Addis Ababa University

School of Graduate Studies

AMHARIC WORD SENSE DISAMBIGUATION USING WORDNET

Segid Hassen Yesuf

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF THE ADDIS ABABA UNIVERSITY IN PARTIAL FULFILMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

March 2015

Addis Ababa University

School of Graduate Studies

College of Natural Sciences

Department of Computer Science

AMHARIC WORD SENSE DISAMBIGUATION USING WORDNET

Segid Hassen Yesuf

ADVISOR:

Yaregal Assabie (PhD)

Signature of the Board of Examiners for Approval

Name	Signature
1. Dr.Yaregal Assabie, Advisor	_____
2. _____	_____
3. _____	_____

Dedication

In loving memory of my most beloved and most cherished to **my Grandmother**, **Meriem Alemu**, who lived a life of dignity, courage, wisdom, patience and above all affection, and who will remain my personal hero and my inspiration forever. May Allah bless her soul, Amen.

Acknowledgments

First of all and the greatest important, all praises and thanks to ***Allah*** for all his blessings without which nothing of my work could have been done.

I would like to thank my advisor Dr. Yaregal Assabie for all his guidance at every step of the way for patiently listening to me for uncountable hours, for his helpful discussions and comments, for imparting so much valuable knowledge and for all his encouragement and words of kindness.

I would like to thank Mr. Endris Abay for his support on the development of Amharic WordNet and test results concerning the linguistic parts. I also like to thank my friends Ermiyas Woldeyohanes for the invaluable ideas, comments and resources he shared.

The last but not the least, I would like to thanks my wife Zinet Ibrahim and to my family for their precious love and comprehensive support.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ACRONYMS

AI	Artificial Intelligence
AWI	Ambiguous Word Identifier
AWNBWSD	Amharic WordNet Based Word Sense Disambiguation
AWN	Amharic WordNet
BS	Bootstrapping
CS	Context Selection
IE	Information Extraction
IR	Information Retrieval
KBAWSD	Knowledge Based Amharic Word Sense Disambiguation
LDOCE	Longman Dictionary of Ordinary Contemporary English
MI	Mutual Information
ML	Machine Learning
MT	Machine Translation
MRD	Machine Readable Dictionary
NLP	Natural Language Processing
NLU	Natural Language Understanding
POS	Parts Of Speech
QA	Question Answering
SS	Sense Selection
SR	Sense Retrieval
WSD	Word Sense Disambiguation

ABSTRACT

Words can have more than one distinct meaning and many words can be interpreted in multiple ways depending on the context in which they occur**.** The process of automatically identifying the meaning of a polysemous word in a sentence is a fundamental task in Natural Language Processing (NLP). This phenomenon poses challenges to Natural Language Processing systems. There have been many efforts on word sense disambiguation for English; however, the amount of efforts for Amharic is very little. Many natural language processing applications, such as Machine Translation, Information Retrieval, Question Answering, and Information Extraction, require this task, which occurs at the semantic level.

In this thesis, a knowledge-based word sense disambiguation method that employs Amharic WordNet is developed. Knowledge-based Amharic WSD extracts knowledge from word definitions and relations among words and senses. The proposed system consists of preprocessing, morphological analysis and disambiguation components besides Amharic WordNet database. Preprocessing is used to prepare the input sentence for morphological analysis and morphological analysis is used to reduce various forms of a word to a single root or stem word. Amharic WordNet contains words along with its different meanings, synsets and semantic relations with in concepts. Finally, the disambiguation component is used to identify the ambiguous words and assign the appropriate sense of ambiguous words in a sentence using Amharic WordNet by using sense overlap and related words.

We have evaluated the knowledge-based Amharic word sense disambiguation using Amharic WordNet system by conducting two experiments. The first one is evaluating the effect of Amharic WordNet with and without morphological analyzer and the second one is determining an optimal windows size for Amharic WSD. For Amharic WordNet with morphological analyzer and Amharic WordNet without morphological analyzer we have achieved an accuracy of 57.5% and 80%, respectively. In the second experiment, we have found that two-word window on each side of the ambiguous word is enough for Amharic WSD. The test results have shown that the proposed WSD methods have performed better than previous Amharic WSD methods.

Keywords: Natural Language Processing, Amharic WordNet, Word Sense Disambiguation, Knowledge Based Approach, Lesk Algorithm

CHAPTER ONE: INTRODUCTION

1.1 Background

Natural Language Processing (NLP) provides tools and techniques that facilitate the implementation of natural language-based interfaces to computer systems, enabling communication in natural languages between man and machine. In all natural languages, many words can be interpreted in a variety of ways, in accordance with their context. Natural language processing (NLP) involves resolution of various types of ambiguity. Lexical ambiguity is one of these ambiguity types, and occurs when a single word (lexical form) is associated with multiple senses or meanings. Ambiguity is a major part of any human language. Almost every word in natural languages is polysemous, that is, they have numerous meanings or sentences. The Amharic language has many words that have multiple meanings. For instance, the Amharic noun "ለጋ" can mean "አካሏ ሙሉ በሙሉ የማይታይ ገና አዲስ የወጣች ጨረቃ", "ቀለል ያለ አዲስ ደመና" , "ልጅ እግር ፥ወጣት፥ሴታ" or "ያልቆየ ና ያልበሰለ ወይም ያልኮመጠጠ ቅቤ". For example, in the sentence, "ልጅቷ የገዛችው ቅቤ ለጋ ነው::", a human can easily understand which sense of "ለጋ" is intended. It would often be useful if software could also detect which sense of "ለጋ" was intended. Word sense disambiguation involves picking the intended sense of a word for a pre-defined set of words, which is typically a machine-readable dictionary, such as WordNet. The lexical and semantic analysis of words is necessary for computers to make sense of the words. This is known as word sense disambiguation [1].

The word sense ambiguity is a hard problem for the developers of Natural Language Processing (NLP) systems. Words often have different meaning in various contexts. The process by which the most appropriate meaning of an occurrence of an ambiguous word is determined known as Word Sense Disambiguation (WSD), and remains an open problem in NLP. Humans possess a broad and conscious understanding of the real world, and this equips them with the knowledge that is relevant to make sense disambiguation decisions effortlessly, in most cases. However, creating extensive knowledge bases, which can be used by computers to 'understand' the world and reason about word meanings, accordingly, is still an unaccomplished goal of Artificial Intelligence (AI) [3].

The WSD problem is that of associating an occurrence of an ambiguous word with one of its senses. In order to do this, first, an inventory of the senses associated with each word to be disambiguated must be

available; second, a mechanism to associate word senses in context to individual senses must be developed, and thirdly, an evaluation procedure to measure how well this disambiguation mechanism performs must be adopted. Designing the actual disambiguation mechanism involves the construction of disambiguation rules and their subsequent application to a real disambiguation problem, achieving WSD. In general, people are unaware of the ambiguities in the language they use because they are very good at resolving them using context and their knowledge of the world. However, computer systems does not have this knowledge, and consequently do not do a good job of making use of the context [2].

Lexical disambiguation in its broadest definition is nothing less than determining the meaning of every word in context, which appears to be a largely unconscious process in people. As a computational problem, it is often described as "AI-complete", that is, a problem whose solution presupposes a solution to complete natural-language understanding or common-sense reasoning. In the field of computational linguistics, the problem is generally called word sense disambiguation (WSD), and which is defined as the problem of computationally determining which sense of a word is activated by the use of the word in a particular context. Words are assumed to have a finite and discrete set of senses from a dictionary, a lexical knowledge base, or ontology based [2].

WSD has obvious relationships to other fields such as lexical semantics, whose main endeavor is to define, analyze, and ultimately understand the relationships between "word", "meaning" and "context". However, word meaning is at the heart of the problem, WSD has never really found a home in lexical semantics. It could be that lexical semantics has always been more concerned with representational issues and models of word meaning and polysemy so far too complex for WSD [2].

Word Sense Disambiguation (WSD) is the task of automatically identifying the correct meaning of a word that has multiple meanings. In WSD, these meanings are referred to as senses, or concepts, which are obtained from a sense-inventory. The ambiguous word is referred to as the target word and the context in which the target word is used is called an instance. WSD is not thought of as an end in itself, however, as an enabler for other tasks and applications of computational linguistics and natural language processing (NLP) such as parsing, semantic interpretation, machine translation, information retrieval, question answering, text mining, Computational Advertising and the like. The computational identification of meaning for words in context is called word sense disambiguation [50].

Word Sense Disambiguation was conceived as an important sub-problem of Machine Translation as early as the late 1940s. The researchers then also acknowledged the essentials needed for WSD such as the local context in which a target word to be disambiguated occurs, the statistical distribution of words and senses, and the role of knowledge bases. However, due to lack of available computational resources, a bottleneck was reached and not much progress was made. With the availability of lexical resources in the 1980s, WSD saw a revival, with people turning to AI based approaches to tackle the problem. With advances made in statistical modeling and Machine Learning, the 1990s saw three major developments: WordNet became available, the statistical revolution in NLP swept through, and Senseval began. SENSEVAL (now renamed SEMEVAL) is an international competition on WSD organized by the Association for Computational Linguistics (ACL) **S**pecial **I**nterest **G**roup on the **LEX**icon (SIGLEX). There are now many computer programs for automatically determining the sense of a word in context (Word Sense Disambiguation or WSD). The purpose of Senseval is to evaluate the strengths and weaknesses of such programs with respect to different words, different varieties of language, and different languages. Sometimes, a word sense disambiguation method is concerned only with disambiguating a single ambiguous word in a given context. Other methods attempt to disambiguate all the words in a text [4].

There have been different studies regarding to WSD in different languages. WordNet is now widely used in the Natural Language Processing (NLP) community for applications in Information Retrieval, Machine Translation, Word Sense Disambiguation etc. One of the most successful to WSD is to make use of WordNet.

WordNet is a manually constructed lexical system developed by George Miller [29] and his colleagues at the Cognitive Science Laboratory at Princeton University. Originating from a project whose goal was to produce a dictionary that could be searched conceptually instead of only alphabetically, WordNet evolved into a system that reflects current psycholinguistic theories about how humans organize their lexical memories [29]. WordNet contains only open class words (nouns, verbs, adjectives, and adverbs) and which does not contain closed class words such as pronouns, conjunctions, and prepositions. WordNet is organized semantically (as part-of-speech).

Amharic WSD systems that are developed using Amharic WordNet include, Preprocessing component, Morphological Analysis component, Amharic WordNet Database and disambiguation components.

1.2 Statement of the Problem

The absence of Automatic WSD would make the development of NLP and IR applications too difficult. However, a few researchers attempted using machine-learning approach to develop Amharic WSD. Teshome Kassie [1] attempted to develop WSD for Amharic, which tries to resolve lexical ambiguity. The author demonstrated word sense disambiguation based on semantic vector analysis in order to improve the effectiveness of Amharic Information Retrieval system. Solomon Mekonnen [25] who employs supervised machine-learning approach for Amharic WSD. The author used manually annotated training data containing instances of a target word to learn the context in which target words are used. Solomon Assemu [26] who employs unsupervised machine learning approach for Amharic WSD and unannotated training data containing instances of the target word was used. Unannotated training data contains training instances whose concepts are not known. The other researcher was Getahun Wassie [27] who employs semi-supervised machine learning approach for Amharic WSD by using unlabeled training data and limited labeled training data. However, the previous researches [25, 26, 27] have the following limitations. The study was limited to experiment on five ambiguous words only and available sense-annotated corpora are largely insufficient to cover all the senses of each of the target words and corpus used as a source of information for Disambiguation. A sample of words is selected from the corpus and the selected words are disambiguated in a short given context. It is assumed that the word to be disambiguated has a fixed set of senses in the sense inventory, where the sense inventory contains the mapping of words and their different senses. Amharic WSD developed by researchers requires manually labeled sense examples, which is time taking and exhaustive when the number of corpus size increased and cluster the contexts of an ambiguous word into a number of groups. Manual sense tagging is very difficult, time taking and limiting the number of sense tagged words to be used. To deal with this problem knowledge-based approach to Amharic WSD technique is proposed. Knowledge-based methods use information extracted from structured data called a knowledge source. These methods rely on information from the knowledge source about a concept such as its definition or synonym rather than training instances in manually annotated or unannotated training data. We need Amharic WSD for various applications such as machine translation, information extraction, question answering, information retrieval, text classification, text summarization, speech processing, text processing, grammatical analysis, content and thematic analysis and soon. In our study, Amharic WordNet is used as a source of information for disambiguation and knowledge-based Amharic WSD method allows the system to disambiguate words in running text, which is called all-

words disambiguation. All-words disambiguation methods have an advantage over what is termed lexical-sample disambiguation methods that was done by the previous researchers because lexical-sample methods can only disambiguate words in which there exists a set of training data in which ambiguous words may not be known ahead of time. We determine the correct concept of ambiguous words by first identifying the ambiguous words semantic type, which is a broad categorization of a concept. After the semantic type of the ambiguous words is identified, then the correct concept is identified based on its semantic type from Amharic WordNet. The development of Amharic WordNet is an important step for WSD, even for other application areas such as Information Retrieval, Machine Translation and soon. This is the research gap that motivates us to use Amharic WordNet for Amharic WSD.

Finally, we proposed a knowledge-based Amharic WSD method that does not require sense tagged corpus and that identifies senses of all words in sentences or not a small number of words. Our proposed method depends on Amharic WordNet, which is relatively very large, and it is a lexical database in a hierarchical structure. Therefore, the major concern of this research was to investigate knowledge-based approach for Amharic WSD, test the results in order to develop a bit further natural language understanding and compare the results with the previous researches that were studied before [1, 25, 26, 27].

1.3 Objectives

General Objective

The general objective of this research work is to design and develop a system for Amharic word sense disambiguation using WordNet.

Specific Objectives

The specific objectives of this research work are:

- Conducting literature review and related works to understand the approaches of WSD.
- Collect data from Amharic dictionary to develop Amharic WordNet.
- Identifying Amharic ambiguous words and their contextual meaning
- Designing architecture for Amharic Word Sense Disambiguation.

- Designing algorithm for Amharic Word Sense Disambiguation.
- Developing a prototype of the system.
- Testing and evaluating the performance of the developed system.

1.4 Scope and Limitations of the Study

Word Sense Disambiguation is a complex and recent research discipline that requires the effective analysis and processing of ambiguous words in its textual context. There are no publicly available linguistic resources for Amharic language. Researches in WSD for other language use linguistic resources like WordNet, thesaurus and machine-readable dictionaries. However, for Amharic those resources are not yet been developed. The scope of this study is limited to retrieving senses of ambiguous word from Amharic WordNet, Identifying the ambiguous words and its context in the given text and assigning the appropriate sense to the given word in the given context from Amharic WordNet, which is developed manually.

The limitation of this study is that the developed study does not perform grammar and spelling correction and do not work for words, which do not exist in Amharic WordNet.

1.5 Methodology of the Study

Literature Review

Literature review will be done on different areas relevant to this research work. Approaches to word sense disambiguation will be studied to show the current state of the art in this area. Extensive literature review will be conduct on word sense disambiguation in order to obtain an in-depth understanding of the area and to find useful approaches for word sense disambiguation. Specifically focus on reviewing literature for the techniques of word sense disambiguation approaches to develop Amharic WordNet Structure, Amharic ambiguous words, writing system of Amharic languages and existing researches on word sense disambiguation.

Data Collection

Amharic documents and information on Amharic ambiguous words will be collected from different libraries and institutions. To develop Amharic WordNet data will be collected from Amharic dictionary [9] and words selected by a linguistic expert from a list of homonyms collected by Girma Getahun [51].

Tools and Techniques

Different tools and techniques will be used to achieve the goal of the research. The main parts of the system such as Word Sense Disambiguation, Amharic WordNet, Morphological Analyzer and Pre-processing will be developed with Java programming language, Python programming language and SQL Server.

Prototype Development

For evaluating the performance of the system in a fair and logical manner, we will developed prototype. Accordingly, the system will be evaluated by comparing its output against the input sentence.

1.6 Application of Results

WSD is an intermediate language engineering technology which could be used as a component in various NLP applications such as spelling checking, grammar checking, search engine, speech and text processing and other text processing applications. Based on these facts, Amharic WSD using WordNet system plays a crucial role in various NLP applications associated to the language. It also opens the track for future Amharic NLP studies and simplifies development of the following applications for Amharic.

- Information Retrieval
- Machine Translation
- Question Answering
- Computational Advertising.
- Information Extraction

1.7 Thesis Organization

The rest of this thesis is organized as follows. Chapter two presents literature review and gives basic introduction of WSD and discusses about Amharic writing system, morphological structure and ambiguities in the language. Chapter three presents works related to word sense disambiguation system for Amharic and other languages. Chapter four discusses about the design and implementation of the system, which is composed of structure of Amharic WordNet and architecture of the system. The performance of WSD is presented in Chapter five. This Chapter discusses the experimentation and its findings. Finally, Chapter six presents conclusion and recommendations.

CHAPTER TWO: LITERATURE REVIEW

In this chapter, literature in the field of WSD is review and discussed in brief. The chapter covers overview of WSD, its main applications is in the field of Natural language Processing (NLP) and discussion on major approaches that have been employed for WSD research with special focus on knowledge-based approach, which is used in this study. The discussion of different approaches and algorithms would help to understand the central problem in WSD research and facilitates the comparison of existing approaches to the specific solutions that are employed in this study. Finally, Amharic language and Amharic word ambiguity is presented

2.1 Word Sense Disambiguation and Application Areas

The task of Word Sense Disambiguation (WSD) is to assign a sense to an instance of a polysemous word in a particular context. Word sense disambiguation (WSD) is an open research area in natural language processing and artificial intelligence and it is also based on computational linguistics. WSD is to analyze word tokens in context and specify exactly, which sense of several word is being used. However, the problem is to search the sense for a word in a given context and lexicon relation. Most of the work has been completed in English, and now the point of convergence has shifted to others languages. In our research, we are using knowledge-based approach to WSD with Amharic language. A knowledge based approach uses of external lexical resources such as WordNet to disambiguate words. We have developed a WSD tool using knowledge-based approach with Amharic WordNet to disambiguate words. WordNet is built from co-occurrence and collocation and it includes synset or synonyms, which belong to either noun, verb, adjective, or adverb [30, 32].

Words are having different meanings, and each one of them is determined based on context of the word usage in a sentence. Word sense disambiguation is the difficulty to find the sense of a word in a given natural language context, where the words have one or more meanings. The sense of a word in a text depends on the context in which it is used, the context of the ambiguous word is certify by other neighboring words. This is called as local context or sentential context. This task needs a lot of words and word knowledge. WSD is the process of determining the sense of the word and the automated process of recognizing word senses in context [2].

The knowledge-based approach, on the other hand, does not rely on sense-annotated corpora; however, it takes advantage of the information contained in large lexical resources, such as WordNet. The Lesk [10] algorithm is a classic example of the knowledge-based approach. A simplified version of the algorithm counts the number of words that are in both the neighborhood of the ambiguous word and in the definition of each sense in a dictionary. It then chooses the sense with the larger number of words. While simple and intuitive, Lesk's approach is very sensitive to the exact wording of the definitions, so the absence of a certain word can radically change the results. This is a significant limitation as dictionary glosses tend to be fairly short and do not provide sufficient vocabulary to relate fine-grained sense distinctions. These drawbacks are common to most dictionary-based methods, as they have not realized the potential of combining the relatively limited information in such definitions with the abundant co-occurrence information extractable from text corpora.

