study which are relevance feedback, and more recently it has been derived from knowledge models such as ontologies. The contextual information can be acquired with one of the approaches [3].

2.3.1.1 Query Expansion Using Relevance Feedback

Relevance feedback is a technique for modification of the initial query using words from previously top-ranked documents that have been identified as relevant documents by the user [37]. The relevance feedback loop requires the user to enter an initial query which results in a display of ranked documents (usually titles/abstracts). From this display, the user makes relevance judgments and selects the relevant documents. The relevant terms from these documents are added to the initial query. To do such process the relevance feedback should consider the term selection, how to weight the new terms, whether to exclude the original query terms, whether to include all of the new terms or just some of them and if so how many new terms to include [3].

Effectiveness of query expansion using relevance feedback can vary depending on many factors [3], such as choice of parameters in the term weighting process, number of relevant documents in the document collection, facilities provided for users to give good quality relevance feedback with ease and finally whether the collection is domain specific or domain independent.

Even though relevance feedback is the most accepted and widely used approach to query expansion, it has problems [3]. It is ineffective when it comes to incorrect spelling. The relevance feedback is also ineffective, if the user uses one word in search but that word is no-where found in the collection then the query will fail. He and Ounis [33], showed the reasons for query expansion's failure. Using this approaches feedback set contains too many non-relevant documents so that misleading expansion terms are added to the query. Second, documents in the feedback set, although containing relevant information, are sometimes only partially related to the topic, and can therefore yield bad expansion terms. This is also called topic drift [3].

2.3.1.2 Query Expansion Using Ontologies

Relevance feedback relies on additional user input. However, since the users might be reluctant to provide feedback, researchers started focusing on contextual IR [3]. Relevance feedback techniques are also content driven in which the corpus content is analyzed to extract candidate terms for query expansion. This can only work if there are sufficient relevant documents to work

with and also that these documents contain a reasonable set of terms that represent the subject area for the query. Contextual IR integrates the user context into the retrieval process. More recently, ontologies have been used in an interactive manner for supporting faceted search queries [37]. According to Arguably et el, (1997), cited in [4] ontology based query expansion is a more effective and favorable method than relevance feedback.

Ontologies seem to be a promising way forward in query expansion. The success of using ontology for query expansion depends on various factors. These are described below [3].

Knowledge model quality: The model must be accurate, stable, comprehensive and up-to-date. If a data model does not cover the subject area in a comprehensive way then queries which are relevant to a subject area will not get any results because the model is suffering from some omissions.

Knowledge model familiarity: The search process has a higher chance of success if the user is familiar with the knowledge model. The initial query formulation starts within the ontology, so its possible for the user to lose their sense of direction or be distracted by the different number of paths during the navigation process and the time taken to traverse those paths. This means that query expansion using ontology, is only beneficial if the searcher is familiar with the search topic.

Navigability of knowledge model: If a user can navigate a knowledge model with ease, this increases its effectiveness. Some ontologies are hundreds of megabytes in size; so a suitable mechanisms should be used to allow large ontologies to fit onto one screen; otherwise, users may 'lose' their way in the vast information space and have difficulty in navigating large knowledge models. To overcome the difficulties users have in navigating ontologies, a mixed approach might be better whereby the system automatically searches the ontology for expansion terms which are suggested to the user who will then interact with the system by selecting the relevant terms.

Other factors: Firstly query terms need to be mapped onto corresponding ontology concepts. If an exact match is not found then the mapping process must find the 'next best' match. The entry point into the ontology forms the basis of any subsequent expansion, so it is crucial to get this process right. Secondly, query length determines whether there will be any resulting benefit from conducting query expansion. It is widely argued that shorter queries are ideal candidates for query expansion because they tend to be more ambiguous. Thirdly, broader information queries benefit more from query expansion than navigational or transactional queries. Finally, using combined query expansion techniques with ontology produces better results than using a single technique.

2.4 WordNet

According to Bates, (2002) and Soergel (1999) as cited in [3][4] ontology is more accurately described as "a classification, thesaurus or a set of concept clusters". Another definition of ontologies is 'classifications, lists of indexing terms, or concept term clusters'. Ontologies improve the accuracy in fuzzy information search and facilitate mono- and multi-lingual human-computer dialogues by paraphrasing the query of the user through context identification and disambiguation. As cited in Bhogal [3], Gruber (1993) defines ontology is a 'specification of a conceptualization'. Gruber explains further that ontologies were first used in philosophy than Artificial Intelligence.

In a natural language, a word may have multiple meanings depending on the applicable context. The purpose of ontology is to provide a context for the vocabulary it contains. In a computer system, context may be represented and constrained by ontology. Therefore, an ontological model can effectively disambiguate meanings of words from free text sentences. Ontologies provide consistent vocabularies and world representations necessary for clear communication within knowledge domains [3].

WordNet is one of the general ontology used in natural language [38]. WordNet is lexical database with remarkably broad coverage [39]. One of its most outstanding qualities is the construction of a word sense network. It is like a dictionary in that it stores words and meanings. However it differs from traditional ones in many ways [38]. For instance, words in WordNet are

arranged semantically instead of alphabetically. The basic relationship between words in WordNet is the Synonym relation called Synset, which is regarded as a basic object in WordNet [38][40]. Words in the same synset are synonymous in a particular sense. Each such synset therefore represents a single distinct sense or concept. Word sense is the meaning a word can take depending on how it is used. For example, the word “bank” could mean a financial institution in one sense and a river bank in another sense.

Words with multiple senses can either be homonymous or polysemous. Two senses of a word are said to be homonyms when they mean entirely different things but have the same spelling. A word is said to be polysemous when its senses are various shades of the same basic meaning. Words with only one sense are said to be monosemous. In WordNet, each word occurs in as many synsets as it has senses [38]. Besides single words, WordNet synsets also sometimes contain compound words which are made up of two or more words but are treated like single words in all respects

Each synset in WordNet has an associated definition or gloss. Each synset contains just one gloss [41]. A gloss for a word sense is the definition of the word in that particular sense and many (but not all) synsets also contain example sentences that show how the words in the synset may be used in the language. For instance, one of the synsets of ‘bank’ is {depository financial institution, bank, banking concern, banking company} and its gloss is (a financial institution that accepts deposits and channels the money into lending activities; "he cashed a check at the bank"; "that bank holds the mortgage on my home") [42].

There is a standard WordNet for English language. WordNet contains more than 118,000 different word forms and more than 90,000 different word senses, or more than 166,000 pairs. Approximately 17% of the words in WordNet are polysemous; approximately 40% have one or more synonyms. WordNet is a large scale database [42] that has a potential to deal with problems on finding the relevant document in text collection for a given user’s query [43]. A group of psychologists and linguists started to develop a “lexical database” in English language at the Cognitive Science Laboratory of Princeton University [44]. WordNet is still maintained by the Cognitive Science Laboratory. Development began in 1985. WordNet stores information

about words that belong to four parts–of–speech: nouns, verbs, adjectives and adverbs. WordNet is defines a variety of semantic and lexical relation between words and synsets. Semantic relation is defines relationships between two synsets [39]. Lexical relations on the other hand defines a relationship between two words within two synsets of WordNet. Thus, whereas a semantic relation between two synsets relates all the words in one of the synsets to all the words in the other synset, a lexical relationship exists only between particular words of two synsets [33][29]. Table 2.1 depicts the semantic relation provided in WordNet with their syntactic category.

Semantic relation	**Syntactic category**	**Example**
Synonymy; (similar)	N, V, Aj, Av	pipe, tube; rise, ascend; sad, unhappy rapidly, speedily
Antonymy (opposite)	Aj, Av, (N, V)	wet, dry; powerful, powerless friendly, unfriendly; rapidly, slowly
Hyponymy(subordinate)	N	sugar maple, maple; maple, tree tree, plant
Meronymy (part)	N	brim, hat; gin, martini; ship, fleet
Troponomy (manner)	V	march, walk; whisper, speak
Entailment	V	drive, ride; divorce, marry
Note: N = Nouns Aj = Adjectives V = Verbs Av = Adverbs		

Table 2.1 Sematic relation in WordNet

- *Synonymy* is WordNet's basic relation, because WordNet uses sets of synonyms (synsets) to represent sent word senses. Synonymy is a symmetric relation between word forms.
- *Antonymy (opposing-name)* is also a symmetric semantic relation between word forms, especially important in organizing the meanings of adjectives and adverbs.
- *Hyponymy (sub-name)* and its inverse, *hypernymy (super-name)*, are transitive relations between synsets. Because there is usually only one hypernym; this semantic relation organizes the meanings of nouns into a hierarchical structure.
- *Meronymy (part-name)* and its inverse, *holonymy (whole-name*), are complex semantic relations. WordNet distinguishes component parts, substantive parts, and member parts.
- *Troponymy (manner-name*) is for verbs while hyponymy is for nouns, although the resulting hierarchies are much shallower.

- *Entailment* relations between verbs are also coded in WordNet

The 1990s saw three major developments [45]: WordNet became available, the statistical revolution in NLP swept through, and Senseval began. WordNet pushed research forward because it was both computationally accessible and hierarchically organized into word senses called synsets. Today, English WordNet (together with WordNet for other languages) is the most-used general sense inventory in Word Sense Disambiguation (WSD) research. Psychology professor Miller [45] supports the WordNet's potential by given two obvious reasons:

- It offers the possibility to discriminate word senses in documents and queries. This would prevent matching *spring* in its "metal device" sense with documents mentioning *spring* in the sense of *springtime*. And then retrieval accuracy could be improved.
- WordNet provides the chance of matching semantically related words. For instance, s*pring*, *fountain*, *outflow*, *outpouring*, in the appropriate senses, can be identified as occurrences of the same concept, '*natural flow of ground water*'. And beyond synonymy, WordNet can be used to measure semantic distance between occurring terms to get more sophisticated ways of comparing documents and queries.

2.5 Measures of Word Semantic Similarity

The need to determine the degree of semantic similarity, or relatedness, between two words is an important problem in Natural Language Processing (NLP) [46]. Similarity measures are used in such applications as word sense disambiguation, determining discourse structure, text summarization and annotation, information extraction and retrieval, automatic indexing, lexical selection, and automatic correction of word errors in text.

There are mainly two approaches to semantic similarity [46, 47]. First approach is making use of a large corpus and gathering statistical data from this corpus to estimate a score of semantic similarity. Second approach makes use of the relations and the hierarchy of a thesaurus, which is generally a hand-crafted lexical database such as WordNet.

Based on the way of utilizing WordNet, the WordNet-based semantic similarity measures can be classified into three categories [15]: *node-based methods*, *Edge-based methods* and *Hybrid methods*

The *node-based methods* which estimate the semantic similarity by computing the amount of information contained by related words in WordNet. Thus, this kind of methods is also called as information-based methods. Most of node-based methods employ the information content to quantify the amount of information that a concept contained. According the definition in the information theory, the Information Content (*IC*) of a concept *c* can be quantified by *IC(c)*=-*log(P(c))*, where *P(c)* is the probability of *c* appearing in a corpus. The drawbacks of node-based methods include, first it is a time-consuming work to analysis the corpora for estimating the IC values and second unbalanced contents of the employed corpora may significantly decrease the accuracy of the IC values.

The *edge-based methods*, which assess the semantic similarity by calculating the length of edges on the shortest path between the words in WordNet. Edge-based methods utilize the shortest path between concepts (i.e., *c1* and *c2*) in WordNet to estimate the semantic relatedness between *c1* and *c2*. Lengths of all edges on the shortest path are accumulated to quantify the semantic similarity. It is the way of calculating the length of edges that differentiates methods in this category. The accuracy of the edge-based methods is significantly affected by the lack of considering the varieties of semantic distances between adjacent words, which is caused by the uneven word densities in WordNet.

The *hybrid methods*, this methods combine the information from different resources to estimate the semantic similarity between concepts, e.g., combining the *IC* of concepts with the structure information retrieved from WordNet to conduct the estimation.

Some of information content based measures discussed that has better performance on word sense disambiguation. All of these measures rely on information content (IC) values assigned to the concepts in the taxonomy, but their usage of IC are different. [47]. It was first proposed by Resnik [48] in 1995 following information theoretic approach, after which Jiang [49], Lin[50] also proposed two other measures respectively.

Resnik's information content

Resnik [48], is the first to bring together ontology and corpus. Guided by the intuition that the similarity between a pair of concepts may be judged by "the extent to which they share information",Resnik defined the similarity between two concepts lexicalized in WordNet to be

the information content of their lowest super-ordinate. He argues that the links in the hierarchy of WordNet representing a uniform distance in the edge-counting measurement cannot account for the semantic variability of a single link. He assumed that for a concept *c*, let *p(c)* be the probability of encountering and instance of concept *c*. The *IC* value is obtained by considering the negative log likelihood.

The Jiang–Conrath Measure

Jiang and Conrath [49] accumulated the scaled length of all the edges in the shortest path between concepts to estimate the semantic similarity of the concepts. The edge length between concept c (a node in the shortest path) and concept p (the parent node of c in the shortest path) is calculated by *length(c, p)* = log(P(p))−.log(P(c)). They also considered the link type, depth, conceptual density, and information content of concepts

The Lin Measure

The Lin [50] measure of semantic relatedness of concepts is based on his Similarity Theorem. It states that the similarity of two concepts is measured by the ratio of the amount of information needed to state the commonality of the two concepts to the amount of information needed to describe them. The commonality of two concepts is captured by the information content of their lowest common subsumer and the information content of the two concepts themselves. This measure turns out to be a close cousin of the Jiang–Conrath measure, although they were developed independently:

The Lesk Measure

As a solution for word sense disambiguation, Lesk [46] proposes to measure the relatedness between two concepts by the overlap between the corresponding definitions of them, as provided by a dictionary. The application of the Lesk similarity measure is not limited to semantic networks, and it can be used in conjunction with any dictionary that provides word definitions.

2.6 Word Sense Disambiguation (WSD)

Lexical disambiguation in its broadest definition is nothing less than determining the meaning of every word in context, which appears to be a largely unconscious process in people. As a

computational problem it is often described as "AI-complete", that is, a problem whose solution pre- supposes a solution to complete natural-language understanding or common-sense reasoning. In the field of computational linguistics, the problem is generally called word sense disambiguation (WSD), and is defined as the problem of computationally determining which "sense" of a word is activated by the use of the word in a particular context. WSD is essentially a task of classification of word senses into their classes, the context provides the evidence, and each occurrence of a word is assigned to one or more of its possible classes based on the evidence [51].

Word sense disambiguation is the process of choosing the right sense for a word in its occurring context [52] [53]**.** Words are assumed to have a finite and discrete set of senses from a dictionary, a lexical knowledge base, or ontology. Most words in natural languages are polysemous, that is they have multiple possible meanings or senses. When using language humans rarely stop and consider which sense of a word is intended. However, computer programs do not have the benefit of a human's vast experience of the world and language. So automatically determining the correct sense of a polyemous word is a difficult problem. WSD has been recognized as a significant component in language processing applications such as information retrieval, machine translation, speech processing etc.

The task of word sense disambiguation is a historical one in the field of Natural Language Processing (NLP). WSD was first formulated as a distinct computational task during the early days of machine translation in the late 1940s, making it one of the oldest problems in computational linguistics. Weaver introduced the problem in his famous memorandum on machine translation [51][54]. Machine translation is the original and most obvious application for WSD but disambiguation has been considered in almost every NLP application, and is becoming increasingly important in recent areas such as Information Retrieval, Information extraction and text mining.

According to Agirre and Edmonds [51], despite this range of applications where WSD shows a great potential to be useful, WSD has not yet been shown to make a decisive difference in any application. There are various isolated results that show minor improvements, but just as often

WSD can hurt performance, as is the case in one experiment on information retrieval [51]. There are several possible reasons for this. First, the domain of an application often constrains the number of senses a word can have (e.g., one would not expect to see the 'river side' sense of a bank in a financial application), and so lexicons can be constructed accordingly. Second, WSD might not be accurate enough yet to show an effect. Third, treating WSD as an explicit component, as the majority of research does, means that it cannot be properly integrated into a particular application or appropriately trained on the domain. Most applications, such as Machine Translation, do not have a place for a WSD module so either the application or the WSD would have to be redesigned.

2.6.1 Approaches to word sense disambiguation

Approaches to word sense disambiguation are often classified according to the main source of knowledge used in sense differentiation [51].

2.6.1.1 Machine Learning Approaches

In machine learning approaches, systems are trained to perform the task of word sense disambiguation. In these approaches, what is learned is a classifier that can be used to assign as yet unseen examples to one of a fixed number of senses. These approaches vary as the nature of the training material, how much material is need, the degree of human intervention, the kind of linguistic knowledge used, and the output produced. But the system accuracy can definitely be improved by machine learning methods. These approaches can be mainly classified into two [51].

Supervised Learning

In such approaches, a learning system is presented with a training set consisting of feature encoded inputs along with their appropriate label, or category. The output of the system is a classifier system capable of assigning labels to new feature encoded inputs. Here a disambiguated corpus is available for training. There is a training set of exemplars where each occurrence of the ambiguous word *w* is annotated with a semantic label. The task is to build a classifier which correctly classifies new cases based on their context of use. Two of the supervised algorithms applied to WSD in statistical language processing is *Bayesian classification* and *Information Theory [51]*. As stated in Kumar [54], Bayesian classification

method is proposed by Gale et.al. It treats the context of occurrence as a bag of words without structure, but it integrates information from many words in the context window. *Information Theory* is proposed by Brownet et.al. This approach looks at only one informative feature in the context, which may be sensitive to text structure, but this feature is carefully selected from a large number of potential informants.

