ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

WORD SENSE DISAMBIGUATION FOR AMHARIC TEXT RETRIEVAL: A CASE STUDY FOR LEGAL DOCUMENTS

By: **Teshome Kassie**

A THESIS SUBMITTED TO

THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

April, 2009

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

WORD SENSE DISAMBIGUATION FOR AMHARIC TEXT RETRIEVAL: A CASE STUDY FOR LEGAL DOCUMENTS

By: **Teshome Kassie**

A THESIS SUBMITTED TO

THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

April, 2009

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUSTE STUDIES

FACULITY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

WORD SENSE DISAMBIGUATION FOR AMHARIC TEXT RETRIEVAL: A CASE STUDY FOR LEGAL DOCUMENTS

By: Teshome Kassie

Approved by examining board:

1. _____ _____

2. _____ _____

3. _____ _____

DEDICATION

To my family- Kibretie Alemayehu
Samuel Teshome
Heran Teshome
Daniel Teshome
Yordanos Teshome

TABLE OF CONTENTS

List of tables

List of figures

List of appendices

List of Acronyms

ACM	Association of Computing Machinery
API	Application Programming Interface
CACM	Communication of the Association of Computing Machinery
GTF	Global Term Frequency
HAL	Hyperspace Analogue to Language
IDF	Inverse Document Frequency
IR	Information Retrieval
LSA	Latent Semantic Analysis
NLP	Natural Language Processing
POS	Part of Speech
RI	Random Indexing
RSV	Retrieval Status Value
SVD	Singular Value Decomposition
TF	Term Frequency
TREC	Text REtrieval Conference
WSD	Word Sense Disambiguation

Acknowledgement

I would like to thank Dr. Nega Alemayehu, my supervisor, for his encouragement, patience and expert advice to finalize this thesis work. I would like to thank everybody that helped and supported me in one way or another.

Special thanks to Tessema Mindaye, Shibru Belete for their valuable support in the course of the research. I also express my gratitude to Dr. Paul R. Doresy and Caryl Lee Fisher from Delcian Inc. for their reviewing this research and gave me their invaluable comments.

Finally I would like to thank my family especially Samuel Teshome.

ABSTRACT

This study demonstrates how linguistic disambiguation based on semantic vector analysis can improve the effectiveness of an Amharic document query retrieval algorithm.

Accurate document retrieval based on query criteria is important in every knowledge domain. The ability to retrieve appropriate documents is made more difficult by the fact that many words can have different meanings in different contexts. If search engines could disambiguate those words, more accurate retrieval of documents should be able to be achieved.

For this study, an Amharic disambiguation algorithm was developed based on the principles of semantic vectors and implemented in Java. The disambiguation algorithm was then used to develop a document search engine.

A set of 865 Ethiopian Amharic language legal statute documents were selected as the document population that would be searched. Ten queries containing Amharic keywords with ambiguous meaning were selected. An expert was used to identify which documents should ideally be retrieved by each query. Depending on the query, the expert identified between 6 and 25 documents that should be retrieved.

The semantic vector query algorithm created in this study was compared to the well known Lucene algorithm. Each query was run using both algorithms. The 20 most relevant documents were identified for each query from each algorithm.

For each query, the list of documents retrieved by each algorithm was compared to the list of documents identified by the expert. The number of correct (consistent with the expert's choices) documents retrieved by each algorithm was measured.

Results are that the semantic vector algorithm was superior for 6 of the 10 queries (Lucene was superior on 2 queries, and on two they were tied). This difference was not statistically significant. However, if the total number of correct document identifications are taken into account (not just which algorithm was superior for each query) then the semantic vector algorithm averaged 82% correct identification of documents where as the Lucene algorithm was only 49% accurate. This difference was highly statistically significant ($p < 0.02$) less than the level of significant ($p<0.05$) for rejecting null hypothesis. .

The conclusion is that for Amharic legal statute documents, for queries that include ambiguous keywords, the semantic vector algorithm is superior over lucene algorithm.

Keywords: word sense disambiguation, semantic vectors, Information retrieval.

Chapter One

Introduction

1.1. Background

The rapid growth of information technology and capability of information storage devices increase the availability of large collections of texts/documents electronically. The enormous increase in the amount of online text available and the demand for access to different types of information have, however, led to a renewed interest in a broad range of IR-related areas [AL03].

Information retrieval (IR) dealing with the presentation, storage, organization, and access to information items should provide the user with easy access to the information in which he/she is interested. In IR system, the user must first translate his/her information need into a query which can be processed by the search engine (or IR system) [BY99]. Given the user query, the key goal of an IR system is to retrieve information which might be useful or relevant to the user. To be effective in its attempt to satisfy the user information need, the IR system must somehow 'interpret' the contents of the information items (documents) in a collection and rank them according to a degree of relevance to the user query. This 'interpretation' of a document content involves extracting syntactic and semantic information from the document text and using this information to match the user information need. The difficulty of retrieving relevant documents can be tackled by resolving ambiguities of words in the collection of texts.

As in [SMJ03] ambiguity in natural language has long been recognized as having a negative effect on the performance of text based information retrieval (IR) systems. When an ambiguous word is used in a sentence, humans are able to select the correct sense of that word without considering alternative senses. But in any application, a computer cannot be able to identify and resolve the usage of ambiguous words in processing natural language. Sanderson used the word

"bat", which has two meanings: an instrument used in sports to hit balls; or a furry, flying mammal, to explain the concept [SM94].

Searching for a text from a large information repository where ambiguous words are common could result with the inclusion of irrelevant texts where the user should filter again the required ones manually. This occurs mostly when there are polysemy words i.e. words which have more than one meaning in the language used. These types of words create ambiguity in processing where a machine could not identify the semantic of a given query to process according to the need of the user. From this perspective it is important to disambiguate those words in the context of a given domain for the effectiveness of information retrieval.

Processing Amharic language text using computers has become common for more than two decades. This days government and non government organizations as well as individuals are using computer systems for processing documents in Amharic. Documents are the major information sources for researchers and the public in general. The increasing availability of electronic documents creates a problem of information access effectiveness by providing irrelevant information. Developing a word sense discrimination induced information retrieval system for retrieving Amharic legal documents for a query could help users of those documents for easy access.

1.2. Motivation of the work

The main motivation of this work is to test the effectiveness of information retrieval of Amharic texts by resolving ambiguity of words in the Amharic language corpus.

Government regulations should ideally be accessible with ease by legal practitioners as well as users who need to use legal documents. In reality, regulations are voluminous, heavily cross-referenced and often ambiguous.

The importance of information retrieval for the legal system and the jurisprudence is not questionable. Lawyers work with words, sentences, and texts. The language is a central subject of a lawyer's work. Legal thinking is based on the vocabulary of legal terms which are used to express a definite concept. The IR-system cannot distinguish between terms with different meanings and thus retrieves irrelevant texts for a query. The user is not always aware of this deficiency of the information stored electronically and also not able to overcome this problem by special research techniques [SW94]. To overcome this problem one of the solutions used is word sense disambiguation (WSD) in NLP. Various researches have been undertaken in Amharic natural language processing ([NW02]: [AB00]; [ME01]). However, to the best of my knowledge there has never been a work on WSD. The intuitive of WSD in IR is to help users in getting relevant documents for the queries resulting in the effectiveness of electronic documents retrieval systems.

1.3. Statement of the Problem

There are a lot of legal documents in Ethiopia which can be stored electronically and retrieved to support the day to day activities of lawyers and the public in general. Currently, those documents are not well organized and stored electronically for easy access. The Ethiopian penal code which is composed of 865 articles, where in some cases articles are composed of sub articles is one of the legal documents which is used in the litigation process. Considering articles and sub articles as documents, the penal code has more than 865 documents. Users of the penal code search for a document/article among the above specified number of documents manually. Manual search for a document is time consuming. In conventional IR, documents are retrieved by posting a query. Queries may be simply a collection of keywords. Incorporating WSD to the IR system for a massive document repository is supposed to improve the result of the system for a given query.

Law consolidation activities, which compile legal provisions through the identification of the active (working) regulations from the repealed regulations, required references to existing documents. Doing the above mentioned activities manually is cumbersome and time consuming. Furthermore, with the advent of information technology there is a need to establish E-Government with in the country as a result of which federal and regional states could benefit from exchange of legal provisions within the country from electronically stored documents. Therefore, legal document retrieval with some simple mechanism is a necessity with the consideration of Ethiopic script features with the application of WSD.

1.4. Objective of the Project

The general objective of the project is to design and develop an information retrieval system for legal documents with the application of WSD for Amharic language and evaluate its performance. The specific objectives of the research are:

- To collect a test data.To identify best parameters and values words which increase the effectiveness of information retrieval.
- To design and implement a retrieval system with WSD for Amharic legal documents.
- To evaluate the effectiveness of the algorithm developed using relevance judgment.

1.5. Scope of the project

The scope of this study is to develop information retrieval prototype system using word sense discrimination, which includes indexing collection of documents, random indexing which is used for dimension reduction of high dimension of word space created from the test collection. In this research work semantic vectors which include term vectors and document vectors formulation will be created for the implementation of word sense induced information retrieval.

1.6. Organization of the thesis

This thesis is organized into five chapters including the current one:

- Chapter 2 deals with the methodology used in this research. It describes the corpus used for the work and the preprocessing work that was done.
- Chapter 3 includes a literature review of research on word sense disambiguation, information retrieval, and the combination of the two; i.e. the application of word sense for information retrieval systems.
- In Chapter 4, the process of inducing word sense extracted from context is explained along with the components of the system.
- Chapter 5 describes the implementation of the information retrieval system designed for this project, and the experiments with effectiveness measurement conducted using a traditional information retrieval system.
- Chapter 6 summarizes the conclusions based on the research and recommendations for future study of this topic.

Chapter Two

Research Methodology

The research methodology describes the procedures used to conduct the research, including how the corpus was prepared, and how word vectors, context vectors, and sense of words were acquired from the corpus. Weighting and implementing the word meanings in the IR are described.

2.1. Corpus

The corpus used for our experiment is the Ethiopian Penal Code which is composed of 865 articles. An article in a corpus is considered as one document. Each document is identified by its article number with respective title. In some cases, an article contains sub-articles. The corpus has 19,493 words in size without the application of stemming, normalizing, and stop words removal.

Before using a corpus for the retrieval experiment, preprocessing of the test corpus was performed. This preprocessing included normalizing, tokenizing, stemming, and stop word removal. Each preprocessing is described below.

i. Normalizing

Since Amharic language has some characters that represent the same sound, use of these characters differs from user to user. Such characters include:- ሀ፣ሐ፣ኀ፤ ሰ፣ሠ፤ አ፣ዐ and others. In order to exclude such variations in processing the Amharic text, one character representation for the same sound is necessary, so such representation was used for the text included in the experiment.

ii. Stop word removal

In information retrieval systems, the analysis of word distribution in the corpus is important. Accordingly the frequency distribution of words is considered. The usage of words in each sentence has to be analyzed. Sentences are formed by using a sequence of words or terms. The frequency distribution of these terms in the corpus has different values. In other words, there are high and low frequency terms. In information retrieval, content words are a more important factor in the performance of IR systems than function or stop words, which although occurring in high frequencies, have less importance. For this reason, stop words were removed before indexing documents for retrieval.

iii. Tokenizing

The process of dividing the input text into units called tokens, where each token consists of either a word or something else such as a number or a punctuation mark is called tokenization. The input text is tokenized with regard to the Amharic language characteristics used to retrieve the documents.

iv. Stemming

In most cases, morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. For this reason, a number of so-called *stemming algorithms*, or *stemmers*, have been developed, which attempt to reduce a word to its *stem* or root form. Thus, the key terms of a query or document are represented by stems rather than by the original words. This not only means that different variants of a term can be *conflated* to a single representative form, but it also reduces the *dictionary size*, that is, the number of distinct terms needed for representing a set of documents. A smaller dictionary size results in a saving of storage space and processing time.

For IR purposes, it does not usually matter whether the stems generated are genuine words or not. Therefore, "computation" might be stemmed to "comput" provided that (a) different words with the same "base meaning" are conflated to the same form, and (b) words with distinct meanings are kept separate. An algorithm that attempts to convert a word to its linguistically correct root ("compute" in this case) is sometimes called a *lemmatiser*.

2.2. Word sense discrimination for Information retrieval

Since word sense discrimination greatly enhances the results of an information retrieval system query, [SC98], [SP95], this type of method was used for the research described in this project. After the preprocessing stage, highly frequent words were analyzed for optimum selection of a frequency threshold. From these highly frequent words, a term-term matrix C was constructed. The entry c_{ij} is the number of times that word i and word j co-occur in a symmetric window of total size k.

To reduce the high real valued dimension of the matrix, singular value decomposition (SVD) is used. Column elements represent entries of the thesaurus vector. Similarity between thesaurus vectors for each was calculated using a cosine distance function. Cosine distance measures the semantic relation between words. A thesaurus is constructed by associating each word with its nearest neighbors. A context vector is derived from the sum of the thesaurus vectors of the context words. After identifying context vectors corresponding to all occurrences of a particular word, vectors were partitioned into regions of high density, where each region could map according to word meaning.