Words do not have well-defined boundaries between their word of senses, and our task is to determine which meaning of the word is indented in a given context. This is first problem encountered by any natural language processing system, which is referred to as lexical semantic ambiguity. WSD is the technique of natural language processing (NLP) which is a research area in NLP and which is very useful now days. WSD involve more words and word knowledge or common sense, which identifies Dictionary or Thesauri. A dictionary is used to decide if a lexical entry is ambiguous or not, and to specify its set of possible senses. The most widely used for this lexical resource for this task is Amharic WordNet, which is described in the next section.

According to Rohana Sharma [11], WSD involves two steps. The first step is to determine all the different senses for every word relevant to the text or discourse under consideration, i.e., to choose a sense inventory from the lists of senses in everyday dictionary, from the synonyms in a thesaurus, or from the translations in a translation dictionary. The second step involves a means to assign the appropriate sense to each occurrence of a word in a context. All disambiguation work involves matching the context of an instance of the word to be disambiguated either by using information from external knowledge sources or with contexts of previously disambiguated instances of the word. For both of these sources, we need preprocessing or knowledge-extraction procedures representing the information as context features. However, it is useful to recognize that another step is also involved here: the computer needs to learn how to associate a word sense with a word in context using either

machine learning or manual creation of rules or metrics. Disambiguating a word in context is crucial in many natural language processing (NLP) applications [11, 32, 53] such as:

Machine Translation (MT): WSD is required for lexical choice in MT for words that have different translations for different senses and that are potentially ambiguous within a given domain (since non-domain senses could be removed during lexicon development). For example, in an English-Amharic financial news translator, the English noun "**change"** could translate to ጠለፈ ("change the direction of something"), ዞመረ ("change around") or **አሸረፈ** ("change money"). In MT, the senses are often represented directly as words in the target language. However, most MT models do not use explicit WSD. The lexicon is pre-disambiguated for a given domain, handcrafted rules are developed, or WSD is folded into a statistical translation model

Information Retrieval (IR): Ambiguity has to be resolved in some queries. For instance, when the information related to the word "ሳለ" is scanned through the documents, some sentences that involve "ሞሳል" can be found in these documents.

Information Extraction (IE): In many applications, text analysis should be made accurately. The Semantic Web requires automatic annotation of documents according to reference ontology. All textual references must be resolved to the right concepts and event structures in the ontology. For instance, if the word "fare" is used in a text related with computers, the sense to be perceived should be the tool that facilitates computer use, not an animal.

*Content and thematic analysis***:** a common approach to analyze the destination of predefined categories of words i.e., words indicative of a given concept idea, etc. the need for disambiguation in such analysis has long been recognized in order to include only those instances of word in its proper sense.

Speech and Text processing: sense disambiguation is required for correct phonetization of words in a speech synthesis and the sense disambiguation is necessary for spelling correction in text processing.

Computational Advertising: is a new scientific sub-discipline, at the intersection of information retrieval and machine learning. Its central challenge is to find the best ad to present to a user engaged in a given context, such as querying a search engine ("sponsored search") or reading a Web page. WSD identifies the appropriate meanings of the main terms in the given context, thus ultimately assisting in

finding the best ad to match the given query or page [2]. Word sense disambiguation is itself often divided into distinct tasks. Sometimes, a word sense disambiguation method is concerned only with disambiguating a single ambiguous word in a given context. Other methods attempt to disambiguate all the words in a text.

From a historical perspective, there are two broad tasks have been defined for WSD [3].These are lexical sample tasks and all word disambiguation.

Lexical Sample Task: This is a more target oriented WSD task where specific words are selected from the lexicon and the selected words are disambiguated in a short given context, generally occurring one per sentence, are required to be sense tagged. The set of words is small and hence does not enforce a wide coverage system. Supervised systems have historically flourished in this task mainly because of the availability of hand-tagged examples for the limited number of words to be disambiguated. However, lexical sample disambiguation is limited, in that it is only able to disambiguate a few words at a time.

All Words Disambiguation Task: A more general approach is to disambiguate all of the open-class words in a context. An "all words" approach is often more useful for certain natural language tasks.. Supervised systems have historically suffered in the task, mainly because of the unavailability of sufficient tagged training instances for every open-class word. Hence, a lot of attention has been paid to unsupervised, semi-supervised, knowledge-based approaches to the problem, which can make use of large-scale knowledge resources for disambiguation. This thesis focuses on all words disambiguation tasks but also highlights target word tasks.

2.2 Knowledge Sources for WSD

Knowledge sources used for WSD are either lexical knowledge released to the public like WordNet, or world knowledge learned from a training corpus [12, 17]. The lexical knowledge usually has form of some sort of dictionary and which is used in knowledge based approaches.

2.2.1 Lexical Knowledge

In the late 80s and throughout the 90s, large efforts have been carried out on developing manually large-scale knowledge bases: WordNet is example of such resources. Currently, WordNet is the best known and the most used resource for WSD in any language. In WordNet the concepts are defined as synonymy sets called *synsets* linked one to each other through semantic relations (hyperonymy, hyponymy, meronymy, antonymy, and so on). Each sense of a word is linked to a *synset*. In this research, we use Amharic WordNet (AWN) to disambiguate words. *Sense Frequency* is the usage frequency of each sense of a word. Interestingly, the performance of the naïve WSD algorithm, which simply assigns the most frequently used sense to the target, is not very bad. Thus, it often serves as the benchmark for the evaluation of other WSD algorithms. *Sense glosses* provide a brief explanation of a word sense, usually including definitions and examples, by counting common words between the gloss and the context of the target word. *Concept Trees* represent the related concepts of the target in the form of semantic networks as is done by WordNet. The commonly used relationships include hypernym, hyponym, holonym, meronym, and synonym. Many WSD algorithms can be derived on the basis of concept similarity measured from the hierarchal concept tree [3]. *Selectional Restrictions* are the semantic restrictions placed on the word sense. LDOCE (Longman Dictionary of Contemporary English) senses provide this kind of information. For example, the first sense of run is usually constrained with human subject and an abstract thing as an object. *Part of Speech (POS)* is associated with a subset of the word senses in both WordNet and LDOCE. That is, given the POS of the target, they may fully or partially disambiguate word sense.

WordNet

WordNet is like a dictionary in that it stores words and meanings. However, it differs from traditional ones in many ways. For instance, words in WordNet are arranged semantically instead of alphabetically. Synonymous words are grouped together to form synonym sets, or synsets. Each synset represents a single distinct sense or concept. The basic object in WordNet is a set of strict synonyms called a synset. By definition, each synset in which a word appears is a different sense of that word. WordNet is a large lexical database. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and

concepts can be navigated with the browser. WordNet is like a dictionary in that it stores words arranged semantically instead of alphabetically [5, 13].

WordNet's structure makes it a useful tool for computational linguistics and natural language processing. WordNet specially resembles a thesaurus, in that it groups words together based on their meanings. However, there are some important distinctions. First, WordNet interlinks not just word forms strings of letters but also specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the groupings of words in a thesaurus do not follow any explicit pattern other than meaning similarity [6, 13].

The English WordNet that was created and being maintained at the Cognitive Science Laboratory of Princeton University under the direction of psychology professor Miller [29] inspire the idea of Amharic WordNet. Its development began in 1985. English WordNet is organized by semantic relations. Since a semantic relation is a relation between meanings, and since meanings can be represented by synsets, it is natural to think of semantic relations as pointers between Synsets.

Unlike most dictionaries, WordNet contains only open-class words (nouns, verbs, adjectives, and adverbs). WordNet does not contain closed class words such as pronouns, conjunctions and prepositions. WordNet is organized semantically (as part of speech) and the central object in WordNet is a synset, a set of synonyms. WordNet organizes the lexical information in terms of word meanings and it can be termed as lexicon based on psycholinguistic principles. Each word may have one or more senses and these are classified as Homonynms, Monosemous and Polysemous. Homonynms- A case of homonymy is one of an ambiguous word, where different cases are related to each other in any way. Words that are identical in sound and spelling are called homonyms, e.g. እርሳስ (ስም) (ክብደት ና ለስላሳነት ያለው፥በቀላሉ የሚቀልጥ የማእድን አይነት), እርሳስ (ስም) (የጥይት አረር) and እርሳስ (ስም) (ለመፃፊያ፥ለመሳያ የሚያገለግል ከእርሳስ ማእድን የሚሰራ የፅህፈት መሳሪያ). Monosemous-word with only one sense are said to be monosemous. e.g., "ሀውልት" has only one sense as "በሰው ወይም በሌላ አምሳል ከድንጋይ ወይም ከነሀስ... የተሰራ መታሰቢያ".

Polysemous-they are words with multiple senses. In WordNet, each word occurs in as many synsets as it has senses. Word sense disambiguation is the task of assigning sense labels to occurrences of an ambiguous word. This problem can be divided into two sub problems [30]: sense discrimination and

sense labeling. Word sense discrimination is easier than full disambiguation since we need only determine which occurrences have the same meaning and not what the meaning actually is. For example, the word "ሳለ" occurs in two synsets as noun and three synsets as verb.

Parts of Speech in WordNet

WordNet stores information about words that belong to four Par-Of-Speech: nouns, verbs, adjectives and adverbs [11]. These are arranged in their respective synsets. Prepositions and conjunctions don't belong to any synset.

*Nouns in WordNet***:** Noun words have various relations defined in WordNet for the Noun Part of speech. These relations are Hypernymy and Hyponymy, Meronymy and Holonymy and Antonymy.
Hypernymy and Hyponymy: These are two most common relations for nouns. They are semantic relationships that connect two synsets if the entity referred to by one is a kind of or is a specific example of entity referred to by other. Specifically, if synset A is kind of B synset, then B is a hyponym of A, and A is the Hypernym of B. The number of hypernym links is equal to the number of hyponym linkes since for every hypernym link there is a corresponding hyponym link. Meronymy and Holonymy: These are also semantic relationships that connect two synsets if A is part of B conversely B is a holonymy of A [11].

*Verbs in WordNet***:** Verb words have various relations defined in WordNet for the Verb Part of speech. These relations are Troponymy, Antonymy, Entailment, and Cause. These Troponym and Antonymy are analogous to the noun hypernymy ad hyponymy respectively. Synset A is the hypernym of B, if B is one way to A; A is then the troponym of B. Antonymy: Like nouns,verbs are also related through the relationship of antonymy that links two verbs that are opposite to each other in the meaning. This is a lexical relationship and does not belong to the other words in the synsets that both belongs to. Entailment and Cause: Other relations defined for verbs include those of entailment and cause, both of which are semantic relations. A synset A is related to synset B through the entailment relationship if A entails B [11].

*Adjectives and Adverbs in WordNet***:** *Adjectives and Adverb words have* various semantic relations defined in WordNet are- Similar-to and Also-see.
Similar-to**:** It is defined for Adjectives. This semantic relationship links two adjective synsets that are similar in meaning, however not close enough to be put together in the same synset**.** Also-see**:** This

relation is common to both adjective and verbs. All links of this type of adjective are semantic in nature but they are not lexical relations

2.2.2 Learned World Knowledge

World knowledge is too complex or trivial to verbalize completely. Therefore, it is a smart strategy to acquire automatically world knowledge from the context of training corpora on demand by machine learning techniques [12, 15]. The frequently used types of contextual features for learning are listed below.

Indicative Words surround the target and can serve as the indicator of target senses. In general, the closer to the target word, the more indicative to the sense. There are several ways, like fixed-size window, to extract candidate words. Domain-specific Knowledge, like selectional restrictions, is about the semantic restrictions on the use of each sense of the target word. However, domain-specific knowledge can only acquire from training corpora, and can only be attached to WSD by empirical methods, rather than by symbolic reasoning. There are no significant distinctions between lexical knowledge and learned world knowledge. If the latter is general enough, it can be released in the form of lexical knowledge for public use. Usually, unsupervised approaches and knowledge-based approaches use lexical knowledge only, while supervised approaches employ learned world knowledge for WSD [12, 15]. For this study, we use Lexical Knowledge as knowledge source.

2.3 Approaches for WSD

Most disambiguation approaches tend to focus on the identification of word-specific contextual indicators that can be used to distinguish between a word's senses. Efforts to acquire these clues or indicators have been characterized by their need for intensive human involvement for each word, which creates the associated problem of limited vocabulary coverage. This is termed as the knowledge acquisition bottleneck in WSD literature. Approaches to WSD are often classified according to the main source of knowledge used in sense distinction. Methods that rely primarily on dictionaries, thesauri, and lexical knowledge bases, without using any corpus evidence, are termed dictionary-based or knowledge-based. Methods that rely on external information and work directly from raw un annotated corpora are called unsupervised methods (adopting terminology from machine learning). Finally, supervised and semi-supervised WSD make use of annotated corpora to train from, or as seed

data in a bootstrapping process. WSD systems can thus be classified based on how they attempt to deal with the knowledge acquisition bottleneck, by considering how they acquire disambiguation information. Based on this a WSD system can be classified as knowledge-based, Machine learning (Corpus-Based) or hybrid [2] and each of these approaches will be discussed briefly in the following sub-sections.

2.3.1 Knowledge-based WSD

Knowledge-based methods rely on information that can be extracted from a knowledge source, such as a dictionary, thesaurus or lexical database. These methods learn based on information from curated and structured data where as supervised and clustering methods learn from example instances. The advantage of the knowledge-based methods over the supervised and the clustering methods is that training data is not required for each word that needs to be disambiguated. This allows the system to disambiguate words in running text, referred to as all-words disambiguation. All-words disambiguation methods have an advantage over what is termed lexical-sample disambiguation methods because lexical-sample methods can only disambiguate words in which there exists in a sufficient set of training data. All-word disambiguation methods are scalable and can be used in real-word practical applications in which ambiguous words may not be known ahead of time and training data is difficult to obtain. The disadvantage to this method is that it is language and domain dependent because a knowledge source is required in the appropriate language and domain. Historically, it has also not obtained as high of disambiguation accuracy as supervised methods. The knowledge-based approach is preferable because it does not require training and is capable of disambiguation arbitrary text and the ambiguous words are resolved using contextual information found in the sentence [2].

The knowledge-based method disambiguates words by matching context with information from a prescribed knowledge source. WordNet is used because it combines the characteristics of both a dictionary and a structured semantic network, providing definitions for the different senses of the English words and defining groups of synonymous words by means of synsets, which represent distinct lexical concepts. WordNet also organizes words into a conceptual structure by representing a number of semantic relationships (hyponymy, hypernymy, meronymy, etc.) among synsets [14].

The main techniques of knowledge based WSD are the measures of semantic similarity, selection restriction, heuristic based WSD and overlap based approach. A review knowledge based approaches

can be found in [6]. Knowledge sources used for WSD are either lexical knowledge or world knowledge, in which lexical knowledge released to the public, or world knowledge learned from a training corpus [16]. These methods mainly try to avoid the need of large amounts of training materials required in supervised methods. Knowledge-based methods can be classified in function of the type of resources they use: 1) Machine-Readable Dictionaries; 2) Thesauri; or 3) Lexical Knowledge Bases.

Machine-Readable Dictionaries (MRDs) provide a ready-made source of information about word senses and knowledge about the world, which could be very useful for WSD and Natural Language Understanding (NLU). Since Lesk [10], the first WSD based on MRD and many researchers have used machine-readable dictionaries (MRDs) as a structured source of lexical knowledge to deal with WSD. These approaches, by exploiting the knowledge contained in the dictionaries, mainly seek to avoid the need for large amounts of training material. As stated by Agirre and Martinez [15], ten different types of information, which is useful for WSD, can be obtained from MRDs. This information includes part of speech, semantic word associations, syntactic cues, selection preferences, and frequency of senses. Since the work by Lesk [10], many researchers have used MRDs as structured source of lexical knowledge for WSD systems. However, MRDs contain inconsistencies and are created for human use, and not for machine exploitation. WSD techniques using MRDs can be classified according to: the lexical resource used (mono or bilingual MRDs); the MRD information exploited by the method (words in definitions, semantic codes, etc.); and the similarity measure used to relate words from context and MRD senses. Lesk created a method for guessing the correct word sense counting word overlaps between the definitions of the word and the definitions of the context words, and selecting the sense with the greatest number of overlapping words. Although this method is very sensitive to the presence or absence of the words in the definition, it has served as the basis for most of the subsequent MRD-based disambiguation systems.

Thesauri provide information about relationships among words, especially synonymy. Like MRDs, a thesaurus is a resource created for humans and, therefore, is not a source of perfect information about word relations. However, thesauri provide a rich network of word associations and a set of semantic categories potentially valuable for large-scale language processing. Roget's International Thesaurus is the most used thesaurus for WSD. Yarowsky [7] used each occurrence of the same word under different categories of a thesaurus as representations of the different senses of that word. The resulting

classes are used to disambiguate new occurrences of a polysemous word. Yarowsky notes that his method concentrates on extracting topical information.

The Lesk method [10] is the seminal dictionary-based method. It is based on the hypothesis that words used together in text are related to each other and that the relation can be observed in the definitions of the words and their senses. Two (or more) words are disambiguated by finding the pair of dictionary senses with the greatest word overlap in their dictionary definitions without using any corpus evidence. In the following section, we will discuss briefly four main types of knowledge-based methods for word sense disambiguation [15].

Measures of Semantic Similarity for WSD

Words that share a common context are usually related in meaning, and therefore the appropriate senses can be selected by choosing those meanings found within the smallest semantic distance. This category includes methods for finding the semantic density or distance between concepts. Depending on the size of the context they span, these measures are in turn divided in to two main categories. These are local context and global context [2].

Local context is one of the kind of semantic constraint which able to provide unity to an entire discourse, its scope has been usually limited to a small number of words found in the immediate vicinity of a target word, or to words connected by syntactic dependencies with the target word. These methods target the local context of a given word, and do not take into account additional contextual information found outside a certain window size. Methods applicable to a local context, semantic measures are used to disambiguate words connected by syntactic relations or their locality.

Other methods rely on a global context and attempt to build threads of meaning throughout an entire text, with their scope extended beyond a small window centered on target words. Lexical chains are an example of such semantic relations drawn across several words in a text. Methods applicable to a *global context*, where lexical chains are derived based on measures of semantic similarity (a lexical chain is a thread of meaning drawn throughout an entire text).

Lexical chains are some of the most widely known structures of meaning. A lexical chain is a sequence of semantically related words, which creates a context and contributes to the continuity of meaning and

the coherence. They are useful for various tasks in natural language processing, including text summarization, text categorization, and word sense disambiguation of a discourse. Similar to the Lesk algorithm, these similarity methods become extremely computationally intensive when more than two words are involved. However, solutions designed to increase the efficiency of the Lesk algorithm are equally applicable here, in which each ambiguous word in disambiguated individually, using a method similar in spirit with the simplified Lesk algorithm. The application of measures of semantic similarity to the disambiguation of words in unrestricted text is not always a straightforward process. A text usually involves more than two ambiguous words [2, 15].