This method suffers from several problems. As mentioned previously, no set of rules can completely disambiguate any word. Moreover, one has to depend on the human tagging of the data which exercise is both error prone and exceedingly tedious. Further, words for which there are no hand–tagged examples needs to pass through unsupervised learning before disambiguation.

Unsupervised Learning

In unsupervised learning we do not know the classification of the data in the training sample. This method works directly from raw unannotated corpora [51]. It can often be viewed as a clustering task. This method is able to induce word senses from training text by clustering word occurrences, and then classifying new occurrences into the induced clusters/senses. Hyperlex and Lin‘s Approach are the main two algorithms used in these techniques several disambiguation systems have been developed for various languages like English, Tamil, Malayalam, Hindi, Chinese [54].

2.6.1.2 Knowledge Based Approaches

The knowledge-based proposals of the 1970s and 80s are still a matter of current research [51]. These approaches are mainly using external lexical resources such as dictionaries, thesaurus, WordNet etc. These are easy to implement because they require simple look up of a knowledge resources like a machine readable dictionary. Here no need of a corpus-tagged or untagged, since no training is involved. So many algorithms are suggested with this approach.

Walker's algorithm

It is a thesaurus based approach. Walker's algorithm works as follows [54]. First, it finds the thesaurus category to which that sense belongs. Then calculate the score for each sense by using the context words. Contexts will be added to the score of the sense if the thesaurus category of the word matches that of the sense. By using WordNet, it is possible to find the conceptual distance by analyzing the hyponyms. Once we find out the conceptual distance, conceptual density can be measured. If the conceptual distance is smaller, conceptual density will be higher. Let w be the word to be disambiguated. w_1, w_2,….w_n etc are the words in context. Each symbol represents the different senses of the word in context. Highest density will be obtained for the sub hierarchy containing more senses

Random Walk algorithm

In a sentence there may be more than one word which has different senses. In this approach, a vertex is created for each possible sense of each word in a text. By using definition base similarity, we can add weighted edges. A graph based ranking algorithm is then applied to find score of each vertex. Then the highest score vertex is selected as the correct sense (for each word).

Lesk Algorithm

This method is suggested by the scientist M.Lesk [54]. According to Lesk, a word is disambiguated by comparing the gloss of each of its senses to the glosses of every other word in the phrase. The algorithm disambiguates a target word by comparing its gloss with those of its surrounding words. The target word is assigned the sense whose gloss has the most overlapping or shared words with the glosses of its neighboring words [55]. The sense whose gloss shares the largest number of words in common with the glosses of other words is selected as the correct sense. Suppose that 'bark' is the target word and it is surrounded by 'dog' and 'tail'. The original Lesk algorithm checks for overlaps in the glosses of the senses of dog with the glosses of bark. Then it checks for overlaps in the glosses of 'bark' and 'tail'. The sense of 'bark' with the maximum number of overlaps with 'dog' and 'tail' is selected. The adaptation of the Lesk algorithm considers these same overlaps and adds to them the overlaps of the glosses of the senses of concepts that are semantically or lexically related to dog, bark and tail according to WordNet.

Hybrid approaches by combining multiple knowledge sources and using tagged data are also one of the approaches to WSD [54].

2.7 Amharic Language

Ethiopia is a linguistically diverse country where more than 80 languages are used in day-to-day communication. Amharic was the national language of Ethiopia until 1995. Following the declaration of constitution of Ethiopian federal democratic government on Article 5(1), it becomes the working/official language. The language is a Semitic language and uses the Ethiopic script for writing. The script of Amharic is taken from Ge'ez. Amharic did not discriminate in adopting the Ge'ez Fidel; it took all of the symbols [56] and added some new ones that represent sounds not found in Ge'ez. These added alphabetic characters are ቾ, ጨ, ፇ, ኘ, ቨ, ሸ, ኸ, and ዠ.

In Amharic each symbol represents a consonant and vowel combination and the symbols are organized in groups of similar symbols on the basis of both the consonant and the vowel. For each consonant in each symbol, there is an unmarked symbol representing that consonant followed by a canonical or inherent vowel [27]. Currently, the writing system of Amharic language contains 34 base characters each of which occurs in a basic form and six other forms known as orders. The seven orders represent syllable combinations consisting of a consonant following vowel. This makes the writing system consists in general 238 unique symbols. In addition, there are forty others that contain a special feature usually representing labialization e.g. ቿ, ቋ. In Amharic there is no Capital-Lower case distinction. There are also 8 punctuation marks and 20 numeration system.

Amharic writing system got some problems [17]. The first problem is the presence of "unnecessary" alphabets (Fidels) in the language's writing system. These Fidels (alphabets) have the same pronunciation but different symbols, while the meaning is the same. The Fidels are ኣ and ዐ, ጸ and ፀ, ሰ and ሠ and ሀ, ሐ, and ኀ. The other problem is in the formation of compound words. Compound words are sometimes written as two separate words and sometimes as a single word. For example, the word "kitchen" can be written as "ወጥ ቤት" or "ወጥቤት". There are many such compound words, which need some effort to have a standard way of forming them. Another

problem of the language is, there are different ways of writing a single word because of regional dialect with their spoken form; “ሂጂ” vs. “ሂጅ”, “አይዶለም” vs. “አይደለም”, “ዓጤ” vs. “ዓፄ” [57]. The other problem is related to if the word is loan word from foreign languages. For example, the word Computer can be written as ኮምፒዩተር, ኮምፒውተር, ኮምፒዉተር, etc. Any application tools that are developed for Amharic language need to consider the characteristics of the language mentioned above in one way or another.

2.7.1 Amharic WordNet

Princeton WordNet is a great inspiration for the development of WordNet in different languages. There have been many efforts to develop a WordNet for different languages. There are two approaches of developing a WordNet [17]: the merge approach and extended approach. Extended approach translate the synsets in the Princeton WordNet to your own language, take over the relations from Princeton and revise and merge approach define synsets and relations in your own language and then align your WordNet with the Princeton WordNet using equivalence relations. Kassie [58], suggests extended approach to develop Amharic WordNet because of the following two reasons:

- It reduces the cost and time of developing Amharic WordNet from scratch.
- It gives an opportunity to integrate the language WordNet with other languages WordNet.

It is therefore wise to use the information in the Princeton WordNet for such under-resourced languages like Amharic.

So far there is no standard WordNet for Amharic language. Many applications in Amharic language, such as Amharic search engine, Amharic automatic text categorization, Amharic Word Sense Disambiguation develop their tools without the use of WordNet. It is argued that if there was an Amharic WordNet, the application would have increased the performance of the tools they developed [17].

2.7.2 Word Sense Disambiguation for Amharic

Ambiguity is defined as the property of being ambiguous, where a word, term, notation, sign, symbol, phrase, sentence, or any other form used for communication, can be interpreted in more

than one way (Mihalcea and Pedersen, 2005) as cited in [17]. When language is capable of being understood in more than one way by a reasonable person, ambiguity exists. Ambiguity is inherent to human language. Successful solutions for automatic resolution of ambiguity in natural language often require large amounts of annotated data/knowledge resources to achieve good levels of accuracy.

A study done by Kassie [58], tries to develop a tool for Amharic word sense disambiguation In the study, Amharic Penal Code document was used for experimentation by applying Semantic Vectors of words of dimension 200. The term vectors are built from index of terms using lucene IR library. Using those term vectors thesaurus can be constructed by calculating the k nearest neighborhood from the word space by applying the distance measure between points of term representation according to the usage of terms in documents. In other words, a query that is one word is run using the prototype where the system retrieves words by applying the similarity calculation of nearest neighborhoods from documents according to their usage. The neighborhood is calculated from the co-occurrence frequency of words in documents. The average precision and recall of the system is 58% and 82%, respectively.

2.8 Related Works

In the past 20 years the area of information retrieval has grown well beyond its primary goals of indexing text and searching for useful documents in a collection [4]. In order to improve the efficiency and effectiveness of retrieval system, various approaches with different models have been used. Since word ambiguity presents an important issue in Information Retrieval community, there has been a lot of efforts invested to discover how to deal with the problem. In IR research, QE has received significant attention and previous work was done to explore methodologies that can enhance the performance of information retrieval system using different query expansion techniques.

2.8.1 Global Researches

Eldin and Elsayed [11], proposed a model that considers the major problem in IR, formulation of queries on the part of the user. The work introduces Query Expansion mechanism involving new

expansion process that is guided by conceptual representation approach using Concept Mapping tool for expanding the original query, in the context of the utilized corpus.

The dataset is from MEDLINE collection. It has 1033 number of documents, 5481 number of index terms and 30 numbers of queries. Extracting additional terms for expansion process by using Cmap tool and applying both the linguistic-based and domain-based approach in the expansion process and show their effects on the recall of retrieved results. Experimental result shows that, a recall 97% and 94% archived using linguistic based and domain based respectively. The recall result affected by user experience constructing the maps. Experimental results, using the MEDLINE test collection data, show the effect of using conceptual representation approach via linguistic and domain based on the recall of retrieval results to enhance its performance.

Word Sense Disambiguation in queries by Liu. *et al* [59] present a new approach to determine the senses of words in queries by using WordNet. In this paper, they utilize word sense disambiguation to improve retrieval performance in two aspects. First, it helps bring in new terms and phrases to the query. Suppose w is a term, and (w w') is a noun phrase in a query. After the sense of w is determined, the selectively chosen synonyms, hyponyms, similar words, and compound concepts of w are added to the query. New terms that are brought in by w form phrases with w' or terms brought in by w'. Second, they assign an additional weight to a feedback term if it can be semantically related to some disambiguated query term. Experiments are performed on the most recent TREC queries in the robust track. This set consists of 250 queries and 333 of them are ambiguous terms in the queries. Experimental results show that the integration of word sense disambiguation algorithm the retrieval system yields an improvement of a 13.7%. Their retrieval effectiveness is 7% better than the best reported result in the literature for short queries.

Bhogal [3] examines the use of ontology based query expansion for defining query context. In this work user's relevance feedback and pseudo relevance feedback for query expansion are considered. The IR system used is based on the probabilistic retrieval model and the query expansion method is extended using information from news domain ontology. The result shows that ontology based query expansion has resulted in higher number of relevant documents being

retrieved compared to the standard relevance feedback process. Ontology based query expansion improves recall but does not produce any significant improvements for the precision results. Pseudo-relevance feedback has achieved better results than user's relevance feedback. The study found that reducing or increasing the relevance feedback parameters (number of terms or number of documents) does not correlate with the results. The study also identifies factors that influence the success of ontology based query expansion; such factors include quality, familiarity.

Hoang and Tjoa [60], surveyed several ontology based query systems on various aspects of using ontologies, including faceted search, query reformulation and refinement. Bhogal [10] provided a comprehensive review of ontology based query expansion, which presents several query expansion approaches, focusing on examples using corpus dependent or independent ontologies.

2.8.2 Local Researches

Different researches attempts to develop information retrieval system for Amharic language. A number of IR systems developed so far for retrieving Amharic texts. Betelihem [18] developed n-gram-based automatic indexing for Amharic text retrieval. This research mainly conducted to solve the problem of not having standard stemming procedure and stop-word list for Amharic language. A work followed by Tewodros [19] developed Amharic text retrieval using latent semantic indexing (LSI) strategy with singular value decomposition. His work mainly focuses on indexing process, to solve the problem found in exact term matching retrieval system. Then Tessema and Solomon [20], designed and implement Amharic search engine, which retrieve web documents written in Amharic language. Amanuel [21] design and develop a probabilistic based information retrieval system for Amharic language in order to enhance the retrieval performance of Amharic IR system considering the advantage of the probabilistic model.

Researches for query expansion with local languages are very few. So far three researches have been done for enhancing the performance of Amharic retrieval system.

Alemayehu [24] developed a prototype for automatic query expansion based on thesaurus as a source to find similar terms for reformulation of user's query. He constructed thesaurus using

WORDSPACE model. His finding increased the recall of the IR system by 44% but this was at the cost of reducing precision by 34%.

Abey [25] followed the footsteps of Alemayehu [24] and explored query expansion using sematic word relationship in an attempt to handle synonyms and polysemous Amharic words. He used a pseudo relevance feedback by using statistical co-occurrence analysis and bi-gram analysis. The experiment he conducted showed the statistical co-occurrence analysis is the best technique registering an improvement of 10% F-measure from the initial result of the experiment without query expansion. He suggested as further research direction ontological query expansion technique for query expansion.

Towards implementing ontology based query expansion for information retrieval, Iman [23] made the first attempt for information retrieval. She used a WordNet for finding the word sense and accordingly for word sense disambiguation to identify the correct senses of terms and expand the query using the term's definition or synonym sets of terms.

Two experiments were carried using synset and gloss definition of terms. The first experiment using gloss expansion shows 50% decline in the overall F-measure while the recall of the system remained the same. The second experiment using synset expansion shows an increase in its general performance. She suggested as further research to advance the WordNet and to investigate other ways of similarity measurements for identifying the sense.

Thus, in this study an attempt is made to develop effective word sense disambiguation using semantic similarity for query expansion to enhance the performance of the Amharic information retrieval. Different approaches of semantic similarity measures are investigated in identifying the sense of the Amharic terms during word sense disambiguation using Amharic lexical database.

CHAPTER THREE

Designing Word Sense Disambiguation based Query Expansion

In information retrieval, adding appropriate synonyms to a query can improve retrieval effectiveness. However, most query terms have multiple meanings and adding a synonym of the query term which has a different meaning in the context of the query would cause deterioration in retrieval effectiveness. Therefore, determining the sense of each query term is essential for effective retrieval. Once a query term's sense in a query context is determined, synonyms with the same meaning as that of the query term are added to the query, so that documents having these synonyms but not the actual term may be retrieved. In this study an attempt is made to design word sense disambiguation using semantic similarity measure for query expansion to enhance Amharic information retrieval.

3.1 Architecture of Amharic Query Expansion for Information Retrieval

As depicted in figure 3.1, the designed query expansion system has different components. Given a query from users the IR system is expected to search and return relevant documents. To enhance effectiveness of the IR system, there is a need to apply query expansion. In this study, lexical resource like WordNet is constructed as a reference for identifying the senses and meaning of the query using Word Sense Disambiguation by semantic similarity measure. The identified word senses are used during query reformulation. Finally, the query expansion module is integrated with the Amharic IR system to see the performance of the system in retrieving relevant documents for the users query.

.

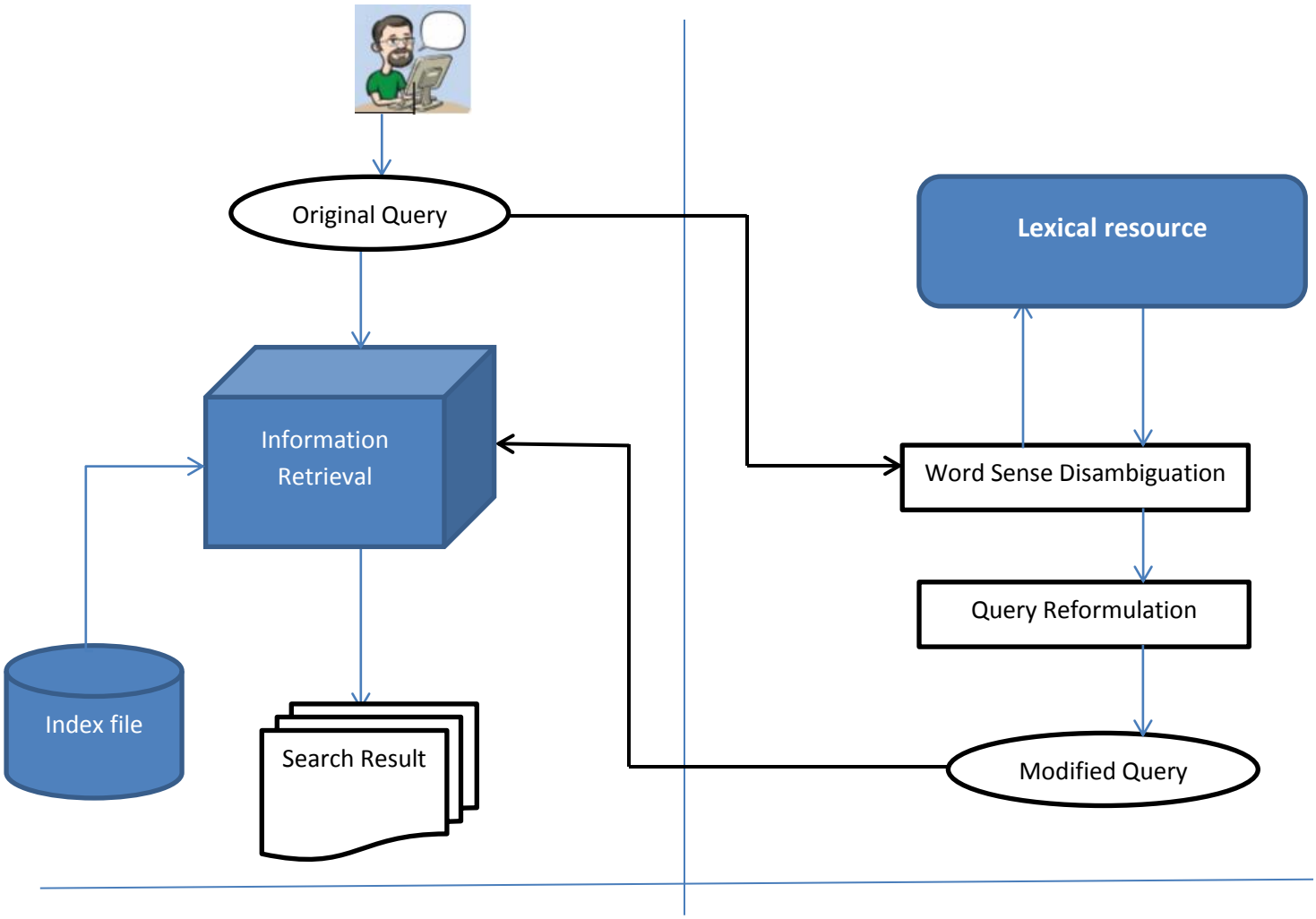

Figure 3.1 WSD based query expansion Architecture

3.2 Word Sense Disambiguation

The Word Sense Disambiguation is performed on the original query terms of the user. It is needed because the user queries are assumed to be ambiguous. The task of disambiguating words in the query begins after initially entering the query to the information retrieval system. The system takes this query and performs word sense disambiguation on each query word. This is done by using lexical resource to disambiguate between query senses.