For disambiguation of a given word, the context vector of its occurrence was computed. Then the meaning of the occurrence was assigned for the closest centroid (average) of the region.

For the final stage of this experiment, word meanings was used for information retrieval by replacing words by their meanings in the representation of documents and indexing. This should improve the precision and recall measurements of information retrieval for Amharic texts.

2.3. Data Analysis

There are different ways to evaluate the performance of word sense disambiguation (WSD) algorithms: Sense tagged data, pseudo words, upper and lower bounds on performance, SENSEVAL are among the techniques used for evaluation.

Sense tagged data are used to compare the accuracy of the disambiguation algorithm in contrasting the result of a disambiguated word with the predefined word sense. Due to laborious and time consuming of sense tagging manually of a large volume of corpus, pseudo words which are artificial words are used for evaluating word sense disambiguation. Pseudo words are created by conflating two or more natural words. For example, one can replace banana and door by an artificial word banana-word in the corpus. Using the text with pseudo words as ambiguous source text and the original one with ambiguous words disambiguated is used for evaluation [SM94], [MS99].

For our experiment, the evaluation of information retrieval system was done based on the precision and recall measurements by formulating a set of queries for retrieval purposes. Domain expert was involved to evaluate the prototype system.

Chapter Three

Litrature Review

3.1. Information Retrieval

Information retrieval (IR) involves the representation, storage, organization of, and access to information items. The representation and organization of the information items in the system should provide the user with easy access to the desired [SM83].

The need for effective methods of automated IR has grown because of the tremendous explosion of the amount of unstructured data, including internal, corporate document collections, and the growing number of document sources on the Internet [EG01].

Information retrieval systems typically use index terms to index and retrieve documents. An index term is any word which appears in the text of a document in the collection. The retrieval of documents using index terms is based on notion that the semantics of the documents and the user's information need can be naturally expressed through sets of index terms [BY99].

IR technology can be categorized into two broad categories; semantic and statistical approaches. Statistical approaches deals with several different techniques: Boolean, extended Boolean, vector space, and probabilistic. Statistical approaches break documents and queries down into terms. These terms are the population that is counted and measured statistically. Most commonly, the terms are words that occur in a given query or collection of documents [EG01].

The classical problem in IR is the ad-hoc retrieval problem where a user enters a query describing his or her desired information. The system returns a list of documents with respect to the query. Exact match system return documents that precisely fulfill some

structured query expression, of which the best known type is Boolean queries. But for large, heterogeneous document collections, the result sets of exact match queries are usually empty or huge and unwieldy, some recent researches have concentrated on systems which rank documents according to their estimated relevance to the query. For such systems probabilistic methods are useful [MS99].

Most IR models are statistical in nature; they will either explicitly or implicitly assume a certain distribution of the textual data. Assuming that the data has certain statistical properties makes it possible to draw statistical inferences [KW04]. The wells known statistical distributions consist of the normal (Gaussian) distribution and the binomial distribution. The former is a continuous distribution, where the random variable has a continuous domain; the latter is a discrete distribution, in this case with two possible values for the random variable.

Some of the distributions which have been used to model textual data in the context of IR are Zipf's law, the binomial distribution, the multinomial distribution, the poison distribution, and the 2-poison distribution.

Zipf's law: describes the distribution of words in a corpus. The histogram of words occurring in a corpus with the words stored in descending frequency shows a non linear curve. The distribution is not homogeneous but skewed. Zipf approximated the shape of the histogram with the formula:

f X r = c, where ,f is frequency, r is rank and c is constant. Figure 3.1 shows the hyperbolic curve of the histogram[KW04].

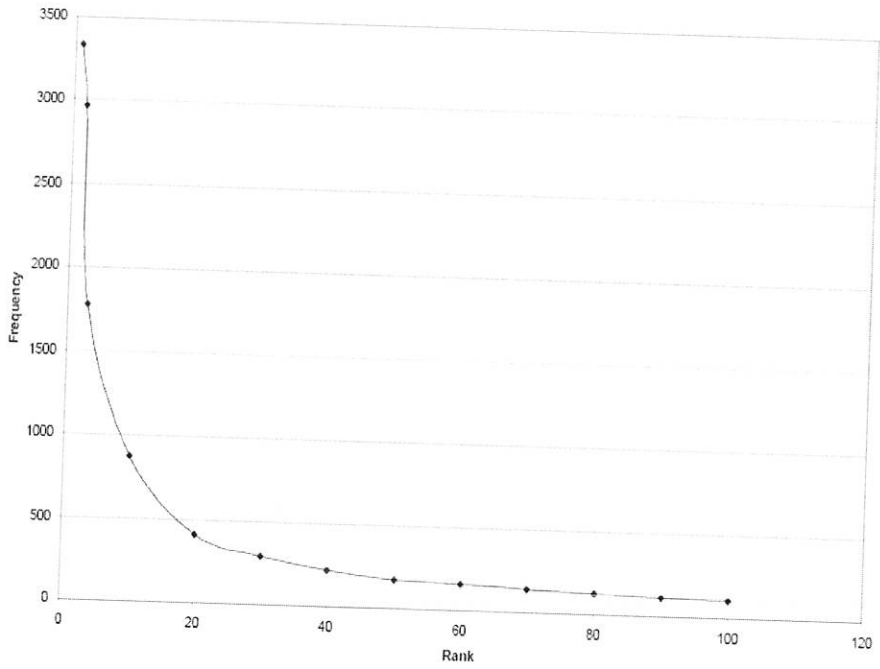

Figure 3.1. Type frequencies, sorted in descending order

The hyperbolic curve reflects the fact that there is a small vocabulary which accounts for a large part of tokens in the text. These words are mainly function words. Zipf explains the hyperbolic distribution by what he calls the "least effort principle," assuming that it is easier for a writer to repeat certain words instead of using new words.

The Zipf distribution shows that the major part of the types in a text are quite rare, which poses practical problems for parameter estimation in statistical IR models: the sparse data problem. On the other hand, the reciprocal relationship between rank and frequency could be taken as a starting point for index term selection [KW04]. This idea is to sort word types according to their frequency in a corpus. As a second stem, the high frequency words can be eliminated because they do not discriminate well between the documents in the collection.

Thirdly low frequency terms below a certain threshold can be removed because they occur so infrequently that they are seldom used in the user's queries. This approach results in the reduction of an index size significantly.

Binomial distribution: The binomial distribution is one of the standard statistical distributions. It concerns the outcome of a series of (independent) Bernoulli trials. For example, one could use a t-test to investigate whether the number of occurrences of a word in a document is significantly higher than what could be expected on the basis of global, collection wide, word counts. High significance could be an indicator for a good index term [KW04]. From Zipf's law that most words occur very rarely, the assumptions required for normality approximation often do not hold because of the sparse data problem.

The multinomial distribution: With this distribution, a discrete sample space is taken, where a trial can have m outcomes instead of two in binomial. We can model the probability that each of the m outcomes occur with a frequency f_i after n trials as shown here:

$$m(f_1, f_2, f_3, \ldots f_m; n, p_1, p_2, p_3, \ldots p_m) = \frac{n!}{f_1! f_2! f_3! \ldots f_m!} \quad {p_1}^{f_1} {p_2}^{f_3} p_3 \ldots {p_m}^{f_m}$$

where $\sum_{i=1}^{m} p_i = 1$ and $\sum_{i=1}^{m} f_i = n$

The above equation can be formulated as follows:

$$m(S) = \frac{n!}{\prod_{t=1}^{m} f_t!} \prod_{t=1}^{m} p_t^{f_t}$$

where m(S) denotes the probability that the sentence S is drawn from a multinomial distribution.

The probability of a certain sequence of events (assuming that the events are independent) can be modeled by the multiplication of the probabilities of the individual events as shown here:

$$P(T_1, T_2, \ldots, T_n) = \prod_{i=1}^{n} P(T_i)$$

A multinomial distribution is a word unigram model application in an example, which corresponds to a zeroth order Markov Model, without any state history. The multinomial distribution intuition is that the probability of relevance of a document with respect to a query can be modeled by the probability that the query is generated by a unigram model of which the parameters are estimated from the document. This can be explained as follows: for each document we build a small statistical language model and estimate the probability that it generated the query

The Poisson distribution: The Poisson distribution is used to model the number of occurrences of a certain random event in fixed size samples.

The Poisson distribution is described as follows:

$$p(k; \lambda_i) = e^{\lambda_i} \lambda_i^k / k!$$

where p(k; λi) is the probability that a certain event i occurs k times in a unit. The Poisson distribution has the interesting property that both expectation and variance are equal to λi. The Poisson distribution is a limit of the binomial distribution where the number of trials approaches to infinity and the probability p is approaching zero, while n.p remains equal to λ_i .

The Poisson distribution has been used in IR to model the distribution of terms over documents, i.e. we apply the model to predict the probability of the term frequency k of a certain term i in a random document: Pi(k) = p(k; λi). The parameter λi is the average term frequency of term i in

the collection which is equal to the global term frequency gtf_i (number of occurrences of term i in the collection) divided by the number of documents.

The Poisson distribution makes the following assumptions which do not hold for actual text data [KW04].

1. The probability of more than one occurrence of a term is much smaller than the probability of one occurrence. In reality, when a term is used, it is often used more than once (burstiness). The fact that terms are not independent is an assumption of Poisson. The deviation between predicted and observed frequency is especially prominent for content terms, which are of prime importance for IR.

2. Poisson models the frequency of occurrence in a fixed interval. In reality however, the length of documents in a collection is extremely variable, since length differences of a factor of 100 or more do occur quite frequently.

The 2-Poisson model: This model provides a better fit of the term frequency distribution of content terms. It is assumed that a collection of documents can be divided in to two classes, a document is either about a certain term or it is not. Both document classes are modeled by a Poisson distribution, but in this case, the probability of a term *i* occurring *k* times is modeled by combining the estimates from both models as shown here:

$$2p(k;\lambda_1,\lambda_2) = \alpha e^{\lambda_1}\frac{\lambda_1^k}{k!} + (1-\alpha)e^{\lambda_2}\frac{\lambda_2^k}{k!}$$

where λ_1 and λ_2 are the average number of occurrences in the two class documents, α is the probability that a document is relevant. The 2-Poisson model postulates that a word can either be of central importance for the content of a document, or can occur spuriously and should not be considered as an index term.

The Conceptual scheme of the IR process is shown in the Figure 3.2.

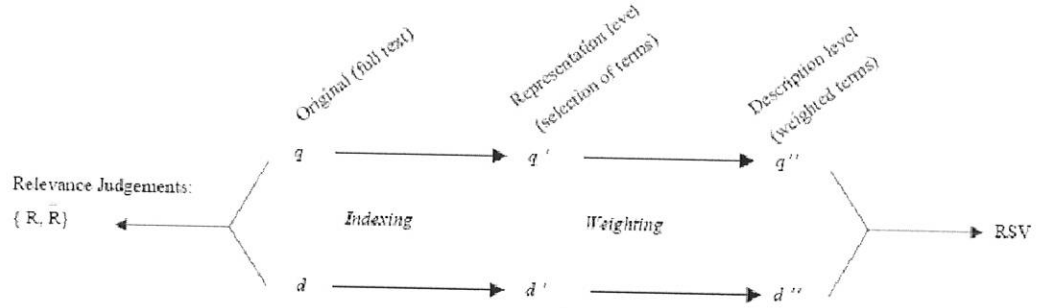

Figure 3.2 Conceptual scheme of IR, (Fuhr, 1992). d = document, q = query.

The IR task consists of a user that poses a certain query q, a collection of documents $d1, d2, ..., dN$ and an IR system. The *indexing* process consists basically of term selection, because conventional automated IR systems work with full text documents in a post-coordinated retrieval setting. The indexing process thus extracts the representations q' for the query and d' for each document. This representation level is used by the classical Boolean retrieval model, more advanced IR models apply term *weighting*, yielding the descriptions q" and d". Finally, the IR system applies a matching function $R(q_i,d_j)$ which computes a ranking score (retrieval status value: RSV) for each document d_j given a query q_i. Apart from the query and document descriptions, the ranking function usually uses global collection statistics. The results of the retrieval process can be evaluated by judging the relevance relation between the document and the query [KW04].

IR models: As described by Kraaji [KW04] the three main IR models are:

- Logical models
- Vector space models
- Probabilistic models

The best known a logical retrieval model is Boolean retrieval model. The query q can be expressed using index terms and operators from the Boolean algebra: conjunction, disjunction and negation. These logical operators have an intuitive set-theoretic semantics: each index term refers to a set of documents indexed by that term. The AND operator restricts the query result to the intersection of two sets, the OR operator yields the union and the NOT operator provides the difference between the sets.

In the vector space model the relevance of a document d for a query q is defined as a *distance* measure in a high-dimensional space, therefore vector space models could also be called algebraic models. The distance measure actually serves as a metric to compute the *similarity* between queries and documents. In order to compute this similarity measure it is necessary to first project documents and queries in the high-dimensional space defined by the vocabulary of index terms.