Selectional Preferences (restrictions) for WSD

A knowledge-based approach is one, which attempt of Selection preferences to restrict the number of meanings of a target word occurring in context. Selection restrictions are constraints on semantic type that a word sense imposes on the words with which it connects usually through grammatical relationships in sentences. They have frequently useful information for WSD. However, their use is limited and the additional sources of knowledge are required for full and accurate WSD. Indeed, the exemplar or sense disambiguation in most computational setting is Katz and Fodor's use of Boolean selection restrictions to constrain semantic interpretation [10]. Automatically or semi-automatically acquired selectional preferences as a means of constraining the possible meanings of a word, based on the relation it has with other words in context.

Some of the earliest algorithms for word sense disambiguation rely on selectional preferences as a way of constraining the possible meanings of a word in a given context. Selectional preferences capture information about the possible relations between word categories, and represent common sense knowledge about classes of concepts. EAT-FOOD, DRINK-LIQUID, are examples of such semantic constraints, which can be used to rule out incorrect word meanings and select only those senses that are in harmony with common sense rules. For instance, given the sentence "ከበደ ወይን ጠጣ", the 'color' sense of "ወይን" does not fit in context since the verb "ጠጣ" requires a liquid as a direct object [15].

The main reason seems to be the circular relation between selectional preferences and WSD: learning accurate semantic constraints requires knowledge of the word senses involved in a candidate relation, and vice versa. WSD can improve if large collections of selectional preferences are available.

Lesk Algorithm for WSD

The Lesk algorithm, in which the most likely meanings for the words in a given context are identified based on a measure of contextual overlap among dictionary definitions pertaining to the various senses of the ambiguous words. The Lesk algorithm is one of the first algorithms developed for the semantic disambiguation of all words in unrestricted text. The only resource required by the algorithm is a set of dictionary entries, one for each possible word sense, and knowledge about the immediate context where the sense disambiguation is performed. It is a simple knowledge-based approach, which relies on the calculation of the word overlap between the sense definitions of two or more target words. This approach is named gloss overlap or the Lesk algorithm after its author Lesk [10].

These require a Machine Readable Dictionary (MRD). They find the overlap between the features of different senses of an ambiguous word (sense bag) and the features of the word in its context. Its basic functioning involves finding an overlap amongst the features of different senses of an ambiguous word and the features in its content. These features could be assigned weights. The sense with the maximum overlap is nominated as the contextually appropriate word. The original Lesk algorithm performs WSD by calculating the relative word overlap between the context usage of a target word, and the dictionary definition of each of its senses in a given MRD. The sense with the highest overlap is then assumed the correct one. The Lesk algorithm, in which the most likely meanings for the words in a given context are identified based on a measure of contextual overlap among dictionary definitions pertaining to the various senses of the ambiguous words [10].

The main idea behind the original definition of the algorithm is to disambiguate words by finding the overlap among their sense definitions. Lesk first determines the overlap of the corresponding definitions by counting the number of words they have in common. Next, the sense pair with the highest overlap is selected, and therefore a sense is assigned to each word in the initial word pair. The Lesk algorithm is an appealing solution for identifying word senses when the only resource available is a set of dictionary definitions. For example, the senses of the word ***"ሳለ"*** *and* ***"ቢላዋ"*** listed below and words, which overlap with the "አበበ ቢላዋ ሳለ" input sentence, are marked in bold Italic:

The senses of the word "ሳለ" are:

1. የአንድን ነገር መልክ፡ቅርጽ ወይም አንድን አይነት ሀሳብ በስእል አሳየ፣ ምስሉን በወረቀት ፣በሸራ፣በግድግዳ ወዘተ ቀረፀ፣ አሰፈረ፣ነደፈ
2. ሞረደ ፣አሾለ፣ ***መቁረጫ*** ጠርዝን አተባ ፣ **ስለት** አወጣ
3. ይህን ካደረክለኝ ይህን አደርግልሀለው በማለት ለአምላክ፡ለመላእክት የሚቀርብ ለመና፡ብፅአት
4. ከጉሮሮ አየርን በሀይል ና በተደጋጋሚነት አስወጣ፡ ትክትክ አደረገ፣ኡህ ኡህ አለ
5. ካለ

The senses of the word "ቢላዋ" are:

1. ካራ፣ሰንጢ
2. በአንድ ጎኑ ***ሰለት*** ያለው ለስጋ ለሽንኩርት ***መቁረጫ***፡መክተፊያ የሚያገለግል የወጥ ቤት እቃ
3. በደረቅ ሳር ጫፍ ላይ የሚገኝ እንደ ምርቅ ያለ እሾህማ ጨጎንት

The overlap of Sense 2 of the word "***ሳለ***" and Sense 2 of the word "**ቢላዋ**" has two overlaps, whereas the other senses have zero overlap, so that the second sense is selected for an ambiguous word.

Banerjee and Pedersen [53] introduced a measure of extended gloss overlap, which expands the glosses of the words being compared to include glosses of concepts that are known to be related through explicit relations in the dictionary (e.g., hypernymy, meronymy, hyponyms, etc.). The range of relationships used to extend the glosses is a parameter, and can be chosen from any combination of WordNet relations. The overlap scoring mechanism is also parameterized and can be adjusted to take into account gloss length (i.e., normalization) or to include function words. As stated by Banerjee and Pedersen disambiguation greatly benefits from the use of gloss information from related concepts, which increase the accuracy from 18.3% for the original Lesk algorithm to 34.6% for extended Lesk. However, the approach does not lead to state-of-the-art performance compared to competing knowledge-based systems. This is due to the maximum accuracy obtained by this method is less than other knowledge based methods. For example, Agirre and Martinez [15], obtain accuracy of 86.7% for word-to-class, 97.3% for class to-class methods using selectional preference approach.

Heuristics for Word Sense Disambiguation

Heuristic methods consist of simple rules that can reliably assign a sense to certain word categories, an easy and yet precise way to predict word meanings is to rely on heuristics drawn from linguistic properties observed on large texts. One such heuristic, which is often used as a baseline in the evaluation of many WSD systems, is the most frequent sense heuristic. The other two Heuristics for WSD refer to the tendency of a word to exhibit the same meaning in all its occurrences in a given discourse (one sense-per-discourse), in the same collocation (one-sense-per-collocation) [2].

Most frequent sense is among all possible meanings that a word may have, it is generally true that one meaning occurs more often than the other meaning. This is a very simple method, which is used as a baseline for WSD, and according to Gale *et al.* [56], "most reasonable systems should outperform this baseline."

Gale *et al.* [57] introduced one Sense per Discourse heuristic. It states that a word tends to preserve its meaning across all its occurrences in a given discourse. This is a rather strong rule since it allows for the automatic disambiguation of all instances of a certain word, given that its meaning is identified in at least one such occurrence.

The one-sense-per-collocation heuristic is similar in spirit to the one-sense per-discourse hypothesis, but it has a different scope. Yarowsky [20] introduced it, and it states that a word tends to preserve its meaning when used in the same collocation. In other words, nearby words provide strong and consistent clues to the sense of a target word. It was also observed that this effect is stronger for adjacent collocations, and becomes weaker as the distance between words increases.

Yarowsky [7] used both one-sense-per-discourse and one-sense per-collocation in the developed iterative bootstrapping algorithm, which improved the performance of WSD from 90.6% to 96.5%.

2.3.2 Corpus-based WSD

The types of NLP problems initially addressed by statistical and machine-learning techniques are those of language-ambiguity resolution, in which the correct interpretation should be selected from among a set of alternatives in a particular context. These techniques are particularly adequate for NLP because they can be regarded as classification problems, which have been studied extensively in the ML community. A corpus based approach extracts word senses from a large annotated data which is a sense tagged. Corpus-based approaches are those that build a classification model from examples.

These methods involve two phases: *learning* and *classification*. The learning phase consists of learning a sense classification model from the training examples. The classification process consists of the application to new examples in order to assign the output senses. One of the first and most important issues to take into account is the representation of the examples by means of features or attributes. That is, which information could and should be provided to the learning component from the examples [47]. We can broadly distinguish three main approaches to WSD based on statistical methods [8].

Supervised Corpus-Based Method

Supervised WSD methods rely on the use of manually annotated training data. The instances in the training data are annotated manually with their appropriate concepts from a sense-inventory. A supervised learning algorithm learns to recognize the context surrounding these concepts, creating a model, which is used to automatically assign concepts to instances containing the target word in the test data. Supervised learning methods in general obtain very high disambiguation accuracy, outperforming other WSD methods. The disadvantage of these methods though is that they require manually annotated training data for each word that needs to be disambiguated. This is a labor intensive and time-consuming process. In this method, an evaluation component takes manually annotated training data as input and splits the data into a training and test portion. Supervised models fall roughly into two classes, hidden models and explicit models based on whether or not the features are directly associated with the word sense in training corpora [12].

Supervised methods are based on the assumption that the context can provide enough evidence on its own to disambiguate words (hence, common sense and reasoning are deemed unnecessary). Probably every machine-learning algorithm has been applied to WSD, including associated techniques such as feature selection, parameter optimization, and ensemble learning. Support Vector Machines and memory-based learning have been shown to be the most successful approaches, to date, probably because they can cope with the high-dimensionality of the feature space. However, these supervised methods are subject to a new knowledge acquisition bottleneck since they rely on substantial amounts of manually sense-tagged corpora for training, which are laborious and expensive to create. These approaches use machine-learning techniques to learn a classifier from labeled training sets, that is, sets of examples encoded in terms of a number of features together with their appropriate sense label (or class). Supervised WSD uses machine-learning techniques for inducing a classifier from semantically annotated corpora. Generally, supervised systems have obtained better results than unsupervised ones,

a conclusion that is based on experimental work and international competitions. This approach uses semantically annotated corpora to train machine learning (ML) algorithms to decide which word sense to choose in which contexts. The words in such annotated corpora are tagged manually using semantic classes taken from a particular lexical semantic resource. Corpus-based methods are called "supervised" when they learn from previously sense-annotated data, and therefore they usually require a large amount of human intervention to annotate the training data. Although several attempts have been made to overcome the knowledge acquisition bottleneck (too many languages, too many words, too many senses, too many examples per sense) it is still an open problem that poses serious challenges to the supervised learning approach for WSD.

In supervised approaches, a sense disambiguation system is learned from a representative set of labeled instances drawn from sense-annotated corpus. Input instances to these approaches are features encoded along with their appropriate labels. The output of the system is a classifier system capable of assigning labels to new feature encoded inputs.

Unsupervised Corpus-Based Method

Unsupervised methods rely on unannotated training data. In general, these methods perform word sense discrimination rather than disambiguation. Discrimination seeks to cluster instances of a given target word such that instances that use the same concept of the target word are in the same cluster, while disambiguation seeks to determine the appropriate concept of an instance given a sense-inventory. In order to evaluate clustering methods, though, the disambiguation of the words in a test data set is required. One advantage to clustering is that a large amount of manually annotated training data is not required; the labeling of the instances in the training data is done using clustering algorithms, rather than by human annotators as with the supervised methods. Another advantage of clustering is that it is language and domain independent requiring only a corpus in the language and domain of interest. The disadvantage is that training data is required for each word that needs to be disambiguated and historically this method does not obtain as high of disambiguation accuracy as supervised methods. Unsupervised methods have the potential to overcome the knowledge acquisition bottleneck, that is, the lack of large-scale resources manually annotated with word senses. These methods are based on unlabeled corpora, and do not exploit any manually sense-tagged corpus to provide a sense choice for a word in a context. These approaches to WSD are based on the idea that the same sense of a word will have similar neighboring words. They are able to induce word senses from input text by clustering

word occurrences, and then classifying new occurrences into the induced clusters. They do not rely on labeled training text and, in their purest version, do not make use of any machine-readable resources like dictionaries, thesauri, ontologies, etc. However, the main disadvantage of fully unsupervised systems is that, as they do not exploit any dictionary, they cannot rely on a shared reference inventory of senses. While WSD is typically identified as a sense labeling task, that is, the explicit assignment of a sense label to a target word, unsupervised WSD performs word sense discrimination, that is, it aims to divide "the occurrences of a word into a number of classes by determining for any two occurrences whether they belong to the same sense or not". Consequently, these methods may not discover clusters equivalent to the traditional senses in a dictionary sense inventory [26].

Admittedly, unsupervised WSD approaches have a different aim than supervised and knowledge-based methods, that is, that of identifying sense clusters compared to that of assigning sense labels. However, sense discrimination and sense labeling are both sub-problems of the word sense disambiguation task and are strictly related, to the point that the clusters produced can be used at a later stage to sense tag word occurrences.

Semi-Supervised Corpus-Based Method

Semi-supervised learning method falls in the middle between unsupervised learning and supervised learning that makes use of both a small amount of labeled and a large amount of unlabeled data for training. The semi-supervised or minimally supervised methods are gaining popularity because of their ability to get by with only a small amount of annotated reference data while often outperforming totally unsupervised methods on large data sets. There are a host of diverse methods and approaches, which learn important characteristics from auxiliary data and cluster or annotate data using the acquired information. The bootstrapping approach starts from a small amount of seed data for each word: either manually tagged training examples or a small number of surefire decision rules (e.g., play in the context of bass usually indicates the musical instrument). The seeds are used to train an initial classifier, using any supervised method. This classifier is then used on the untagged portion of the corpus to extract a larger training set, in which only the most confident classifications are included. The process repeats, each new classifier being trained on a successively larger training corpus, until the whole corpus is consumed, or until a given maximum number of iterations are reached. Bootstrapping (BS) looks like supervised approaches, however it needs only a few seeds instead of a large number of training examples [27].

The bootstrapping approach is situated between the supervised and unsupervised approach of WSD. The aim of bootstrapping is to build a sense classifier with little training data, and thus overcome the main problems of supervision: the data scarcity problem specially lack of annotated data. The bootstrapping methods use a small number of contexts labeled with senses having a high degree of confidence. This could be accomplished by hand tagging with senses the contexts of an ambiguous word for which the sense of ambiguous word is clear because some seed collocations [11] occur in these contexts.

Other semi-supervised techniques use large quantities of untagged corpora to provide co-occurrence information that supplements the tagged corpora. These techniques have the potential to help in the adaptation of supervised models to different domains. In addition, an ambiguous word in one language is often translated into different words in a second language depending on the sense of the word. Word-aligned bilingual corpora have been used to infer cross-lingual sense distinctions, a kind of semi-supervised system.

2.3.3 Hybrid Approaches

Several models of WSD have been proposed, mainly for monolingual contexts, including knowledge-based approaches, which make use of linguistic (and eventually extra-linguistic) knowledge manually codified or extracted from lexical resources; corpus-based approaches, which make use of knowledge automatically acquired from text using Machine Learning (ML) algorithms; and hybrid approaches, which merge characteristics from both approaches. These approaches can neither be properly classified as knowledge or corpus-based, since they obtain disambiguation information from both corpora and explicit knowledge bases.

Luk's [22] system is an example of a hybrid approach that combines information in MRD definitions with statistical information obtained from raw corpora. The researcher uses textual definitions of senses from the LDOCE to identify relations between senses. To determine which of these relations are most useful for WSD, the researcher uses a corpus to compute Mutual Information (MI) scores between these related senses.

Bootstrapping approaches where initial (seed) data comes from an explicit knowledge source, which is then augmented with information derived from corpora, are another example of hybrid systems. Yarowsky's [7] unsupervised system is a good example of a bootstrapping approach. The researcher

defines a small number of seed dentitions for each of the senses of a word (the seeds can also be derived from dictionary definitions or lexicons such as WordNet synsets). The researcher uses the seed definitions to classify the 'obvious' cases in a corpus. Decision lists are used to learn generalizations based on the corpus instances that have already been classified. This process is repeatedly iteratively to the corpus, classifying more instances. Learning proceeds in this way until all corpus instances of the ambiguous word have been classified.

Hybrid systems aim to harness the strengths of the individual approaches while at the same time, overcoming specifics limitations associated with a particular approach, to improve WSD accuracy. They operate on a 'knowledge-driven, corpus-supported' theme, utilizing as much information as possible from different sources. For example, Luk successfully exploits a lexical resource to reduce the amount of training data required for WSD, while Yarowsky's seeds provide initial knowledge, critical to the statistical learning phase.

2.4 Design Requirements

The design and realization of every WSD must consider the language feature that it is intended for. In designing Amharic WordNet based word sense disambiguation using, typical features of the language composition is a crucial role. In addition to this, four main elements are required in designing every WSD system: the selection of word senses, the use of knowledge sources, the representation of context, and the selection of an automatic classification approach [21].

2.4.1 Knowledge Sources

Knowledge source provides data, which are essential to associate senses with words. Knowledge sources used for WSD are either lexical knowledge released to the public, or world knowledge learned from a training corpus [21]. Lexical knowledge is usually released with a dictionary where as world knowledge is too complex to be verbalized completely. Therefore, it is a good strategy to automatically acquire world knowledge from the context of training corpora on demand by machine learning techniques. For this study, we use Amharic WordNet as knowledge source that we have developed.

2.4.2 Selection of Word Senses

A *word sense* is a commonly accepted meaning of a word. For example, consider the following two sentences:

- ልጅቷ የገዛችው ቅቤ ለጋ ነው::
- ደመናው ለጋ ነው::

The word "ለጋ" used in the above sentences with two different senses "ያልቆየ ና ያልበሰለ ወይም ያልኮመጠጠ ቅቤ" for the first sentence and "ቀለል ያለ አዲስ ደመና" for the second sentence. The example makes it clear that determining the sense inventory of a word is a key problem in word sense disambiguation. A ***sense inventory*** partitions the range of meaning of a word into its senses [21].

2.4.3 Representation of Context

As text is an unstructured source of information, to make it a suitable input to an automatic method it is usually transformed into a structured format. To this end, preprocessing of the input sentence is usually performed, which typically includes normalization, tokenization and stop-word removal.

We observe two of the most important information sources for determining the sense of a polysemous word are the topic of context and relational information representing the structural relations between the target word and the surrounding words in a local context.

2.4.4 Choice of a Classification Approach

Three main approaches have been applied in the field of WSD. These are knowledge-based approaches, corpus based approaches and hybrid approach. Knowledge based approaches use WordNet. It relies on information provided by Amharic WordNet. Corpus based approaches can be divided into three types, supervised, unsupervised and semi-supervised learning approaches. Supervised learning approaches use information gathered from training on a corpus that has sense tagged for semantic disambiguation. Unsupervised learning approaches determine the class membership of each object to be classified in a sample without using sense tagged training examples. The semi-supervised learning approaches starts from a small amount of seed data for each word: either manually tagged training examples or a small

number of surefire decision rules. For this study, we use knowledge-based approach to disambiguate words.

2.5 Amharic Language and Amharic Word Ambiguity

2.5.1 Amharic Language

Amharic is the working language of the Federal Government of Ethiopia. Amharic is one of Semitic languages, which is spoken in many parts of Ethiopia. The language is written in the unique and ancient Ethiopic script, which is called fidel, which is inherited from the Geez language and which is currently used only in Ethiopian Orthodox Tewahedo Church as worshiping language. *Ethiopic ("Fidel") is the* name given to the script used by Ethiopians and Eritreans to write their language. It is the only script indigenous to the African continent for official purpose. Foreign linguists [35] mostly use the name Ethiopic.