3.2.1 Lexical Resource Preparation

During WSD or query expansion, the first important decision is the choice of the source of candidate expansion terms [12]. Lexical resources like WordNet are used as the source for disambiguation and expansion the given terms. Lexical resources used for query expansion may be constructed either manually (e.g. WordNet), or automatically (usually based on co-occurrence information). Since automatically constructed thesauri are usually based on corpus statistics, they may contain linguistic flaws, like sentence structure, language rules. In contrast, resources like WordNet that are handcrafted by experienced lexicographers are expected to contain less noise [12].

3.2.1.1 Preparation of Amharic WordNet

The main focus of this study is on the use of lexical resource like WordNet to identify the sense of the given word and use it for query expansion. Since there is no standard WordNet for Amharic language, here manually constructed lexical resource is used. The lexical resource is constructed by consulting language experts and with the use of Amharic Dictionary by Desta Tekleweld [61] and Amharic Context Dictionary [62]. This Amharic WordNet is limited to contain only two information associated with each term which are words with list of synonyms that have similar senses and the gloss definition those groups of words with similar sense.

WordNet synset:-The basic relationship between words in WordNet is the Synonym relation called Synset. Words in the same synset are synonymous in a particular sense. Word sense is the meaning a word can take depending on how it is used. For example the word "በራ" could mean 'giving light' in one sense and 'bold' in another sense. Each synset of a word contains one or more words including the word itself and has a gloss associated with it. Then, the synset of the word "በራ" will be "ብርሃን, ነጸበራቅ" for the first sense and for the second one is 'መላጣ'.

WordNet Gloss:- A gloss for a word sense is the definition of the word in that particular sense and typically includes example sentence(s). For example, for the word "በራ" the first sense definition is, "ሻማ መብራት ጸሃይ የሚሰጠዉ ብርሃን", and for the second one is "የወንድ ልጅ ጸጉር ሲመለጥ".

There is an issue to be considered when dealing with the WordNet as source of terms for expansion [12]. One is if a query word occurs in multiple senses, which sense should be selected? How to select it? Once some sense has been selected, which words should be added to the query? Should only synonyms contained in these synsets be added or the one with the gloss definition be added?

To understand in what sense the word is used and to identify the meaning of words in contextual manner a Word Sense Disambiguation (WSD) using semantic similarity measure is applied.

3.2.2 Semantic Similarity Measure for WSD

Given a query containing multiple words, the aim is to find the precise meaning (sense) of each word in the context of other query words. If the query consists of a single word and the word has multiple meanings, it is usually not possible to determine the sense of the query word. Thus, this work concentrates on multi-word queries.

In WordNet, there are several pieces of information associated with each content word and they can be used for word sense disambiguation [59]. As discussed above, in this study the lexical resource constructed like a WordNet that includes its synonymous set and gloss definition with its examples. By comparing these pieces of information associated with the terms, it may be possible to assign senses to these terms. To do this the Lesk algorithm is used to measure semantic similarity.

3.2.3 Lesk Algorithm

Lesk algorithm involves looking for overlap between the words in given definitions with words from the text surrounding the word to be disambiguated. The original Lesk algorithm disambiguates words in short phrases [39]. In this algorithm to disambiguate the given word, the gloss of each of its senses is compared to the glosses of every other words in the phrase. A word is assigned that sense whose gloss shares the largest number of words in common with the glosses of the other words. The algorithm begins a new for each word and does not utilize the senses it previously assigned.

The algorithm exploits the similarity or relatedness between the sense definitions of the ambiguous word (MA) and the definitions of the words of its context $\{M_1, M_2, \ldots, M_i, \ldots, M_n\}$. Figure 3.2 below provides the architecture of the Lesk algorithm, where S_A^i is the gloss definition corresponding to the i^{th} sense of the ambiguous word.

The original algorithm uses a dictionary as a resource. For every possible meaning of the word to disambiguate S_j, a definition $D(S_j)$ is attributed. The word M (belonging to the context of the ambiguous word) is represented by the dictionary definitions E(M).

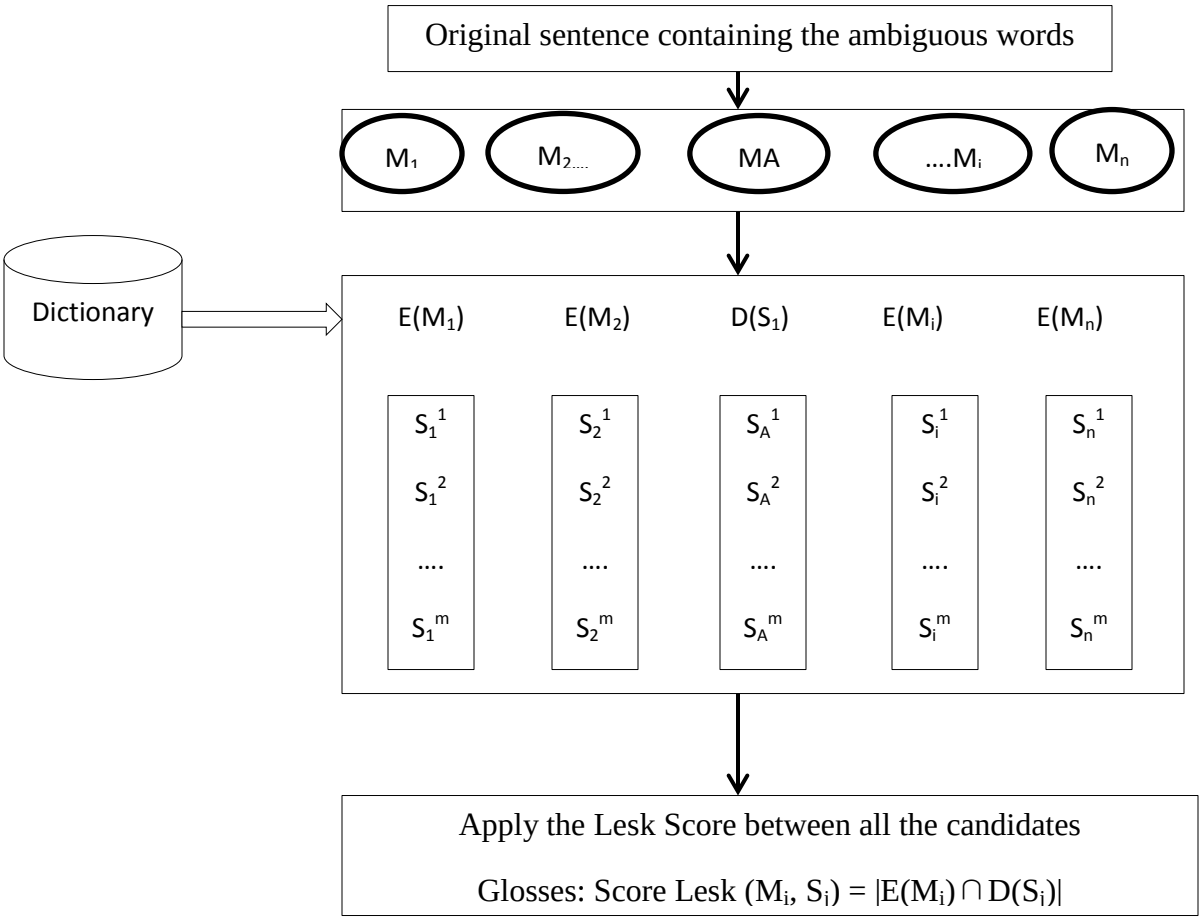

Figure 3.2. Original Lesk Algorithm Architecture

For two given words, equation 3.1 below allows us to calculate the overlap between each possible definition of the ambiguous word and the definition of words contained in the same sentence as the ambiguous word.

$$\text{Score Lesk } (M_i, S_j) = |E(M_i) \cap D(S_j)| \text{.....3.1}$$

For each word to disambiguate, the algorithm assigns initially, as the best candidate, the most common sense. Another meaning is chosen if and only if its score is higher than the current best candidate.

Lesk [39] demonstrates this algorithm on the words *pine cone*. Using the Oxford Advanced Learner's Dictionary, it finds that the word *pine* has two senses:

*Sense 1: kind of **evergreen tree** with needle–shaped leaves*

Sense 2: waste away through sorrow or illness.

On the other hand, the word *cone* has three senses:

Sense 1: solid body which narrows to a point

Sense 2: something of this shapes whether solid or hollow

*Sense 3: fruit of certain **evergreen tree***

Each of the two senses of the word *pine* is compared with each of the three senses of the word *cone* and it is found that the words evergreen tree occurs in one sense in each of the two words. These two senses are then declared to be the most appropriate senses when the words *pine* and *cone* are used together.

While Lesk's algorithm restricts word sense disambiguation using the gloss definition in this study the idea of the Lesk algorithm is extended to disambiguate the sense using synset and the gloss definition.

Method One: ***There are content words in common between the definition of one sense of w_i = {D(w_i)} with one sense of w_j = {D(w_j)}.***

Since the definition of a term may contain quite a few words, it is not uncommon that multiple pairs of the definitions of w_i and w_j have words in common. The assumption here is if the two terms are semantically related and have similar context, the gloss used to define those terms" senses can contain at least one same word as opposed to other words used by different senses. It

is assumed that words used describe the same idea or context can have common terms that can be found in common definition of their glosses. Even if there is no common word in the definition, the glosses also include examples so the common word can be described on the example too.

Method Two: ***One of its synonyms appears in the definition of the jth sense of w_j $D(w_i)$.***

The assumption here is if the synonym of the *ith* sense of *w* appears in ***$D(w_i)$.*** which means ***$D(w_i)$*** uses the *ith* sense of *w* on its definition.

Example: መማሪያ@ትምህርት ቤት:ተማሪዎች የሚማሩበት ትምህርት ቤት ሚማሩበት መጻህፍት;ርኅራሄ ሃዘኔታ ይቅር ማለት:ያጠፋ የበደለ ሰዉ ምህረት ይቅርታ እንዲደረግለት

ክፍል@የትምህርት ቤት ደረጃ: በትምህርት ቤት የሚገኙት እያንዳንዱ ክፍሌ;ቤት:በግድግዳ የተከፈለ አንድ ውስጥ አካል;መለያ ምዕራፍ:በመጻህፍት በፊልሞች ላይ መረጃዎች በአንቀጽ ወይም በምዕራፍ ሲከፋፈሌል

One of the synonyms of word መማሪያ is ትምህርት ቤት which also found on the definition of the word ክፍል.

When applying word sense disambiguation three kinds of answers are expected. The method might identify either the same sense for each ambiguous word, different senses or it might not identify at all. The new query that will be used for expansion is formed from the combination of these two methods used for disambiguation. The frequency of each sense calculated based on the one term's identified sense with the other term that helps to identify the sense. The frequency of identified sense calculated in each case. The new expanded queries formed by comparing and take the one with the highest frequency.

Let $Q=(w_1, w_2, \ldots\ldots w_i \ldots w_j \ldots wk,)$, the original query contains a number of words. Those are the words needed to be disambiguated if they are ambiguous words. Therefore, the words should be found from the WordNet having their own synset and gloss definition. If the word is ambiguous it has two or more senses.

The first word found on the given query with its multiple sense sets $W_i=\{(S_1,G_1), (S_2, G_2)\ldots.\}$

the second word found on the given query with its multiple sense sets $W_j=\{(S_1, G_1), (S_2, G_2)\ldots\}$

The synset of each word can have one or more term sets.

S={t_1, t_2…} the synset of the first sense with the terms (t1, t2= set of synonyms terms)

The gloss of each word is also a combination of words. It is always more than one word as long as it is the definition of the word.

G= {t_1, t_2, t_3……t_n} the gloss of the first sense with the (t_1, t_2..= terms that defines the synonyms (s) and (w))

The *S* and *G* are synset and gloss respectively. Each synset(S) and gloss(G) has terms of their set. During disambiguation the algorithm count the common terms found in each. It is the number of common words found given as the weight of that sense. The two methods used for disambiguation might identify two different senses for the same word, in such cases there is a need to calculate the frequency of the senses.

Therefore, as shown in equation 3.2 below to calculate the frequency, the weight of the selected sense divided by the number of terms found on the identified sense's synset or gloss (Wis) depends on the method. If it is synset to gloss method, it counts the number of terms on the synset and if it is gloss to gloss method, it counts the number of terms on the gloss) then multiplied (Wis) divided by the number of terms found on the word's sense of synset or gloss that helped to identify the sense.

$$fre = \left(\frac{weight}{Wis}\right) * \left(\frac{Wis}{Wjs}\right) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots 3.2$$

Where*:*

Weight= the weight of the chosen sense (number of terms that make them similar with the word)

W_is= the number of terms found on the identified sense's synset or gloss (synset if it is Method 2. Gloss if it is Method 1)

W_is= the number of terms found on the word's sense of synset or gloss that helped to identify the sense.

Steps for Word Sense Disambiguation

Step 1: Get the query word

Step 2: Disambiguate using gloss to gloss definition method

***Step 3*:** Calculate frequency of sense using gloss to gloss

Step 4: Disambiguate using synset to gloss method

Step 5: Calculate frequency of sense using synset to gloss

Step 6: Select the sense with the highest frequency

Step 7: Combine to formulate the new query

3.3 Query Expansion

Using the Word Sense disambiguation based on semantic similarity measure the sense or meaning of the term of the given query is determined. The next step is to use the identified sense for expansion. For query expansion the researcher uses three methods; synset expansion, gloss expansion and combination of the two methods expansion. The first two methods are adopted from the previous work [23].

Method one: Expansion using the gloss definition

Here it is the terms found from gloss definition which the researcher uses for expanding the query. Glosses are short definitions providing proper meanings of words and thus whole synsets. This method also used on previous work [23], however it was implemented with no text operation like stemming and stop-word removal.

Method two: Expansion using the sysnet

In this method the synsets of the term used to expand the query. The modified query is reformulated by using the terms found on the synset of the selected sense term. This method also used on the previous work.

Method three: Expansion using combination of the two methods

This method combine the above two methods. This means the modified query is formed from the synset and gloss of the identified sense.

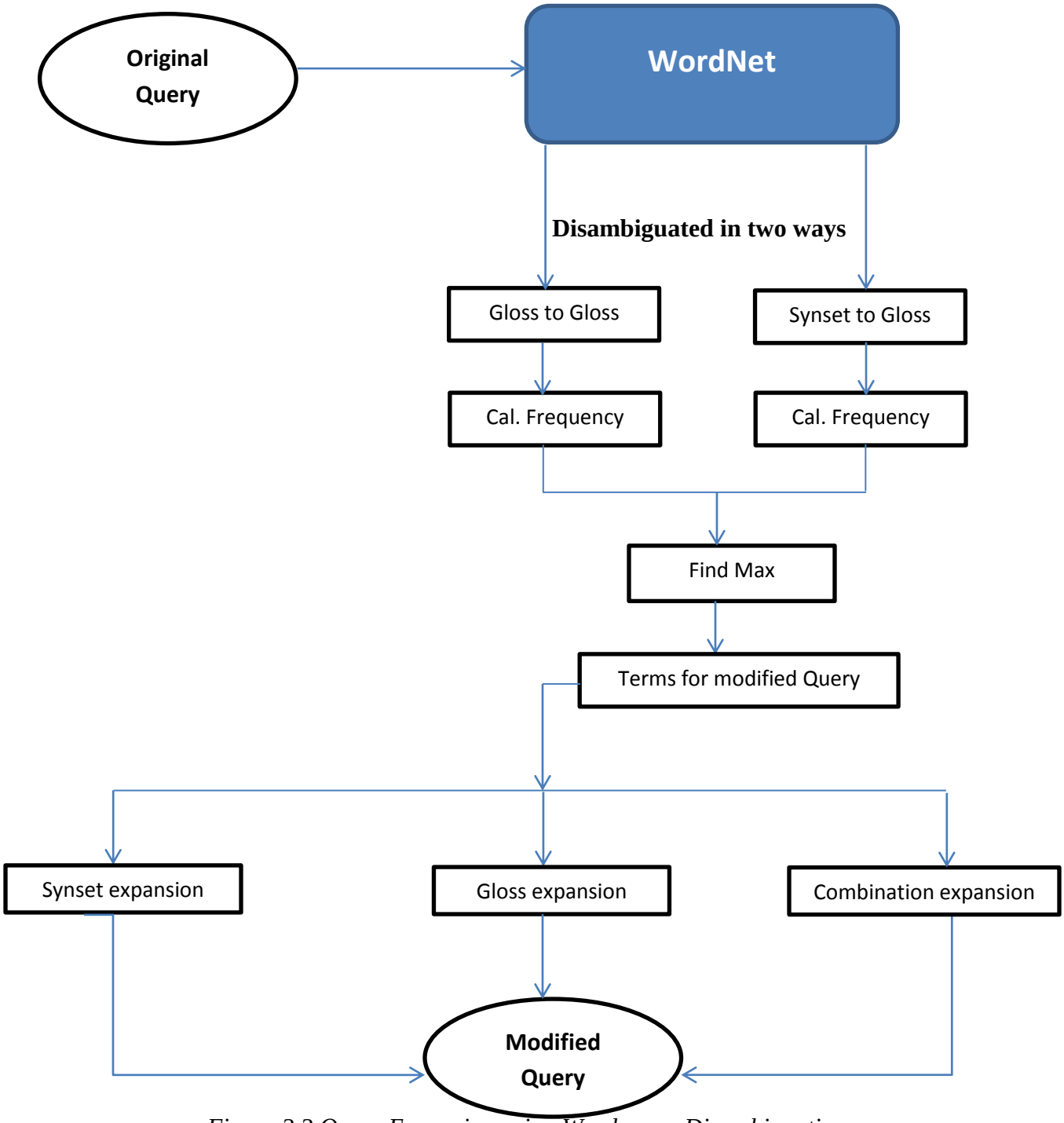

Figure 3.3 Query Expansion using Word sense Disambiguation

3.4 System Evaluation

The goal of IR system is retrieving relevant documents from the collection that satisfies user's information need to evaluate the performance of Amharic IR, the commonly used effectiveness measures, such as precision, recall and F-measure. In this study, the three widely used retrieval effectiveness measure such as precision, recall, and F- measure are used.