The classical probabilistic models exploit the different distributions of terms in the class of relevant and the class of non-relevant documents. They calculate query term weights which are directly related to the term in question present in a relevant document. Recently, another probabilistic approach to IR based on statistical language models has proven quite successful. The intuition here is that the probability that a document is relevant with respect to a query can be modeled by the probability that the query is generated by a combination of two language models: a model estimated on the document in question smoothed by a model which is estimated on the complete document collection [KW04].

Primary data structure in most IR system is an inverted index [MS99]. Inverted index is a data structure that lists (for each word in the collection) all documents that contain it and the postings and frequency of occurrence in each document. An inverted index makes it easy to search for lists of a query word. One just goes to the part of the inverted index that corresponds to the query word and retrieves the document listed there. Inverted index also contains all occurrences position in the document. An inverted index with position information lets users search for phrases.

In some IR systems all words are not represented in the inverted index. A stop list or function words, which are unlikely to be useful for searching, are excluded from the index. A stop list has the advantage of reducing the size of the inverted index. According to Zipf's law a stop list that covers a few dozen words can reduce the size of the inverted index by half [MS99].

The semantic approach of IR is based on knowledge of the syntax/semantics of the natural language in which the document text is written. This approach attempts to address the structure and meaning of textual documents directly instead of using only statistical measures for representation.

One aspect of semantic approach deals with interpreting meaning at the level of clauses and sentences, rather than just analyzing individual words. Disambiguation of words having multiple senses is a semantic approach where a word can only be disambiguated in the context of the phrase, sentence, or larger text unit in which it occurs.

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. For example, documents related to an IR query sometimes contain only the synonyms of the query words instead of the query words themselves. A simple IR system with no knowledge of synonyms fails to recognize the relevance of these documents to the query. In such cases IR systems can be improved by considering the synonyms of the query words as part of the IR query. However, only relevant synonyms of the query words in the given context contribute useful information to the query to which this thesis work pertains.

Research has been done in natural language processing in various languages to support the development of efficient information retrieval (IR). In Amharic language processing for instance, stemming [NW02], Part of Speech Tagging [ME01], and Parsing [AB00] are among the efforts to develop and support IR and other language technology systems.

3.2. Natural Language Processing (NLP)

Natural language processing (NLP) is the branch of computational linguistics which is concerned with building models and tools that process human language. Theoretical linguistics is concerned with describing (and explaining) expressions of natural language using a rule-based symbolic, Hidden Markov Model framework. Traditionally there are several levels of linguistic analysis distinguished as follows:

Morphology: is concerned with assigning an internal structure to words.

Syntax: is concerned with assigning an internal structure to sentences in terms of grammatical relationships.

Semantics: is concerned with interpreting the meaning of a sentence in terms of an unambiguous formal language.

Discourse analysis: is concerned with the analysis of language phenomena that exceed the sentence level e.g., referring expressions.

3.3. Word sense disambiguation (WSD)

3.3.1. Ambiguity

Definition: 1. Ambiguity is the property of being ambiguous, where a word, term, notation, sign, symbol, phrase, sentence, or any other form used for communication, is called ambiguous if it can be interpreted in more than one way [AW08].

2. Something is ambiguous when it can be understood in two or more possible senses or ways. If the ambiguity is in a single word it is called lexical ambiguity. In a sentence or clause, it is called structural ambiguity [CQ03].

Classification of Ambiguity

With the given definition ambiguity is context dependent where same communication may be ambiguous in one context and unambiguous in another context. For a word, ambiguity refers to an unclear choice of different definitions which may be found in dictionaries.

As described by Qing-liang [ZQ05] ambiguities can be divided into two main categories: Ambiguous words and ambiguous sentences. Each main category has different types of ambiguities.

Ambiguous words:

Ambiguous words can be further classified as homonymy and polysemy

1. **Homonymy:** A case of homonymy is one of an ambiguous word, where different cases are related to each other in any way. Homonymy can be further divided into three types.

(i) **Homographs:** words that have the same spelling but differ in sound and meaning are called homographs, e.g. bow/baU/ v. (bend the head and body in respect)—bow/bEU/n. (a device for shooting arrows).

(ii) **Homophones:** Words that are identical in sound but differ in spelling and meaning are called homophones, e.g. air—heir, see—sea.

(iii) **Full homonyms:** Words that are identical in sound and spelling are called full homonyms, e.g. ball n. (a round object used in games)—ball n. (a gathering of people for dancing).

2. **Polysemy:** A case of polysemy is one where a word has several clearly related senses, e.g. mouth (of a river vs. of an animal), the two senses are clearly related by the concept of an opening from the interior of some solid mass to the outside, and of a place of issue at the end of some long narrow channel.

Ambiguous sentences:

A sentence is ambiguous if it has two (or more) paraphrases which are not themselves' paraphrases of each other [ZQ05]. For example the sentence "Visiting relatives can be boring" can be paraphrased into two ways:

- It can be boring to visit relatives.
- Relatives who are visiting can be boring

Word sense disambiguation is the problem of selecting a sense for a word from a set of predefined possibilities [MP05]. The possibilities include Sense Inventory which usually comes from a dictionary or thesaurus, knowledge intensive methods, supervised learning, and (sometimes) bootstrapping approaches.

Machine translation, information retrieval, question answering and knowledge acquisition are some of the applications of word sense disambiguation [MP05].

The automatic disambiguation of word senses shown in Figure.3.3 is an "intermediate task" which involves two steps [IV98]:

- The determination of all the different senses for every word relevant to the text or discourse under consideration.
- a means to assign each occurrence of a word to the appropriate sense

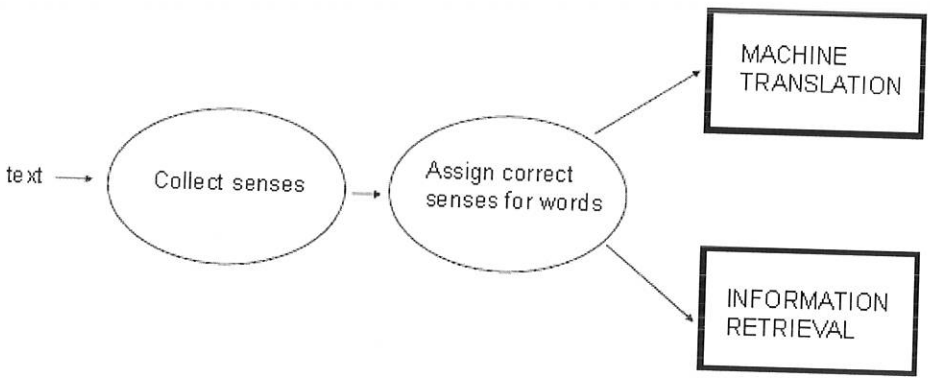

Figure 3.3 The task of word sense disambiguation (Nikumen, 2007).

In order to address ambiguity in natural languages, there are three main approaches: knowledge-based, corpus-based and hybrid approaches. Each will be discussed in turn.

3.3.2. Corpus-Based approach

A corpus based approach extracts word senses from a large annotated data which is a sense tagged. The possible means of identifying attributes of senses of the ambiguous words are the distributional information, context and additional knowledge that has been annotated in the corpus or added during preprocessing. Distributional information about an ambiguous word

refers to the frequency distribution of its senses. Context is composed of words found nearest to a certain word which can be to the left or the right of a word. These contexts can be refereed as collocational or co-occurrence information. With corpus-based additional sources, such as lemmas, part of speech (POS), syntactic annotations, etc can be used. Based on these approaches many research have been done.

The major problem with this approach is the availability of sense tagged corpora where raw corpora do not indicate which sense is applicable for a word in a given text. In order to use corpora as information resource for WSD, they have to be annotated with word senses, which is a very labor intensive process. A search of the literature indicates that there has not been a lot of sense-tagged material available publicly, especially for languages other than English. One approach to solve the problem is to manually sense tag corpora using a given sense inventory such as WordNet's hierarchies or dictionary sense listings. The other approach is the application of less data-intensive methods with respect to annotated data approaches to WSD, such as bootstrapping or unsupervised techniques. Within the corpus-based approaches there are two possibilities: supervised and unsupervised WSD.

3.3.3. Supervised approaches

The supervised approach uses the annotated data for training which basically refers to a classification task. In the training phase on the disambiguated corpus, information about context words and other knowledge sources included in the system as well as distributional information about the different senses of an ambiguous word are collected in the testing phase, the sense with the highest probability or similarity computed on the basis of the training data selected. Training and evaluating such algorithms requires a sense tagged corpora.

Exemplar Based, rule based and probabilistic approaches are some of the techniques used in supervised WSD systems. Use of different probabilistic classifiers is the other technique used in supervised WSD. Due to the relative simplicity, naïve Bayes has been frequently applied in WSD with good results.

3.3.4. Unsupervised approaches

Unsupervised approaches have been applied for a raw text material where annotated data is only needed for evaluation purposes. These approaches refer to the clustering task rather than classification task sense tagging, which is not possible in a completely unsupervised way some characterization of senses must be provided.

Disambiguation as sense discrimination can be achieved through unsupervised clustering. This involves clustering the contexts of an ambiguous word into a number of groups and discriminates among them without labeling them [PB97], [SC98].

Bootstrapping is used in the unsupervised approach. Bootstrapping means that a small corpus is sense tagged by hand and statistical information is extracted from the context of these occurrences. Iteratively, large amounts of unlabeled data are then labeled using the information, and the new correctly labeled data is then used as input to collect statistical information. Using this method labeled data can be acquired quickly and incrementally. The quality of bootstrapping is assured through hand correction. The bootstrapping method achieved good results [KY98], [MM01], [YD95].

The other means to solve the need for hand tagged data are parallel corpora [DI94], [[NWC03]. In a bilingual corpus, word correspondences are identified and the translations are used as sense tags. As noted by Ng et. al [NWC03], tying sense distinctions to the different translations in a target language introduces a more data oriented view of sense distinction and also adds some

more objectivity to defining senses [NWC03]. Using parallel corpora mainly include the size of the parallel corpus required with the quality of the word alignment. The other problems with this method are the limited coverage (words do not appear in the corpus or lack of examples for secondary senses) and mutual ambiguity across languages. Another problem noted by Diab and Resinik [DR02], is that even though a word-sense combination is translated with some consistency into a relatively small set of words, it rarely contains unambiguous words. They therefore assume that words having the same translation at least share a dimension of meaning if not the exact sense [DR02]. Diab and Resinik use WordNet as a sense inventory for the translations and an algorithm which reinforces the correct sense of a word by the semantic similarity of other words with which it shares those dimensions of meaning.

3.4. WSD and Information Retrieval

As noted by Krovetz and Croft, word senses provide a significant separation between relevant and non-relevant documents provided that other factors contribute to the performance improvement in applying disambiguation [KC92]. Krovetz and Croft have done experiments on two standard test collections of information retrieval, communication of the ACM (CACM) and Time Magazine with four types of information retrieval mechanisms:

1. Coordination match: Base line; documents quickly scored according to number of words that matched the document
2. Frequency weighting: A standard TF.IDF weighting bases on the probabilistic Model
3. Sense weighting: using the standard method by replacing the IDF component by a sense weight
4. Combined: Modification of frequency weighting to incorporate a term's degree of ambiguity

Experimental result shows that word sense weighting improved retrieval effectiveness by a small amount in one collection, and had no difference in the other collection. Their reasoning regarding the results was that general vocabulary words used which led to that anomalous frequency distribution can be useful for detecting domain specific word senses. In their analysis they stated that stemming in queries couldn't capture all of the variants a word can have such as actor, actress which can be stemmed to ''act''. Describing all of the pitfalls in their experiment, they argue that word sense disambiguation will have the greatest impact on a search that requires a high level of recall.

Replacing the words in a standard ''bag of words'' text representation is used for information retrieval application. Text is analyzed into words where each word occurrence is annotated by a disambiguation algorithm [SP95]. Their disambiguation algorithm was tested on queries of 51-75 of the category B TREC-1 collection which contains 170,000 documents from the wall street journal, the queries containing 1013 different terms. They considered 50 occurrences per sense as the minimum amount of data necessary to distinguish between senses. Using this statistic, the number of senses for word w was defined as f/50, f indicating the number of occurrences of w in the corpus. The 1013 query terms in the corpus were clustered with the predetermined number of clusters. Context vectors for each term of a query in the corpus were computed and assigned to the closest centroid (average of context vectors). Then documents in corpus were indexed with the automatically assigned senses rather than terms. The results of the word based, sense based and combined based searches are shown in Table 3.1.

Table 3.1 Experimental result (Schutze and Pederson 1995).

recall	word-based	sense-based		combined	
at 0.00	0.693	0.788	+13.7	0.854	+23.2
at 0.10	0.540	0.629	+16.5	0.645	+19.4
at 0.20	0.453	0.503	+11.0	0.506	+11.7
at 0.30	0.385	0.427	+10.9	0.434	+12.7
at 0.40	0.315	0.340	+7.9	0.347	+10.2
at 0.50	0.264	0.260	-1.5	0.291	+10.2
at 0.60	0.206	0.198	-3.9	0.229	+11.2
at 0.70	0.165	0.145	-12.1	0.174	+5.5
at 0.80	0.118	0.110	-6.8	0.129	+9.3
at 0.90	0.085	0.076	-10.6	0.091	+7.1
at 1.00	0.061	0.057	-6.6	0.059	-3.3
average precision (not-interpolated) over all real docs.					
	0.299	0.321	+7.4	0.342	+14.4

Schutze & Pederson concluded that word sense-based retrieval with the average precision of 11 points of recall increased by 4% where the combined ranking of word-based and sense-based retrieval incorporating WSD achieved a relative improvement of 14% over the basic vector similarity model as shown in figure 3.4.