According to the 2010 census, Amharic is the most spoken language in Ethiopia. Over 17 million people speak Amharic in Ethiopia and about 2.7 million immigrants outside Ethiopia including Israel, Egypt and Sweden [33, 34]. The alternative word for Amharic is Abyssinian, Amarigna, and Ethiopian [33]. It is one of the semantic languages having its own script. The scripts are orthographic representation of the phonemes in the language. Amharic is widely spoken language even in the world and it is the second most spoken language next to Arabic [34].

The total Fidel in Amharic becomes 276 distinct symbols today, 20 numerals and 8 punctuations [38, 55]. These Fidel is written left-to-right unlike Arabic [39]. Although the speakers have used Amharic scripts, the writing system has problem of standardizations. The first problem is the presence of "unnecessary" alphabets (fidels) in the language writing system. These fidels (alphabets) have the same pronunciation but different symbols. These different fidels used interchangeably without meaning change. The fidels are ኦ and ዐ, ጸ and ፀ, ሰ and ሠ, and ሀ, ሐ, and ኀ. For example, the word "force" can be written as, ሀይል, ሃይል, ኃይል,ኀይል, ሐይል, etc. all meaning the same thing, although written differently [35, 40].

Amharic has its own writing system, a semi-syllabic system. There is no agreed translation of Amharic symbols to Roman characters (used in English) [54]. Hence, understanding Fidel usage in words is the expected task to deal with words. If a word is written in a consistent Fidel, it provides efficient and

accurate natural language processing systems. Such types of word ambiguities can be handled through character normalization.

The Amharic writing system holds 17 punctuation marks in addition to alphabets of which only a few of them are commonly used and have representations in Amharic software [39]. These different Amharic punctuation marks are used for different functions. The following are some punctuation marks commonly used today [41]. **ሁለት ነጥብ** (**:**), this mark is used for separating words. For the modern Amharic, it is left except for hand writing purpose. Its place is almost completely taken over by white space. In English, words are separated by white space. This shows that modern Amharic borrows some writing style from English. **አራት ነጥብ** (**::**) is Sentence separator which is basic punctuation marks in Amharic writing system. It shows the end of a sentence, as a single dot implies end of a sentence in English. **ነጠላ ሰረዝ** (፣) separates lists in Amharic text with equivalent function of comma in English. **ድርብ ሰረዝ** (፤), has the equivalent function of semi-colon of English.

In addition to indigenous punctuation marks, some marks have been borrowed from foreign languages. For instance, the **exclamation mark** "!" and the **question mark** "?" are borrowed from English and used in Amharic language [39]. Unlike English Amharic does not have upper and lower case representations.

The removal of punctuation marks increases the effectiveness and efficiency of natural language processing systems [33] as morphological analyzer and stop word removal does. Like punctuation removal, morphological analyzer and stop word removal increases WSD systems` performance. Asker *et al.* [42] define stemming as "a technique whereby morphological variants are reduced to a single stem". For languages, as if Amharic with very rich morphology it is intuitively assumed that morphological analyzer will have a positive effect for WSD tasks.

Amharic has a complex morphology. Sentences in Amharic are often short in terms of the number of words they are formed [38]. This nature of the language makes the window size (bag of context words) narrow. In other token, context words surrounding the word have more advantage for disambiguation purpose in WSD area. Studying morphological aspects of languages helps to distinguish between lexical components of words, which are accountable for the semantics of the words, and grammatical words. Verbs are morphologically the most complex word class in Amharic with many inflectional forms. Sentences in Amharic are often short in terms of the number of words they are formed [38].

Verbs are morphologically the most complex word class in Amharic with many inflectional forms. Even though Amharic sentences are short, it is morphologically rich with inflectional and derivational variants [39].

Syntactically, Amharic is an SOV language i.e. subject + object+ verb [36]. For example the sentence in English "Abebe sharpen the knife" can be written in Amharic as (አበበ ቢላዋ ሳለ).We know that the Amharic word sale(ሳለ) has four meanings that is "sharp", "to vow", "to drew", and the other is "to cough". Therefore, it is an ambiguous word. Disambiguation can be performed to identify what the sentences are talking about after considering neighboring words (አበበ, ቢላዋ).

2.5.2 Ambiguities in Amharic Language

Ambiguit**y** is an attribute of any concept, idea, statement or claim whose meaning, intention or interpretation cannot be definitively resolved according to a rule or process consisting of a finite number of steps. In ambiguity, specific and distinct interpretations are permitted (although some may not be immediately apparent), whereas with information that is vague, it is difficult to form any interpretation at the desired level of specificity.

Lexical ambiguity occurs whenever a word has more than one sense in a given text where as the syntactic ambiguity occurs when the order of words have more than one grammatical relationship that change the meaning of the text. Listeners or readers [43] can disambiguate the ambiguity words in a language raised by speakers or writers for understandability.

Getahun [44] identified six types of ambiguity in Amharic: Lexical Ambiguity, Phonological Ambiguity, Structural Ambiguity, Referential Ambiguity, Semantic Ambiguity, and Orthographic ambiguity. We will summarize each type of ambiguity as follows and the examples adopted from [1].

Lexical Ambiguities

Word meanings in more than one sense can lead to different interpretations by different individuals. The scope of lexical ambiguity is on individual words or word-level understanding [43, 45]. Lexical ambiguity comes into being when two or more of the meanings of a word are applicable in a given situation. In other words, lexical ambiguity refers to a case in which either a lexical unit belongs to different part-of-speech categories with different senses, or to a lexical unit for which there is more

than one sense, while these different senses fall into the same part of-speech category [37, 38]. Some factors that can cause lexical ambiguity are synonymy, homonymy, homophonous affixes and categorical ambiguity [1, 25, 26, 27].

Categorical Ambiguity: Caused from lexical elements having the same phonological and homographic form which belongs to different word class. It can be explained using the following ambiguous word: consider the Amharic sentences, *Beklo ayehugn,* "በቅሎ አየሁኝ" The word "*beklo*" is ambiguous since it has either a noun or a verbal meaning. When we interpret the word, it provides senses as:

1. When the word *beklo* used as noun, it means, "I saw a mule".

2. When the word comes with verbal meaning, it means "I saw something which is grown (may be for a type of plant)".

Synonymy: Words those are very close in meaning [39]. Some words are found to be used with different meaning in different areas in synonyms way**.** For instance, the concepts denoted by all three Amharic words, "*ካራ*", "*ሰንጢ*" and "*ቢላዎ*" are synonymous. These words are used in exclusion in different areas to refer to the English word "knife". The knowledge of one such synonymy words in this area can be ambiguous to speakers in the other areas.

Homonymy: refers to words that share similar lexical nature. These can be ambiguous words that are spelled and pronounced alike but they have different meanings [1]. For instance, the word sale (ሳለ) has the same spelling and sound for two meanings. Consider Amharic text "አበበ ስእል ሳለ", from this example the word *sale* (ሳለ) can be interpreted as "draw" or "to paint". It can also be understood as "cough" based on the context words around the ambiguous word "ሳለ" in the sentence, "አበበ ጉንፋን ስለያዘው ሳለ". It can also be understood as "to sharp" based on the context words around the ambiguous word "ሳለ" in the sentence, "አበበ ቢላዋ ሳለ".

Homophonous Affixes: This ambiguity results when affixes are used for different word classes. The word can be morphologically analyzed into three separate morphemes: the prefix, the root and suffix. Since the meaning of each morpheme remains the same across words, it creates similar words that are difficult to understand [45].

Semantic ambiguity

It determines the possible meanings of a sentence by focusing on the interactions among word level meanings in the sentence. It is caused by polysemy, idiomatic and metaphorical constituents of sentences [39]. It needs semantic disambiguation of words with multiple senses. Semantic disambiguation permits one and only one sense of polysemous words to be selected and included in the semantic representation of the sentence [45].

Polysemy- Many ambiguity words can have different meaning by stressing some characters in the word while reading because most words are polysemy having a range of meanings. Since polysemy, word is a word with different meanings and, therefore results in the rise of ambiguity problems, that again becomes the first issue whenever these words are used in natural language possessing (WSD) systems. Polysemous words depend on the linguistic context to determine their meaning. This makes polysemous words more difficult to disambiguate than other words. They can cause ambiguity not only for semantic ambiguity but also for others too.

Idioms- refer to an expression that means something other than the literal meanings of its individual words. Ambiguity of idioms can be illustrated using the following example which is adapted from [38]. *Hullu agerish* "ሁሉ አገርሽ". The literal meaning of the expression is "every country is hers". Can a person have all countries on the world in practical? , however the idiomatic expression refers to "she is adaptable".

Metaphors- have literal or non-literal (metaphoric) senses. The following is an example of metaphoric ambiguity: *qal seTe* "ቃል ሰጠ". It has two different senses. One sense can be "He makes conversation" and the other provides sense as "He promised".

Phonological Ambiguity

Interpretation of speech sounds within and across words may cause ambiguity. Phonological Ambiguity occurs when the speakers pronounce by creating pause sound. Speaking using pauses and without it leads to word ambiguity [1, 25, 26]. It can be illustrated from the adopted example [1], ደግ ሰው ነበር "*deg + sew neber"v*. In the sentence "+" sign shows where the pause is. When the sentence is

pronounced with pause it senses as “He was a kind man”, however if it is pronounced without pause.It will provide different sense from the previous i.e. “They had preparation for a ceremony”.

Structural (syntactic) Ambiguity

Structural ambiguity results when word order becomes in unorganized manner and holds more than one possible position in the grammatical structure of the sentence. Syntactic disambiguation of words that can function as multiple parts-of-speech is accomplished by reorganizing the order at the syntactic level [1, 25, 26]. Consider the Amharic sentence, “የሀበሻ ታሪክ አስተማሪ” this sentence can have two different interpretations: “*a person who teaches Abyssinian history”* and “*an Abyssinian who teaches history”.* It can be further illustrated using structural organization of the sub-constituent */tarik/* ‘History’.

Referential Ambiguity

This ambiguity arises when a pronoun stands for more than one possible antecedent. For example, “ካሳ ስለተመረቀ ተደሰተ” Kasa sletemereke tedesete. In Amharic, pronoun is understood by default even if it is not written grammatically. The above sentence has two different readings [1, 25]:

1. Kasa was pleased himself because he graduated. “ካሳ ስለተመረቀ ራሱ ተደሰተ።”
2. Somebody was pleased because Kasa graduated. “ካሳ ስለተመረቀ ተደሰተ።”

Orthographic Ambiguity

Orthographic Ambiguity results from geminate and non-geminate sounds. The ambiguity can be resolved using context [1, 25]. In some cases it might not be possible to disambiguate. For example: *liju yislal* ‘ልጁ ይስላል’’.’’ The word “*yislal*” is the cause of ambiguity. The sentence is ambiguous between the following meanings. He draws (“*yslal* “) and He coughs (“*yslal* “). Let us also take the Homograph word which is adapted from [43], '*Gena*'(ገና), to mean “yet”. When the character ‘ና’ is stressed during pronunciation, the meaning is changed as “one of Ethiopian festivals celebrated once a year”. Therefore, depending on the context, the word ገና can have the meaning “still/yet” or “Christ mas”. To sum up, all these ambiguities need to be dealt for Amharic. In the Previous researches [1, 25,

26] limited their works on only lexical ambiguities. In our thesis, we focus on lexical semantic ambiguities and their semantic relation of words.

2.6 Summary

This chapter discussed the application areas of WSD and Knowledge sources for WSD, approaches of WSD as well as Amharic language and Amharic word ambiguity such as Amharic writing system and Amharic Punctuation marks and capitalizations was presented. The field of Knowledge Based with associated algorithms is used for WSD research and WordNet and Part-of-Speech of WordNet has been discussed. We also presented different types of Amharic word ambiguity with examples. For this study, we focus on lexical semantic ambiguity, which we believed to be resolved by WSD.

CHAPTER THREE: RELATED WORKS

3.1 Introduction

This Chapter gives brief description of the various solutions proposed for word sense disambiguation problem and word sense disambiguation using WordNet. Research work on Word Sense Disambiguation started in the 1940's and a wide range of Word Sense Disambiguation algorithms has been proposed over the years. Some of them follow the supervised approach in which labeled training set is utilized, some of the algorithms follow unsupervised approach, which attempts to disambiguate a word without previous training, or labeled corpora. In knowledge-based approach, the algorithm uses the underlying meaning of the text to disambiguate a word. The task of disambiguation system is to resolve the lexical ambiguity of a word in a given context. To put it more precisely, the term "lexical ambiguity" refers to two different concepts "homonymy" and "polysemy". The distinction between bank ("river edge") and bank ("financial institution") has been used as an example of homonym, and rust (verb) and rust (noun) for polysemy. Reflecting the rapid growth in utilization of machine-readable texts, word sense disambiguation techniques have been explored variously in the context of "Corpus-Based and Knowledge-based approaches". This Chapter surveys past research associated with Corpus-Based and Knowledge-based word sense disambiguation. Knowledge-based disambiguation is carried out by using information from an explicit lexicon or knowledge base. The lexicon may be a machine readable dictionary, thesaurus or hand-crafted. [5,18,19] use WordNet as the knowledge-base to disambiguate word senses, and [20] uses Roget's International Thesaurus.

Corpus-based approaches attempt to disambiguate words by using information gained from training on some corpus, rather than taking it directly from an explicit knowledge source [21]. Training can be carried out either on a disambiguated corpus or a raw corpus. In a disambiguated corpus, the semantics of each polysemous lexical item has been marked, while in a raw corpus, the semantics has not been marked yet.

For hybrid Approaches a good example is Luk's system [22] which uses the textual definitions of senses from a machine-readable dictionary to identify relations between senses. It then uses a corpus to

calculate mutual information scores between the related senses in order to discover the most useful information. In this way, the amount of text needed in the training corpus is reduced.

3.2 WSD for Amharic Language

Teshome Kassie [1] has done on WSD for the Amharic language. The researcher has studied how linguistic disambiguation can improve the effectiveness of an Amharic document query retrieval algorithm. The author developed Amharic disambiguation algorithm based on the principles of semantic vectors analysis for improving the precision and recall measurements of information retrieval for Amharic legal texts and implemented in Java. The researcher used the Ethiopian Penal Code which is composed of 865 Articles was used as a corpus in the study. The disambiguation algorithm was used to develop a document search engine.

The researcher developed an algorithm based on distributional hypothesis stating that words with similar meanings tend to occur in similar contexts. For disambiguation of a given word, the author computed the context vector of each occurrence of the words. The context vector was derived from the sum of the thesaurus vectors of the context words. The author constructed the thesaurus by associating each word with its nearest neighbors.

However, for evaluating WSD the author used pseudo words, which are artificial words, rather than real sense tagged words reasoning that it is costly to prepare sense annotation data. The researcher compared the developed algorithm with Lucene algorithm and reported that the algorithm is superior over the Lucene's one. The researcher achieved 58%-82% average precision and recall, respectively.

Solomon Mekonnen [25] used corpus based, supervised machine-learning approach using Naïve Bayes algorithm for Amharic WSD, which is used to check standard optimal context window size, which refers to the number of surrounding words sufficient for extracting useful disambiguation. Based on Naïve Bayes algorithms, experiment found that three-word window on each side of the ambiguous word is enough for disambiguation. The author used a monolingual corpus of English language to acquire sense examples and the sense examples are translated back to Amharic, which is one approach of tackling the knowledge acquisition bottleneck. Based on Naïve Bayes algorithm, the experiments were conducted on WEKA 3.6.2 package. The author concluded that, Naïve Bayes methods achieve higher accuracy on the task of WSD for selected ambiguous words and achieved an accuracy within

the range of 70% to 83% for all classifiers. This is an impressive accuracy for supervised WSD; however, it suffers from knowledge acquisition bottleneck. However, supervised machine learning approach of WSD performs better by human intervention; however, this research has limitations of knowledge-acquisition bottleneck, i.e., it requires manually labeled sense examples which takes much time, very laborious and therefore very expensive to create when the corpus size increases.

Solomon Assemu [26] used a corpus-based approach to word sense disambiguation that only requires information that can be extracted automatically from untagged text. Unsupervised machine learning technique was applied to address the problem of automatically deciding the correct sense of an ambiguous word. The author used corpus of Amharic sentences, based on five selected ambiguous words to acquire disambiguation information automatically. A total of 1045 English sense examples for five ambiguous words were collected from British National Corpus (BNC). The sense examples were translated to Amharic using Amharic-English dictionary. The author tested five clustering algorithms:- simple k-means, hierarchical agglomerative: single, average and complete link and expectation maximization algorithms, in the WEKA 3.6.4 package. Based on the selected algorithms, the author concluded that simple k-means and EM clustering algorithms achieved higher accuracy on the task of WSD for selected ambiguous words. The author achieved accuracy within the range of 65.1 to 79.4 % for simple k means, 67.9 to 76.9 for EM and 54.4 to 71.1 for complete link clustering algorithms for five ambiguous words. The strength of this research is that the context of an ambiguous word is cluster in to a number of groups and discriminate these groups without actually labeling them. However, the limitation of this research is that training data is required for each word that need to be disambiguated and unsupervised method cannot rely on a shared reference inventory of sense. The approach used by the researcher is that the same sense of a particular word will have alike neighbor words and clustering word occurrences and classifying new occurrences into induced clusters.

Getahun Wassie [27] has implemented the other WSD for Amharic language. The author used Semi-supervised method for five words only. To disambiguate words the author used abundant unlabeled training data and limited labeled training data, because labeling the training examples requires human efforts that are costly. So, semi-supervised learning which tries to exploit many unlabeled examples with some seed examples to improve learning performance. Although seed examples selection was a challenging task, semi-supervised learning that tries to exploit many unlabeled examples with some seed examples to improve learning has been implemented for senses disambiguation task in semi-

supervised study. The author used WEKA package for clustering and classifying data. The problem of word ambiguity in Amharic is being tried to be solved by preparing a five selected words corpus after a total of 1031 Amharic sentences were collected. Two clustering algorithms, expected maximization and k-mean, were employed for clustering of sentences into their senses. The average performance of the employed on five classifying algorithms specifically AdaboostM1, bagging, ADtree, SMO, and Naïve Bayes were 83.94%, 78.28%, 88.47%, 87.40% and 47.98% respectively. The strength of this research is that unlike supervised approach, semi-supervised approach needs only a few seeds instead of a large number of training examples. However, the limitation of this research is that training data is required for each word that needs to be disambiguated.

The previous study has a limitation of less data, less coverage of ambiguous words and ambiguity types and corpus was used as a source of information for disambiguation. In our study, Amharic WordNet is used as a source of information for disambiguation. The system to disambiguate words in running text, referred to as all-words disambiguation due to lexical-sample methods can only disambiguate words in which there exists on a sample set of training data in which ambiguous words may not be known ahead of time and the real sense of ambiguous word is retrieved from AWN.