Recall

Recall is a measure of the ability of a system to present most relevant items that are available in the corpus.

$$Recall = \frac{number\ of\ relevant\ items\ retrieved}{Number\ of\ relevant\ items\ in\ collection} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 3.3$$

It is important to measure recall for circumstances where the searcher wants as much information on the topic as possible and therefore is interested in retrieving as many relevant results as possible. Recall on its own is not very useful, we need to compare it with the number of non-relevant documents by calculating precision [4].

Precision

Precision is a measure of the ability of a system to present only relevant items taking into account all retrieved documents.

$$Precision = \frac{number\ of\ relevant\ items\ retrieved}{Total\ number\ of\ items\ retrieved} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 3.4$$

IR systems aim to have high precision because this means that the majority of documents retrieved are relevant to the user needs. It should be noted recall and precision clearly trade off against one another. Precision usually decreases as the number of documents retrieved is increased. The ideal would be to achieve high recall and high precision. To identify the point where recall and precision are maximized, F-measure is recommended.

F-Measure

The F-measure is used to measure the performance of the system since it balances the precision and recall values. F-measure is a single measure that trades off precision versus recall. It is the weighted harmonic mean of precision and recall.

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 3.5$$

CHAPTER FOUR

Experimentation and Discussion

In this research an attempt has been made to design Word Sense Disambiguation using semantic similarity based query expansion to enhance the effectiveness on Amharic information retrieval. The system is developed with four basic components. Lexical resource like WordNet is the first one which is used as reference used as knowledge based to understand the meaning of concepts. Word Sense Disambiguation is the second one which is used to identify the word sense of the given query. The third one is Query reformulation which helped to expand the query with the identified sense using word sense disambiguation from the lexical resource. Finally, the query expansion module is integrated with Information Retrieval system to show its contribution to the enhancement of Amharic IR system performance.

4.1 Data Preparation

To test the prototype system developed, 300 Amharic News articles were used as a document corpus. The articles are taken from the previous work of Iman [23] which was orginally used by Amanuel [21] for designing probablistic information retrival. All news articles are taken from the web site of Walta Information Center. As shown in table 4.1, the news articles contain seven clusters of news, which are accident, health, education, sport, tourism, justice and politics.

No.	Types of news	No of documents
1	Accidents	40
2	Health	70
3	Education	40
4	Sport	30
5	Tourisim	40
6	Justice	40
7	Poletics	40
Total		300

Table 4.1: Types and sizes of news article used for experiment

In addition, as shown below table 4.2 the previous researchers selected 10 test queries to evaluate the performance of the system. These ten queries also are used in this research for conducting the experiment.

No.	List of Queries
1	የአደጋ ጊዜ እርዳታዎች
2	የኤችአይቪ ምርመራ
3	የመማሪያ ክፍል ግንባታ
4	ቅርሶች እንክብካቤና ጥበቃ
5	ጤና ጣቢያ ማስ�ፉያ ስራዎች
6	የእግርኳስ ስልጠና
7	ቴክኒክና ሙያ ማሰልጠኛ ተቋም
8	የሞትና የአካል ጉዳት አደጋ
9	የወባ በሽታ መከላከልና ቁጥጥር
10	ድርጅቶች የሚያደርጉት የልማት እንቅስቃሴ

Table 4.2 List of queries

4.2 Word Sense Disambiguation

4.2.1 Lexical Resource Preparation

Given a query containing multiple words, the aim of this research is to identify the meaning of each word in the context of other query words, so that to use those meanings or senses for expansion purpose. The senses of the query terms are prepared on the WordNet. The sample WordNet is constructed using the Amharic to Amharic dictionary [61] and Amharic context dictionary [62]. Figure 4.1 depicts sample Amharic Lexical resource like WordNet constructed in this study.

ምርመራ@ህክምና ማግኘት፡የታመመ ሰዉ በህክምና ቦታ ሲመረመር፡ፍተሻ ጥየቃ ፍለጋ፡ህግን የተላለፈ ድርጊት ሲፈጸም በፖሊስ ጣቢያ ወይም በፍርድቤት የሚደረግ ምርመራ

አደጋ@አደጋ ቸግር ሲቃይ ጉዳት፡በህመም ባልታሰበ አደጋ በህዘን በቸግር ምክንያት የሚመጣ በተፈጥሮ የሚከሰት እንደ ነርፍ መሬት መንቀጥቀጥ ጉዳት፡ተገቢ ያልሆነ ድርጊት፡ትክክል ተገቢ ተመጣጣ ባልሆነ ሁኔታ የተፈጸመ

እርዳታ@እርዳታ እገዛ አገልግሎት፡ለድንገተኛ ችግር ጉዳት ወይም አደጋ ሲኖር የሚደረግ እርዳታ በተለይ በህክምና፡እገዛ ማገዝ፡ስራ የበዛበትን ሰው ማገዝ፡ድጎማ፡በእቃ እና በቁሳቁስ መልክ የሚሰጥ እርዳታ

መማሪያ@መማሪያ መጽሃፍ ደብተር፡ትምህርት ለመማር የሚያስፈልጉ መሳሪያዎች መማሪያ መጽሃፍ፡መማሪያ ትምህርት ቤት፡ተማሪዎች የሚማሩበት የትምህርት ቦታ፡ርኅራሄ ሃዘኔታ ይቅር ማለት፡ያጠፉ የበደለ ሰዉ ምህረት ይቅርታ እንዲደረግለት

ክፍል@መማሪያ ክፍል ክላስ፡በትምህርት ቤት የሚገኙት እያንዳንዱ የመማሪያ ክፍሎች፡ክፍል ድርሻ ፋንታ፡የሚከፋፈል ነገር በዕድል መልክ ሲደርስ፡መለያ ምዕራፍ፡መጻህፍት በፊልሞች ላይ ያሉ መረጃዎች በአንቀጽ ወይም በምዕራፍ ሲከፋፈል

Figure 4.1 Sample WordNet with basic Words and their sense of meaning

The WordNet contains the term, synonyms term and the definition of the synonyms. The term before '@' is a reference term about which the different senses are given, for example, term "ምርመራ". The synsets are defined between '@' and ':' in this case the first synset for the word "ምርመራ" is "ህክምና ማግኘት" then followed with gloss definition which is "*የታመመ ሰዉ በህክምና ቦታ ሲመረመር*". If the given term has multiple senses, it is separated with ';' on the WordNet. The second sense for the term "ምርመራ" is "ፍተሻ ጥየቃ ፍለጋ፡ህግን የተላለፈ ድርጊት ሲፈጸም በፖሊስ ጣቢያ ወይም በፍርድቤት የሚደረግምርመራ. "ፍተሻ ጥየቃ ፍለጋ" is the second sense and the rest is the gloss definition. The format of the WordNet is presented in figure 4.2.

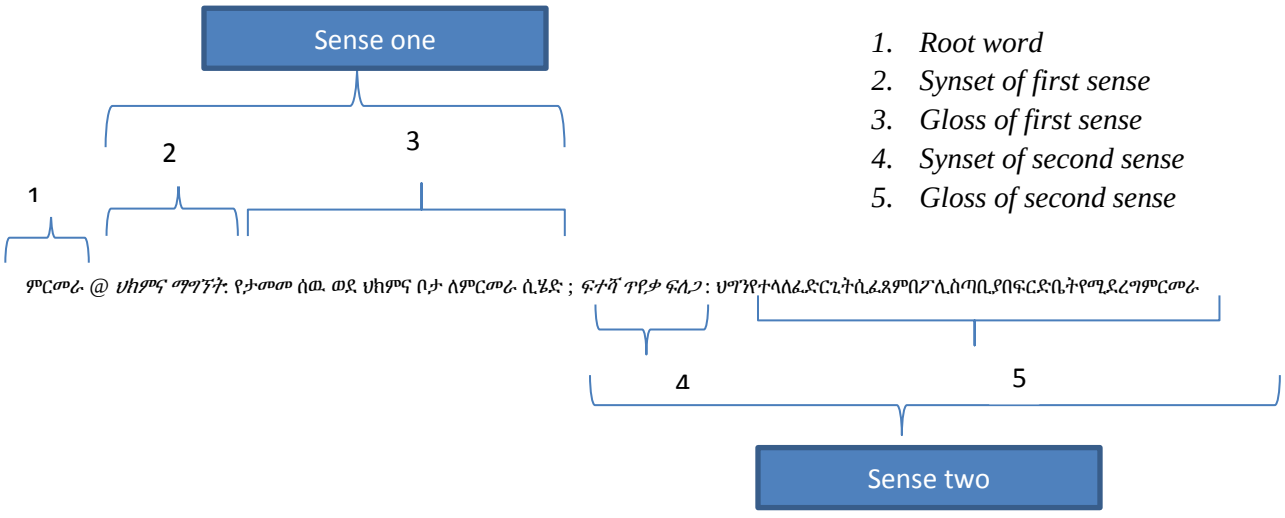

Figure 4.2 the format for term, synset, gloss and sense on WordNet

The design and construction of lexical resource like WordNet is labor intensive, time consuming and difficult. The sources for constructing the Amharic WordNet are one of the challenges of this study. The constructed WordNet contains only the synonymous and gloss definition information of the term.

4.2.2 Appling Semantic Similarity for Word Sense Disambiguation

To enhance the performance of the IR system this study integrates query expansion. The query expansion involves word sense disambiguation based on Amharic WordNet constructed in this study. There are several pieces of information associated with each content word and they can be used for word sense disambiguation. In this study the WordNet includes for each word, its synonymous and gloss definition with example. By comparing these pieces of information associated with the query terms, it may be possible to assign senses to these terms.

As discussed in *section 3.3*, there are two proposed methods for the selection of senses that are similar context for word sense disambiguation using Lesk algorithm semantic similarity measure. The main idea behind the algorithm is to disambiguate word sense by finding the overlap among their sense definition (gloss) in the WordNet.

For example let us consider the query *ለአደጋ ጊዜ እርዳታዎች*

For the word አደጋ the senses are:

1. **አደጋ ችግር ሰቃይ ጉዳት:**በህመም ባልታሰበ አደጋ በሀዘን በችግር ምክንያት የሚመጣ በተፈጥሮ የሚከሰት እንደ ጎርፍ መሬት መንቀጥቀጥ ጉዳት;
2. **ተገቢ ያልሆነ ድርጊት :**ትክክል ተገቢ ተመጣጣ ባልሆነ ሁኔታ የተፈጸመ

For the word ጊዜ the senses are:

1. **ጊዜ ዘመን አመት ወር ቀን ሰዓት ወቅት:** ዘመን አመት ወር ቀን ሰዓት ሲፈራረቅ
2. **ጊዜ:** ሳይመሸ ሳይጨልም ጸሃይ ሳለ እንግዳ በጊዜ መጣ ያለ ጊዜ እንደ ጊዜ

For the word እርዳታ the senses are:

1. **እርዳታ መስጠት ልገሳ እገዛ አገልግሎት**:ለድንገተኛ ችግር ጉዳት ወይም አደጋ ሲኖር የሚደረግ እርዳታ በተለይ በህክምና

2. **እርዳታ እገዛ ማገዝ :**ስራ የበዛበትን ሰው ማገዝ

There are three words from original query; each word has two senses with a total six comparisons required to identify similar sense. What the Lesk Algorithm does is compare each sense with each other. The two proposed ways are gloss to gloss which Lesk algorithm originally applied for, and synset to gloss. From the above example the one written with bold format is the synset and the rest is gloss definition of a term.

With the first proposed way (gloss to gloss), the first sense definition of the word '**አደጋ**' that is **"**በህመም ባልታሰበ አደጋ በሀዘን በችግር ምክንያት የሚመጣ በተፈጥሮ የሚከሰት ጉዳት**"** is compared with the two gloss definition of the term "**ጊዜ**" and the other two definition of the term "**እርዳታ**", that is compared with four of gloss definition in total. Then there it gets an overlap with gloss definition of the first sense of the word '**እርዳታ**' that is "ለድንገተኛ ችግር ጉዳት ወይም አደጋ ሲኖር የሚደረግ እርዳታ በተለይ በህክምና. The common words found in each sense definition are takes as the weight of the sense. In this example three words are found and the weight of the sense going to be 3. Then the second sense of the word '**አደጋ**' gloss definition which is **"**ትክክል ተገቢ ተመጣጣ ባልሆነ ሁኔታ የተፈጸመ**'** continued the same process. In this case there is no common word to be found for this sense; the weight for this sense is going to be 0. The one with the highest weight is taken as the identified sense that will be used for the expansion. The same process is done for each term of the query and the one with the highest weight is assigned as the sense of the word based on the given query context.

The second approach is finding an overlap between synset of one term with the other term's gloss definition. For example to find the words of the synset of the word **አደጋ** with the first sense that is "**አደጋ ችግር ስቃይ ጉዳት"** on the gloss definition of the terms **ጊዜ** and **እርዳታ**. The weights are assigned by counting the number of words found on the definition. The one with the highest weight is the one the identified sense to be used. From the above example the terms "**አደጋ ችግር ስቃይ ጉዳት"** can be found on the gloss definition of the word "**እርዳታ**" that is "በህመም ባልታሰበ አደጋ በሀዘን በችግር ምክንያት የሚመጣ በተፈጥሮ የሚከሰት እንደ ጎርፍ መሬት መንቀጥቀጥ ጉዳት**"**.

The other proposed method is (synset to synset) and the process is all the same with the first one except the way finding overlaps are between synsets of the term with the other synset of the other term. For example "**አደጋ ችግር ስቃይ ጉዳት"** is the synset of the first sense of the word **አደጋ.**

This synset is compared with each synset of the other terms of the query **ጊዜ** and **እርዳታ**. One of the challenges of this study is using this approach. It is very rare to find common synonymous terms of two words in the WordNet. Therefore, this synset to sysnet way of identifying sense is not giving good result. In this study it only works for one query, that is *''**መማሪያ ክፍል ግንባታ**''*.

The Lesk Algorithm is applied as explained above. The relevance judgment is prepared to show the right sense for each word. The ten queries formed in this study contains a total of 32 terms, out of which 18 of them are ambiguous terms and their correct sense is prepared in table 4.2 below.

	Words	No of Senses	The correct sense from WordNet
1	አደጋ	2	አደጋ ችግር ስቃይ ጉዳት
2	ጊዜ	2	ጊዜ ዘመን አመት ወር ቀን ሰዓት ወቅት
3	እርዳታ	2	እርዳታ መስጠት ልገሳ እገዛ አገልግሎት
4	መማሪያ	3	መማሪያ ትምህርት ቤት
5	ክፍል	3	መማሪያ ክፍል ክላስ
6	ምርመራ	2	ህክምና ማግኘት
7	ግንባታ	2	መስራት መገንባት
8	ቅርስ	2	የአገር ሃብት
9	ጤና	2	ጤና ጣቢያ
10	ጣቢያ	4	ጤና ጣቢያ ሆስፒታል የጤና ኬላ
11	በሽታ	2	መታመም መታወክ
12	ቁጥር	3	መከላከል
13	አካል	2	የሰውነት ክፍል
14	ጥበቃ	2	ክትትል ቁጥር
15	ስልጠና	2	ትምህርት
16	ሙያ	2	ችሎታ ስራ
17	ቴክኒክ	2	ቴክኒክ
18	እንቅስቃሴ	2	እንቅስቃሴ

Table 4.2 Ambiguous Words with their correct sense

Experiment result of the word sense disambiguation is presented in table 4.3. The first method which is gloss to gloss, finds the common words between the gloss definitions of the query terms and is applied to disambiguate 72.22% of the ambiguous terms. The 5 of them could not be identified because of the absence of common words with the other words in the query. Out of the 13 ambiguous terms 84.61% of them are correctly identified. And two terms; "*መማሪያ*" and "*ቴክኒክ*" disambiguated incorrectly.

The second method that is synset to gloss for identifying the sense is to find if the terms of synset of the word found on the gloss definition of the other word. The method applied to only 38.89% of the ambiguous terms. All of the 7 terms are disambiguated correctly. Even if the 11 of them has a multiple sense, this method is unable to identify them.