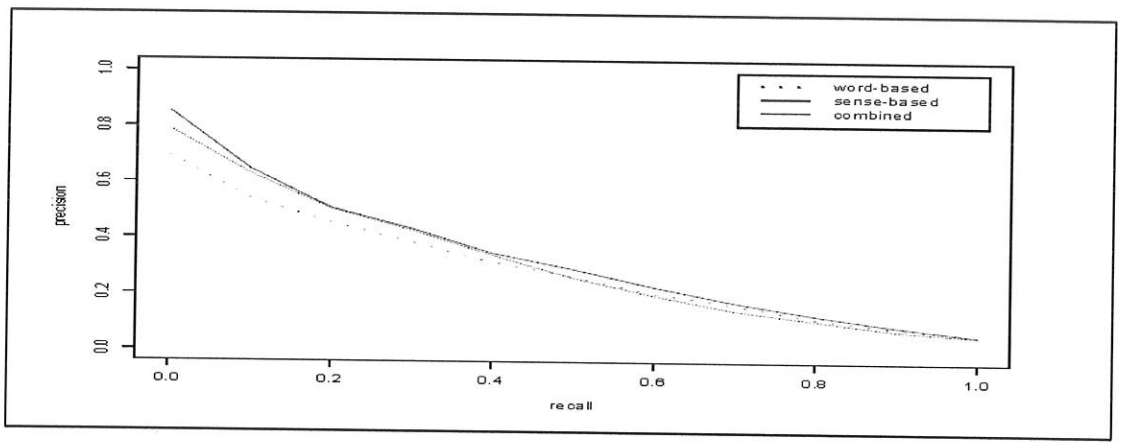

Figure 3.4. Precision- Recall graph [SP95].

Sanderson (1994) used artificial pseudo-words to measure the effects of ambiguity on CACM, Cranfield, and TREC-B collections. Introducing artificially ambiguous terms to the collections he measured the retrieval performance against a baseline for the original collection. According to his best result with these collections, he found that queries with one or two terms are readily affected by ambiguity. For longer queries there was little measurable effect as longer queries can be considered as disambiguated which concurrence word of the query terms. Although it has an effect on one to two terms, errors of disambiguation more than 10% degrades IR performance. The result of his experiment on 5 words pseudo words is shown in Figure 3.5 with f-max/length of query words [SM94].

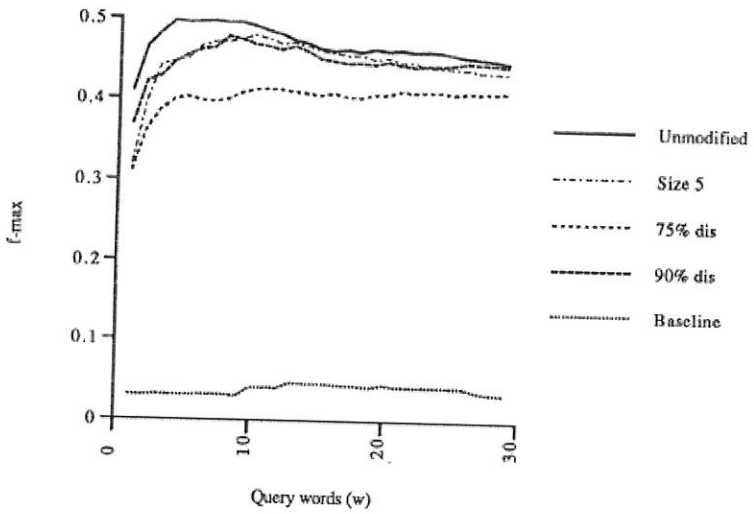

Figure 3.5 results of IR base line and application of errors (Sanderson 1994).

3.5. Amharic Language

Amharic which belongs to the Semitic family of languages, is written in the unique and ancient Ethiopic script inherited from Geez language). Amharic is the second most spoken Semitic language in the world, next to Arabic. Amharic is spoken by roughly 30% of the population as a

first language, and an additional 20% as a second language, totaling about half of the population in Ehiopia which is estimated 67 million according to the 1998 census. Amharic is the language of some 2.7 million people living in Egypt, Israel and Sweden, and is spoken in Eritrea. In general, there are more than 34 million speakers of Amharic [MMJ].

As Bloor [BT06] described in his paper, scholars who believe that Ethiopic is derived from Sabean claim that when Geez adopted the Sabean system, a number of symbols were dropped. This didn't happen when Amharic and Tigrinya took up the system from Geez. They kept all the symbols (later adding 8 more consonants not used in Geez, even later adding a modified form of the /b/ symbol to represent the foreign phoneme /v/). This resulted in a considerable redundancy, particularly in the case of Amharic, which lacks several consonant sounds found in the phonology of Geez. Because of this 4 distinct sets, which each set contains 7 symbols can represent the sound /h/+vowel: ሀ፣ ሐ፣ ኸ ፣ ኀ; two sets represent /s/: ሰ, ሠ and two /s'/ (ejective) ጸ, ፀ. The 44 labialized consonant symbols (/kwa/, etc) are also arguably redundant, wholly or partially, and there is considerable variation in the spelling of many words that may involve them. 2 sets of 7: አ, ዐ are also considered consonant-less vowels as a redundancy [BT06].

Amharic has thirty four consonants with seven vowels [BY87]. Redundancy of Amharic symbols results in some confusion of written Amharic words. For example ፀባይ, ጸባይ literal meaning "conduct"; ሀይማኖት, ኃይማኖት, ሐይማኖት, ሃይማኖት literal meaning "religion" are various writing forms of a single word with different symbols existing in the Amharic language.

Bloor [BT06] explained that Ethiopic writing system was passed on from Geez to Amharic and Tigrinya. The writing system of the Amharic language is syllabary, which uses one character per syllable.

Unlike Arabic and Hebrew, Amharic is written from left to right. The Amharic system does not have distinctions between upper case and lower case letters.

Word boundaries were originally unmarked and later indicated by two vertically placed dots like a colon (፡) [BT06]. Nowadays, letter spaces are used for the separation of words especially to process Amharic documents electronically. However frequent use of a colon for the separation of words still prevail in hand written form. Sentence boundary is indicated by four dots (፨). The old form of question mark, three vertically placed dots (፧) has been superseded by the question mark (?). Quotes are usually in the French style <<...>> and parentheses and exclamation marks are as in the Roman system: (...),!. There are also additional punctuation marks: Paragraph separator (፠), comma (፣), semi colon (፤) and preface colon (፥) in the Amharic writing system [ETUN].

Word units are classified as phoneme, morpheme, root, stem and word [NW02]. The morphemes are of two types either free morpheme or bound morpheme. Free morphemes are those which can have meaning by themselves where as bound morphemes do not have a meaning by themselves unless attached to free morphemes [BY87].

The stem is a sequence of consonants or sequence of consonant-vowel a root which is the basis of words derivation is a sequence of consonants. Most Amharic words and their variants are derived from the root words. Words in Amharic are divided into content-bearing and non content-bearing (stop words).

Amharic language ambiguities:

In Amharic, causes of different types of ambiguities have been identified. The most common causes are phonological, lexical, structural, referential, semantic and orthographic. Some of the causes cover a range of meanings, where lexical ambiguity covers four types of ambiguity [GA01]. Amharic ambiguity types with examples as given by Getahun are summarized in Table 3.2.

Table 3. 2 Types of ambiguities of Amharic

Ambiguity type	Sub type	Example (Amharic phrases)	English Meaning	
			1	2
phonological ambiguity		ደግሰው ነበር	They had made a preparation for for a banquet	He was kind
Lexical ambiguity	Categorical ambiguity	አክርማ ሰጠችኝ	She gave me akirma (kind of grass)	She gave me some thing after delaying it for some time
	Homonymy	በወራ አልፈታም	I will not be released in a month	I will not get frustrated by any rumour
	Homophonous affixes	ቤቱ ፈረሰ	The house is destroyed	His house is destroyed
Structural ambiguity		የአበሻ ታሪክ አስተማሪ	A person who teaches Abyssinian history	An Abyssinian who teaches history
Referential ambiguity*		ካሣ ለአስቱር ወደ ጎጃም እንደሚሄዱ ነገራት		
Semantic ambiguity	**Polysemic**	መብራቱ ጠፋ	The light went off	Mebratu (person) disappeared
	Idiomatic	በሬ ወለደ		
Orthographic Ambiguity		መኪናው ይሰራል	The car works	The car will be repaired

*Referential ambiguity:

A referencial ambiguity may arise when a pronoun has more than one possible antecedents, having as many readings as there are antecedents. The referencial ambiguity example on the table has an English equivalent "Kasa told Aster that they will go to Gojjam." The pronoun "they" can have five different variants (references): Kassa and Aster, or with others, or with either of them and others [GA01].

Chapter Four

Design of word sense induced information retrieval protoype

For the experimentation of our work we have designed the system where it is composed of processing components implemented with the retrieval system. Since Amharic is poor in resources which are in need for disambiguation such as WorldNet, annotated data , lexicon etc we design the prototype which can be implemented using only the raw corpus which we select for the experiment.

The design of the system is shown in Figure 4.1 below followed by explanation of the functionality of each component.

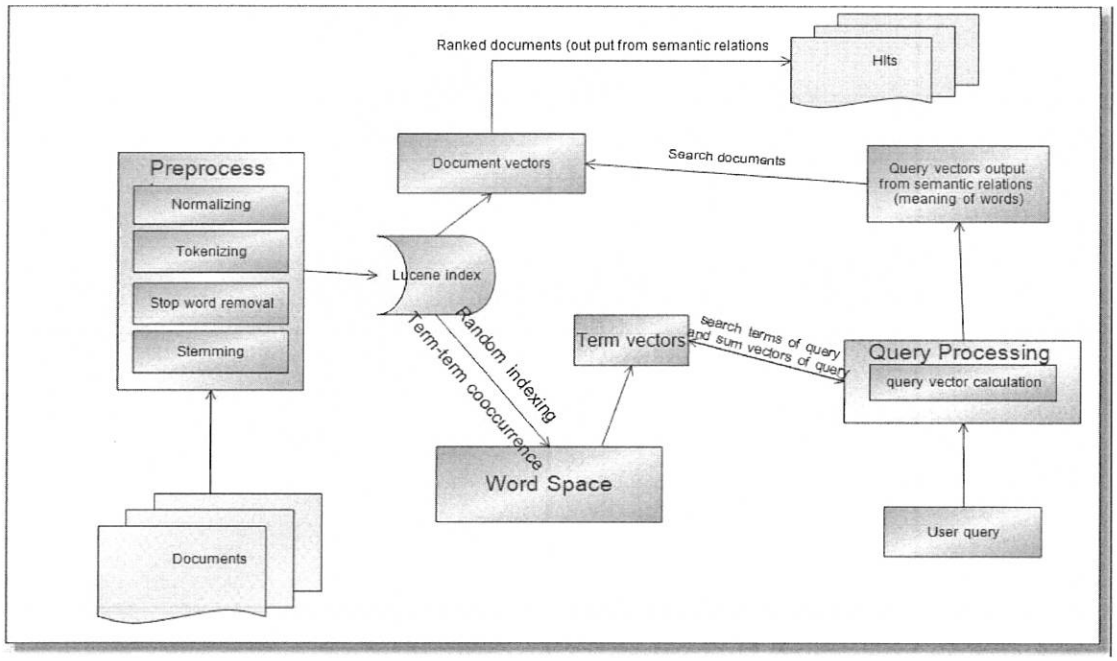

Figure 4.1 Design of the system

4.1. Preprocessing

Tokenizing, stemming, normalizing, and stop word removal are performed at this stage. Each of these processes are described below.

4.1.1. Normalization

The normalization component of the preprocessing step handles the problem of Amharic text writing with consideration of punctuation marks. The main function of the component is to replace the alphabets that have the same pronunciation and use with one of the alphabets. For example the letters ሀ፣ሐ፣ኀ፣ኸ are represented by letter ሀ and its variants with all their respective variants. The normalization algorithm is shown bellow.

While (! End of text)

1. *Set a buffer to empty*
2. *Read a character*
   - I. *If the character is one of* ሀ,ሐ,ሰ,ሠ,ኀ,ኘ,ዐ,ጸ,ፀ, or their orders *call a component that can handle their replacement.*
     - i. *add the character to the buffer*
     - ii. *go to step 2*
   - II. *Else if the character is "/" or"."*
     - i. *call a component that can handle such characters*
     - ii. *return the string as a word*
     - iii. *Go to step 1*
   - III. *Else if the character is a white space or one of the punctuation marks*
     - i. *Return what is in the buffer as a word*
     - ii. *Go to step1*
   - IV. *Else*
     - i. *Add the character to the buffer*
     - ii. *Go to step 2*

Character replacement algorithm

Read a character

- ***If the character is one of ሀ፣ሐ፣ኀ _replace them with ሀ;_***
  - ***Return the replaced character***
- ***If the character is one of _ሰ_or ሠ replace them with ሰ__***
  - ***Return the replaced character***
- ***If the character is one of ኧ or_ዐ_replace them with ኧ***
  - ***Return the replaced character***
- ***If the character is one of _ፀ_or ጸ replace them with ጸ_ _***

4.1.2. Tokenization

Tokenization is the process of splitting stream of characters in to raw terms or tokens. This process detects the boundaries of a written text. Tokenizing of a given text depends on the characteristics of language of the text which it is written.