3.3 WSD for Afaan Oromo Language

Tesfa Kebede [28] has done WSD for the Afaan Oromo language. The researcher used a corpus-based approach to disambiguation where supervised machine learning techniques were applied to a corpus of Afaan Oromo language, to acquire disambiguation information automatically. It also applied Naïve Baye's method to find the prior probability and likelihood ratio of the sense in the given context. The author used corpus of Afaan Oromo sentence based on five selected ambiguous words to acquire disambiguation information automatically a total of 1240 Afaan Oromo sense examples were collected for selected five ambiguous words and the sense examples were manually tagged with their correct senses.

A standard approach to WSD is to consider the context of the ambiguous word and use the information from its neighboring or collocation words. The contextual features used in this study were co-occurrence feature, which indicate word occurrence within some number of words to the left or right of the ambiguous word and k-fold cross-validation statistical technique was applied for performance

evaluation. However, supervised machine learning approach of WSD performs better by human intervention; however, this research has limitations of knowledge-acquisition bottleneck, i.e., it requires manually labeled sense examples which takes much time, very laborious and very expensive to create when the corpus size increases and training data is required. The researcher achieved an accuracy of 79% and found four-word window on each side of the ambiguous word is enough for Afaan Oromo WSD.

3.4 WSD for English Language

Faris and Cheng [23] implemented word sense disambiguation for the English language using a knowledge-based approach. The authors propose a robust knowledge-based solution to the word sense disambiguation problem for English language. Ambiguous words are resolved using not only the word's part-of-speech, but also the contextual information found in the sentence. The researchers described a two-phase word-sense disambiguation solution. The first phase is responsible for identifying and locating all the possible knowledge objects corresponding to each term in the given sentence. It uses morphology rules taught to the system to convert words into their base forms, and happens at the first step of parsing a sentence, namely, the syntax step. The second phase is responsible for resolving the ambiguity among all possibilities to correctly identify the intended meaning. Due to the nature of the ambiguities, the researchers classify ambiguities into two categories. One category may be resolved based on the grammar's requirement that a certain *POS* be at a specific place of the given sentence, and hence can be resolved during the parsing stage of a sentence. The other category is resolved during the understanding of the thought, which uses the context information available from the rest of the sentence. Finally, the researcher uses POS and contextual information found in the sentence. Therefore, resolving an ambiguous word based on the word's POS is possible when the parse tree is unambiguous. However, problems may arise when multiple parse trees can be formed due to the absence of an optional term and the presence of a term with an ambiguous POS.

Hee-Cheol Seo et al. [24] did unsupervised word sense disambiguation using WordNet for English language. Word sense disambiguation method for a polysemous target noun using the context words surrounding the target noun and its WordNet relative's words, such as synonyms, hypernyms and hyponyms used to disambiguate. The result of sense disambiguation is a relative that can substitute for that target noun in a context. The selection was made based on co-occurrence frequency between

candidate relatives and each word in the context. Since the co-occurrence frequency is obtainable from a raw corpus, the researchers used unsupervised learning algorithm, unsupervised learning use a raw corpus and therefore does not require a sense-tagged corpus. Finally, the researchers evaluated the developed system on 186 documents in Brown Corpus and achieved 52.34% of recall and the researchers does not considering a way to utilize the similarity between definitions of words in WordNet.

3.5 WSD for Hindi Language

Rohana Sharma [11] developed WSD for Hindi language. The author uses different approaches of word sense disambiguation (WSD) like knowledge-based approaches, machine learning based approaches and hybrid based approaches. In this research the problem of word sense disambiguation is being tried to solve by using Hindi WordNet developed at IIT, containing different words and their sets of synonyms called synsets. By the help of the words in these synsets; the researcher made an attempt to resolve the ambiguity by making the comparisons between the different senses of words in the sentence with the word present in the synset form of the WordNet and the information related to these words in the form of parts-of-speech.

Finally, the researcher found the best approach for word sense disambiguation is knowledge-based approaches and Lesk's algorithm was taken as an example to show its applicability for WSD of Hindi Language. The author has taken an example paragraph and created its context bag and then extracted the semantic bag for the word to be disambiguated and the author has done the overlap between both bags corresponding to each sense of the word and then the appropriate sense of the word is found out. However, the researcher does not evaluate the performance of the developed Hindi WSD system using knowledge-based approach.

Bala [30] has implemented the other WSD for Hindi language. The researcher developed a WSD tool using knowledge-based approach with Hindi WordNet. The researcher studied WSD according to the area to which some words in context are sense tagged. WSD tasks fall into two types: firstly, tag all considerable words (nouns, verbs, adjectives and adverbs) and secondly, tag some considerable words (usually nouns or verbs). In this research the system can improved by using parts-of-speech (POS) and Hindi WordNet. If a word occurs multiple times in different senses in the same text, it is highly likely that the system would assign the same synset or synonyms to all its occurrences. The researcher used a

small corpus with word occurrences and collocations to evaluate word sense disambiguation. The researcher achieved 58% of precision and 50% of recall.

3.6 WSD for Swahili Language

WSD for the Swahili language done by Ng'ang'a [31] is one of them who developed WSD for Swahili using supervised machine learning approach. The researcher addresses the problem of word sense disambiguation within the context of Swahili-English machine translation. The goal of disambiguation is to choose the correct translation of an ambiguous Swahili noun in context. A corpus-based approach to disambiguation is taken, where machine-learning techniques were applied to a corpus of Swahili, to acquire disambiguation information automatically. In particular, the Self-Organizing Map algorithm was used to obtain a semantic categorization of Swahili nouns from the data. The researcher exploits these semantic classes to automatically obtain annotated training data, addressing a key problem facing supervised word sense disambiguation. The semantic and linguistic characteristics of these classes are modeled as Bayesian belief networks, using the Bayesian Modeling Toolbox. The researcher developed a disambiguation solution, which does not make extensive resource requirements. However, rather capitalizes on freely available lexical and computational resources for English as a source of additional disambiguation information. A semantic tagger for Swahili was created by altering the configuration of the Bayesian classifiers. The disambiguation solution was tested on a subset of unambiguous nouns and a manually created gold standard of sixteen ambiguous nouns, using standard performance evaluation metrics. Finally, the researcher achieved an accuracy of 80.4% for sixteen ambiguous words.

3.7 WSD for Turkish Language

A number of researchers have developed WSD for Turkish. Ozdemir [32] is one of them who developed WSD for Turkish using supervised machine learning approach. The researcher uses four example words, which have more than one sense, and the WSD studied for these words. Due to lack of sense-annotated text to be able to do these types of studies, the researcher first collected the data composed of sentences containing the sample words chosen. Four sample words were chosen as ambiguous word and the average number of sense per word was two. A total of 1008 sentences were collected for the sample words. These sentences were selected from different sources in accordance with senses for the target word. Then, the features that distinguish the sense of the word were

identified. According to the researcher, Turkish is a language with an agglutinative structure, there are several features affecting the word sense. Hence, structural features like the suffixes of target words, the type of words that are used with them and the suffixes they get have been examined and based on this correct sense of the sentence was identified.

Supervised learning algorithms were applied to the data by the researcher, and the results obtained using evaluation methods have been interpreted. For sense disambiguation, Naïve Bayes, K-Star, Simple Cart and Bagging algorithms have been used in the test processes performed. The data were evaluated separately as test and train to measure the effects of different evaluation methods, and also evaluations were made with Cross Validation (CV) method. The best results were obtained with Naïve Bayes algorithm.

Furthermore, the features for the words were identified that are believed to be effective in the study. The researcher observed that the most effective feature was the type of the word, and this was followed by the suffix on the target word, the preceding and succeeding words types and their suffixes.

3.8 WSD for Nepali Language

WSD for the Nepali language developed by Shrestha *et al.* [48] is one of them who developed WSD for Nepali language using knowledge-based approach. Nepali language has words, which have many meanings so there is also the problem of WSD. The researchers try to find the impact of NLP resources like morphology analyzer, MRD in ambiguity resolution. The Lesk algorithm was used to solve the WSD problem using a sample Nepali WordNet. The sample Nepali WordNet contains only a few set of Nepali noun and the system is able to disambiguate those nouns only. The system was tested on a small set of data with limited number of nouns. The experiment was performed on 29 words on 40 sentences. The accuracy is about 50% - 70% depending on the data provided and when the same data was tested with manual morphological analysis then accuracy is considerably higher.

The other WSD scheme using Nepali language developed by Roy *et al.* [49].They developed WSD for Nepali language using knowledge-based approach. The surrounding words of the target word in a sentence provide the context for the target word and this context provides consistent clue as regards to the appropriate sense of the target word. They presented some approaches to word sense disambiguation using Nepali WordNet. These approaches are overlap-based approach and conceptual

distance and semantic graph based approach, which fall under knowledge based approach. Conceptual distance and semantic graph distance were used as measures to score their WSD algorithm. The researchers presented that the performance of overlap-based approach is less than the combination of conceptual distance and semantic graph method because overlap based approach suffers from sparse overlap. In the case of nouns, the overlap based approach gives better performance than the overlap based approach with machine readable dictionaries because not only the gloss and examples of the target and context synsets are taken but also the gloss and examples from their hypernyms have been taken into consideration

3.9 Summary

This chapter reviewed different word sense disambiguation that are related to our study either in WSD approach, techniques they used to resolve ambiguity, evaluation of those works or language behavior. The review has shown that knowledge-based and corpus-based approaches successfully used for word sense disambiguation to identify the exact sense of ambiguous word based on context. In order to create a good WSD, all the major areas present in the related work should be considered. To accomplish this, different approaches have been proposed in previous knowledge-based and corpus based works. We also use knowledge-based approach in this thesis to picks the sense whose definition is most similar to the context of the ambiguous word, by means of textual overlap or using ontology based related words. However, corpus-based WSD works dependent on corpus evidence, which is used to train a model using tagged or untagged corpus and hence, they are very costly, we use Amharic WordNet used as source of information and does not require training corpora and is capable of disambiguating arbitrary text.

Finally, we learn from the related work WSD research should incorporate preprocessing, context selection and sense selection techniques for disambiguation. Even though all of them consider words around the ambiguous words such as a collocation and co-occurrence, features some of them have considered as used to identify the context of the target word additional information like POS of words and morphological forms of ambiguous word were used.

.

CHAPTER FOUR: DESIGN OF AMHARIC WORD SENSE DISAMBIGUATION

In this Chapter, the design of knowledge based WSD system for Amharic is discussed. The Chapter describes the design of WSD for Amharic language. It mainly focuses on architecture of Amharic WSD and design and structure of Amharic WordNet. Finally, the detailed description of components on the architecture and their algorithms are also presented.

4.1 System Architecture

The system architecture of the proposed Amharic word sense disambiguation system is composed of four essential components that are preprocessing component, morphological analysis component Amharic WordNet database and disambiguation component. The architecture of the proposed system is as shown in Figure 4.1. The diagram shows the overall functionality of the Amharic Word Sense Disambiguation system. The system takes text as an input and identifies the ambiguous words and its sense form Amharic WordNet. The texts are preprocessed to make suitable for further processing.

Morphological analysis is important for morphologically complex languages like Amharic because it is impossible to store all possible words in WordNet. We used morphological analysis to reduce various forms of a word to a single root word. Morphological analysis produces root word and provides the root word-to-word sense disambiguation component particularly to the ambiguous word identification. The Amharic WordNet database contains Amharic words along with their different meanings, synsets and semantic relations within concepts. This component helps to implement the components of WSD.

Word sense disambiguation component is responsible to identify the ambiguous word and to assign the appropriate sense to ambiguous word. To accomplish this, it incorporates various components such as Ambiguous Word Identification, Context Selection, Sense Selection and Sense retrieval components. The WSD components are integrated with Amharic WordNet. The detailed explanation of the processes is given in the next subsections.

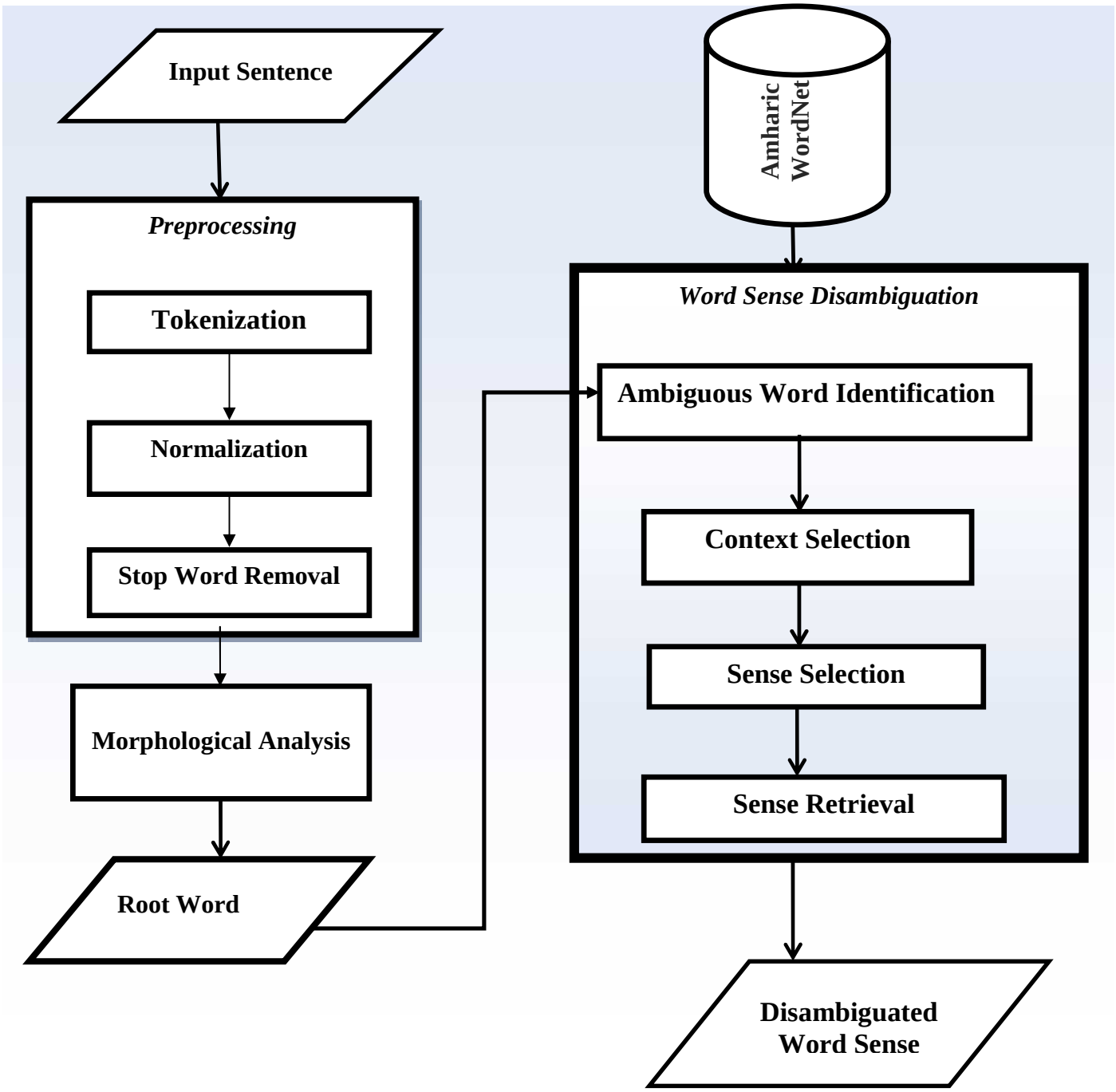

Figure 4.1: Proposed Architecture of Amharic WordNet Based Word Sense Disambiguation (AWNBWSD)

4.2 Preprocessing

Preprocessing tasks are data preparation procedures that should be done before dealing with different text mining techniques. Pre-processing is involved in preparing the input sentence into a format that is suitable for the morphological analysis. The pre-processing stage consists of steps such as tokenization, normalization and stop-word removal. This work makes use of the preprocessing component from Teshome [1], which is adopted from Tessema [46].

4.2.1 Tokenization

The first step in the preprocessing of the input sentence is tokenization, which is also known as lexical analysis. The tokenization takes the input text supplied from a user and tokenizes it into a sequence of tokens, which is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called token and finally it gives the tokens to the next phase. Token is the smallest unit that will be extracted from the input sentence before performing word sense disambiguation. In order to find the boundaries of the sentence the input sentence is segmented in to tokens. For most languages, white spaces and punctuation marks are used as boundary markers. The Amharic language has its own punctuation marks which demarcate words in a stream of characters which includes 'hulet neTb' (:), 'arat neTb '(: :), 'derib sereze' (፧), 'netela serez'(፨) , exclamation mark '!' and question mark'?'. These punctuation marks do not have any relevance to identify the meaning of ambiguous words using WSD. Therefore except 'arat neTb'and 'question mark' which are used to detect the end of the sentence, all other punctuations are detached from words in tokenization process.

4.2.2 Normalization

Normalization is performed on the word tokens that result from text segmentation. In the Amharic language, two types of normalization issues arise [46]. The first one is the identification and replacement of shorter forms of a word that is written using forward slash "/" or period ".". An example is the replacement of "ሥ/ር" by "ሥዎህር". The second normalization issue is the identification and replacement of Amharic alphabets that have the same use and pronunciation, but they have different representations of alphabets. The replacement is made using a representative alphabet from a set of similar alphabets. For example, the word "cough" can have two representations in Amharic: "ሳለ" and

“ሣለ”. These two words differ only by their first characters: “ሰ” and “ሣ”and have similar usages and different forms. They need to be converted to a single representative character such as “ሰ” [46].

4.2.3 Stop Word Removal

Stop words are low information bearing words such as “ነው” or “ና”, typically appearing with high frequency. Stop words may be context dependent. High frequency words have higher variance and effective weight in many methods, causing them to be erroneously selected as features due to sample noise. Overly common words, such as pronouns, prepositions and conjunctions in Amharic, occur so frequently that they cannot give any useful information about the content and be discriminatory for a specific class. These words are called stop words. Stop words are words which are filtered out prior to, or after, processing of natural language data. There is not one definite list of stop words, which all tools use, and such a filter is not always used. Some tools specifically avoid removing them to support phrase search. Like other languages, Amharic has non-content bearing words, which are called stop words. Usually words such as articles (e.g. ‘ያኛው’, ‘ይሄ’), conjunctions (‘ና’, ‘ነገርግን’, ‘ወይም’) and prepositions (e.g. ‘ውስጥ’, ‘ላይ’) do not have a significant discriminating power in the meaning of ambiguous words, we filtered the sense examples with a stop-word list, to ensure only content words are included. In addition to stop words, names of people and places also filtered from the sense examples, as they are not related to the meaning of words. In our approach, “stop words” like ‘ነው’, ‘እስከ’, ‘እንደ’, etc. are discarded from input texts as these words are meaningless to derive the “sense” of the particular sentence. Then, the text containing meaningful words (excluding the stop words) pass through morphological analysis.