	Case 1	**Case 2**
Disambiguation terms	32	32
Ambiguous terms	18	18
Applicability	13 terms (72%)	7 terms (38%)
Correct terms	11	7
Accuracy	84%	100%

Table 4.3 Overall performance of the Word Sense Disambiguation

One of the terms disambiguated incorrectly using the first method is disambiguated correctly on the second method. The term "*መማሪያ*" has three sense" *መማሪያ መጽሃፍ ደብተር*", "*መማሪያ ትምህርት ቤት*", "*ምህረት ማድረጊያ*". While the first method "*መጽሃፍ ደብተር*" which is not the right one based on the context of the given query, the second one propose "*መማሪያ ትምህርት ቤት*" which is correct sense. This shows that a term that is not disambiguated with the first method may be disambiguated with the second one. At the same time there is a challenge, every gloss definition of each term's sense may not be defined with common word and every synonymous term may not be found on the gloss definition of the other term. Because of this, even the term with multiple senses may not be disambiguated using this semantic similarity measure at all with the two methods. This is because of the information associated to each term in WordNet is limited that includes only synonymous and gloss definitions.

4.3 Experiment on Query Expansion

The main aim of this study is to develop technique that adds more relevant search terms to the user's query for improving the retrieval results. To this end, this research attempts to identify the sense of the query term using semantic similarity for word sense disambiguation and expand the query term based on the identified sense. The Amharic Information retrieval system with WordNet based query expansion has been built using python version IDLE 3.1.

The very first step of the system is to get query from the user. Figure 4.3, presents a screen shot which shows the first list of retrieved document using first given query.

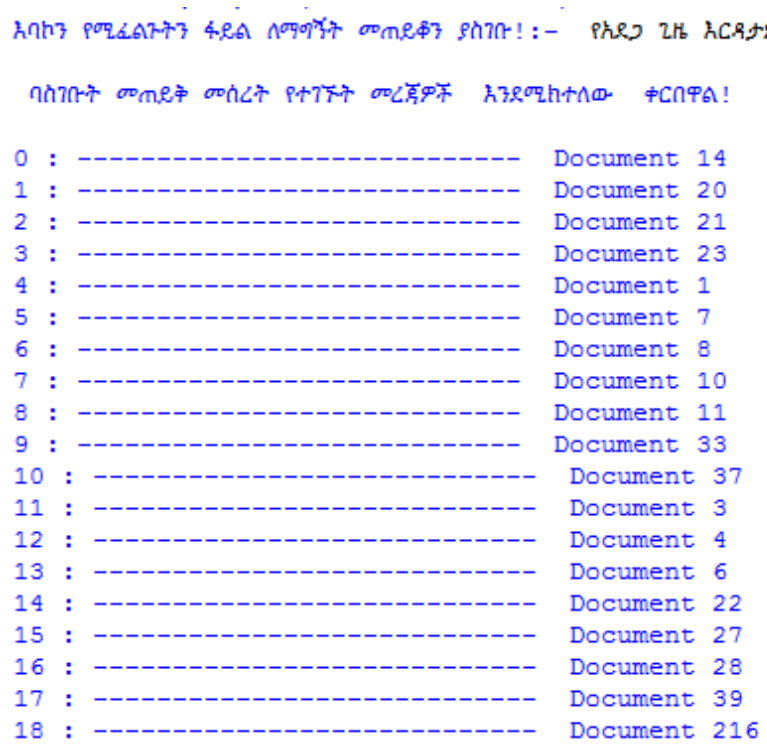

Figure 4.3: Retrieved documents for a given query 'የአደጋ ጊዜ እርዳታ'

The information retrieval system used in this study is developed based on the probabilistic retrieval model. The procedure of the system is that first documents are pre-processed for removing stop words, stemming and normalization in order to extract content bearing index terms. Then using probabilistic model a comparison is made between documents and query, calculating the weight of each query terms based on the notion implemented by probabilistic

model and finally computing the score of each document and ranks in decreasing order. The above figure shows the result of this process without applying query expansion. For the query ‘‘የአደጋ ጊዜ እርዳታ”, it retrieves 25 documents, out of which 14 of them are relevant (document numbers 14, 20, 21, 23, 7, 10, 11, 33, 3, 4, 22, 18, 30, 39), however, in the corpus there are 21 relevant documents for the query.

The modified query for expansion is formed from the terms found by combining the two methods; gloss to gloss and synset to gloss that is used to disambiguate the word. There are terms of original query. These terms have one or more senses. Each of the sense of the term is compared with each of the other senses of the original query term. When it compares either gloss to gloss or synset to gloss the target is to find the common words in between so as to identify the correct sense of that term based on the context given from the original query. The number of common words found to query terms is declared to be the score of the sense of the target word and assign as the weight of the term.

The new query is reformulated by adding terms with similar sense with each term in original query to form this new query the frequency of the identified sense with both methods compared and the one with the highest frequency is chosen to form the new query. The following figure 4.4 is a python code used to calculate the frequency of the sense.

```
            print ("your new query will be")
            aa=int(l3[xx][0][0])
            ab=int(l3[xx][0][1])
            ac=int(l3[xx][0][2])
            lens=len(sense[aa][ab][ac][1])
            aaa=int(l3[xx][1][0])
            aab=int(l3[xx][1][1])
            aac=int(l3[xx][1][2])
            leng=len(sense[aaa][aab][aac][1])

#            print (lens)
#            print (leng)

            nf1=(w[xx]/lens)*(lens/leng)          #calculating frequency

            case1.append(l3[xx])
            case1.append(nf1)
            expandedQ=expandedQ+sense[aa][ab][ac][1]
#            print (expandedQ)
            temp=l3[0][0]
            l=len(l3)
            for j in range(len(l3)):
                    for i in range(len(l3)-1):
                            if l3[i][0]==temp:
                                    del l3[i]
                                    break
```

Figure 4.4 python code used to calculate the frequency of the sense.

For example, in the query *"መማሪያ ክፍል ግንባታ"* the words *መማሪያ*, *ክፍል* and *ግንባታ"* has three, three and two senses respectively.

The senses of the word *መማሪያ* are

መጽሃፍ ደብተር: *ትምህርት ለመማር የሚያስፈልጉ መሳሪያዎች;*

መማሪያ ትምህርት ቤት: *ተማሪዎች የሚማሩበት የትምህርት ቦታ ክፍል;*

ርጎራሄ ሃዘኔታ ይቅር ማለት: *ያጠፋ የበደለ ሰዉ ምህረት ይቅርታ እንዲደረግለት*

And the sense of the word ክፍል

መማሪያ ክፍል ክላስ: *በትምህርት ቤት የሚገኙት እያንዳንዱ የመማሪያ ክፍሎች;*

ድርሻ ፋንታ: *የሚከፋፈል ነገር በዕድል መልክ ሲደርስ;*

መለያ ምዕራፍ: *መጻህፍት በፊልሞች ላይ ያሉ መረጃዎች በአንቀጽ ወይም በምዕራፍ ሲከፋፈል*

Using the first method gloss to gloss comparison the sense of *መማሪያ* identified as *መጽሃፍ ደብተር* with weight of 1. The frequency of this term is 16%.

Using the second method synset to gloss the sense identified as it means of *መማሪያ ትምህርት ቤት* with weight of 3. The frequency of this is 50%. So for query reformulation one with 50% frequency is chosen.

Frequency is calculated only if the two methods identify two different senses for the same term so that the one with the highest frequency is taken for the expansion. There are some cases like the sense can be identified on the first method but not on the second one and vise-versa. On such cases when the two methods combined to form the new query the one that is identified with one of the method is taken as it is.

Queries		
	Method 1	**Method 2**
የአደጋ ጊዜ እርዳታዎች	67%	33%
የኤችአይቪ ምርመራ	100%	50%
የመማሪያ ክፍል ግንባታ	67%	67%
ቅርሶች እንክብካቤና ጥበቃ	100%	0
ጤና ጣቢያ ማስፊፉያ ስራዎች	25%	0
የእግርኳስ ስልጠና	100%	50%
ቴክኒክና ሙያ ማሰልጠኛ ተቋም	100%	25%
የሞትና የአካል ጉዳት አደጋ	100%	75%
የወባ በሽታ መከላከልና ቁጥጥር	75%	25%
ድርጅቶች የሚያደርጉት የልማት እንቅስቃሴ	50%	0
Average	78%	33%

Table 4.4. The performance of the modified queries using the two methods

The above table 4.4 shows the percentage of how the original query changed to the modified query. For example for the query "የኤችአይቪ ምርመራ" the new query constructed by the first method is "ኤችአይቪ ምርመራ ኤድስ ህክምና ማግኘት". 'ኤድስ' the expansion term for the word 'ኤችአይቪ' and 'ህክምና ማግኘት' is for word 'ምርመራ' that gives 100% of expansion terms for the new query.

However, the second method for the same query it gives "ኤችአይቪ ምርመራ ህክምና ማግኘት" without identifying expansion term for the word 'ኤችአይቪ that gets 50% of expansion terms. The same for all query like 'የመማሪያ ክፍል ግንባታ' has 67% because the expansion terms are added for only the terms of 'መማሪያ' and 'ክፍል' no expansion terms for the word 'ግንባታ'. This is to show to what extent the modified queries are formulated.

Therefore, the combination of the two methods of disambiguation gives better modified query for query reformulation. The modified query then submitted to the probabilistic IR and gives new retrieval result.

For query expansion system three methods of expanding used, this means the modified queries reformulated in three ways. Expansion using the synset that means after the correct sense is identified and when it forms the modified query the one that is added to original query is only synset of the terms. For example query 'መማሪያ ክፍል ግንባታ' expanded to 'መማሪያ ክፍል ግንባታ ትምህርት ቤት መማሪያ ክላስ'. 'ትምህርት ቤት' is a synonymous of the word 'መማሪያ' and 'መማሪያ ክላስ' is for the word 'ክፍል'. The word 'ግንባታ' is one of the words that are not disambiguated using the two word sense disambiguation methods so that the system did not give any expansion term for this word. Figure 4.5 shows the modified query using synset

```
መጠይቅዎ ሲጠናከር የሚከተለዉን ይመስላል
['መማሪያ', 'ክፍል', 'ግንባታ', 'ምህር', 'ቤት', 'ክላስ']
----------
```

Figure 4.5 The modified query using synset

The second method is expansion using the gloss definitions. Unlike the first method of expansion here the terms concatenate with the original query is the terms found on the gloss definitions. For example the new query for 'መማሪያ ክፍል ግንባታ' using this method 'መማሪያ ክፍል ግንባታ ተማሪዎች የሚማሩበት የትምህርት ቦታ በትምህርት ቤት የሚገኙት እያንዳንዱ የመማሪያ ክፍሎች'. This 'ተማሪዎች የሚማሩበት የትምህርት ቦታ' is a gloss definition of the word 'መማሪያ' and the synonymous word of 'ትምህርት ቤት'. And the gloss definition for the word 'ክፍል' and its synset 'መማሪያ ክላስ' is 'በትምህርት ቤት የሚገኙት እያንዳንዱ የመማሪያ ክፍሎች'. Figure 4.6 shows the modified query using gloss

```
መጠይቅዎ ሲጠናከር የሚከተለውን ይመስላል
['መማሪያ', 'ክፍል', 'ግንባታ', 'ተማሪ', 'ሚማሩበ', 'ምህር', 'ቦታ', 'ቤት', 'ሚገኙ', 'ንዳንዱ', 'ክፍሎ']
----------
```

Figure 4.6 The modified query using gloss

The third method is the combination of the two methods. This means the modified query composed of the synset and gloss definition of the identified sense of the query terms. For example query '*መማሪያ* ክፍል ግንባታ' expanded to '*መማሪያ* ክፍል ግንባታ ትምህርት ቤት *መማሪያ* ክላስ' using synset and '*መማሪያ* ክፍል ግንባታ ተማሪዎች የሚማሩበት የትምህርት ቦታ በትምህርት ቤት የሚገኙት እያንዳንዱ የ*መማሪያ* ክፍሎች' using gloss. Therefore, using this method the modiefd query is '*መማሪያ* ክፍል ግንባታ ትምህርት ቤት *መማሪያ* ክላስ *መማሪያ* ክፍል ግንባታ ተማሪዎች የሚማሩበት የትምህርት ቦታ በትምህርት ቤት የሚገኙት እያንዳንዱ የ*መማሪያ* ክፍሎች'. The repeated words are deleted, rearrange and reformulate the query. The following figure 4.7 shows the result of this method.

```
መጠይቅዎ ሲጠናከር የሚከተለውን ይመስላል
['መማሪያ', 'ክፍል', 'ግንባታ', 'ምህር', 'ቤት', 'ተማሪ', 'ሚማሩበ', 'ቦታ', 'ክላስ', 'ሚገኙ', 'ንዳንዱ', 'ክፍሎ']
----------
```

Figure 4.7 the modified query using the combination method

4.3 Performance Evaluation of Amharic IR system

Precision, Recall and F-measure are the most frequent and basic statistical measures which are widely used to assess the effectiveness of IR system. Ten original and expanded Amharic test queries are used for experimentation. The relevance judgment is prepared to construct document query matrix that shows all relevant documents for each test queries. The performance of the system is evaluated before expansion and after expansion by using synset expansion, gloss expansion and combined expansion.

Table 4.5 shows total relevant documents available in the Amharic corpus for the ten test queries used to evaluate the performance of the system.

Q. No.	List of queries	List of relevant documents for the query
1	የአደጋ ጊዜ እርዳታዎች	3, 4, 7, 9, 10, 11, 13, 14, 15, 17, 18, 20, 21, 22, 23, 25, 30, 31, 33, 34, 39
2	የኤችአይቪ ምርመራ	243, 249, 250, 262, 267, 268, 276, 293, 291, 298
3	የመማሪያ ክፍል ግንባታ	191, 194, 197, 198, 199, 201, 203, 205, 216, 208, 221, 222, 225, 228, 229
4	ቅርሶች እንክብካቤና ጥበቃ	41, 42, 45, 49, 50, 51, 52, 53, 55, 57, 62, 64, 71, 75, 76, 78,79
5	ጤና ጣቢያ ማስፊፉያ ስራዎች	233, 236, 239, 247, 248, 255, 256, 263, 264, 271, 279, 284, 297
6	የእግርኳስ ስልጠና	162, 164, 167, 168, 173, 174, 177
7	ቴክኒክና ሙያ ማሰልጠኛ ተቋም	191, 201, 210, 208, 213,
8	የሞትና የአካል ጉዳት አደጋ	1, 8, 19, 26, 27, 28, 35, 36, 38
9	የወባ በሽታ መከላከልና ቁጥጥር	4, 11, 238, 239, 240, 241, 250, 251, 258, 260, 267, 266, 275, 286, 287, 288, 296
10	ድርጅቶች የሚያደርጉት የልማት እንቅስቃሴ	123, 129, 130, 137, 143, 149, 148

Table 4.5. Test query with relevant document list

The IR model used in this work is probabilistic IR. Based on this system the initial result without expansion for those ten queries are presented in table 4.6 below

Query	corpus	Retrieved	Relevant	R	P	F
የአደጋ ጊዜ እርዳታዎች	21	26	14	0.67	0.54	0.60
የኤችአይቪ ምርመራ	10	17	10	1	0.59	0.74
የመማሪያ ክፍል ግንባታ	15	30	8	0.53	0.27	0.36
ቅርሶች እንክብካቤና ጥበቃ	17	25	14	0.82	0.56	0.67
ጤና ጣቢያ ማስፊፉያ ስራዎች	13	24	8	0.61	0.33	0.43
የእግርኳስ ስልጠና	7	4	4	0.57	1	0.73
ቴክኒክና ሙያ ማሰልጠኛ ተቋም	5	22	4	0.8	0.18	0.30
የሞትና የአካል ጉዳት አደጋ	9	29	5	0.56	0.17	0.26
የወባ በሽታ መከላከልና ቁጥጥር	18	37	17	0.94	0.46	0.61
ድርጅቶች የሚያደርጉት የልማት እንቅስቃሴ	7	49	5	0.71	0.10	0.19
			total	**0.72**	**0.42**	**0.53**

Table 4.6 Initial retrieved result with before query expansion

As it is observed from table 4.5, the average result of precision and recall of the system using the initial guess made by the model about the relevance of documents are 42% and 72% respectively.

This shows that the percentage of recall dominates the percentage of precision by 30%. Finally, the F-measure, score is 53%, which indicates the performance of the system is not satisfactory.

The result depicts the system retrieved most of the relevant documents in the collection out of the total relevant documents in the corpus. However, the result of the precision indicates that, the non-relevant documents retrieved are higher than the relevant documents retrieved. This is because documents containing one of query terms but not-relevant are retrieved. Documents are irrelevant because, the query term found in those documents not express the meaning of the query with respect to other terms found in the query. For example, for query "የአደጋ ጊዜ እርዳታዎች" which express the aid given at accidental time, the system retrieved irrelevant documents such as "doc216", "doc245","doc254" and "doc253" because they contains query term "አደጋ". However, in these documents the term "አደጋ" is used to expresses different accidental cases which are not related with aid. The same problem is also revealed for other queries just because it contains the query term. On the other hand some documents are relevant for the query but the document may not contain the query term on the document. Because of this the relevant documents are also not retrieved. For example "የመማሪያ ክፍል ግንባታ", "doc194" is relevant document which is not retrieved because it expressed by synonym words "የትምህርት ቤት ማስፊፊያ".

Therefore, in order to enhance the performance of the IR system, the query expansion by applying Word Sense Disambiguation using semantic similarity measure for identifying the sense from the lexical resource which is WordNet. To identify the sense of a word, two method are used, synset to gloss and gloss to gloss. And to expand three methods used, synset, gloss and combined.