The Amharic language has its own punctuation marks which demarcate words in a stream of characters. The tokenization of text on this component uses Amharic punctuation marks and white spaces. The Amharic punctuation marks used in the tokenization process are shown in appendix III.

4.1.3. Stop word removal

Since stop and function words have no discriminating power for information retrieval they must be removed. The Amharic language has its own stop words. Since there is no a standard stop words for Amharic language we took list of stop words shown in appendix VI including numbers from 1 to 865 (article numbers) from our corpus which is considered to be no of importance for indexing for our work.

4.1.4. Stemming

Morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. For this reason, a number of so-called stemming Algorithms, or stemmers, have been developed, which attempt to reduce a word to its stem or root form. Thus, the key terms of a query or document are represented by stems rather than by the original words. This not only means that different variants of a term can be conflated to a single representative form, but it also reduces the dictionary size, that is, the number of distinct terms needed for representing a set of documents. A smaller dictionary size results in a saving of storage space and processing time.

Amharic is morphologically rich language. Amharic can have many words with the attachment of different affixes to a stem. Stemming can be applied to both inflectional morphology and derivational morphology or on either of the two. Derivational morphology usually results in a change in class of word which may result in some loss of semantic. This semantic loss of a word may create a negative effect on the performance of information retrieval system. For example the Amharic word ዳኛ (a judge or an arbiter) and ዳኝነት (judging) has the same stem ዳኝ. But these two words have different meanings. Amharic language includes some prefixes and combination of prefixes and suffixes which create negative meaning when they are applied to a given stem. This kind of semantic loss is not usually witnessed during inflectional morphology, which usually involves grammatical features such as, singular/plural, tense.

For this thesis work a stemming algorithm which is used by Tessema Mindaye [TM07] was adopted.

4.2. Word Space

The idea of word space model is to use distributional statistics to generate high dimensional vector spaces. In word space words are represented by context vectors whose relative directions

are assumed to indicate semantic similarity. The assumption comes from the distributional hypothesis stating that words with similar meanings tend to occur in similar contexts [MS05].

According to this hypothesis, observing two words that constantly occur with the same contexts, we are justified in assuming that they mean similar things. Note that the hypothesis does not require that the words co occur with each other; it only requires that the words co-occur with the same other words.

In the standard word space methodology, the high-dimensional vector space is produced by collecting the data in a co-occurrence matrix F, such that each row F_w represents a unique word w and each column F_c represents a context c, typically a multi-word segment such as a document, or another word. In the former case, where the columns represent documents, we call the matrix a words-by-documents matrix, and in the latter case where the columns represents words, we call it a words-by-words matrix. LSA is an example of a word space model that uses document-based co-occurrences, and Hyperspace Analogue to Language is an example of a model that uses word-based co-occurrences.

The cells F_{wc} of the co-occurrence matrix record the frequency of co occurrence of word w and document or word c. As an example, if we use document-based co-occurrences, and observe a given word three times in a given document in the data, we enter 3 in the corresponding cell in the co occurrence matrix. By the same token, if we use word-based co-occurrences and observe that two given words occur close to each other five times in the data where the window length of co-occurrence is the document length, we enter 5 in the corresponding cell. The frequency counts are usually normalized and weighted in order to reduce the effects of high frequency words.

The point of the co-occurrence matrix is that the rows *Fw* effectively constitute vectors in a high-dimensional space, such that the elements of the vectors are (normalized) frequency counts, and the dimensionality of the space is determined by the number of columns in the matrix, which is identical to the number of contexts (i.e. words or documents) in the data. We call the vectors *context vectors*, since they represent the contexts in which words have occurred. The context vectors are representations of the distributional profiles of words, which means that we may define distributional similarity between words in terms of vector similarity.

The distributional hypothesis allows for very straight forward computation of semantic similarity between words. We simply compare their context vectors using any of a wide range of possible vector similarity measures, such as the cosine of the angles between vectors, or the City-Block Metric [MS05]. The co occurrence matrix is generated using the random indexing algorithm which is described in the following section.

4.2.1. Random Indexing

As discussed in section 4.3 the rows (and the columns) of the frequency matrix can be interpreted as multi-dimensional context vectors where the elements are (normalized) frequency counts, and the dimensionality is the number of contexts in the text data. The inherent problem with using co-occurrence representations in natural language processing is that the size, or dimensionality, of the representations will grow with the size of the data. This means that the model will not scale very well, and that the co-occurrence matrix will soon become computationally intractable when the vocabulary and the document collection grow. To make the method practically feasible, it is necessary to reduce the dimensionality of the matrix.
Dimension reduction techniques (e.g. SVD) tend to be very computationally costly. Where efficiency is considered important, it may not be practical to use such techniques. Furthermore,

dimension reduction is a one-time operation, with a rigid result. New data cannot be added to the model once a dimension reduction has been performed. As an alternative to vector-space models that use local co-occurrence matrices and some form of dimension reduction, Random Indexing, which accumulates a words-by-contexts co-occurrence matrix by incrementally adding together distributed representations in the form of high-dimensional (i.e., on the order of thousands) sparse random index vectors can be used. The index vectors contain a small number of non-zero elements, which are either +1 or -1, with equal amounts of both. For example, if the index vectors have eight non-zero elements in1,800 dimensions, they have four +1s and four -1s the rest being 0 [MS05].

Magnus Sahlgren [MS06],described the RI methodology and dimension reduction as follows:
RI represents a novel way of conceptualizing the construction of context vectors. Instead of first collecting co-occurrences in a co-occurrence matrix and then extracting context vectors from it, RI incrementally accumulates context vectors This methodology can be used to assemble both a words-by-documents and a words-by words co-occurrence matrix. The ability to use both types of contexts makes RI unique in word-space research.
RI accumulates context vectors in a two-step operation:

1. Each context (i.e. each document or each word type) in the text is assigned a unique and randomly generated representation called an index vector. In RI, these index vectors are sparse, high-dimensional, and ternary, which means that their dimensionality r is on the order of thousands, and that they consist of a small number (ϵ) of randomly distributed non-zero elements (as many +1s as -1s). Each word also has an initially empty context vector of the same dimensionality r as the index vectors.

The context vectors are then accumulated by advancing through the text one word token at a time, and adding the context's (the surrounding word types' or the current document's) r-

dimensional index vector(s) to the word's r-dimensional context vector. When the entire data has been processed, the r-dimensional context vectors are effectively the sum of the words' contexts.

If we then want to construct the equivalent of a co-occurrence matrix, we can simply collect the r-dimensional context vectors into a matrix of order w x r, where w is the number of unique word types, and r is the chosen dimensionality of the vectors. The dimensions in the RI vectors are randomly chosen, and thus do not represent any kind of context (which is the case in the original co occurrence matrix) - they constitute a distributed representation. Furthermore, r is chosen to be much smaller than the size of the vocabulary and the number of documents in the data, which means that RI will accumulate (roughly) the same information in the w x r matrix as other word-space implementations collect in the w x w or w x d co-occurrence matrices, but that $r << d,w$.

The methodology described above can also be used to produce a words-by words or a words-by-documents co-occurrence matrix by using unary index vectors of the same dimensionality n as the number of contexts. These unary index vectors have a single 1 in a different position for each context. Mathematically, these n dimensional unary vectors are orthogonal, whereas the r-dimensional random index vectors are only nearly orthogonal. This near-orthogonality of the random index vectors is the key to the RI methodology. Since there are many more nearly orthogonal than truly orthogonal directions in a high-dimensional space [KA99], choosing random directions as we do when generating the index vectors can get us very close to orthogonality. This means that the r-dimensional random index vectors can be seen as approximations of the n-dimensional unary vectors.

Consequently, the r-dimensional context vectors produced by RI can be interpreted as approximations, in the sense that their mutual similarities are (nearly) equal, of the n-dimensional context vectors extracted from the co-occurrence matrix. The near-orthogonality of

random directions in high-dimensional spaces is exploited by Random Projection which rely on Lindenstrauss lemma [JL84], which states that if we project points in a vector space into a randomly selected subspace of sufficiently high dimensionality, the distances between the points are approximately preserved. Thus, the dimensionality of a given matrix F can be reduced to F' by multiplying it with (or projecting it through) a random matrix R:

$$F'_{wxr} = F_{wxd}R_{dxr}$$

The index vectors serve as indices or labels for words or documents, depending upon the type of co-occurrences we want to use. When using document-based co-occurrences, the documents are represented by high-dimensional sparse random index vectors, which are used to accumulate a words-by-contexts matrix with the following procedure:

Every time a given word occurs in a document, the document's index vector is added to the row for the word in the matrix. The procedure is similar when using word-based co-occurrences: first, we assign a high-dimensional sparse random index vector to each word type in the data. Then, every time a given word occurs in the data, the index vectors of the surrounding words are added to the row for the focus word. Words are thus represented in the co-occurrence matrix by high-dimensional context vectors, that contain traces of every context (word or document) with which the word has co-occurs.

In our work Random indexing is used for dimension reduction which is done with indexing on the fly. The idea behind using Random indexing is that word is the sum of it's contexts and context is the sum of it's words.

Algorithm of Random Indexing

1) Take a context of words
2) Generate a context index vector (context index vector generation is representing a a context vector by (+1, -1, 0) where we use 4 +1s, 4 -1s and 0 for the rest for 200 word dimensions
3) Add index to all the word vectors
4) Go to 1)

Random Indexing on the space vector model is preferred because it is fast, scalable and it is incremental which is not rigid as SVD.

Random indexing example:

የወንጀለኛ መቅጫ ሕግ can be represented as follows:

የወንጀለኛ and መቅጫ as contexts ሕግ as target word.

ሕግ = $\{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\ldots,0,0,0,0\}_{200}$ *initialized to zero vector*

መቅጫ= $\{0,1,0,0,0,-1,0,0,0,0,0,1,0,-1,0,0,\ldots,0,1,0,1\}_{200}$

የወንጀለኛ = $\{0,0,-1,0,0,1,0,0,0,-1,0,0,0,0,0,1,\ldots,-1,0,0,1\}_{200}$

ሕግ = $\{0,1,-1,0,0,0,0,0,0,-1,0,1,0,-1,0,1,\ldots,-1,1,0,2\}_{200}$ *(sum of two context vectors)*

The Random Indexing output can be used to compare words to words where it can be applied for query expansion. It can be also used for comparing documents to documents for clustering, searching; for comparing documents to words for keyword extraction.

4.3. Lucene

Lucene is an information retrieval library that is written in pure Java. It provides a core Application Programming Interface (API) for adding full-text indexing and searching functionalities for applications [LUL]. In our work it is used to index the corpus which is a collection of text files. The index output of lucene which is stored in the disk is used as input for the construction of word space.

Chapter Five

Implementation

Using lucene index the term vectors are built from lucene index of terms. 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 which 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 different meaning of an ambiguous word often belongs to different semantic categories. The context within which the sentence appears provides valuable clues for sense disambiguation. The "context" refers to the other words presence in the sentence containing the ambiguous word. These words provide valuable clues for identifying the correct meaning of the word. In most cases, these informative words occur near the ambiguous words and can be used reliably. The goal is to identify the context words for each meaning of the ambiguous word that will uniquely represent one particular meaning for that word in the given context.

A sample of term vectors is shown in appendix IV. For example, we can take a word **ወሰን** to retrieve words related to this word, there by forming a thesaurus. Word vectors are compared using cosine similarity. Since each word is represented by a vector with a suitable number of reduced dimensions (in our case which is 200), we can compare words with one another to find out if they are similar or different. To compare vectors, let vector a have coordinates (a1, ... , an) and let the vector b have coordinates (b1, ... , bn). Their scalar product is defined as the sum

$a.b = a1.b1 + a2.b2 + \cdots an.bn.$

If we divide this by the product of moduli ||a|| and ||b|| we obtain the cosine of the angle between the two vectors, which is called their cosine similarity as shown in the formula:

cos(a,b) = (a1.b1 + ... + an.bn) / (||a|| ||b||).

This enables us to find the nearest neighbors of a given word with the highest cosine similarity. The result is more similar words to form the thesaurus based upon their frequency.

The great advantage of the vector formation is that it allows us to combine words into sentences or documents by adding their vectors together. If article vectors are built into the system, the word vector with the document vector search can be used to find nearby documents in the information retrieval system.

The composite vectors are context vectors because they gather information from the context surrounding a particular word. When a query has more than two words, those query words are parsed into a number of words where each word is used to find context vectors according to parsed words. The sum of those context vectors is used to retrieve those highly related documents from the document vectors built. A sample of document vectors formed with the component of the prototype is shown in Appendix V.