4.3 Morphological Analysis

Morphological analysis is the segmentation of words into their component morphemes and the assignment of grammatical morphemes to grammatical categories and lexical morphemes to lexemes. Morphological analysis is important for morphologically complex languages like Amharic because it is practically impossible to store all possible words in a lexicon. This becomes obvious in the context of machine translation to a morphologically simple language such as English, where the correspondence between words in Amharic, Oromo, or Tigrinya and the other language will often be many-to-one [58]. Amharic root words can generate hundreds of lexical forms of different meanings. The Amharic language makes use of prefixing, suffixing and infixing to create inflectional and derivational word

forms as indicated in Appendix II. Removing stop words causes an efficient reduction in the dimensionality of the feature space; however, we also need root word to reduce the dimensionality of the feature space to a reasonable number. Morphological analysis is used for finding the root morphemes of the words and it is a one of the components used to reduce various original forms of a word to a single root or stem of these words. It is necessary to represent different word forms in a single format and to reduce memory usage for storing the words. In morphologically complex language like Amharic, a morphological analysis will lead to significant improvements in WSD systems. In this thesis, we used Hornmorpho morphological analyzer developed by Gasser [64].

4.4 Amharic WordNet

WordNet for Amharic language has a significant impact on search engine, automatic text categorization and Amharic word sense disambiguation [35]. In Amharic WordNet, the words are grouped together according to their similarity of meanings.

Amharic WordNet is a system for bringing together different lexical and semantic relations between the Amharic words. The design of the Amharic WordNet is based on the principle of English WordNet and constructed Amharic WordNet in a way that it can be used as an input to build the Knowledge-Based Amharic Word Sense Disambiguation (KBAWSD) system. WordNet's structure makes it a useful tool for computational linguistics and natural language processing [29].

Structure of Amharic WordNet

The main relation among words in WordNet is synonymy, as between the words ካራ and ቢላዋ or ሀሴት and ደስታ. Synonyms are words that denote the same concept and are interchangeable in many contexts and are grouped into unordered sets (synsets).Two words that can be interchanged in a context are synonymous in that context. For each word there is a synonym set, or synset in the Amharic WordNet, representing one lexical concept. This is done to remove ambiguity in cases where a single word has multiple meanings. Synsets are the basic building blocks of WordNet. Amharic WordNet (AWN) is divided into 4 broad hierarchies based on POS, one each for nouns, verbs, adjectives and adverbs. Every synset is described by a brief definition. This is usable to remove ambiguity in cases where a single word has multiple meanings. For example the word "ሳይ" has multiple meanings such as "ሠዓሊ" and "ስመለከት " and to handle such cases the word is associated to different concepts each representing separate senses. Synsets in WordNet are connected by relations, which can be categorized into two

kinds. The introduction of a “frequently used” or “highly expected” field in the synset structure of WordNets can scale-up the efficiency in determining winner sense of a polysemous word, as these highly related words will enrich the sense bag with more information, thereby enhancing the chances of appropriate overlap. For example consider the word “ኮምፒውተር” is “highly expected” or “frequently used” with the concept of “ማውዝ”,”ኪይቦርድ” and “ሴንትራል ፕሮሰሲንግ ዩኒት”. Thus putting the above list in the synset structure of the most appropriate sense of “ኮምፒውተር” will result in attaining high degrees of overlap with sentences comprising of the word.

WordNet defines the relations between synsets and relations between word senses. A relation between synsets is a semantic relation, and a relation between word senses is a lexical relation. The distinction between lexical relations and semantic relations is somewhat subtle. The difference is that lexical relations are relations between members of two different synsets, however semantic relations are relations between two whole synsets.

Antonymy: Synset A is an antonym of synset B if A and B have senses of opposite meaning. The hypernym of a hypernym of a word is also a hypernym of the word (hypernym chain). If a synset is connected to another synset through is-a-kind-of relation then the relation is Hyponym. Synset A is a hypernym of synset B if B is a “kind of” A. Hyponymy and Troponymy: Hyponymy is the inverse relation of hypernymy for nouns, while troponymy is the inverse relation of hypernymy for verbs. For example “ጽጌረዳ” is a kind of “አበባ” then “ጽጌረዳ” is child or hyponymy and “አበባ” is parent or hypernymy. Meronymy: Sometimes a meronym relation is called part or whole relation. This relation is only applied in nouns. An Amharic word A is a meronym of another Amharic word B if A is-a-part of B. The structure of meronym consists of two synsets IDs as arguments, it is like hypernym structure. Holonymy is the inverse relation of meronymy. For example “ቅጠል” is part of “ዛፍ” here “ዛፍ” is whole or meronym and “ቅጠል” is part or holonym. The AWN database has six basic tables that are Words, POS, Synset, ontology, lexicalRelations and Semantic Relations. The Structure of the Amharic WordNet adopted from English WordNet [13].

Amharic WordNet Database Schema

Word table is used to maintain the unique words of the Amharic language and to maintain the synonymous words in a synset, which are used to describe a concept in a language, by maintaining the principle of coverage and minimality. The purpose of this table is to identify ambiguous words and the sense of the ambiguous word through the field WordName and synsetID. Synset table is used to

maintain the details of a synset (concept in a language). A synset (or concept) has a gloss and synonym word set. The purpose of this table is to maintain concepts which are used to describe a sense of words through the field synsetID and DomainID. Lexical relations table is used to maintain the lexical relations with respect to the synsets. POS table used maintains the part of speeches such as Noun, Adjective, Adverb and Verb of the language.

Semantic Relation is used to maintain the semantic relation like Hyponyms, Hypernym, Holonym, and Meronym between pair of synsets/concepts, which is a IS-A-KIND-OF/ PART-WHOLE type of a semantic relationship between synsets. Ontology tableused to maintain the source from which a concept or synset has been take or belongs like medical concept, marine concept, technology concept, language specific concept, etc.

Figure 4.2 shows the database schema of AWN. Different words in Amharic language are stored in the Word table identified by WordId field. synset are included in the synset table and each synset is identified by SynsetId. POS table hold the name and symbol that represent the main POS of Amharic language such as Noun, Adjective, Adverb and Verb. A word with a particular sense is identified with the help of Word table. SemanticRelation table hold pair of concepts along with the relation they have. For example, the word "ሳለ" has a WordId: w1, w2, w3, w4, w5 in the AWN. Since this word has five senses, the synset table holds gloss of those senses of the word under different synsetId with a gloss "የአንድን ነገር መልክ፡ቅርጽ ወይም አንድን አይነት ሀሳብ በስእል አሳየ፣ ምስሉን በወረቀት ፣በሸራ፣በግድግዳ ወዘተ ቀረፀ፣አሰፈረ፣ ነደፈ" WordId: w1 and SynsetId: s1, the word "ሳለ" with a sense. "ሞረደ ፥አሾለ፥መቁረጫ ጠርዝን አተባ ፥ ስለት አወጣ" WordId: w2 and SynsetId: s2, the word "ሳለ" with a sense. "ይህን ካደረክለኝ ይህን አደርግልሀለው በማለት ለአምላክ፡ለመላእክት" word "ሳለ" with a sense. "ከጉሮሮ አየርን በሀይል ና በተደጋጋሚነት አስወጣ፡ ትክትክ አደረገ፥አህ አህ አለ" WordId: w4 and SynsetId: s4, word "ሳለ" with a sense. "ካለ" WordId: w5 and SynsetId: s5 has ontology : DomainID which is a combination of WordId and SynsetId.

For example, "አበበ ቢላዋ ሳለ።" From this sentence the ambiguous word is ሳለ and ቢላዋ is the context used to identify the sense of the ambiguous word, due to this the synset of "ሳለ" is indicated in the above gloss, and synset of "ቢላዋ" is "ካራ፣ሰንጢ","በአንድ ጎኑ "*ሰለት* ያለው ለስጋ ለሸንኮርት *መቁረጫ፥መክተፊያ* የሚያገለግል የወጥ ቤት እቃ" and "በደረቅ ሳር ጫፍ ላይ የሚገኝ እንደ ምርቅ ያለ እሾህማ ጨጎጎት". In addition to sense overlap we use ontology to identify the context of the ambiguous word "ሳለ" so that, the ontology of the word "ቢላዋ" is የቤት *መገልገያ ቁሳቁስ*, *ሰውሰራሽ*, *መቁረጫ መሳሪያ*, የሚሞረድ or የሚሳል *መሳሪያ*.

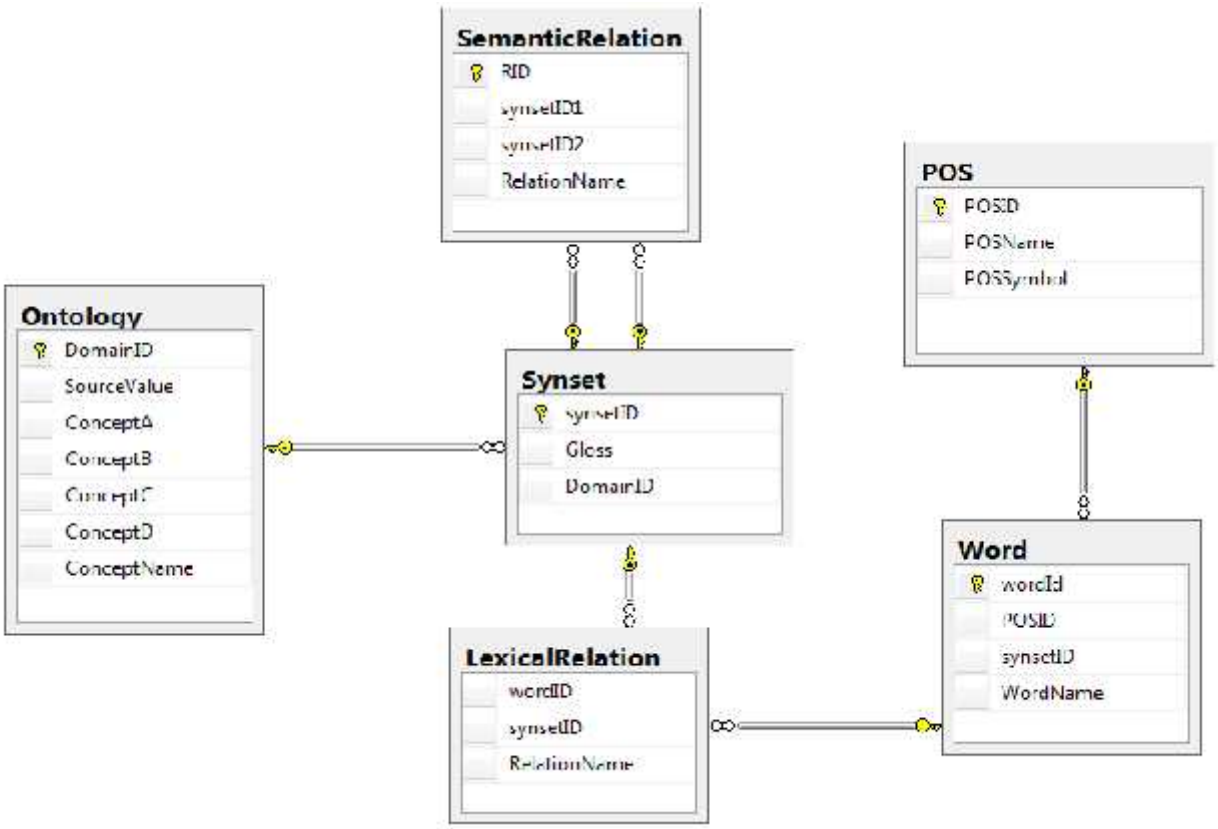

Figure 4.2: Database Schema of Amharic WordNet

4.5 Word Sense Disambiguation (WSD)

WSD is the main component of the Knowledge-Based Amharic Word Sense Disambiguation (KBAWSD) composed of Context Selection, Ambiguous Word Identification, Sense Selection and Sense Retrieval components. We discussed each of them as follows:

4.5.1 Ambiguous Word Identification (AWI) Component

Each word in the input text is disambiguated separately, starting with the first word and working left to right. At each stage, the word being disambiguated is called the target word, and the surrounding words form the context window. Ambiguous word Identification is a component used to identify the ambiguous word from the input sentence based on information provided on Amharic WordNet and Amharic WordNet is used as a knowledge source for this study. Amharic WordNet (AWN) is used to

identify the ambiguous word in this study and contains a list of senses for given words from the input sentence.. The Ambiguous Word Identification is to be checked whether each root word exist in the Amharic WordNet or Not. Words that are found in Amharic WordNet have their own sense on AWN. If words do not exist in AWN the word is discarded. For example, if the following sentence is the input sentence: "ተማሪው ጉንፋን ስለያዘው ሳለ።"

First, the input sentence is preprocessed. After morphological analysis, only four words will be left (i.e. ምእር, ጉንፋን, ይእዝ and ስእል) in the input sentence. Then each root word with respect to it sense in the input sentence is counted in AWN. In our case, the root word **"ሰ-እ-ል"** and its sense exist five times. So that, "ስ-እ-ል" is detected as ambiguous word in the input sentence and "ስ-እ-ል" is the root word for the word "ሳለ".Therefore, "ሳለ" is ambiguous word and their sense is retrieved in AWN based on the context of the sentence. The algorithm is shown in Algorithm 4.1

Input: root words

Index=0

Read words from Amharic WordNet

Add words In Array of Buffers

For word in array of buffer

Count word of Index and its sense in Amharic WordNet

Index++

End For

Count the total number of each Words and its Sense in Amharic WordNet

Compute the largest occurrence of words and its sense

Assign the largest occurrence of words to Ambiguous word

Return ambiguous word

Stop

Output: ambiguous word

Algorithm 4. 1: Ambiguous Word Identification (AWI) Algorithm

4.5.2 Context Selection Component

Context in WSD refers to the words surrounding the ambiguous words, which are used to decide the meaning of the ambiguous word. For example the sentence "ተማሪው ጉንፋን ስለያዘው ሳለ።" after morphological analysis, it will be "ም-እ-ር, ጉንፋን, ይ-እ-ዝ and ስ-እ-ል". From the sentence the ambiguous word is "ሳለ" based on AWI component and the contexts are the surrounding words or neighbor words on the ambiguous word "ሳለ" which are {ም-እ-ር, ጉንፋን, ይ-እ-ዝ} which are identified using the ontology based related words and a sense overlap. Therefore, the correct sense of a word is obtained from the context of the sentence. This component uses the words of the sentence itself as context, including ambiguous words and selects the context that contains ambiguous words from Amharic WordNet (AWN). We use the context selection algorithm to select a subset of the context words to be

used for sense selection. Based on this analysis, the ambiguity of words is known as lexical ambiguity. The algorithm is shown in Algorithm 4.2.

```
Input: root word
Read root word w[i] in the morphological analysis
Open Amharic WordNet
Read root words from Amharic WordNet
For each root word in the sentence
      If   w[i] is ambiguous word
            Find the sense and ontology of words in Amharic WordNet
              Extract the sense of the ambiguous word and related word
      Else
                  Assign empty value to array buffer
End for
If end of Amharic WordNet not reached
        Read root word in Amharic WordNet
  Else
                  Return sense and related words
End if
            Return sense and related words
Stop
Output: sense of ambiguous word + root word
```

Algorithm 4.2: Context Selection Algorithm

4.5.3 Sense Selection Component

The sense selection component finds the number of overlapping of the words from the set of words output by the context selection component with the sense of ambiguous word. The root word definition with the ambiguous words having highest overlapping is selected as the sense of the ambiguous word. For each of the root words in sense selection set, all sense of root word and ambiguous words were identified in the entire Amharic WordNet (AWN) and identify all sense definitions of the words to be disambiguated from AWN and determine the definition overlap for all possible sense combinations. Finally, sense selection chooses senses that lead to highest overlap or identify the related words of the ambiguous word from the ontology table. For example, "ሞረደ ፥አሾለ፥መቁረጫ ጠርዝን አተባ" is the sense of ambiguous word for "ሳለ",for the input sentence "አበበ ቢላዋ ሳለ።" to identify whether the given sense is as a sense of the given ambiguous word. First, the context of ambiguous word is identified by using context selection component and sense overlap. The context of the ambiguous word based on the input

sentence is “ቢላዋ”. Finally, the sense overlap of “ቢላዋ” and the ambiguous word “ሳለ” are selected in Amharic WordNet. Therefore, the highest overlapping is selected as the sense of the ambiguous word or one of the root words is related to the ambiguous word then the sense of ambiguous word is selected based on synset table. The algorithm is adopted from [10] and the modification is shown as italics in Algorithm 4.3.

```
 Input: definition of root word or root word
For every word w[i] in the sentence
      let overlap= 0
      let BEST_SENSE = null
      Open Amharic WordNet
      Assign words in to array of Buffer
      Read root words from Amharic WordNet
    For every sense sense[j] of w[i]
          If word[i] is ambiguous word
                let maxoverlap = 0
            For every other word w[k] in the sentence, k != i
                overlap = overlap + number of words that occur
                In the gloss of both sense[j] and sentence
            End for
      IF overlap> maxoverlap
                 maxoverlap = overlap
                BEST_SENSE = w[i]
      End IF
   End for
      IF maxoverlap > 0
            Extract  BEST_SENSE
      Else
            Output "Could not disambiguate w[i]"
     End If
  End for
 For each root word
           If root word is related word of the ambiguous word
                Extract sense
                Else
                 Assign empty value to array buffer
           End if
End for
   If end of Amharic WordNet Not reached
        Read root word from Amharic WordNet
       Else    sense of ambiguous word
 End if
Stop
```

Algorithm 4. 3: Sense Selection Algorithm

4.5.4 Sense Retrieval Component

The sense retrieval component is responsible for extracting a sense of ambiguous word from Amharic WordNet, and associating the extracted sense with the ambiguous words. For example, if the input sentence is "አበበ ቢላዋ ሳለ", the sense of ambiguous word is "ሞረደ ፥አሾለ፥መቁረጫ ጠርዝን አተባ ፥ ስለት አወጣ" which is selected by sense selection component .This sense is retrieved from Amharic WordNet using sense retrieval component.

4.6 Summary

In this Chapter, the design and implementation of Amharic WordNet Based WSD was presented. It includes Amharic WordNet, Proposed System Architecture. The main components of Amharic WordNet Based WSD are Preprocessing, Morphological analysis, Ambiguous word identification, Context Selection, Sense Selection and Sense Retrieval involves a means to assign the appropriate sense to ambiguous word. Finally, we proposed and implemented three algorithms AWI, CS, and WSS for Word Sense Disambiguation. AWI is used to identify the ambiguous word from the input sentence based on information provided on the AWN. CS is used to context selection algorithm to select a subset of the context words to be used for sense selection. By removing the unimportant words, the computational complexity of the algorithm is reduced. WSS is used to select the possible senses of ambiguous words in the given input sentence.

CHAPTER FIVE: EXPERIMENT

5.1 Introduction

As discussed in the previous chapter, knowledge-based word sense disambiguation was selected for this study. Knowledge based approach; the system uses information on Amharic WordNet to assign senses of words. Hence, we developed Amharic WordNet (AWN) from the scratch. The Amharic WordNet contains 10,000 synsets and 2000 words as shown in sample words and it sense in Appendix I. Python and Java programming language have been used to develop the prototype because Java is dynamic in nature and it can be run in any platform. Root word extraction component of the system was written in Python programming language and SQL server is used to develop Amharic WordNet.

In this Chapter, experiments are conducted to evaluate the performance of the proposed approach. Evaluating the performance of the WSD system is an important part of the research, which discusses the actual work of the research. However, it is a very difficult task since there is no standard rule for WSD evaluation for all languages.