Experiment 1: Experimental result using synset for query expansion

After expansion to get the maximum optimal performance, a recursive testing has been made to fix threshold. In order to get the optimal threshold four experiments are done for each expansion methods by increasing the weight of the terms. The need of optimal threshold is to limit the retrieved result, because the gloss definition has few words and if all those words used when it expand the query every term appear on the query term will be retrieved even if it is not relevant. This will increase the number of irrelevant numbers on the retrieved result and this reduces precision of the system.

Table 4.7 shows the experiments on finding the optimal threshold. The experiment starts with weight greater than 0(>0) which gives 17%F-measure, 30% f-measure with weight greater than one (>1), then weight greater than two (>2) gives a better result with 59% F-measure. But further test with weight greater than three (>3) gives 35% F-measure which starts decreasing. So weights greater than two (>2) is selected as a threshold.

Thresholds		**Q 1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Avg**
>0	Corpus	21	10	15	17	13	7	5	9	18	7	
	retrieved	159	61	213	74	195	61	183	119	88	154	
	relevant	21	10	11	12	12	7	4	7	17	7	
	R	1	1	0.8	0.65	0.92	1	0.8	0.78	0.94	1	**0.88**
	P	0.13	0.16	0.06	0.14	0.06	0.11	0.02	0.06	0.19	0.04	**0.09**
	F	0.23	0.29	0.10	0.24	0.11	0.20	0.04	0.10	0.32	0.09	**0.17**
>1	Corpus	21	10	15	17	13	7	5	9	18	7	
	retrieved	57	61	40	56	91	27	32	58	48	73	
	relevant	17	10	12	11	10	7	4	7	17	4	
	R	0.80	1	0.8	0.64	0.76	1	0.8	0.78	0.94	0.58	**0.81**
	P	0.30	0.16	0.3	0.20	0.10	0.26	0.12	0.12	0.35	0.05	**0.20**
	F	0.44	0.28	0.44	0.30	0.19	0.41	0.21	0.20	0.51	0.1	***0.30***
>2	Corpus	21	10	15	17	13	7	5	9	18	7	
	retrieved	22	14	9	15	33	5	11	27	17	21	
	relevant	13	8	9	5	12	4	4	9	14	3	
	R	0.62	0.8	0.6	0.30	0.92	0.57	0.8	1	0.78	0.42	**0.68**
	P	0.60	0.58	1	0.33	0.36	0.8	0.36	0.33	0.82	0.14	**0.53**
	F	0.60	0.67	0.75	0.31	0.52	0.67	0.5	0.5	0.8	0.21	***0.59***
>3	Corpus	21	10	15	17	13	7	5	9	18	7	
	retrieved	4	1	1	6	22	3	7	13	2	0	
	relevant	4	1	1	5	8	3	4	6	2	0	
	R	0.19	0.1	0.06	0.30	0.61	0.42	0.8	0.67	0.11		**0.32**
	P	1	1	1	0.83	0.37	1	0.58	0.46	1		**0.72**
	F	032	0.18	0.16	0.43	0.46	0.6	0.67	0.54	0.2		***0.35***

Table 4.7 experiment on finding the optimal threshold using ten queries (Q1-Q10)

Using the first method that is expansion using *synset*, after identifying the sense for each query term, the synset of each term concatenate together and form a modified query. This can help to

find the relevant document with the synonyms word of that term even if the given term is not found on the document. Based on this approach the researcher found the result depicted in table 4.6 with 2 optimal threshold.

As can be seen from Table 4.6, after expansion using synset method, the average percentage of precision is increased from 42% to 53%, recall is decreased from 72% to 68%. Thus, the performance of the system is improved from 53% to 59% F-measure.

The overall performance of the system increased when it is compared with before expansion (see table 4.6), however, from the above result it can be observed that the Recall is always high and the Precision is very low compared to Recall. That means the retrieval of non-relevant documents is very high. This shows that most of the words found on the synset are polysemy that means a multiple meanings of words. This leads to retrieval of irrelevant document, because the documents contain a word which similar in shape and different in meaning. The IR model also has its own effect on the performance of the system. The IR model used for this study is probabilistic model, which uses a binary weighting technique. Even if it applies term reweighting, the use of binary weight biased the system to assign equal importance ignoring frequency of occurrence of words in a document

One of the problem found on previous work of Amanuel [21] is the problem of documents having synonym terms of query word. For instance, for the query "የመማሪያ ክፍል ግንባታ" relevant document '*doc194*' is not retrieved at first because it is expressed by synonym words of query terms called "የትምህርት ቤት ማስፊፉያ". Because of identifying the sense correctly and use the other terms found on the synset overcome such kind of problems. For the query term "መማሪያ" and "ግንባታ" the synonym word is "የትምህርት ቤት" and "ማስፊፉያ" respectively. This shows that the synonymous is controlled.

On the other hand there are also problems. Even if the sense of a term is identified some terms have no synonymous terms at all but can have different sense. On such cases those senses have the same word for each or one of the sense can have the same word with the one that need synonymous word, which means the term can be used as it is for those different senses. For example in this study the word 'ቴክኒክ' has two senses. The first sense has a synonymous term 'ዘዴ' and the second one is 'ቴክኒክ' by itself because there is no synonymous word for this sense.

Therefore expansion using synets may not make a big difference on information retrieval. However to solve this kind of gaps expansion using gloss definition is used. Such that the term ‘ቱክኒክ’ is further expanded using ‘ከመካኒክ ከእንጨት ስራ ጋር የተያያዘ የሙያ ስራ’

Experiment 2: Experimental result using gloss definition for query expansion

The second method experimented for query expansion is using the gloss definition. For this method also it is needed to fix the threshold. Experimental result shows that there is continues increase in F-Measure, 21%, 36%, and 47% for the weights >1, >2, and >3 respectively. However at the forth level the performance decreases since 0 relevant documents retrieved for some queries. While the threshold value >3 give a better performance, the value >2 takes to compare it with the original performance and with the first method.

Table 4.8 shows the experimental result of query expansion using gloss definition.

Thresholds		**Q 1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Avg**
>1	*Corpus*	21	10	15	17	13	7	5	9	18	7	
	retrieved	147	57	62	62	57	167	108	95	108	98	
	relevant	21	10	15	11	8	5	6	17	5	4	
	R	1	1	1	0.64	0.61	1	0.66	1	0.71	0.61	0.84
	P	0.14	0.17	0.24	0.17	0.14	0.02	0.05	0.17	0.04	0.05	0.12
	F	0.24	0.29	0.38	0.42	0.22	0.03	0.09	0.29	0.29	0.07	0.21
>2	*Corpus*	21	10	15	17	13	7	5	9	18	7	
	retrieved	63	20	19	22	42	20	67	46	33	29	
	relevant	21	7	10	10	5	4	5	6	16	2	
	R	1	0.7	0.67	0.59	0.38	0.57	1	0.67	0.89	0.29	**0.68**
	P	0.33	0.35	0.53	0.45	0.11	0.2	0.07	0.13	0.48	0.07	**0.27**
	F	0.5	0.47	0.59	0.51	0.18	0.29	0.13	0.21	0.62	0.11	**0.36**
>3	*Corpus*	21	10	15	17	13	7	5	9	18	7	
	retrieved	38	10	12	11	19	8	31	17	22	11	
	relevant	13	3	10	7	8	3	5	6	13	1	
	R	0.62	0.3	0.67	0.41	0.62	0.42	1	0.67	0.72	0.14	**0.57**
	P	0.34	0.3	0.83	0.64	0.42	0.38	0.16	0.35	0.59	0.09	**0.41**
	F	0.44	0.3	0.74	0.5	0.5	0.4	0.28	0.46	0.65	0.11	**0.47**

Table 4.8 experiment on finding the optimal threshold for gloss using ten queries (Q1-Q10)

When compared with the initial performance (see table 4.6) as well as with the first experiment (see table 4.7), the overall performance decreased to 36%. The reason for decreasing the performance is because of the large number of terms used for modifying the original query.

It is possible that the query expansion process generate a large number of terms that it might not be practical to use all of those terms. The removal of stop-words are applied and gloss definition contain a number of terms as it is the expression (definition) of the word, however, still there are unimportant terms used for expansion. This leads to retrieve many irrelevant documents. To overcome this problem there is a need to identify the most important words found on the gloss definition. This method need further research on discriminating the most important words related to users query.

Experiment 3: Experimental result using combined approach for query expansion

The third method experimented for query expansion is using the combination of gloss and synset expansion. For this method also it is needed to fix the threshold. Experimental result shows that in F-Measure, 33%, 42%, and 51% for the weights >2, >3, and >4 respectively. However at the forth level the performance decreases since 0 relevant documents retrieved for some queries. As it is observed from the result table 4.9 below, this method registered 33% F-score. In this method the query is formed by merging the two methods. The challenges faced for the two experiments also have an impact on this experiment. If the challenges fixed for the two experiments this experiment could give a better performance.

Thresholds		**Q 1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Avg**
>2	*Corpus*	21	10	15	17	13	7	5	9	18	7	
	retrieved	62	34	19	23	74	20	74	50	36	33	
	relevant	18	7	10	10	12	4	5	6	14	2	
	R	0.86	0.7	0.67	0.59	0.92	0.57	1	0.67	0.78	0.28	**0.70**
	P	0.29	0.20	0.52	0.43	0.16	0.2	0.06	0.12	0.39	0.06	**0.25**
	F	0.43	0.31	0.59	0.5	0.27	0.29	0.12	0.20	0.57	0.1	**0.33**
>3	*Corpus*	21	10	15	17	13	7	5	9	18	7	
	retrieved	33	14	12	12	37	8	36	21	22	13	
	relevant	12	6	10	7	11	3	5	5	11	1	
	R	0.57	0.6	0.67	0.41	0.84	0.42	1	0.56	0.61	0.14	**0.58**
	P	0.36	0.42	0.83	0.58	0.30	0.37	0.13	0.23	0.5	0.07	**0.38**
	F	0.44	0.5	0.74	0.48	0.44	0.4	0.24	0.33	0.55	0.1	**0.42**
>4	*Corpus*	21	10	15	17	13	7	5	9	18	7	
	retrieved	15	5	6	6	22	3	15	11	11	1	
	relevant	8	2	6	5	8	3	4	3	11	1	
	R	0.38	0.2	0.4	0.30	0.62	0.42	0.8	0.33	0.61	0.14	**0.42**
	P	0.53	0.4	1	0.83	0.36	1	0.27	0.27	1	1	**0.67**
	F	0.44	0.27	0.57	0.43	0.46	0.6	0.4	0.3	0.78	0.25	**0.51**

Table 4.9 experiment on finding the optimal threshold for combined expansion using ten queries (Q1-Q10)

4.5 Finding and Challenges

The obtained result indicates using word sense disambiguation on WordNet using sematic similarity to identify the word sense for the given query register encouraging performance. This shows the need of standard Amharic WordNet is much useful for determining the correct sense and to use the information found on these senses for query expansion. The use of query expansion in information retrieval using the first method gives a good result when it compares to the system without the expansion and the two methods decreases the performance of the IR system as shown in table 4.10.

Measures	Original query	Synset expansion	Gloss expansion	Combined Expansion
Recall	0.72	0.68	0.68	0.70
precision	0.42	0.53	0.27	0.25
F-Measure	***0.53***	***0.59***	***0.36***	***0.33***

Table 4.10 Summarized result of the overall performance

Table 4.9 presented the overall performance of the designed performance. From the analysis, WSD based query expansion has achieved low recall, high precision and F-measure when compared the synset expansion methods with the original query. The goal of the system is to retrieve relevant documents as much as possible. Even if the recall decreased the precision is increased, this shows the system decrease the retrieval of irrelevant documents to some extent. And the F-measure balanced the results of both by registering a good performance.

In other hand, with the second method the result shows same recall and decrement in precision and F-measure result when it compares with the second expansion method. This method drops the overall performance. There are a large number of expansion terms in gloss definition when compared with the synset. For this method there is a need of selecting the most important terms instead of using all terms for expansion.

The need of third method is to see if the combination of both methods gives a good result. The method increases the recall but decrease the overall performance. In this method the two methods merged to form the new query, however instead of merging there is a need of choosing the best in each method and combine them for a better result.

In general, this study shows an effective use of Word Sense Disambiguation using semantic similarity for identifying the sense and to form the new query. The algorithm used for semantic similarity performs well for the second method which is synset to gloss comparison. This shows the Lesk algorithm can be applied using different information associated with the term/word from the WordNet. For query expansion the technique using synset expansion registered a better performance when it comes to the overall performance, this study increases in 6% F-score.

In general when this study compared with previous work of Iman [23], from the two approaches she used to determine the sense the use of word sense disambiguation is a failed approach. Therefore, she used the vector space model for identifying the sense. This study shows a ways of an effective use of Word Sense Disambiguation using semantic similarity for identifying the sense and to form the new query.

When it come to the overall performance as shown from table 4.12 this study increases in 6% F-measure expansion with gloss definition and using expansion synsets remain the same, But there is an increase in recall.

Works	***Methods***	***R***	***P***	***F***
Pervious Work	*Synset*	0.55	0.76	***0.59***
	Gloss	0.37	0.28	***0.30***
This study	*Synset*	0.68	0.53	***0.59***
	Gloss	0.68	0.27	***0.36***

Table 4.12 Comparison of this work with previous work

However, there are several challenges faced which limits to register the optimum performance expected from the model in order to outperform the entire Information Retrieval system developed for Amharic language. The main problem is standardization of every component used for designing the system.

There is no any developed standard WordNet for Amharic language. Lack of resources is the main challenge for constructing the lexical resource in WordNet form. Constructing a WordNet with a single person is difficult, especially when there are no enough resources. Even if this research attempts to show the possible use of WordNet and its information associated with each term for Query expansion in IR system, there is a need of constructing WordNet by including different information associated with the given term.

In addition to this the way phrases writing also make a difference on the performance. Example if the query is “እግርኳስ ስልጠና” and “እግር ኳስ ስልጠና” the documents that contain like “እግርኳስ” can be retrieved but not the documents that contains “እግር እጅ እና መረብ ኳስ ስልጠና” with the first query, however, with the second query that is “እግር እጅ እና መረብ ኳስ ስልጠና”, can be retrieved but not

documents with “እግርኳስ”. Very common phrases in the language should be considered when WordNet is constructed and at the same time the normalization should be implemented effectively.

Finding standard corpus and query for Amharic language is also another challenge in this study. This result not only weakens the performance of the system but also makes it difficult to compare the result obtained with several researches since there is different in test queries, document content and size used for testing. Even if this study uses the same corpus and queries taken from the previous work for the purpose of evaluating the performance, there is still a challenge of on using information retrieval system, because there is no standard probabilistic information retrieval system to be used for every research.

CHAPTER FIVE

Conclusion and Recommendations

5.1 Conclusion

It is obvious that the main goal of information retrieval system is to retrieve relevant information. Information space on the web is comparatively larger and combined with the ambiguity of the Amharic language, a long list of results is returned, much of which is not always relevant to the user's information need. Short queries, ambiguity of natural language and the vocabulary mismatch are the main problems in information retrieval system [23].

To increase the number of relevant documents retrieved queries need to be disambiguated by looking at their context. The search engines attempt to determine the context of the user query and allow the user to obtain more meaningful results. In other words these search engines are focusing more on achieving high precision. The most recent query expansion technique involves the use of lexical resources like ontology/WordNet to infer context for ambiguous queries. The concepts in the lexical resource like WordNet can be used for word sense disambiguation using semantic similarity and subsequent query expansion [3].

This research investigates the effectiveness of word sense disambiguation using semantic similarity to identify the correct sense from lexical resource like WordNet and to use it in query expansion techniques for Amharic Language. To this end, query expansion is designed in order to enhance the retrieval performance of Amharic IR system. For query expansion word sense disambiguation is performed using Amharic WordNet by applying semantic similarity measure in this study.

The first objective of this study is to prepare Amharic lexical resource built from two dictionaries available for Amharic language that has a format of WordNet. The words used in the WordNet are limited to include the terms used for the prepared queries. The information associated with each terms in the WordNet also limited to two information only. It contains the synset, synonymous terms of phrases of the word and the gloss definition of the word.

The second objective is to choose a suitable semantic similarity technique for Word Sense Disambiguation to identify the correct sense from the WordNet. The Lesk Algorithm is chosen based on the information constructed in the WordNet. The original Lesk algorithm is applied on gloss to gloss, but in this study the idea of the algorithm is extended to be used on other approaches too.

The third objective is to design and conduct experiment to see the performance of the system. The study provided two semantic similarity approaches that use the idea of Lesk algorithm to identify the sense from WordNet using word sense disambiguation in short queries and demonstrated that is applied 72% and 38% to disambiguate terms using synset and gloss definition, respectively.

The query reformulation is constructed by combining the sense's identified using the two methods used for word sense disambiguation. Three experiments are carried out to test the performance of the IR system by expanding the queries using terms found from synset, gloss and the combination of synset and gloss. System evaluation has been done to discover the extent to which the designed system enhances the performance of Amharic IR system based on the F-measure.

The final objective of the study is to summarize the findings of this research and recommend areas for future work. As the experimental result show, gloss definition based expansion register 36%. The problem is many unimportant documents are retrieved due to the large set of expansion terms used for expansion. The second method using synset for query expansion register performance of 59% F-measure. This method registered an improvement of 6% from original query. This shows the possibility of controlling the effect of synonymous on IR performance. However, regarding polysemous it needs further research to control the effect. In this study, a promising result is registered to design an applicable IR system for Amharic language considering polysemous and synonymous nature of Amharic words with the help of Ontology. The third method using combined expansion register performance of 33% F-measure. This shows the method depends on the performance of the two methods. For this method instead of merging it can be used by selecting the best in each method ad combine them.