As defined and described in the problem definition section of this thesis , the purpose is to retrieve documents which satisfy the user queries efficiently and effectively from a large volume of text documents stored, using of the meaning of ambiguous words. As mentioned earlier, the corpus used for this work was the Ethiopian penal code which is comprised of 865 articles. The average size of each article is 2.5 KB. These texts are legal texts where many of the words and terms used are part of a specialized legal vocabulary. The purpose of the experiment was to examine the contribution disambiguating a word which is polysemous

(having different meanings) in enhancing an information retrieval system in the Amharic language.

The user interface screenshot shown in Figure 5.2 is the same as that shown in Figure 5.1, except that the index used for the retrieval is different. The indexes used for this purpose were term vectors intended to retrieve related words with respect to the context of the query terms. Indexed document vectors are then searched for documents relevant to a user's query. The query « **የስልጣን ወሰን** » is used to retrieve documents and displayed indicates articles relevant to the query.

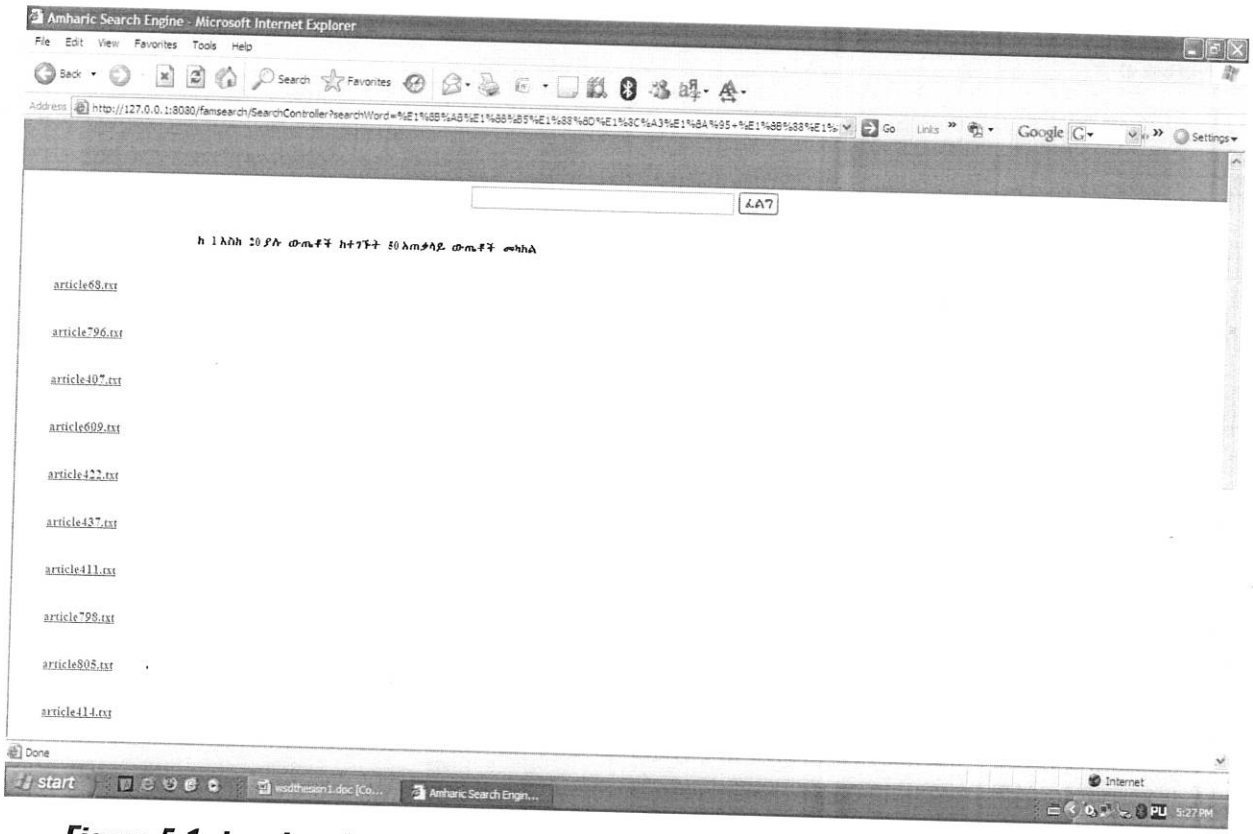

Figure 5.1 showing document retrieval snapshot for a given query

One can observe that when the above query words are used separately (number of words used to formulate query is less than 2) the result displayed is different which shows that the

contribution of summing vectors from parsed query terms has a contribution for relevancy of queries and documents to be retrieved.

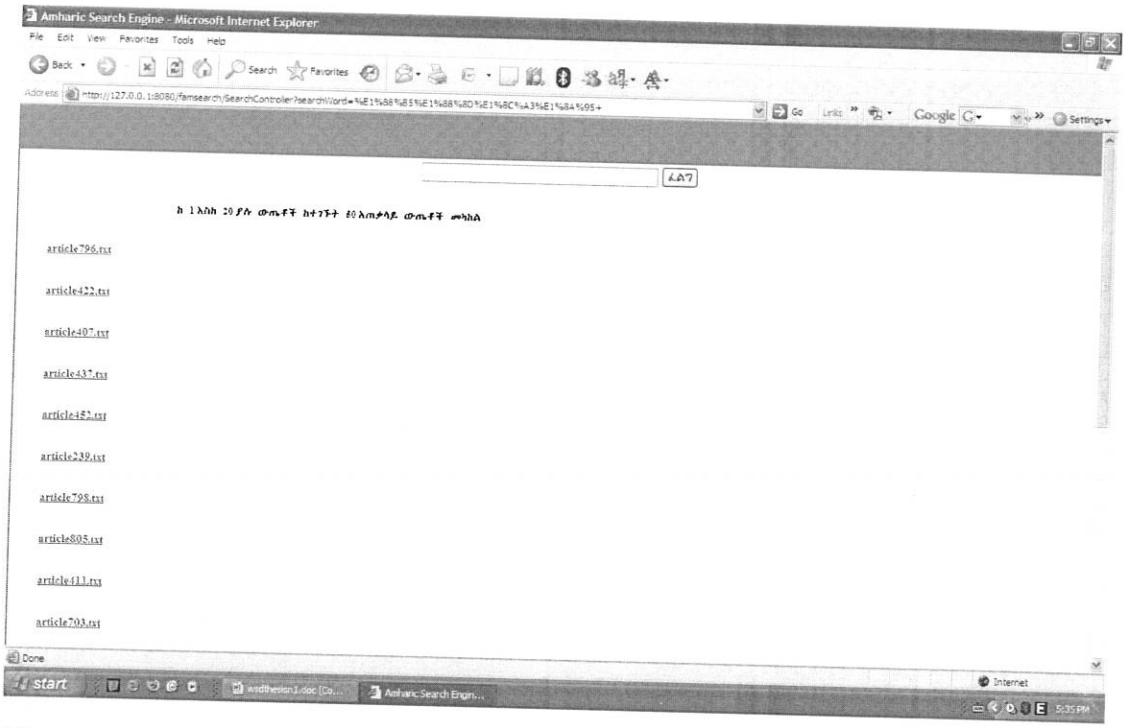

Figure 5.2 showing results for query «ስልጣን»

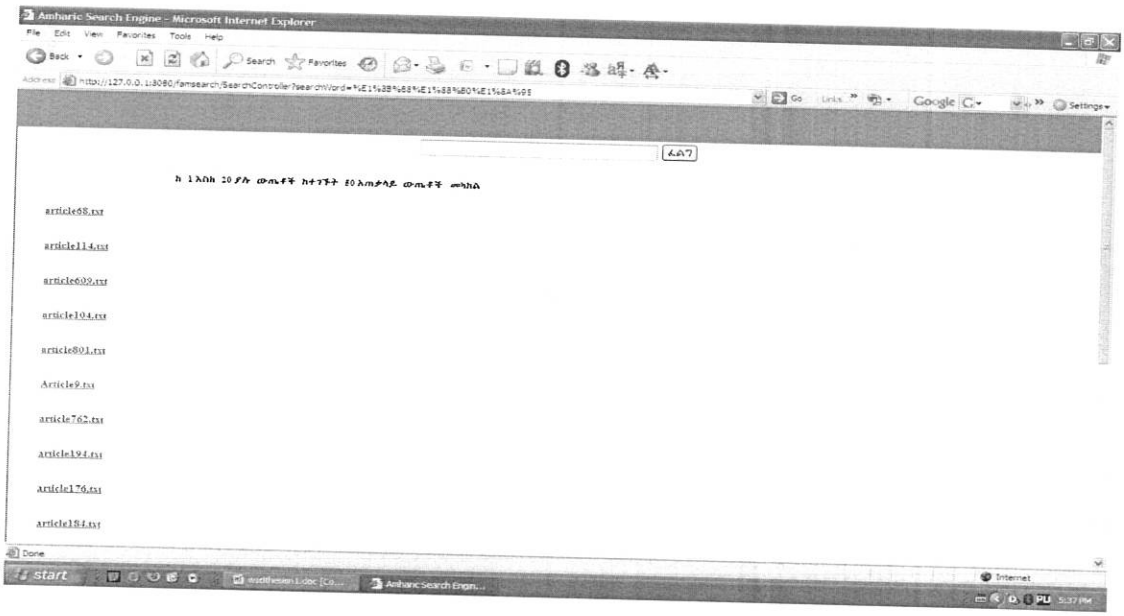

Figure 5.4 showing results for query « ወሰን »

In the screenshots shown above, it is evident that when a combination of more than two words for a query is used, the result differs. This provides the most relevant documents ranked and displayed in descending order, which is the result of the combination of the individual words context vectors and sum of these context vectors. The system searches for the most relevant documents by calculating the sum vector with the document vector by applying a cosine similarity measurement method. Highly similar documents have high cosine value whereas dissimilar documents have low cosine values. Note that a document compared to itself has value 1. To compare the results of word meaning induction for retrieval system with the others. We used context vector sums and the Lucene default ranking method for information retrieval.

5.1. Experimental Results

For the evaluation of our work, we used another system with a Lucene default ranking method. Our system included inducing word meanings where ranking is done by the cosine relationship of term vectors and document vectors. The system queries used needed to be selected based on the ambiguities of words found in the document collection. This was done with the help of a domain expert. A lawyer, who has an experience in using the collection used for our experiment and was involved in the evaluation process. Accordingly, the expert selected the 10 queries shown in Table 5.1. These queries were run in both systems and the results were analyzed by using the information retrieval measurement mechanism of precision and recall. The measurement of precision and recall are admittedly subjective due to differences among users in judging the relevance of the document retrieved. With this limitation in mind, we measured the precision and recall of the retrieval system as judged by the domain expert. The results of the document retrieval by the two systems are shown in Table 5.1. We only examined the two systems using the top 20 retrieved documents.

Table 5.1 precision recall results

query	Relevant documents in corpus	Using Lucene default ranking				Using semantic vectors for nearest neighborhood ranking			
		Retrieved	Relevant retrieved	Precision (top 20 retrieved)	Recall	Retrieved	Relevant retrieved	Precision (top 20 retrieved)	Recall
የመሬት ወሰን	6	20	5	0.25	0.83	20	5	0.25	0.83
በሰው አካል ጉዳት ማድረስ	19	20	6	0.3	0.32	20	19	0.95	1
ቅጣት መፈጸም	25	20	10	0.5	0.4	20	7	0.35	0.28
ስነድ ማጥፋት	11	20	3	0.15	0.27	20	8	0.4	0.72
ገንዘብ መሰብሰብ	11	20	0	0	0	20	11	0.55	1
የተዘራ መሬት	6	20	6	0.3	1	11	5	0.45	0.83
ተገቢ አርምጃ	18	20	16	0.8	0.89	20	18	0.9	1
አገልግሎት ማቋረጥ	12	20	8	0.4	0.67	20	8	0.4	0.67
ስልጣን መያዝ	17	20	8	0.4	0.47	20	15	0.75	0.88
የስልጣን ወሰን	16	20	1	0.05	0.06	20	16	0.8	1
Average				0.32	0.49			0.58	0.82

As shown in Table 5.1, the average precision is 32% and recall is 49% for documents retrieved for the 10 queries ran by a system using the Lucene ranking method. In the case of our prototype the precision is 58% and recall is 82% which is an improvement for an information retrieval system using Amharic word meaning extraction from the corpus. Although the result is encouraging, more queries must be used in order to reach a more robust conclusion. This would require the participation of a linguist in the experiment.

To perform the data analysis, it was first necessary to identify an appropriate statistics. Since we were trying to establish that the semantic vector algorithm would support improved query results, the statistic needed to be a surrogate for query quality.

In any search result set, there may be Type 1 (an important document was not retrieved) and/or Type 2 (an irrelevant document was returned) errors. In this study, the number of documents retrieved was fixed to be larger (in all cases) than the potential result set. As a result, no data is available to investigate Type 2 errors. This study only focuses on Type 1 errors.

we first investigated a simple better/worse/equal categorical statistic. As seen in Table 5.1, the semantic vector algorithm outperforms the Lucene algorithm in 6 out of the 10 queries; on 2 they tie, and on 2 the Lucene algorithm provided superior results. Using a simple categorical analysis sign test at this point does not yield statistically significant results (p=0.15).