We performed an evaluation of the proposed knowledge-based WSD algorithm using the context window and morphological analyzer effect for word sense disambiguation. A test sentence of 200 random sentences containing the ambiguous words from the knowledge base is created. Some sentences are taken from linguistic experts and some are taken from newspapers. Testing the same set of random sentences over different strengths of knowledge base gives the idea of disambiguation quality when magnitude of knowledge base increases. Out of several senses for each ambiguous word, we considered only two or three senses that are most frequently used.

The system takes the sentence as input and process each sentence one at a time. The system finds the suitable meaning of the ambiguous word or target word in Amharic WordNet. We have followed some set of procedures to conduct the experiment. The test environment, the set of activities defined under the procedures, and findings of the experiment are described in detail in the following sub sections.

5.2 The Prototype

Developing a prototype used to demonstrate the usability of the proposed Amharic word sense disambiguation sense disambiguation using Amharic WordNet is one of the objective of this study. Hence, we develop the prototype of Amharic word sense disambiguation using Amharic WordNet Using Java, Python and SQL Server. Figure 5.1 shows the result of disambiguated sense of the ambiguous word for the given input sentence.

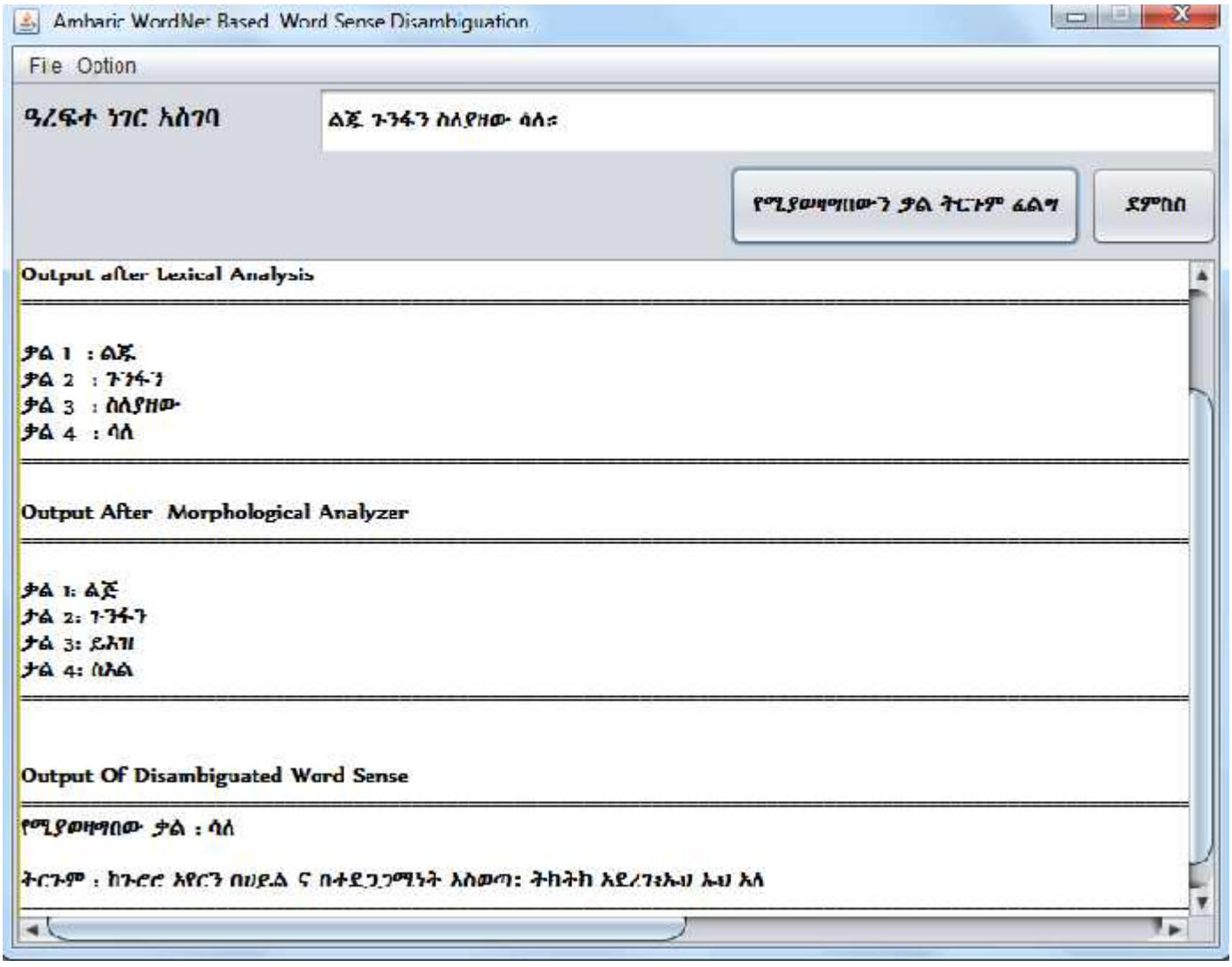

Figure 5.1: Screenshot of the result of Disambiguated sense of the ambiguous word

5.3 Performance Evaluation Criteria

To evaluate the performance for word sense disambiguation there are two measures of performance these measures are precision and recall. Both are calculated from the number of objects relevant to the query determined by some other method, e.g., by manual annotation of given collection and the number of retrieved objects. Based on these numbers we define precision (P) as a fraction of retrieved relevant objects in all retrieved objects and recall (R) as a fraction of retrieved relevant objects in all relevant objects. The performance evaluation criteria were based on the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). TP counts the number of words that are recognized by WSD system and are found in the test data. TN counts the number of words that are not recognized by WSD system and are found in the test data. FP counts the number of words that are wrongly recognized by WSD system; however, they are not in the test. FN counts the number of words that are left unrecognized by WSD system; however, they are in the test data.

The performance evaluation criteria were as follows.

1. Precision is used to measure how much of the information the system returned is correct.

$$\text{Precision} = \frac{\text{No.of correct answers returned}}{\text{No.of total answers returned}}$$

2. Recall is used to measure how much relevant information is the system has correctly disambiguated.

$$\text{Recall} = \frac{\text{No.of correct answers returned}}{\text{No.of total test cases}}$$

3. Accuracy is defined as the proportion of instances that were disambiguated correctly, and is often compared to a baseline. Accuracy puts *equal* weights on relevant and irrelevant documents and it is common that the number of relevant documents is very small compared to the total number of documents.

$$\text{Accuracy} = \frac{TP + FP}{TP + FP + TN + FN}$$

4. The F score is the harmonic mean of recall and precision, a single measure that combines recall and precision. The function ensures that F- score will have values within the interval [0, 1]. The F score is zero when no relevant documents have been retrieved, and it is one when all retrieved

documents are relevant. Furthermore, the harmonic mean F assumes a high value only when both precision and recall are high. Therefore, determination of the maximum value for F can be interpreted as an attempt to find the best possible compromise between recall and precision

$$\text{F-Measure} = \frac{2 * P * R}{P + R}, P + R \neq 0$$

5.4 Test Results

In this study, two experiments were conducted. The first experiment was to measure to what extent morphological analyzer in the Amharic WordNet will affect the accuracy of WSD. The first experiment was conducted on Amharic WordNet with and without morphological analyzer since knowledge-based methods do not use any manually or automatically generated training data, however, use information from an external knowledge source so that the sense inventory for these methods comes from the knowledge source being used were Amharic WordNet. The second experiment is investigating the effect of different context sizes on disambiguation accuracy for Amharic to point out the optimal window size. For the purpose, different dimensionally variant data sets were tested starting from 1-left and 1-right to 5-left and 5- right window sizes.

Experiment I: The effect of Morphological analyzer on the accuracy of the WSD

As we discussed in the previous Chapter, morphological analysis has been found a significant improvement on the performance of word sense disambiguation for morphologically complex languages like Amharic language. This experiment is performed to test whether morphological analyzer improve the performance of Amharic WSD or not. We have carried out a number of evaluations of WSD algorithms using different linguistic resources in various combinations. The linguistic resources that we have used are Amharic WordNet without morphological analyzer and Amharic WordNet with morphological analyzer. A total number of 200 sentences used to evaluate performance of the system.

Amharic WordNet without Morphological Analyzer: The result from this experiment is obtained by comparing the instance with context. The system determines the system tags words that were correctly disambiguated and words that have wrong sense.

Amharic WordNet with Morphological Analyzer: This experiment shows the importance of the morphological analyzer in WSD. In the experiment, ontology based related word and sense overlap is used to identify the context of the ambiguous word in the instance and ontology based related word is created manually due to lack of linguistic resources. We apply morphological analyzer in the WSD the accuracy of the system increases. The system determines the system tags words that were correctly disambiguated and words that have wrong sense.

Table 5.1: Performance of the KBAWSD system with and without Morphological Analyzer

	Amharic WordNet with Morphological Analyzer	**Amharic WordNet without Morphological Analyzer**
Recall	90.3%	74.5%
Precision	84.8%	66.67%,
F-Measures	87.5%	70.37%
Accuracy	80%	57.5%,

As shown in Table 5.1, we can say that for all word sense disambiguation task, Amharic WordNet with morphological analyzer improved the accuracy of knowledge based word sense disambiguation. As we can see Amharic WordNet with morphological analyzer the context of word is found by determine the meaning of a word by using ontology based related words and the overlap of the sense of target word to each words, we achieved an accuracy of 80%. Morphological analyzer reduces various forms of word into their common root or stem word. This minimizes the consideration of the variants of a word as different word by WSD. If morphological analyzer is done, the variant of a word is taken as the same pattern, which will improve the accuracy of the knowledge based word sense disambiguation algorithms. For example, before morphological analyzer, surrounding words *"ሳለ","መሳል","አሳሳሉ"*, and *"ተሳለ"* would be assumed as different however, basically they are the variants of the same word " ስእል". After Morphological analyzer applied, these words are taken as the same pattern. Therefore, in subsequent experiments the Morphological analyzer was used as it enhanced the performance of the WSD systems. Recall and precision denote completeness of retrieval and purity of retrieval respectively. Unfortunately, it was observed that with the increase of recall, the precision usually

decreases. This means that when it is necessary to retrieve more relevant objects, a higher percentage of irrelevant objects will be probably retrieved.

Experiment II: Determining optimal context window

In other languages, an optimal context window size, which refers to the number of surrounding contexts for sense disambiguation, is obtained through researches. For example in English, a standard two-two word window (2-2) on either side of the ambiguous word is found to be enough for disambiguation [52]. Solomon Mekonnen [25] reported that Window size of three-three words(3-3) is considered to be effective using supervised learning method with achieved accuracy within the range of 70 to 83% on five ambiguous words (*መሳሳት, መጥራት, ቀረፀ, አጠና* and *መሳል*).

Solomon Assemu[26] tested the optimal window size using unsupervised learning methods and the author advised that window size of 3-3 or 2-2 is enough for disambiguation depending on the algorithms used. Window size of 3-3 was effective for Simple K means and EM clustering algorithms achieved accuracy ranged from 65.1 to 76.9 % where as windows 2-2 was effective for agglomerative SL and CL clustering algorithms achieved accuracy range from 51.9 to 71.1% on the same five ambiguous words (*መሳሳት, መጥራት, ቀረፀ, አጠና* and *መሳል*).

Getahun Wassie [27] tested the optimal window size using semi-supervised learning methods and the author advised that window size of 3-3 or 2-2 is enough for disambiguation depending on the algorithms used. Window size of 3-3 was effective for bootstrapping algorithms (adabostM1, ADtree, and bagging) with achieved accuracy of 84.90%, 81.25% and 88.45% respectively where as windows 2-2 was effective for Naïve Bayes and SMO algorithms achieved an accuracy of 67.01% and 87.89% respectively on five ambiguous words (*አጠና, ደረሰ , ተነሳ , ፈለ* and በላ). However, this has not been tested for Amharic words using Knowledge based approaches. For this study, an experiment is carried out to test 1-1 window up to 5-5 window on both side of the target word for some ambiguous words. In the Previous, research [25, 26, 27] does no evaluate the precision and recall of the window size for Amharic language.

Table 5.2: Summary of experiment in different window sizes

Window size	**1-1**	**2-2**	**3-3**	**4-4**	**5-5**
Precision	**78.5%**	**77.6%**	**66.5 %**	**64.7%**	**65.7%**
Recall	**70.5%**	**84.7%**	**87.4%**	**89.8%**	**90.4%**
F1-Measure	**74.28%**	**80.99%**	**75.53%**	**75.23%**	**78.37%**
Accuracy	**71.5 %**	**86.5 %**	**80.4 %**	**78.2%**	**79.9 %**

As shown in the Table 5.2, for the knowledge based Amharic word sense disambiguation the maximum accuracy of, precision and recall achieved 86.5 %, 84.7% and 77.6% on two-two word window size respectively. So that to the determining the optimal window size for all word sense disambiguation is that, a small window size tends to result in high precision, however low recall. In terms of recall, if there are more words in the context, the chance of finding related word with ambiguous word at least one of them is higher and hence increased window size would lead to a higher recall. We observed especially good results for window size 2. This is because for window size=2, we can assign the sense to the first instance in a sentence. For example, in the sentence "አበበ ቢላዋ ሳለ", if window size=2 and the target is "ሳለ", since there is no word in the right context, we assigns the sense to the target word. This induces enormous sense of resulting in good precision for window size = 2. From the above sentence, "ቢላዋ" is in the context while disambiguating the target word "ሳለ". We define a window size around the target polysemous word and calculate the number of words in that window that overlap with each sense of the target polysemous word. The Performance of knowledge Based methods is high and also they do not face the challenge of new knowledge acquisition since there is no training data required.

To compare the test results with the Previous researches [1, 25, 26] used stemmer to improve the performance of WSD, however when we compare to morphological analyzer with the stemmer. The stemmer does not improve the performance of WSD, which is better than morphological analyzer.

When we apply stemmer and morphological analyzer for the following sentence morphological analyzer works very well to stemmer. For Instance the "የበሽታው የተለመደ ምልክት ያለማቋረጥ መሳል ከደረት ህመም ጋር የተያያዘ ነው።" after applying stemming and stop word removal the sentence will be "በሽታ ተለመደ ምልክት ማቋረጥ መሳል ደረት ህመም ያዝ". From the given sentence, the stemmer removes affixes that are usually used for changing the tense, number, gender and case of a word. Further, more in the case of removing suffixes with vowels, the last character of the word after removal of the suffix is changed to sades (the six order of character a character). That the variant of ተማሪ, ተማሪው, ተማሪዎች,ተማሪዎቹ and ተማሪዋ are change in to their stem word ተማር. However, when we apply stop word removal and morphological analyzer the sentence will be **"በሽታ ልምድ ምልክት ቁርጥ ስእል ደረት ህመም ይእዝ"** from here the morphological analyzer changes each word in to the root word. The comparison of the test result on the previous research [25, 26] on Amharic WSD, we have evaluated the effect of Amharic WordNet with morphological analyzer. We achieved an accuracy of 80%, but for the previous researches [25, 26] they evaluate the effect of stemmer on Amharic WSD, they achieved an accuracy of 40% to 70% and 61% to 75% respectively. However, the researcher does not evaluate the precision and recall values on the effect of WSD.

5.5 Summary

In this Chapter the experimentation procedure together with presentation and discussion of two experiments are covered. The prototype of the system was presented. We have implemented a Python and Java programming language was used to develop the prototype. In the first experiment, it has been shown that morphological analyzer significantly improved the accuracy of WSD. The second experiment was also conducted to determine optimal window size for the Knowledge-Based Amharic Word Sense Disambiguation (KBAWSD) and two-two window size has been found as most favorable window size. Using two-two window, the final accuracy of the algorithm has achieved 86.5%, which is a very encouraging in knowledge-based WSD.

CHAPTER SIX: CONCLUSION AND RECOMMENDATION

In this thesis, we developed a WSD system, called Amharic WNBWSD, for Amharic language. Before developing Amharic WordNet Based Amharic Word Sense Disambiguation (AWNBWSD), we have studied some of the WSD systems developed for Amharic language and others. WSD approaches have also been studied to select an approach that can give the best performance given the constraints of Amharic language. The nature, structure and pattern of Amharic language has also been studied before developing the system.

6.1 Conclusion

This research work is the first attempt to develop a word sense disambiguation system for Amharic language using Amharic WordNet. Since there is no linguistic resources prepared i.e. WordNet, Thesaurus, Machine Readable Dictionaries and others for Amharic Language, which is important for WSD purpose, we prepared Amharic WordNet manually for this study. During this preparation, we have selected 2000 words including ambiguous words. Based on these ambiguous words, we extracted Amharic sentences from newspaper as test set by the help of language experts.

The architecture of the system includes four main components: preprocessing, morphological analysis, Amharic WordNet database and word sense disambiguation phase. The system takes Amharic sentence as an input. In the preprocessing stage, it segments the input sentence by using tokenization and removes all words that can be stop words from the input sentence. By considering the morphological variants of the language, morphological analyzer is also applied after the preprocessing component gives the input sentence, which is used to reduce various forms of word to common root word. After gathering information in the morphological analyzer step, the system uses the remaining words in the input sentence as context, which used ontology based related words and overlap features in this thesis to identify the sense of ambiguous words. disambiguation component is used to identify the ambiguous word and its sense based on information found in the Amharic WordNet. Then, the system identify the context of ambiguous word using Ontology based related words and the overlap of the sense of target word to each words and decides the most appropriate sense for a given ambiguous word in the input sentence.

We have conducted two experiments; the first one is evaluating the effect of Amharic WordNet with and without morphological analyzer and the second one is determining an optimal windows size for Amharic WSD. For Amharic WordNet with morphological analyzer and Amharic WordNet without morphological analyzer we have achieved an accuracy of 57.5% and 80% respectively. We can conclude that Amharic WordNet with morphological analyzer can enhance the accuracy of Amharic word sense disambiguation as Amharic is a morphologically complex Language. For the second experiment, for Amharic there is no standard optimal context window size which refers to the number of surrounding words that is sufficient for extracting useful disambiguation. Based on our experiment we have found that two-word window on each side of the ambiguous word is enough for Amharic WSD.

6.2 Recommendations

The underlying hypothesis of the technique used in this thesis is that Ontology based related words and sense overlap tells us about the intended meaning of a word. Thus for an accurate disambiguation, selecting the appropriate context is crucial. Word sense disambiguation researches require variety of linguistic resources like thesaurus, WordNet and Machine Readable Dictionaries in which we faced a significant challenge as Amharic lacks those resources. The other challenge we faced was lack of word searching software to collect texts with ambiguity words to evaluate knowledge-based Amharic WSD for the study. Therefore, we forward the following recommendations for WSD for Amharic texts:

- Researches in WSD for other language use linguistic resources like thesaurus and machine readable dictionaries. For Amharic those resources are not yet been developed. We recommend those resources to be included in the future work.
- We have used manually developed Amharic WordNet for Synsets, Word-sysnset pairs and relationships. No full-fledged Amharic WordNet is available and constructing it manually is tedious. Constructing such lexical knowledge base for WSD is important and time efficient.
- We have used manually developed Ontology for some Amharic words, which is used to identify the context of sentences. We recommend the development of this resource used to enhance Amharic word sense disambiguation.
- In addition to knowledge based and corpus-based approach, there is also hybrid approach, which has been used in WSD systems for other languages. This approaches need to be investigated for Amharic as well.