The use of lexical resource as a reference is the core of this study. The word sense disambiguation and query expansion process is based on this lexical resource. The main challenge of the study is on constructing the lexical resource like WordNet. The number of information associated to each terms is limited because of the lack of resource. Therefore, the use of similarity measure and the use of query expansion terms are limited based on the information available on the WordNet.

5.2 Recommendation

Query expansion using lexical resource like WordNet has been successful to a certain extent for English language, but for our local language the query expansion, WordNet and Word sense disambiguation are new area of research. Even the information retrieval idea needs more research to improve the techniques for selecting and designing algorithms for better performance.

The first goal in integrating query expansion with information retrieval is to improve the retrieval system in many ways. This study showed a promising performance if the following observed points taken into consideration for future work on enhancing the information retrieval effectiveness for Amharic language.

- In this work the WordNet included only a set of synonyms with each set of synonyms representing a meaning of the word. It also has a definition associated with each set of synonyms terms. On the other hand it lacks hyponym synset. A hyponym is a set of words or phrases which have the same meaning but are narrower than the given word in a specific sense. This can control the polysemous effects on the effectiveness of the system. Hence, there is a need to construct a standard Amharic WordNet
- This study used Lesk algorithm for similarity which only focuses on co-occurrence of multiple words in gloss definition. The idea of this algorithm is extended and applied on one more approach in this study. However, there is a need to come up with a well-constructed WordNet with more information available for each term that can help for applying the similarity measures in identifying the sense of synonymous and polysomous query terms in different ways by applying the Lesk algorithm
- Expansion using gloss definition needs further experiments. The method does not discriminate the significance of a term it uses to expand the query. Methods to

discriminate words used for expansion like calculating weights for each term can immensely increase the performance of the IR system.

- Finding a standard corpus, test queries with relevance judgment, standard IR system with a better stemmer for testing the designed system is one of the challenges faced in this research. Therefore, future research need to consider the development of standard Amharic corpus, test queries and IR system that can be used by every researcher to evaluate progress made in designing techniques for enhancing effects of Amharic IR system.
- This system tested on Probabilistic model which makes 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. Hence, there is a need to build hybrid system that uses vector space model to guess relevant documents for user query using non-binary weighting technique.

Reference

[1] A. Andreou, "Ontologies and Query expansion," M.S. thesis, School of Informatics, University of Edinburgh, South Bridge, UK, 2005.

[2] D. Katta, "A study of relevance feedback in vector space model," MSc, thesis, Computer Scince, Howard R. Hughes College of Engineering, University of Nevada, Las Vegas,USA 2009.

[3] J. Bhogal, "Investigating ontology based query expansion using a probabilistic retrieval model," City University London, 2011.

[4] R. Baeza-Yates and B. Ribeiro-Neto, Modern information retrieval. ACM press New York, 1999.

[5] D. Hiemstra, "Information retrieval models," in Information Retrieval, Searching in the 21st Century, 2009, pp. 1-19

[6] J. Xu and W. B. Croft, "Improving the effectiveness of information retrieval with local context analysis," ACM Transactions on Information Systems (TOIS), vol. 18, pp. 79-112, 2000.

[7] B.Billerbeck “Efficient Query Expansion” PhD, School of Computer Science and Information Technology, RMIT University, Melbourne, Victoria, Australia, 2005.

[8] D.Wollersheim, “Dynamic Query Expansion for Information Retrieval of Imprecise Medical Queries,” Ph.D, Dept. Comp. Sci. and Comp. Eng., La Trobe Univ., Bundoora, Victoria 3086 Australia, 2005

[9] D. Wollersheim and J. W. Rahayu. (2005, Mar.) "Ontology based query expansion framework for use in medical information systems," International Journal of Web Information Systems, vol. 1, pp. 101-115.

[10] J. Bhogal, A.Macfarlane and P.Smith. "A review of ontology based query expansion," Information Processing & Management, vol. 43, pp. 866-886, 2007.

[11] S. S. Eldin and A. Elsayed. (2012, Aug.) "Using of Conceptual Representation Approach for Query Expansion in Information Retrieval," International Journal of Electrical & Computer Sciences, vol. 12.

[12] D. Pal, M. Mitra, and K. Datta, "Improving Query Expansion Using WordNet", presented at CoRR, 2013.

[13] C. D. Manning, P.Raghavan & H.Schütze. . “ Introduction to information retrieval” vol. 1: Cambridge university press Cambridge, 2008.

[14]. Jinxi Xu and W. Bruce Croft. “Query Expansion Using Local and Global Document Analysis,” in SIGIR conference on Research and development in information retrieval, 1996, pp,1-11

[15] D. Fensel, Ontologies: Springer, Springer Berlin Heidelberg, 2001.

[15] G. Liu, R.Wang, J.Bucklay and Helen M.Zhou “A WordNet-based Semantic Similarity Measure Enhanced by Internet-based Knowledge.” in Proceedings of the 23rd Inter. Conf. on Software Eng. & Knowledge Eng. (SEKE'2011), Eden Roc Renaissance, Miami Beach, USA, July 7-9, 2011

[16] T. Mindaye, M.Sahlemariam, T.Kassie, “The Need for Amharic WordNet.”

[17] “Internet World Stats.” Internet: www.internetworldstats.com/stats1.htm, Sep. 29, 2014.

[18] Betelihem M. "N-Gram-Based Automatic Indexing for Amharic Text", MSc Thesis, Addis Ababa University, School of Information Science, Ethiopia, 2002

[19] Tewodros H., "Amharic text retrieval: An Experiment Using Latent Semantic Indexing (LSI) with Singular Value Decomposition (SVD)" , MSc Thesis, Addis Ababa University, School of Information science, Ethiopia, 2003.

[20] Tesema M. and Solomon A. , "Design and Implementation of Amharic Search Engine", fifth international conference on signal image technology and internet based systems, 3(1):318-325, 2009

[21]A. Hirpa, "Probabilistic Information retrieval System for Amharic Language," Msc, Thesis, School of Information Science, Addis Ababa University, Addis Ababa, 2012.

[22] M. Barathi and S. Valli, "Ontology based query expansion using word sense disambiguation,", International Journal of Computer Science and Information Security, IJCSIS, Vol. 7, No. 2, pp. 022-027, February 2010, USA

[23] Iman. M. Yusuf, "Query Expansion Based on Proper Sense Disambiguation for Amharic language," Msc, School of Information Science, Addis Ababa University 2013.

[24] Alemayehu, "Application of query expansion for Amharic information retrival," Msc, School of Information Science, Addis Ababa University, Addis Ababa, 2002.

[25] A. Bruck, "Semantic based query expansion technique for Amharic Information retrieval," School of Information Science Addis Ababa University, Addis Ababa, 2011.

[26]C.R.Kothari, “Research Methodology, Method and Techniques” *New age International*, New Delhi, India, 2004

[27] T. Semere, "Tigirigna Amharic Cross language Information retrival System " Msc, School of Information Science Addis Ababa University, Addis Ababa, 2013.

[28] Hui Fang, "A Re-examination of Query Expansion Using Lexical Resources", *Association for Computational Linguistics*, Department of Computer Science and Engineering, Columbus, Ohio, USA, June 2008.

[29] F. Sebastiani, "Machine learning in automated text categorization," ACM computing surveys (CSUR), vol. 34, pp. 1-47, 2002.

[30] D. Hiemstra, Using language models for information retrieval: Taaluitgeverij Neslia Paniculata, 2001.

[31] D. Hiemstra, "Information retrieval models," Information Retrieval: searching in the 21st Century, pp. 1-19, 2009.

[32] M. Bendersky, D.Metzler, W. Bruce Croft, "Effective Query Formulation with Multiple Information Sources," Proceedings of the fifth ACM international conference on Web search and data mining, Pages 443-452, 2012

[33] B. He and I. Ounis, "Studying query expansion effectiveness," in Advances in Information Retrieval, ed: Springer, 2009, pp. 611-619.

[34] X. Xu, "Cluster-based query expansion using language modeling for biomedical literature retrieval," Drexel University, 2011.

[35] J.Wu, I. Ilyas, and G. Weddell., "A study of ontology-based query expansion," Technical report CS-2011-04, University of Waterloo, 2011.

[36] B.Billerbeck "Efficient Query Expansion" PhD, School of Computer Science and Information Technology, RMIT University, Melbourne, Victoria, Australia, 2005.

[37] G. Salton and M. J. McGill. "Introduction to modern information retrieval," 1983.

[38] S.Banerjee" Adapting the Lesk Algorithm for Word Sense Disambiguation to WordNet" MSc, Dept. Computer Science, Uni. of Minnesota, Minnesota, 2002.

[39] G.A.Miller "WordNet: a lexical database for English", Communications of the ACM, Pages 39-41, Volume 38 Issue 11, Nov. 1995

[40] X. Li , S. Szpakowicz , S.Matwin, "A WordNet Based Algorithm for Word Sense Disambiguation" In Proceedings of the 14th International Joint Conference on Artificial Intelligence, pages 1368-1374, 1995.

[41] Jan Nemrava, “Using WordNet Glosses to Refine Google Queries” University of Economics, Czech Republic, 2006

[42] A. B. Muhammad and A. T. Yusuf, "Query Expansion: Is It Necessary In Textual Case-Based Reasoning?," Nigerian Journal of Basic and Applied Sciences, vol. 19, 2011.

[43] J. Gonzalo, et al., "Indexing with WordNet synsets can improve text retrieval," arXiv preprint cmp-lg/9808002, 1998.

[44] “WordNet is a Large lexical database for English” Http://wordnet.prinstone.edu

[45] George A. Miller, R.Beckwith, C.Fellbaum, D. Gross, and K. Miller., "Introduction to wordnet: An on-line lexical database," International journal of lexicography, vol. 3, pp. 235-244, 1990.

[46] S.Torres and A. Gelbukh “Comparing Similarity Measures for Original WSD Lesk Algorithm” Advances in Computer Science and Applications. Research in Computing Science, pp. 155-166. 2009.

[47] L.Meng and J. Gu. “A New Method for Calculating Word Sense Similarity in WordNet” International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 5, No. 3, September, 2012

[48] P. Resnik. “Using information content to evaluate semantic similarity”. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995

[49] J. Jiang and D. Conrath. “Semantic similarity based on corpus statistics and lexical taxonomy”. In Proceedings of the International Conference on Research in Computational Linguistics, Taiwan, 1997.

[50] Lin. “An information-theoretic definition of similarity”. In Proceedings of the 15th International Conference on Machine Learning, Madison, WI, 1998.

[51] E.Agirre and P.Edmonds ,” Word Sense Disambiguation: Algorithms and Applications” University of the Basque Country, 2006.

[52] C. D. Manning and H. Schütze, Foundations of statistical natural language processing: MIT press, 1999.

[53] D. Jurafsky and H. James, "Speech and language processing an introduction to natural language processing, computational linguistics, and speech," 2000.

[54] S.Kumar, N.Sharma, Dr. S. Niranjan, “Word Sense Disambiguation Using Association Rules: A Survey” *International Journal of Computer Technology and Electronics Engineering (IJCTEE),* 2012.

[55] S.Patwardhan,S.banerjee andT.Pedersen, “Using measures of semantic relatedness for word sense disambiguation”, Proceedings of the 4th international conference on Computational linguistics and intelligent text processing, Pages 241-257, 2003

[56] B. Yimam, "Yamarińńa Sáwasáw (Amharic Grammar)," Addis Ababa. EMPDA, 1987.

[57] D. Yacob, "Application of the Double Metaphone Algorithm to Amharic Orthography," in International Conference of Ethiopian Studies, 2006.

[58] T. Kassie, "Word Sense disambiguation for Amharic Text Retrieval, A Case Study for Legal Documents," Msc, Addis Ababa University, Addis Ababa, 2009.

[59] S.Liu, C.Yu, W.Meng, “Word Sense Disambiguation in Queries,” Proceedings of the 14th ACM international conference on Information and knowledge management, Pages 525-532, 2005.

[60] H. H. Hoang and A. M. Tjoa, "The state of the art of ontology-based query systems: A comparison of existing approaches," In Proc. of ICOCI06, 2006.

[61] ደስታ ተክለወልድ, "አማርኛ መዝገበ ቃላት", አዲስ አበባ፣ ኢትዮጵያ

[62] W. Leslau. “An Amharic Dictionery”. Los Angeles, USA: Uni. Of California, 1975

APPENDIXES

Appendix 1: Python code for WSD based query expansion

```
print('''እንኳን ወደ አማርኛ የመረጃ ማእከል በደህና መጡ!')
print("\n")
print("\t'''መረጃ ፍለጋ ሊዘገይ ስለሚችል እባኮን ትንሽ ይጠበቁ')
import re
import math
import string
import os
def ano_docs():
	docs=','
	for i in os.listdir('adct'):
		docs=docs+','+i
	docs=docs.replace(',,',')
	doc_List=docs.split(",")
	return doc_List
def Anormaliz(text):
	h1=['ሀ','ሁ','ሂ','ሃ','ሄ','ህ','ሆ']
	h2=['ሐ','ሑ','ሒ','ሓ','ሔ','ሕ','ሖ']
	h3=['ኀ','ኁ','ኂ','ኃ','ኄ','ኅ','ኆ']
	a1=['አ','ኡ','ኢ','ኣ','ኤ','እ','ኦ']
	a2=['ዐ','ዑ','ዒ','ዓ','ዔ','ዕ','ዖ']
	s1=['ሰ','ሱ','ሲ','ሳ','ሴ','ስ','ሶ']
	s2=['ሠ','ሡ','ሢ','ሣ','ሤ','ሥ','ሦ']
	t1=['ጸ','ጹ','ጺ','ጻ','ጼ','ጽ','ጾ']
	t2=['ፀ','ፁ','ፂ','ፃ','ፄ','ፅ','ፆ']
	for i in range(len(h1)):
		text=text.replace(h3[i],h1[i])
		text=text.replace(h2[i],h1[i])
		text=text.replace(t2[i],t1[i])
		text=text.replace(s2[i],s1[i])
		text=text.replace(a2[i],a1[i])
	return text
def remove_puc(v_String):
    lp=['[',']','{','}','.',',','?','፤','።','፡','፧፧',':','::','፣','፤','፥',':']
    v_String=re.sub('[??o=?.??():“,<>-``/”??"??!‘'' \ufeff]',",v_String)
    for i in lp:
        v_String=v_String.replace(i,")
    v_String= re.sub('[\d+]',", v_String)
    return v_String
def stemtq(text):
	tp=['ኀ','እ','ን']
	for i in tp:
		if text.startswith(i):
			text=text.replace(text[0:1],")
	return text
def asufpre(v_List):
    prfx=open("prefix.txt",encoding='utf-8')
    prefix=prfx.read()
    prefix=prefix.split()
    sfx=open("sufix.txt",encoding='utf-8')
```

```
    suffix=sfx.read()
    suffix=suffix.split()
    for n in range(0,len(v_List)):
        stemed_query="
        stemed_query=stemed_query+v_List[n]
        for preffix_l in range(0,len(prefix)-1):
            if(len(stemed_query)>2):
                if(stemed_query.startswith(prefix[preffix_l])):
                    stemed_query=stemed_query.replace(prefix[preffix_l],'')
                    preffix_l=len(prefix)
        for suffix_l in range(0,len(suffix)-1):
            if(len(stemed_query)>2):
                if(stemed_query.endswith(suffix[suffix_l])):
                    stemed_query=stemed_query.replace(suffix[suffix_l],'')
                    suffix_l=len(suffix)
        v_List[n]=stemed_query
    sfx.close()
    prfx.close()
    return v_List
def Amain():
    prfx=open("prefix.txt",encoding='utf-8')
    prefix=prfx.read()
    prefix=prefix.split()
    prfx.close()
    sfx=open("sufix.txt",encoding='utf-8')
    suffix=sfx.read()
    suffix=suffix.split()
    sfx.close()
    dfile={}
    d_no=1
    stp_w=[]
    v_Index={}
    docu_List=[]
    docu_List=ano_docs()
    d_no=len(docu_List)
    nsw=0
    stopw=open("stopw.txt",encoding='utf-8')
    while stopw.readline()!='':
        nsw= nsw+1
    stopw.close()
    stopw=open("stopw.txt",encoding='utf-8')
    for i in range(1,nsw):
        line=stopw.readline()
        line=line.rstrip()
        stp_w.append(line)
    stopw.close()
    for j in range(1,d_no+1):
        dline='adct\\'+docu_List[j-1]
        ddn=docu_List[j-1]
        ddn=ddn.replace('doc','')
        ddn=ddn.replace('.txt','')
        dname="
        dn=int(ddn)
        dline=dname+dline.rstrip()
        encoding_doc=open(dline,encoding='utf-8')
        while True:
```

```
        v_String=encoding_doc.readline()
        fstring=remove_puc(v_String)
        fstring=Anormaliz(fstring)
        v_List=fstring.split()
        v_List=asufpre(v_List)
        s=0
#to remove empty words
        while s < len(v_List):
            if v_List[s]=='':
                del v_List[s]
            s+=1
        for i in range(0,len(v_List)-1):
            if v_Index.__contains__(v_List[i]):
                if v_Index[v_List[i]].__contains__(dn):
                    continue #v_Index[v_List[i]][j]+=1
                else:
                    t={dn:1}
                    v_Index[v_List[i]].update(t)
            else:
                 if stp_w.__contains__(v_List[i]):
                    continue
                 v_Index[v_List[i]]={dn:1}
        if len(fstring)==0:
            break
    tpf=open('Posting.txt','w',encoding='utf-8')
    tpf.write('\tDoc_ID\tTF \t \t \t position')
    tpf.write("\n")
    for i in v_Index:
        tpf.write('\t')
        tpf.write(i)#DocId and TF for each term
        tpf.write('\t')
        tpf.write('\t')
        tpf.write('\t')
        tpf.write('\t')
        tpf.write(str(v_Index[i]))# The position of each terms
        tpf.write("\n")
    tpf.close()
#########################################################################
def search():
################################## searching ###################
    docu_List=ano_docs()
    aN=len(docu_List)
    av=[]
    ac=0
    posting=open("Posting.txt",encoding='utf-8')
    check=posting.readline()
    while check!='':
        post=check
        post=post.replace('\t','')
        s1=post
        s2=s1.split('{')
        if len(s2)<=1:
            check=posting.readline()
            continue
        else:
            s1=s2[1]
```

```
        s2=s1.split(',')
        l=[]
        for i in range(len(s2)):
            s1=s2[i]
            ll=s1.split(':')
            lll=int(ll[0])
            l.append(lll)
        s=len(s2)
        postt=post.split(':')
        an=(len(postt))-1
        t=postt[0].split('{')
        w=t[0]
        ww=math.log10((aN-an+0.5)/(an+0.5))
        av.append([w,an,ww,l])
        ac+=1
        check=posting.readline()
m=0
while m < len(av):
        n=0
        while n < len(av[m][3]):
            #av[m][3][n]=av[m][3][n].replace(' ','')
            n+=1
        m+=1
posting.close()
d_no=1
stp_w=[]
nsw=0
stopw=open("stopw.txt",encoding='utf-8')
while stopw.readline()!='':
    nsw= nsw+1
stopw.close()
stopw=open("stopw.txt",encoding='utf-8')
for i in range(1,nsw):
    line=stopw.readline()
    line=line.rstrip()
    stp_w.append(line)
stopw.close()
query=input("እባኮን የሚፈልጉትን ፋይል ለማግኘት መጠይቆን ያስገቡ!:-  ")
while len(query)==0:
   query= input(" ምንም ዓይነት መጠይቅ አላስገቡም፤ እባክዎ መጠይቆን እንደገና ያስገቡት! ")
fstring1=remove_puc(query)
fstring1=re.sub('[፡፡.!:።()"፣?/!\ufeff]',",query)
fstring1=Anormaliz(fstring1)
qv_List=[]
qt_List=[]
qt_List=fstring1.split(" ")
qu=len(qt_List)
if qu>15:
   for q in range(15):
        qv_List.append(qt_List[q])
else:
     qv_List=qt_List
indexQ={}
aresult=[]
aall=asufpre(qt_List)
aws={}
```

```
for i in range(len(aall)):
        for j in range(len(av)):
                if aall[i]==av[j][0]:
                        for k in range(len(av[j][3])):
                                t=av[j][3][k]
                                if aws.__contains__(t):
                                        aws[t]+=av[j][2]
                                else:
                                        aws[t]=av[j][2]
for ii, j in aws.items():
        i=str(ii)
        i="Document "+i
        t=[i,j]
        aresult.append(t)
tot=[]
tot=aresult
result=tot
for i in range(len(result)):
        for j in range (len(result)-1):
                if result[j][1] < result[j+1][1]:
                        t=result[j]
                        result[j]=result[j+1]
                        result[j+1]=t
print("\n ባስገቡት መጠይቅ መሰረት የተገኙት መረጃዎች እንደሚከተለው ቀርበዋል!  \n")
for i in range(len(result)):
        if result[i][1]>1:
         print(i,": ---------------------------- ",result[i][0],"")
```