It did appear that when taking the number of type 1 errors in to account that the new algorithm was superior when the number of type errors were taken into account. Therefore, we then attempted to find a continuous statistic that took into account the number of Type 1 errors in each case. The exact documents that should be retrieved for each query ("expected") as well as the actual documents retrieved for each algorithm ("actual") are known. The recall ratio (actual/expected) provides a meaningful statistic bounded by 0..1 where lower numbers closer to 0 indicate a worse result and higher numbers closer to 1 indicate a better result.

Running a paired t-test on the data revealed significant results. Since it was expected that the new algorithm would significantly outperform the Lucene algorithm, only a single tail test was used for analysis. The results were highly significant p=.019 t=2.41. It is well known that t-tests tend to be reasonably robust with regard to violations of normality in the underlying data, but this data set was so obviously not normal (highly skewed, bounded 0-1, severe kurtosis violation) that we were uncomfortable with accepting the results without some other validation.

There exists a well accepted non-parametric equivalent for the t test, namely the Wilcoxon Sign Rank test. It is not paired, so one might expect it to deliver a slightly weaker p-value than the paired t-test used above. The results of the single tail Wilcoxon test consistent with the t-test. The results were W=22.5 p=0.020.

We had considered using a transformation (ln or similar) on the data to attempt to normalize it, but the strength of the non-parametric results clearly established the superiority of the new algorithm.

Since the experiment produced significant results, clearly "something" has been demonstrated. The interpretation of the results can only be considered definitive for the population selected (legal statutes).

The queries selected were not truly random, but instead were selected to require disambiguation. The results should not be considered to be extendable to queries encountered in a real-world environment. As the queries are not truly a random sample the conclusion should be only accepted after further validation.

Chapter Six

Conclusion and Recommendation

6.1. Conclusion

An effort was made to develop an Amharic information retrieval system which considers Amharic word ambiguities and resolve the ambiguities in order to improve information retrieval. Considering the morphological variants of the language, stemming was applied in the preprocessing stage. Character normalization and stop words removal was used while implementing and designing the system. Evaluation of the system was done by comparing the top 20 retrieved documents from the newly designed system and existing system.

Since the experiment produced better results, clearly "something" has been demonstrated. The interpretation of the results can only be considered definitive for the population selected (legal statutes).

Therefore, we conclude: "The new system (as measured by correctly identifying proper documents) will outperform the Lucene algorithm for queries that require disambiguation on legal statute documents." With the following observations:

1. Using a query of a single word to retrieve documents does not provide results that are adequately relevant to a user's query which using semantic vectors in this case does not help in getting good results.

A query composed of two or more words provided the most relevant documents within our retrieval system.

6.2. Recommendations

Word sense disambiguation does facilitate information retrieval and information translation in the use of information in day-to-day life. To satisfy those users in accessing information from various Internet resources, research and development of information retrieval systems greatly contributes to the ease with which information can be searched and retrieved. However, in the case of Amharic language, there are no resources such as Corpora, Thesaurus/WordNet available for conducting research on information retrieval, especially for disambiguation. Therefore, the following recommendations are made:

1. Annotated standard corpus could help for word sense disambiguation in the Amharic language. This justifies the usefulness of research on word sense disambiguation different heuristics by different researchers can be employed for research using the standard corpus.

2. From the available literature, it is understood that most researchers use WordNet for word sense disambiguation and information retrieval. The Amharic language does not have a resource of this type. Due to the importance of using WordNet for ongoing research in information technology, concerned institutions should develop this resource.

3. A thesaurus is another resource which could be used in word sense disambiguation research in the Amharic language. But since this resource has not yet been developed, this project should be undertaken to make this resource available for future researchers.

4. Future studies should attempt to extend the conclusion by:

   I. Applying word sense clustering algorithm

   II. Testing other document domains

   III. Testing a true random sample of "real" queries collected from actual users

IV. Controlling the number of documents returned so that both Type 1 and Type 2 errors can be assessed.

Evaluation of word sense disambiguation is a difficult task which needs assessors of having different knowledge from different disciplines with the inclusion of linguists. This could bring a better result of evaluation for a research of this kind. Further, the assessor of this experiment is only one domain expert which could affect the experimental result, additional domain experts which we lack are needed for good experimental work. Hence due attention should be given for selecting those capable resource persons.

References:

[AB00] Abiyot Bayou (2000), Design and Development of Word Parser for Amharic Language, Masters Thesis, Addis Ababa University.

[AW08] http://en.wikipedia.org/wiki/Ambiguity, browsed 5/1/2008

[BT06] Thomas Bloor, The Ethiopic Writing System: a Profile, Journal of the Simplified Spelling Society, J19, 1995/2, p30-36.]

[BY87] ባዬ ይማም ፤ 1987, የአማርኛ ሰዋሰው

[BY99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* ACM Press / Addison-Wesley, 1999.

[CQ03] *Cecilia Quiroga-Clare,* Language Ambiguity: A Curse and a Blessing. Translation Journal, Vol. 7, No. 1 January 2003

[DI94] Dagan, I., and A. Itai. 1994. Word Sense Disambiguation Using a Second Language Monolingual Corpus Computational Linguistics 20:4, pp. 563--596.

[DR02] Mona Diab and Philip Resnik, An Unsupervised Method for Word Sense Tagging using Parallel Corpora , 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02), Philadelphia, July, 2002.

[EG01] Ed Greengrass. Information retrieval: A survey. DOD Technical Report TR-R52-008-001, 2001.

[ETUN] http://www.alanwood.net/unicode/ethiopic.html, browsed on 02/05/2008

[GA01] Getahun Amare, Towards the analysis of Ambiguity in Amharic, Journal of Ethiopian Studies , vol. XXXIV, No. 2, December 2001

[IV98] Ide, N. and Véronis, J. 1998. Introduction to the special issue on word sense disambiguation: the state of the art. *Comput. Linguist.* 24, 1 (Mar. 1998), 2-40

[JL84] Johnson, W., & Lindenstrauss, J. (1984). Extensions of lipshitz mapping into hilbert space. Contemporary Mathematics, 26, 189-206.

[KA99] Kaski, S. (1999). Dimensionality reduction by random mapping: Fast similarity computation for clustering. In Proceedings of the International Joint Confer ence on Neural Networks, IJCNN'98 (pp. 413-418). IEEE Service Center

[KC92] Krovetz, R. and Croft, W. B. 1992. Lexical ambiguity and information retrieval. ACM Trans. Inf. Syst. 10, 2 (Apr. 1992), 115-141.

[KW04] W. Kraaij - Enschede: Neslia Paniculata.Variations on Language Modeling for Information Retrieval Thesis Enschede - With ref. With summary. ISBN 90-75296-09-6 ISSN 1381-3617; No. 04-62 (CTIT Ph.D. -thesis series)

[KY98] Karov, Yael and Shimon Edelman. 1998. Similarity-based word sense disambiguation. Computational Linguistics, 24(1):41–60

[LUL] http://lucene.apache.org/java/docs/, browsed on 20/05/2008

[ME01] Mesfin Getachew (2001), Automatic Part of Speech Tagging for Amharic: An Experiment Using Stochastic Hidden Markov (HMM) Approach, Master's thesis, Addis Ababa University.

[MMJ] Million Meshesha and C.V. Jawahar. Indigenous Scripts of African Languages, *Proceeding of the African Journal of Indigenous Knowledge Systems, Vol. 6, Issue 2, pp. 132-142, 2007.*

[MM01] Rada Mihalcea and Dan Moldovan. A highly accurate bootstrapping algorithm for word sense disambiguation. International Journal on Artificial Intelligence Tools, 10(1-2):5–21, 2001.

[MP05] Rada Mihalcea, Ted Pedersen Advances in Word Sense Disambiguation Tutorial at AAAI-2005 July 9, 2005

[MRN02] El Achkar Mona ,RAMMAL Mahmoud , NABHAN Philipe Arabic Intelligent Retrieval System For Legal Database AIRS – LD. http://gandalf.aksis.uib.no/lrec2002/pdf/ws13/mona-AIRS-LD2.doc

[MS05]. Magnus Sahlgren, An Introduction to Random Indexing. In Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005), Copenhagen, Denmark, August 16 2005.

[MS06]. Sahlgren, M. (2006): **The Word-Space Model:** ***Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces.*** Ph.D. dissertation, Department of Linguistics, Stockholm University.

[MS99] Christophor D. Manning, Hinrich Schutze. Foundations of Statistical Natural Language Processing, 1999 MIT Press. PP 233

[NW02] Nega A. and Willet P. Stemming of Amharic Words for Information Retrieval. In Literary and Linguistic Computing. Oxford, Oxford University press, Vol. 17, No.1, pp 1-17, 2002

[NWC03] Ng, Hwee Tou, & Wang, Bin, & Chan, Yee Seng (2003). Exploiting Parallel Texts for Word Sense Disambiguation: An Empirical Study. Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03). (pp. 455-462). Sapporo, Japan.

[PB97] Pedersen, T. & Bruce, R. (1997) Distinguishing Word Senses in Untagged Text. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing.* Providence, RI, 197-207.

[SC98] Schütze, H. (1998) Automatic Word Sense Discrimination. *Computational Linguistics,* 24, 97-123.

[SM83] Salton, G. and Micheal J. McGill. *Introduction to Modern Information Retrieval*, New York: McGraw-Hill Book Company, 1983.

[SM94] Mark Sanderson. Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval Word Sense Disambiguation and Information Retrieval - Sanderson (1997)

[SP95] H. Schutze and J. Pedersen. Information retrieval based on word senses. In Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval, pages 161--175, Las Vegas, NV, 1995

[SW94] E. Schweighofer, W. Winiwarter, Intelligent Information Retrieval: KONTERM - Automatic Representation of Context Related Terms Within A Knowledge Base for a Legal Expert System, Proc. of the 25th Anniversary Conf. of the Istituto per la documentazione giuridica of the CNR: Towards a Global Expert System in Law, Padua, Cedam, 1994.

[TM] Tessema Mindaye (2007), Master's thesis, Design and Implementation of Amharic Search Engine, Addis Ababa university.

[YD95] Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, pages 189-- 196, Cambridge, MA.

[ZQ05] ZHANG Qing-liang1A Discussion on Ambiguity in English (Foreign Language School, Linyi Normal University, Linyi, Shandong 276005, China)

Appendices:

Appendix I. Amharic Alphabets

Orders							Labialized				
1st	2nd	3rd	4th	5th	6th	7th					
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ			ሏ		
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ					
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ			ሟ		
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ			ሧ		
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ			ሯ		
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ			ሷ		
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ			ሿ		
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቍ	ቋ	ቊ	ቌ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ			ቧ		
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ			ቷ		
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ			ቿ		
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኍ	ኋ	ኊ	ኌ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ			ኗ		
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ					
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ					
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	ኵ	ኳ	ኲ	ኴ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ					
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ			ዟ		
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ					
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ					
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጒ	ጓ	ጔ	ጕ
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ			ዷ		
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ					
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ			ጧ		
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ			ጯ		
ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ			ጷ		
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ			ፇ		
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ			ፏ		
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ					
ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ					

Appendix II. Amharic Numerals

፩	፪	፫	፬	፭	፮	፯	፰	፱		
1	2	3	4	5	6	7	8	9		
፲	፳	፴	፵	፶	፷	፸	፹	፺	፻	፼
10	20	30	40	50	60	70	80	90	100	1000

Appendex III. Amharic Punctuation Marks

፡	-hulet netib	-Amharic word space
።	-mulu arat netib	-Amharic full stop
፣	-netela serez	-Amharic comma
፤	-dirib serez	-Amharic semi-colon
“ “	-temiherte tikes	-Amharic quotation mark
!	-timiherte ankro	-Amharic exclamation mark
()	-qinif	-Amharic bracket
_	-cheret	-Amharic underscore
-	-neus cheret	-Amharic Hyphen
...	-netebtab	-Amharic etc.
?	-temhrte tiyakie	-Amharic question mark
·	-yizet (netib)	-Amharic dot

Appendix IV. Sample Normalized term to term vectors

-dimensions|200
????|0.0|0.0|0.0|0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0
|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.
0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|-
0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|-
0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0
.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|
0.0|0.0|0.0|0.0|0.0|0.0|-
0.31622776|0.0|0.0|0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.3162277
6|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|
0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|-
0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.31622776|0.0|0.0|0.31622776|0.0|0.
0|0.0|0.0|0.0|0.0|0.0|-0.31622776|0.0|0.0|0.0
???|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.
0|-
0.31622776|0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|
0.0|0.0|0.0|0.0|0.0|0.0|0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|-
0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0
.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|
0.0|0.0|0.0|0.0|0.0|0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0
.0|0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|-
0.31622776|0.0|-
0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0
.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.31622776|0.0|0.
0|0.0|0.0|0.0|0.0|0.0|-0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0
?????|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.316
22776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0
|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|
0.0|0.0|0.0|0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|-
0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0
.0|0.0|0.0|0.0|0.0|0.0|0.0|-0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|-
0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|-
0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|
0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.
0|0.0|0.0|0.0|0.0|0.0|0.0|-
0.31622776|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0
.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0|0.0