REFERENCES

[1] Teshome Kassie (2008). Word Sense Disambiguation For Amharic Text Retrieval: A Case Study for Legal Documents, *Unpublished Master's Thesis*, Department of Computer Science, Addis Ababa University, Addis Ababa, Ethiopia.

[2] Rada Mihalcea, E. Agirre and P. Edmonds Eds (2007). *Word Sense Disambiguation Algorithms and Applications Text, Speech and Language Technology*, Springer, VOLUME 33, Université de Provence and CNRS, France.

[3] Shruti Ranjan Satapathy (2013). Word Sense Disambiguation, *Unpublished Master's Thesis*, Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, India.

[4] Jason Michelizzi (2005). A Semantic Relatedness Applied to All Words Sense Disambiguation, *Unpublished Master's Thesis*, Department of Computer Science, University Of Minnesota, Daluth, USA.

[5] Ellen M. Voorhees (1993). Using WordNet to Disambiguate Word Senses For Text Retrieval, *Siemens Corporate Research*, Inc. 755 College Road East Princeton, NJ 08540

[6] Kerem Celik (2012). A Comprehensive Analysis of Using WordNet, Part-Of-Speech Tagging, And Word Sense Disambiguation in Text Categorization". *Unpublished Master's Thesis*, Department of Computer Science, Bogazici University, Turkey.

[7] Yarowsky D. (1995). Unsupervised word sense disambiguation rivaling supervised methods, in *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Cambridge, M.A.

[8] Roberto N. (2009).Word Sense Disambiguation: A Survey. *ACM Computing Surveys*, Vol 41 No. 2, Universit `a di Roma La Sapienza, Italy.

[9] የኢትዮጵያ ቋንቋዎች ጥናት ና ምርምር ማእከል(1993)። *አማርኛ መዝበ ቃላት*፣አዲስ አበባ፣አርትስቲክ ማተሚያ ቤት, ኢትዮጵያ.

[10] Lesk M. (1986). Automatic sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone, In *Proceedings of the 5th SIGDOC*. pp.24–26.

[11] Rohana Sharma (2008), Word Sense Disambiguation for Hindi Language, *Unpublished Master's Thesis*, Thapar University, Patiala, India.

[12] X. Zhou and H. Han. (2005). Survey of Word Sense Disambiguation Approaches, *Proceedings of the 18th International FLAIRS Conference*.

[13] Fellbaum Christiane (1998). *WordNet: an Electronic Lexical Database*, Cambridge, MIT press, USA.

[14] Kleinberg, M. Jon (1998). Authoritative sources in a hyperlink environment . *Proc. of ACM-SIAM Symposium on Discrete Algorithms*.

[15] Agirre, E. and Martinez, D. (2001). Learning class-to-class selectional preferences, In *Proceedings of the 5 Conference on Computational Natural Language Learning*, pp. 15–22

[16] Mark Stevenson (2003). Word Sense Disambiguation: The Case for Combining Knowledge Sources, *CSLI Publications*, Stanford, CA.

[17] Agirre, E., Stevenson, M. (2006). *Knowledge sources for WSD, In Word Sense Disambiguation Algorithms and Applications*, E. Agirre and P. Edmonds Eds. Springer, New York, NY, pp. 217–251.

[18] Agirre E., Rigau G (1996). Word sense disambiguation using conceptual density. *Proc. Of COLING*

[19] Richardson R. and Smeaton A. (1995). Using WordNet in a knowledge-based approach to information retrieval. *Proc. of the BCS-IRSG Colloquium, Crewe*.

[20] Yarowsky, D. (1992): Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. *Proc. of the 14th International Conference on Computational Linguistics (COLING-92)*, Nantes, France, pp. 454-460.

[21] Brown P., Pietra S., Pietra V. and Mercer R. (1991). Word sense disambiguation using statistical methods. *Proc. of the 29th Meeting of the Association for Computational Linguistics (ACL- 91)*, Berkley, C.A. pp 264-270

[22] Luk, A (1995). Statistical sense disambiguation with relatively small corpora using dictionary definitions. *Proc. of the 33rd Meetings of the Association for Computational Linguistics (ACL-95)*, Cambridge, M.A. pp 181-188.

[23] W. Faris and K.H. Cheng (2013). A Knowledge-Based Approach to Word Sense Disambiguation Computer Science Department, University of Houston, Houston, Texas, USA.

[24] Hee-Cheol Seo, Hoojung Chung, Hae-Chang Rim, Sung Hyon Myaeng and Soo-Hong Kim (2004), unsupervised word sense disambiguation using WordNet relatives, Department of Computer Science and Engineering, Korea University, Published.

[25] Solomon Mekonen (2010). Word Sense Disambiguation for Amharic Text: A Machine Learning Approach, *Unpublished Master's Thesis*, Department of Information Science, Addis Ababa University, Addis Ababa, Ethiopia.

[26] Solomon Assemu (2011). Unsupervised Machine Learning Approach for Word Sense Disambiguation to Amharic Words, *Unpublished Master's Thesis*, Department of Information Science, Addis Ababa University, Addis Ababa, Ethiopia.

[27] Getahun Wassie (2012). Semi-supervised Machine Learning Approach for Word Sense Disambiguation to Amharic Words, *Unpublished Master's Thesis*, Department of Information Science, Addis Ababa University, Addis Ababa, Ethiopia.

[28] Tesfa Kebede (2013). Word Sense Disambiguation for Afaan Oromo Language, *Unpublished Master's Thesis*, Department of Computer Science, Addis Ababa University, Addis Ababa, Ethiopia.

[29] WordNet:A lexical database for English Language; Available at: http://WordNet.princeton.edu/index.html. Accessed on 10 October, 2013

[30] Prity Bala (2013). Knowledge Based Approach for Word Sense Disambiguation using Hindi WordNet, In *The International Journal Of Engineering And Science (IJES)*, Apaji Institute, Banasthali Vidyapith Newai, Rajesthan, India, pp. 36-41.

[31] Wanjiku NG'ANG'A (2005).Word Sense Disambiguation of Swahili: Extending Swahili Language Technology with Machine Learning, Faculty of Arts of the University of Helsinki, in *auditorium XII*, *Unioninkatu* 34, on the 18th of November.

[32] Vildan Ozdemir (2009). Word Sense Disambiguation for Turkish Lexical sample, *Unpublished Master's Thesis*, Department of Computer Engineering, Fatih University, Istanbul, Turkey.

[33] Atelach Alemu Argaw and Lars Asker (2007). An Amharic Stemmer: Reducing Words to their Citation Forms, Department of Computer and Systems Sciences, Stockholm University/KTH, Sweden.

[34] Maurice van Keulen and Mena B. Habib (2013). Handling uncertainty in information extraction, University of Twente, Enschede, The Netherlands.

[35] Tessema Mindaye, Meron Sahlemariam and Teshome Kassie (2010). The Need for Amharic WordNet, *the 5th International Conference of the Global WordNet Association*, Mumbai, India.

[36] Daniel Gochel Agonafer (2003). An Integrated Approach to Automatic Complex Sentence Parsing for Amharic Text, *Unpublished Master's Thesis*, Department Of Information Science, Addis Ababa University, Addis Ababa, Ethiopia.

[37] Dawit Bekele (2003),The Development and Dissemination of Ethiopic Standards and Software Localization for Ethiopia, The ICT Capacity Building Programme of the Capacity Building Ministry of the FDRE and United Nations Economic Commission for Africa ,Addis Ababa, Ethiopia.

[38] Solomon Teferra Abate and Wolfgang Menzel (2005). Syllable-Based Speech Recognition for Amharic, University of Hamburg, Department of Informatik. Vogt-Kölln-Strasse, 30, D-22527 Hamburg, Germany.

[39] Saba Amsalu Teserra (2007). Bilingual Word and Chunk Alignment: A Hybrid System for Amharic and English, *Unpublished Master's Thesis*, ,Universitat Bielefeld, UK.

[40] Sebsibe H/Mariam, S P Kishore, Alan W Black, Rohit Kumar, and Rajeev Sangal (2004). Unit Selection Voice For Amharic Using Festvox , Carnegie Mellon University, *Institute for Software Research International*, Carnegie Mellon University, Hyderabad, India.

[41] R. M.Voigt (1987). The classification of central Semitic, *Journal of Semitic Studies*, No. 32, pp.1-2.

[42] Lars Asker, Atelach Alemu Argaw, Björn Gambäck and Magnus Sahlgren (2006). Applying Machine Learning to Amharic Text Classification, Stockholm University and Swedish Institute of Computer Science.

[43] Tayebeh Mosavi Miangah and Ali Delavar Khalafi (2005). Word Sense Disambiguation Using Target Language Corpus in a Machine Translation System, *literary and linguistic Computing,* Vol. 20, No. 2, Shahre Kord University, Iran.

[44] Getahun A. (2001). *Towards the Analysis of Ambiguity in Amharic*, JES Vol XXXIV No 2.

[45] Liddy, E. D. *Natural Language Processing*, Library and Information Science, 2nd Ed. Marcel Decker.

[46] Tessema Mindaye (2007). Design and Implementation of Amharic Search Engine, *Unpublished Master's Thesis*, Addis Ababa University, Addis Ababa, Ethiopia.

[47] Gerard Escudero Bakx (2006). Machine Learning Techniques For Word Sense Disambiguation *Ph.D. thesis*, Department of Computer Science, Universitat Politµecnica de Catalunya.

[48] Niraj Shrestha, Patrick A.V. Hall and Sanat K. Bista (2008). Resources for Nepali Word Sense Disambiguation, Department of Computer Science & Engineering, Kathmandu University, Dhulikel, Nepal.

[49] Arindam Roy, Sunita Sarkar and Bipul Syam Purkayastha (2014). Knowledge Based Approaches to Nepali Word Sense Disambiguation, Department of Computer Science, Assam University, Silchar.

[50] Mukti Desai (2013). Word Sense Disambiguation, Department of Computer Engineering Dwarkadas J. Sanghvi College of Engineering, Mumbai University, India.

[51] Girma Getahun (2007), በአማርኛ ሥርዓተ-ጽሕፈት ውስጥ የድምፀ-ሞክሼ ሆሄያት አጠቃቀም ማስታወሻ. Retrieved on 15 November, 2013 from: http://www.nlp.amharic.org/resources/lexical/word-lists/homonyms

[52] Kaplan A.(1955).*An experimental study of ambiguity and context*, Machine Translation, Vol 2 No.2.

[53] Banerjee S., Pedersen T. (2002). Extended gloss overlaps as a measure of semantic relatedness, *Unpublished Master's Thesis*, Department of Computer Science, University of Minnesota, Duluth, USA.

[54] Amanda Wimsatt and Rachel Wynn (2011). Amharic Language and Culture Manual ,National Language of Ethiopia , Texas State University .

[55] ባዬ ይማም (2000)፤ *የአማርኛ ሰዋሰው*.

[56] Gale William, Ken Church & David Yarowsky (1992a). Estimating upper and lower bounds on the performance of word-sense disambiguation programs. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Newark, U.S.A., 249–256.

[57] Gale, William, Ken Church & David Yarowsky (1992b). One sense per discourse. *Proceedings of the DARPA Speech and Natural Language Workshop*, New York, U.S.A, 233–237.

[58] Gasser M. (2012). HornMorpho: A System for morphological processing of Amharic, Oromo, and Tigrinya. *Conference on Human Language Technology for Development*, Alexandria, Egypt.

[59] Esha Palta (2010). Word Sense Disambiguation, Indian Institute of Technology, Powai, Mumbai, India.

APPENDICES

Appendix I. The senses of Sample Amharic words

Words	Word Senses
ሳለ	1. የአንድን ነገር መልክ፡ቅርጽ ወይም አንድን አይነት ሀሳብ በስእል አሳየ፣ ምስሉን በወረቀት ፣በሸራ፣በግድግዳ ወዘተ ቀረፀ፣አሰፈረ፣ነደፈ 2. ሞረድ ፥አሾለ፥መቁረጫ ጠርዝን አተባ ፥ ስለት አወጣ 3. ይህን ካደረከለኝ ይህን አደርግልሀለው በማለት ለአምላክ፡ለመላእክት የሚቀርብ ለመና፡ብፅኣት 4. ከጉሮሮ አየርን በሀይል ና በተደጋጋሚነት አስወጣ፡ ትክትክ አደረገ፥ኡህ ኡህ አለ 5. ካለ
መሳሳት	1. መቅጠን፣ ዘርዛራ ወይም ስስ መሆን 2. በስስት ወይም በጥንቃቄ 3. ትክክል አለመኾን፤ አቅጣጫን አለማወቅ
ቁር	1. ሄልሜታ፤ቆብ፤ኮፍያ፤ባርኔጣ፤ኻሽ 2. ቀዝቃዛ፤ብርድ፤ውርጭ
ሀቅ	1. እዉነት፣ትክክል የሆነ፣የተረጋገጠ ነገር 2. የራስ ድርኻ፣አንጡር ንብረት 3. ሶስት ክንድ ርዝመት ያለው የብትን ልብስ የሸማ መጠን ፥ ልክ
ሳይ	1. ሰአሊ 2. ስመለከት
ሰየመ	1. ስም አወጣ 2. ፈጨ፤ ሞላ፤ አዘጋጀ 3. ሾመ፤ ማዕረግ ሰጠ
ህብር	1. ቀለም፤ መልክ 2. ትብብር 3. ጥምር ፍቺ ያለው ቃል (ሐረግ) 4. በጠባይ ወይም በግብር ተመሳሳይነት ያለው
ሀይል	1. ብርታት፥ጥንካሬ፥ጉልበት፥አቅም 2. አንድን ተግባር የሚፈጽም ቡድን 3. ዘዴን ያልተከተለ አስገዳጅ ግፊት
ለጋ	1. ቀንበጥ፥ለምለም 2. ያልቆየ ና ያልበሰለ ወይም ያልኮመጠጠ ቅቤ 3. ልጅ እግር ፥ወጣት፥ሶታ 4. ሩርን ወይም ጥንግርን በእጅ ወርወር አድርጎ መሬት ሳታርፍ በዱላ መታ፥ቀላ 5. ኳስን በረጅም መታ(የእግር ኳስ)

ለብታ	1. መጠነኛ ሙቀት፥ለስስታ (የውሃ) 2. ሞቅታ፥መጠነኛ ስካር
ሰላም	1. ረብሻ፥ሁከት፥ጦርነት የሌለበት፥የተረጋጋ ሁኔታ 2. ደህንነት፥ጤንነት፥የአእምሮ ረፍት 3. በቁም ዜማ ና በማህሌት መጨረሻ ላይ የሚባል ዜማ
ልማት	1. ኢኮኖሚያዊ ና ማህበራዊ እድገት ፥ ብልፅግና ፥ግንባታ 2. ጥሩ ፥መልካም ተግባር
ልማድ	1. ባህርይ፥አመል 2. ዘወትር፥ በተደጋጋሚ በመስራቱ ወይም በመደረጉ እንደ ደንብ፥ እንደ ህግ የተወሰደ 3. አንድ ማህበረሰብ የሚከተለው፥ከትውልድ ወደ ትውልድ እየ ተላለፈ የመጣ ህግ፥ ስርአት፥ባህል 4. ዘወትር በተደጋጋሚ የሚደረግ ፥የሚሰራ 5. ሱስ
ሰራዊት	1. የጦር ኀይል፤ ጭፍራ 2. ዋና አግዳሚ፤ ርብራብ
መአት	1. ከፍተኛ እልቂት፣ችግር፣ቁጣ 2. እጅግ በጣም ብዙ፥አያሌ
ሃብት	1. ንብረት፣ጥሪት 2. እድል፣እጣ
እምነት	1. ሃይማኖት፥አምልኮ 2. እርግጠኝነት 3. ሃላፊነት፥ታማኝነት
ሰንደል	1. ሲያጨሱት ጥሩ ሽታ ያለው ና በቀጭን ዘንግ መልክ የሚዘጋጅ የቤት ማጠኛ እንጨት፥ ነድ 2. የነጠላ ጫማ አይነት
አሞራ	1. እንደ ጥንብ አንሳ፥የሎስ፥ጋጋኖ ያሉ ትላልቅ አእዋፍ 2. ፈጣን፥ቀልጣፋ
ዋና	1. አለቃ፥ሃላፊ፥ባለቤት፥ቀዳሚ ከሁሉም በላይ የሆነ፥የበለጠ 2. በጉልህ የሚታይ፥የሚታወቅ አስፈላጊ የሆነ 3. መነሻ፤መነገጃ፤ገንዘብ 4. በእጅ በእግር እየተቀዘፈ በውሃ በባህር ላይ የሚደረግ ስፖርት
ብልህ	ነ. ብነግርኽ፤ ባሳውቅኽ 2. አስተዋይ፤ ዐዋቂ፤ ጮሌ
አሳማ	ነ. ሐሜት እንዲነገር አደረገ 2. ለሥጋ የሚረባ የቤት እንስሳ
አዞ	ነ. አዝዞ፣ ተፈጻሚ ቃልን ተናግሮ 2. ኮምጣጣ እና ቅመማም እርጎ 3. የወንዝ (የሐይቅ) አውሬ

እርሳስ	1. ክብደት ና ለስላሳነት ያለው፥በቀላሉ የሚቀልጥ የማእድን አይነት 2. የጥይት አረር 3. ለመፃፊያ፥ለመሳያ የሚያገለግል ከእርሳስ ማእድን የሚሰራ የፅህፈት መሳሪያ
ዛፍ	1. ግንድ ና ቅርንጫፍ ያሉት፥መጠኑ ትልቅ ና ትንሽ የሆነ፥ለረጅም ጊዜ ሊቆይ የሚችል የእፅዋት አይነት 2. ረጅም (ለሰው)
ግዝት	1. ክልክል፣ሃራም፣እርም የሆነ ነገር
ሀሴት	1. ደስታ
ሃኬት	1. ተንኮል፥ክፋት
ሆድ	1. እንደ ጨጓራ፥ጉበት፥ኩላሊት፥አንጀት የመሳሰሉ የሚገኙበት የሰውነት ክፍል
ፊደል	1. ንግግርን በጽሑፍ ለማስፈር የሚያስችልና ድምጽን ወክሎ የሚቆም ምልክት (ለምሳሌ ሀ፣ ለ …)
ሆሄ	1. ከግእዝ እስከ ሳብዕ (ለምሳሌ ከሀ-ሆ ወይም ከሀ-ፐ) ያለው የፊደላት ስምና መልክ ወይም ቅርፅ

Appendix II: Lists of Sample Affixes removed from the words

የ	ስለ	ለ	ከ
እና	እንደ	ና	ዎች
ዎቹ	ኛ	አች	ቷ
እስከ	ው	ዋ	በ

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 source of materials used for the thesis have been duly acknowledged.

Declared by:

Name: Segid Hassen

Signature: _____

Date: _____

Confirmed by advisor:

Name: Dr.Yaregal Assabie

Signature: _____

Date: _____

Place and date of submission: Addis Ababa, March 2015.