- **Expansion**

```
g=[]
c=0
wn=open("wordnet.txt",encoding='utf-8')
while wn.readline()!='':
   c=c+1
wn.close()
wnn=open("wordnet.txt",encoding='utf-8')
wnn.readline()
for i in range(c):
   ll=wnn.readline()
   g.append(ll)

ww=[]
www=[]

for i in range(len(g)):
   ww.append(g[i])
   www.append(ww)
   ww=[]

#print (www)
wordnet=[]
word=""
#print("\n\n\n\n")
for i in range (len(www)):
   word=www[i][0]
```

```
    ww=word.split("@")
    wordnet.append(ww)

l2=[]
l3=[]
for i in range(len(wordnet)-1):
    word=wordnet[i][1]
    ww=word.split(";")
    wordnet[i][1]=ww

for i in range(len(wordnet)-1):
    for j in range(len(wordnet[i][1])):
        word=wordnet[i][1][j]
        ww=word.split(":")
        wordnet[i][1][j]=ww

for i in range(len(wordnet)-1):
    for j in range(len(wordnet[i][1])):
        word=wordnet[i][1][j][0]
        ww=word.split(" ")
        wordnet[i][1][j][0]=ww
        word=wordnet[i][1][j][1]
        ww=word.split(" ")
        wordnet[i][1][j][1]=ww

for i in range(len(wordnet)):
        for j in range(1, len(wordnet[i])):
                for k in range(len(wordnet[i][j])):
                        for m in range(len(wordnet[i][j][k])):
                                # for n in range(len(wordnet[i][j][k][m])):
                                wordnet[i][j][k][m]= asufpre(wordnet[i][j][k][m])


print (wordnet)
sense=[]
for i in range(len(qv_List)):
    for j in range(len(wordnet)):
        if qv_List[i]==wordnet[j][0]:
            sense.append(wordnet[j])

print("\n\n\n\n")
print(sense)
l1=[]
l2=[]
l3=[]
l4=[]
case1=[]
case2=[]
x=""
y=""
for i in range(len(sense)):
    #print ("\n")
    for j in range(len(sense[i][1])):
        for k in range(len(sense[i][1][j][1])):
            w=sense[i][1][j][1][k]
            for l in range(len(sense)):
```

```
            if i==l:
                continue
            else:
                i
                for ll in range(len(sense[l][1])):
                    for lll in range(len(sense[l][1][ll][1])):
                        ww=sense[l][1][ll][1][lll]
                        if w==ww:
                            x=str(i)+str(1)+str(j)+str(1)
                            y=str(l)+str(1)+str(ll)+str(1)
                            l4.append(x)
                            l4.append(y)
                            l3.append(l4)
                            l4=[]

expandedQ=qv_List
d=l3[0][0]
c=1
for i in range(len(l3)):
    if l3[i][0]==d:
        continue
    else:
        c=c+1
        d=l3[i][0]
w=[]
for iii in range(c):
    for i in range(len(l3)):
        w.append(0)
        for j in range(i+1,len(l3)):
            if l3[i]==l3[j]:
                w[i]=w[i]+1

    x=w[0]
    xx=0
    for i in range(len(w)):
        if x<w[i]:
            x=w[i]
            xx=i
        else:
            continue

    print ("your new query will be")
    aa=int(l3[xx][0][0])
    ab=int(l3[xx][0][1])
    ac=int(l3[xx][0][2])
    lens=len(sense[aa][ab][ac][1])
    aaa=int(l3[xx][1][0])
    aab=int(l3[xx][1][1])
    aac=int(l3[xx][1][2])
    leng=len(sense[aaa][aab][aac][1])

    nf1=(w[xx]/lens)*(lens/leng)
    case1.append(l3[xx])
    case1.append(nf1)
    expandedQ=expandedQ+sense[aa][ab][ac][1]
    temp=l3[0][0]
```

```
        l=len(l3)
        for j in range(len(l3)):
            for i in range(len(l3)-1):
                if l3[i][0]==temp:
                    del l3[i]
                    break
    l=0
    f=0
    key=0
    for j in range(len(expandedQ)):
        for i in range(key+1,len(expandedQ)-1):
            if expandedQ[key]==expandedQ[i]:
                del expandedQ[i]
                break
        key=key+1

##########################AFTER EXPANSION ##########################################
    aresult=[]
#    aall=asufpre(qt_List)
#    aall=expandedQ
    aall=finalq
    aws={}
    for i in range(len(aall)):
        for j in range(len(av)):
            if aall[i]==av[j][0]:
                for k in range(len(av[j][3])):
                    t=av[j][3][k]
                    if aws.__contains__(t):
                        aws[t]+=av[j][2]
                    else:
                        aws[t]=av[j][2]
    for ii, j in aws.items():
        i=str(ii)
        i="Document "+i
        t=[i,j]
        aresult.append(t)
#####

    tot=[]
    tot=aresult
    result=tot
    for i in range(len(result)):
        for j in range (len(result)-1):
            if result[j][1] < result[j+1][1]:
                t=result[j]
                result[j]=result[j+1]
                result[j+1]=t
    print("\n መጠይቅዎ ከተጠናከረ በሁዋላ የተገኙት መረጃዎች እንደሚከተለው ቀርበዋል! \n")
    for i in range(len(result)):
        if result[i][1]>4:
         print(i,": ---------------------------- ",result[i][0],"")
```

Appendix 2: Amharic WordNet

አደጋ@አደጋ ችግር ስቃይ ጉዳት:በህመም ባልታሰበ አደጋ በሀዘን በችግር ምክንያት የሚመጣ በተፈጥሮ የሚከሰት እንደ ጎርፍ መሬት መንቀጥቀጥ ጉዳት;ተገቢ ያልሆነ ድርጊት:ትክክል ተገቢ ተመጣጣ ባልሆነ ሁኔታ የተፈጸመ

ጊዜ@ጊዜ ዘመን አመት ወር ቀን ሰዓት ወቅት:ዘመን አመት ወር ቀን ሰዓት ሲፈራረቅ;ሳይመሽ ሳይጨልም:ጸሃይ ሳለ እንግዳ በጊዜ መጣ ያለ ጊዜ እንደ ጊዜ

እርዳታ@እርዳታ መስጠት ልገሳ እገዛ አገልግሎት:ለድንገተኛ ችግር ጉዳት ወይም አደጋ ሲኖር የሚደረግ እርዳታ በተለይ በህክምና;እገዛ ማጣዝ:ስራ የበዛበትን ሰው ማገዝ

መማሪያ@መጽሃፍ ደብተር:ትምህርት ለመማር የሚያስፈልጉ መሳሪያዎች;መማሪያ ትምህርት ቤት:ተማሪዎች የሚማሩበት የትምህርት ቦታ ክፍል;ርኅራሄ ሃዘኔታ ይቅር ማለት:ያጠፋ የበደለ ሰዉ ምህረት ይቅርታ እንዲደረግለት

ክፍል@መማሪያ ክፍል ክላስ:በትምህርት ቤት የሚገኙት እያንዳንዱ የመማሪያ ክፍሎች;ድርሻ ፋንታ:የሚከፋፈል ነገር በዕድል መልክ ሲደርስ;መለያ ምዕራፍ:መጻህፍት በፊልሞች ላይ ያሉ መረጃዎች በአንቀጽ ወይም በምዕራፍ ሲከፋፈል

ማስፋፊያ@ማጠናከሪያ:የሆነ የተጀመረ ስራ እንዲጠናከር እንዲሻሻል የሚደረግ እንቅስቃሴ

ምርመራ@ህክምና ማግኘት:የታመመ ሰዉ ወደ ህክምና ቦታ ለምርመራ ሲሄድ;ፍተሻ ጥየቃ ፍለጋ:ህግን የተላለፈ ድርጊት ሲፈጸም በፖሊስ ጣቢያ ወይም በፍርድቤት የሚደረግ ምርመራ

ግንባታ@ማስፋፊያ:የተለያዩ የስራ ተቋማት በግንባታ መልኩ ሲስፋፋ;መስራት መገንባት:የህንጻ ቤት መስራት

ቅርስ@ሃብት:ካባት ወደ ልጅ የሚተላለፍ ተካባች ብር መኖሪያቤት;የአገር ሃብት:አንዲት አገር የተለዩ ቦታዎች ሃውልቶች የዱር አራዊቶች እንደ ቅርስ ሲታዩ

ጤና@ፈዉስ ህይወት:በበሽታ አለመተቃት;ጤና ጣቢያ:በሽተኛ ሰዉ ሲታመም ለህክምና የሚኬድበት ማእከል

ጣቢያ@ጤና ጣቢያ ሆስፒታል የጤና ኬላ:በሽተኛ ሰዉ ሲታመም ለህክምና የሚኬድበት ማእከል;ባቡር ጣቢያ:ባቡር የሚያርፍበት የሚነሳበት;መታቆሪያ:በወንዝ ዳር ያለ የዉሃ ማቆሪያ;ፖሊስ ጣቢያ:ህግን የተላለፈ ነገር ሲፈጸም የሚመረመርበት ቦታ

ወባ@የበሽታ ስም:አስራቦ ከሚባል ትንኝ የሚመጣ ንዳድ ትኩሳት በሽታ ህመም

በሽታ@መታመም መታወክ:በህመም በሽታ መጠቃት ጤንነት አለመሰማት;እንቅፋት መሰናክል:ለሆነ ስራ ወይም ድርጊት ማሰናከል ሰዉየዉ እንዳልሰራ በሽታ ሆነ

ቁጥጥር@መከላከል:የሆነ ሰው በበሽታ ወይም በአደጋ እናዳይጠቃ አስቀድሞ መከላከል;መመርመር:ህግን የተላለፈ ድርጊት ሲፈጸም መርምሮ በቁጥጥር ስር ማዋል;መከታተል:ቤተሰብ ልጁን አልባሌ ቦታ እንዳይዉል መቆጣጠር

ራስ@ጭንቅላት የራስ ቅል አኽምሮ:የሰዉ ልጅ በጸጉር የተሸፈነ የሰዉነት ክፍል;መበስበስ መርጠብ:ፈሳሽ የተደፋበት ነገር; ሹመት እንደራሴ ወኪል:የማዕረግ ስም

በራ@ብርሃን ነጸብራቅ:ሻማ መብራት ጸሃይ የሚሰጠዉ ብርሃን;መላጣ:የሰዉ ልጅ ጸጉር ሲመለጥ

ኤችአይቪ@ኤድስ:በአለማችን መድሃኒት ያልተገኘለት በግብረ ስጋ ግንኙንት የሚተላለፍ በሽታ በህክምና በምርመራ የማይድን

እግርኳስ@የስፖርት ዓይነት:በጣም የሚዘወተር የስፖርት ዓይነት

እግር@እግር:አንዱ የሰዉነታችን ክፍል ሲሆን የምንራመድበት የምንሮጠበት እግር ኳስ የምንጫወትበት

ኳስ@ኳስ:ስፖርታዊ እንቅስቃሴ የምንጫወትበት እንደ እግር መረብ ቅርጫት እጅ ኳስ እንደምንጠቅምበት አይነት

እንክብካቤ@ክትትል እድሳት:የተለዩ ቦታዎች እንደ ቅርስዎች ሙዚየም እንክብካቤ ክትትልና ቁጥጥር ሲደረግለት

ስራ@ስራ ተግባር:የሰዉልጅ ኑሮውን ለመደገፍ ብሎ እለትተለት የሚያደርገዉ ስራ እንቅስቃሴ ድርጊት

ሞት@ህይወት ማለፍ:የሰዉልጅ ከዚህ ዓለም ሲለይ እስትንፋሱ ሲቆም መላ የሰውነት ክፍል በድን ሲሆን

አካል@የሰውነት ክፍል:እያንዳንዱ የሰውነት ክፍል;ክፍል አባል:የሆነ የአንድ ነገር ክፍል

ጉዳት@አደጋ:በሰውነት ክፍል በንብረት ላይ አደጋ ሲደርስ

ጥበቃ@ዘበኛ:መኖሪያ ቤት ድርጅት የሚጠብቅ ሰው;ክትትል ቁጥጥር:የተለዩ ቦታዎች እንደ ቅርስዎች ሙዚየም እንክብካቤ ክትትልና ቁጥጥር ሲደረግለት

መከላከል@መጠንቀቅ:የከፋ ነገር እናዳይደርስ አስቀድሞ መጠንቀቅ

ስልጠና@ልምምድ:በማንኛዉም የስፖርት ዓይነት ዙሪያ ያሉ ስፖርተኞች ከዋናው ጨዋታ በፊት የሚያደርጉት ልምምድ;ትምህርት:ሙያዊ ነክ የሆኑ ትምርቶች መውሰድ

ሙያ@ችሎታ ስራ:ሰው በሚሰራ ስራ የታካነ ሲሆን ባለ ሙያ ነው ሲባልለት;ባለሙያ:በቤት ስራዎች ጎበዝ የሆነች ሴት

ቴክኒክ@ዘዴ:ለማሰልጠን ወይም ለማስተማር የሚደረግ ዘዴ ብልሃት;ቴክኒክ:ከመካኒክ ከእንጨት ስራ ጋር የተያያዘ የሙያ ስራ

ማሰልጠ@መማሪያ ቦታ:ሰዎች በተለያየ ስራ ሆነ ሙያ ብቁ ለመሆን የማሩበት ቦታ ማእከል

ተቋም@ድርጅት:የስራ ቦታ

ድርጅት@ተቋም:የስራ ቦታ

ልማት@እድገት:አንዲት አገር በስራ እድገት እንድታሳይ የሚደረግ እንቅስቃሴ

እንቅስቃሴ@እንቅስቃሴ:ለተለያዩ ስራዎች የሚደረግ እንቅስቃሴ;እንቅስቃሴ:ስፖርታዊ እንቅስቃሴ

ማድረግ@ማከናወን:ለሚሰራ ስራ ወይም እንቅስቃሴ የማከናወን ድርጊት