Appendix V. Sample of Document vectors representation

-dimensions|200
article722.txt|0.10592719|-0.04681184|0.034072228|-0.009204118|0.017899094|0.09857544|-0.0011576046|-0.06188319|0.05415879|0.024111066|-0.051672325|0.027174991|0.025491789|0.033576082|0.015813755|-0.00579999953|0.012583083|-0.009628508|-0.18037562|-0.077302486|0.0638136 8|0.0012213525|-0.032612436|-0.036744073|-0.047943316|-0.091275945|-0.00793248|0.031353265|0.06745159|-0.01187234|0.001265702 4|0.01485026 4|0.021365706|-0.0014898059|0.019079125|-0.026401423|-0.026977524|-0.04261838|-0.014349785|-0.14602336|0.027630223|0.065529376|-0.15924543|0.005188246|-0.013957508|0.13841526|-0.033813097|-0.043760832|-0.018655568|0.007798248 4|0.0809531|-0.12556115|-0.017825726|-0.22447953|-0.0091678435|-0.013369686|0.085935 83|0.111821264|-0.078361824|-0.014078796|-0.17662479|-0.07245881|0.20376372|0.07293033|0.064263724|-0.06277343|0.07882532|0.03899157|-0.0010636762|0.09947497|-0.026772555|-0.017524283|-0.10359889|6.7925855E-4|-0.0483928|-0.0784758|0.031175014|0.10405093|0.062977724|0.040497474|-0.040765475|0.092998564|0.073154956|0.017778823|-0.020517718|-0.0014415606|-0.1276021|0.0685612|-0.048751738|0.1706961|-0.0024839747|-0.12199595|-0.087637186|0.069967106|-0.0013568051|-0.017975431|0.06160461|-0.101130985|-0.037653092|0.051034927|0.04808869|-0.011723873|-0.0472502|0.007927974|0.055412706|-0.03642279|-6.6904596E-4|-0.017788166|0.13647075|0.18227404|-0.00367005|-0.008852045|0.0957096|0.0376552|-0.022666084|0.017445693|-0.020240085|0.04247444|-0.041845456|0.09794847|-0.11119833|-0.03733384|-8.418765E-4|-0.005569865|-0.06609293|0.029800072|0.07844179|-0.0011062041|0.03531734|-0.036051054|0.0040443637|-0.011840966|-0.012859555|-0.09859044|0.0991105|0.05026416|-0.032532934|0.011857331|-0.04858969|-0.021375107|-0.036040884|-0.062313307|0.037562005|-0.019225908|0.11060007|0.10250427|-0.0060002995|-0.009616407|-0.008785127|0.017207826|-0.016242031|0.08163543|0.05223354|-0.009546461 5|0.011894241|0.04294765|-0.030418878|0.05011201|-0.061777182|0.11378848|-0.0057404973|0.03686649|0.03363188 7|0.00426567|-0.055713717|-0.045515787|0.0865161|-0.066641256|0.020110676|0.031188475|-0.013149544|-0.27704194|0.013746442|0.0047459034|-0.006871413|0.0062644896|-0.1430295|0.01521231|-0.0096993|0.05540932|0.020281654|-0.053188886|0.04923063|-0.13395655|-0.010833855|0.1526578|0.07200775|-1.21137375E-4|-0.048813924|0.00329075 94|0.008121173|0.065738186|0.027012423|-

Appendix VI. List of Stop words

አንቀጽ	ሀያ	የሆኑትን	ስድስት	እስከ	ይኸኛው	-		205/		-4	8/ሀ/	/አንቀጽ.164/	(ሰ)	336)	2-55/	36//
ወይም	ሆነ	በሆኑ	ምንም	ብሎ	ክርሱ	/1/		90/	248-250)		1አ5	62/	481//	107/	/pubic/	356-34
እንደሆነ	በኑላ	ጀምሮ	በሆነው	ከሰላሳ	መሆኑን	1/		84/	248-22/		784/	613/3//	134/	335/3/	2)%//1/	355-34/
ላይ	በአንድ	እንደዚህ	ስለ	የሚሆኑ	ለዚያው	/		78/	248-22		784-791/	599/	479//	332)	187//ሰ/)	327/ሀ
ማንም	የሆኑ	በመሆን	ይሁን	ላይም	ለዚሁ	//		200/.	247/ሀ/		784-790/	///ሀ/	478/	327/ሀ	/36//	.ስ
በማይበልጥ	ከአስራ	የሌለው	ከዚህ	የሆናል	ለእነርሱም	ሰ/		189/	244/		194/	59/1/	134-153/	32-1	187//ሀ/	32-1
መሰረት	የሆነውን	በማለት	በእነዚህ	ከነዚህ	እዚሁ	180/		769/	243///3/		/1//መ/	589/	468/	3/1/(ሀ)/	184)	3/1/(ሀ)/
ሁኔታ	ከላይ	ባለ	ከማናቸውም	ያህል	.ቢሆንም	3/		768-770	240/		775/	588/	467/	103/1/	18/	3/
ሌላ	ሁሉ	ይህንን	ከነበረው	ከሆነና	ሀ/	/ሀ/		75/	240)		771/	158/	465/	2አ5	///ሀ/	284-337/
እና	መሆኑ	እንዲቆይ	በአንዳንድ	እስከ	ለ/	66/		684/	/ሀ//		19-200/	587/	46(1)/	103/	136//	27/
ይሆናል	ሌላውን	ሌላው	በእያንዳንዱ	ለሆኑት	ሐ/	179/		218/	237/		770/	583/	459/	298/	(አንቀጽ187//ሀ/)	269-24)
ሆኖ	ከሰባት	የሚሆነው	ጊዜም	አለው	መ/	5/		671/	238-260		754/	58/3/	455//	297/	13/	26/1//2/
ነው	ለሌላ	በአንዱ	አስከ	እነዚሁ	ሠ/	/ከአንቀጽ		161/	231/		750/	58-582/	129-15)	ሆን½	129-15)	(1)እስክ
በዚህ	አለበት	አይሆንም	የሌሎች	እንደሆኑ	ረ/	(አንቀጽ		6/	229/		187/1/	(መ)	451/	29/	123/ሀ	
እስከ	ሲል	ማለት	የሚሆኑት	ስለማናቸውም	ሸ/	/3/		16/	228/		/1//ለ/	(1)/	45(2)/	ቀ/	11/	
ውስጥ	ይሆናሉ	ሲባል	ከሆነው	ስለዚህ	-ሀ/		-2	159/	224/1/መ/		73/	58(3)/	129-131/	ሆን&	11-419	
ከአንድ	ያልሆነ	ሳለ	የነበረውን	ከአንዳንድ	-ለ/	12/		156/	224/		187//ለ/)	57/	449/	289-283/	11-154/	
በማናቸውም	በሙሉ	የሆነው	ያሉ	በእነዚሁ	-ሐ/	9/		/መ//	(አንቀጽ187//ሀ/)		ማናቸWM	57-59/	435/	284-337	1/3/	
ወር	አስራ	መሆናቸው	ከሌሎች	በአምስት	-መ/	83/		15//	224-228		728-30/	560/	127/	284-325/	9-	
ከአምስት	አስር	በዋና	አንዲት	ወይም	-ሠ/	188/		145/	22)		72/	150/	426/	284-317/	84ኣ	
በላይ	እንደ	በማቀድ	ለሌሎች	በሆኑ	-ረ/	140/		145-150/	217/		187//ሀ/	56-599	423/	1/ለ/	8/ለ/	
ሲሆን	ቢሆንም	ጊዜና	እንኳን	የሆኑበታል	ሀ.		-1	48/	213/		71/	559/	123/ሀ	28/1/	728-30/	
በሆነ	አንዱ	ለዚህ	ለሆነው	ለነዚህ	ለ.	4/		49/	211-213/		70/	55/3/	(ሀ)/	28(3)/	72/	
ከዚህ	የሌላውን	ሶስተኛ	አንዳንድ	ለማንኛውም	ሐ.	208/		13/	21//		187/	55/	422-424	1/3/	ን/ኣ	
የሆነ	ከሁለት	የነገሩ	ሰኣት	አንደኛ	መ.	20/		407/	/5/		ኘ/	15/1/	420/	273/	690//	

አምስት	ሶስት	እንዲሆን	እንደነዚህ	ናቸው	ሠ.
ማንኛውም	ካልሆነ	እንኳ	የሆኑት	አሁን	ሽ.
ጋር	ቢያንስ	ከሀያ	የማናቸውም	ሰባት	
አንድ	ቢሆን	ከሀምሳ	ይህንንም	እነደሆነ	
ልዩ	እነዚህን	ይኸው	የአንድን	እነደሆነ	
ከሶስት	ናቸው	ለአንድ	በመሉም	ይህችው	
በተለይም	አንዱን	የሚችለውን	በነዚህ	ከእነዚህ	
በሌላ	ሁለት	ወይም	የዚሁ	ከእነዚሁ	
ሺህ	ለዚሁ	በሚገባ	ለእያንዳንዱ	የአንቀጹ	
ወደ	ወይዘሮ	ይህም	ስለሆነ	ወይ	
ማናቸውንም	ተብሎ	እነዚህ	መሆናቸውን	የሆነችን	
ከአስር	ላይሆን	ከዚያ	ማንኛውንም	የለውም	
የማይበልጥ	እንደሆነና	እንዲሆኑ	ሁለቱ	በሚችሉ	
ብቻ	ብሎ	ከሌላ	እንጂ	የሌላቸውን	
እንዲሁም	ከብር	ለሆነ	ከስምንት	በሶስተኛ	
ሌሎች	ሆኖም	በሌሎች	ሁለቱንም	በቀር	
ይህ	በታች	አንደሆነ	በሁለት	በነሱ	
ይህን	የሌላ	እንዲህ	በአስር	የአንዱን	
ከሆነ	ያላቸው	በነዚሁ	በሚል	የአንዱ	
የዚህ	ይህንኑ	በእንደዚህ	ቁጥር	ው	
ማናቸውም	ሆነው	ስምንት	ባሉ	በዚህ	
ከስድስት	በስተቀር	ሲሆንና	ከመቶ	በዚህም	
መቶ	መሆን	ምንጊዜም	እነዚህም	በዚህና	
ያለ	ስም	ለማናቸውም	ሲኖር	ከዚህም	
መሆኑን	እንደገና	የአንድ	ሰላሳ	በሁኔታው	
አንድን	የማያንስ	እነዚህኑ	ማንም	ከነዚሁ	
ያላቸውን	እጅግ	ሲሆኑ	ለሆኑ	ሌሎች	
ሊሆን	ግን	በሁለቱም	አለ	ይህን	

	79/		37/		98/		690/1/		549)		123/ሀ//		-133	68/4/
	88/		11/		95/		690//		543/		42/	(4)/		68/
	9እ		37/3/		94-95/		186/		533/		42-47/	ሀ		676/1//
	183/		110(2)/			937	ረ		146-150/		123/ሀ/	አንቀጽ		66//
	/ፋርማሲስት/		34/		207/		69/		532-534/		41/	12/		656/
	131/		106/		93/		184/		53/	4010		5/		622-31
	123/		40/		201-207/		680/		514-24/		123(ሀ)/	9/		58-582/
	(ሀ)		/በአንቀጽ		9-		68/4/		511///5/		399/	20/		55/3/
	104/		100/		9(3)		ነ/		(ሰ))		121-128/	9እ		55/
	148/		ሆነ&		861/		676/1/		510//		397/	79/		514-24/
		-3	10/		20/)		/1/ለ/		51(ሰ)/		397)	14/		511///5/
	81/		27//		84/1/መ//		670/		144-149/		39/	15//		510//
	19/		27/		84/1/(መ)/		665/		506/3/		39-641/	16/		51(ሰ)/
	/1//ሀ/		269-322		84/1/		180)		505-513/		117/	10/		4እ
	68/		269-24)		2/		ቸ/		143/		379/	766/		4እ7/
	18/		266/		/36//		66/1/ሀ/		4እ		376/	59/		493//
	90/3/		261/1		835/		66//		4እ9		111/	40/		481//
	59/		260/		2-55/		655/		14/3/		375/	27//		48/
	155/		26/1//2/		2)%//1/		654/		4እ7		37)	26/1/		"ሀ"
	154/		26/1/		829/		179/:-		ዚH		367/	254/4/		479//
	54/		257/ሰ/		821/		/አንቀጽ101/		497/		36//	25//		46(1)/
	14/		257)		2(ሰ)		(ሰ)		14/1		11-419	248-22/		455//
	101/		256/		809/		65-687/		494-500		36/	248-22		45(2)/
	/4/		255)		2(ሀ)		640/		493//		በ/	243///3/		41/
	50/		ህ/		/3		168-162		491/		356-34	22)		42/
	232-237/		254/		80/		627/		488/		11-154/	21//		(Manual)
	21/		25//		8/ሰ/		622-31		486/		355-34/	20/)		39/
	98/3/		25/		2(5)		168(2)/ማዘዝ		136//		35/	2/		39-641/

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all sources of materials for the thesis have been duly acknowledged.

TESHOME KASSIE

This thesis has been submitted for examination with my approval as an advisor.

NEGA ALEMAYEHU, PhD

Addis Ababa, Ethiopia

February 2009