2020-03-16

Word Sense Disambiguation for Amharic Sentences using WordNet Hierarchy

Mulugeta, Mieraf

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF RESEARCH AND POSTGRADUATE STUDIES

FACULTY OF COMPUTING

Word Sense Disambiguation for Amharic Sentences using WordNet Hierarchy

MSc. Thesis

By

Mieraf Mulugeta

Program: Computer Science

Bahir Dar, Ethiopia

Word Sense Disambiguation for Amharic Sentences using WordNet Hierarchy

Mieraf Mulugeta Kibret

A thesis submitted to the school of Research and Graduate Studies of Bahir Dar Institute of Technology, BDU in partial fulfillment of the requirements for the degree of Master of Science in the Computer Science in the Faculty of Computing.

Advisor Name: Million Meshesha (PhD)

Bahir Dar, Ethiopia

July 31, 2019

DECLARATION

I, the undersigned, declare that the thesis comprises my own work. In compliance with internationally accepted practices, I have acknowledged and refereed all materials used in this work. I understand that non-adherence to the principles of academic honesty and integrity, misrepresentation/ fabrication of any idea/data/fact/source will constitute sufficient ground for disciplinary action by the University and can also evoke penal action from the sources which have not been properly cited or acknowledged.

Name of the student:MierafMulugetaKibret_______ Signature _____________

Date of submission: ________________

Place: Bahir Dar

This thesis has been submitted for examination with my approval as a universityadvisor.

Advisor Name: ____Million Meshesha (PhD)

Advisor’s Signature: _million___ 19/07/2019

Bahir Dar Institute of Technology-

School of Research and Graduate Studies

Faculty of Computing

THESIS APPROVAL SHEET

Student:

Mieraf Mulugeta Kibret		22/07/2019
Name	Signature	Date

The following graduate faculty members certify that this student has successfully presented the necessary written final thesis and oral presentation for partial fulfillment of the thesis requirements for the Degree of Master of Science in Computer science.

Approved By:

Advisor:		
Million Meshesha (PhD)		19/07/209
Name	Signature	Date
External Examiner:		
Yaregal Assabie (PhD)		22/07/2019
Name	Signature	Date
Internal Examiner:		
Tesfa Tegegne (PhD)		22/07/2019
Name	Signature	Date
Chair Holder:		
Abrham Debasu		23/07/2019
Name	Signature	Date
Faculty Dean:		
Belete Biazen		22/07/2019
Name	Signature	Date

ACKNOWLEDGMENTS

First and foremost, I would like to thank the God for His support and being with me in all directions of my life.

Next, I would like to thank my advisor Million Meshesha (PhD) for helping me work on this research and for his continued support during the time of work. I am very proud of working under his supervision. Thank you for your unreserved guidance, support, encouragement and constructive suggestion.

I would like to thank to Marew Alemu (PhD) and Melese Zegeye (PhD), linguists in Bahir Dar University for their support and professional contribution to the data preparation and in all of the linguistic part of my study. I also would like to thank my mother W/ro Nigist Bizuayehu and my father Professor Mulugeta Kibret for their support and unconditional love in my academic journey and in all aspects of my life. Last but not least I thank my friendsand colleagues for their advice and support.

Table of Contents

List of Figures

List of Algorithms

List of Tables

List of Abbreviations

IC: Information Content

IE: Information extraction

IR: Information retrieval

MRD: Machine Readable Dictionaries

NLP: Natural Language Processing

OL: Ontology Learning

POS: Part Of Speech

QA: Question Answering

SA: Semantic Annotation

SM: Semantic Mapping

SR: Speech Recognition

WSD: Word Sense Disambiguation

ABSTRACT

Word Sense disambiguation (WSD) is an important application which can be integrated with different NLP applications for better performance. The presence of different types of ambiguities has been one of the main challenges for different researches and it is recommended to have the integration of WSD. Accordingly, though different attempts have been done to design Amharic WSD, there are problems on disambiguating all ambiguous words from an input sentence. The works done before can only disambiguate one target word at a time. Few studies also reported that WordNet is used as a knowledge base during the disambiguation process. However, the information contained in the WordNet and in the disambiguation is only definitions of the words, which is equivalent with dictionary based. On the other hand, when we see works which are corpus based, there is problem of knowledge acquisition and they are limited to only verb word class.

Amharic WSD developed in this study is based on WordNet. Amharic ambiguous words used in the previous researchis used by adding relationships which are encoded in the WordNet and tested using augmented semantic space and context-to-gloss overlap implemented using python. Experiments are done to evaluate algorithmsimplemented in this study using Amharic sentences with ambiguous words. Word-level and sentence-level performance for one, two and three target words for different senses of ambiguous words are tested.

Experimental results shows that, context-to-gloss followed by augmented semantic space has achieved the highest recall 87% and 79% for three target words at word and sentence level respectively. And the highest average accuracy 80% and 75% at word-level and sentence-level is achieved by this approach. The major challenge in this study is getting data for both WordNet preparation and testing. The performance of the system can be increased if better stemmer or morphological analyzer is used, standard test sentences are used and fully constructed WordNet containing relationships for non-ambiguous words are used.

Keywords: Word Sense Disambiguation; WordNet; Synset; Amharic Language; Context-to-Gloss

CHAPTER ONE

INTRODUCTION

1.1. Background

Human languages are ambiguous in that there are words with multiple meanings which are interpreted depending on the context they occur (Samta & Monika, 2017). There are also words having the same orthography and/or phonography but different meanings in different natural languages. These words are referred as polysemy or multi-sense words (Udaya & Subarna, 2014). The meaning or interpretation of those words differs as the context changes. This is called sense of the word and the words having different senses are called ambiguous. For example, the Amharic word "ድካም"‹dkam› have two senses; "making a goal of reaching a goal" and "to say losing power because of some work or lacking rest". Generally there exists a sense of ambiguity for a given word when the context of that word is not considered. Knowing the sense of a word from the given context is easy for humans because we can use our experience of the language and infer the meaning by using the surrounding context. But, there are challenges in making machines to understand the senses from the context. So we need to use Word Sense Disambiguation (WSD) to enable machines to automatically recognize the sense of ambiguous words (Roshan &Manoj , 2015).

WSD is determining the most appropriate sense (meaning) of ambiguous words from a given text by analyzing the text contextually from finite number of senses provided for the word(Samhith, Arun, & Panda, 2016). The procedure of any WSD system consists of assigning the most appropriate sense (meaning) for input words by applying techniques which use one or more knowledge sources to acquire the sense information (Swathy, 2017). Word Sense Disambiguation is an intermediate task which aids other NLP tasks by increasing performance and accuracy because many NLP tasks face challenges of ambiguous words (Ravi, Mahesh, & Prashant, 2014). It is alsonecessary for many real world applications in improving the performances of machine translation (MT), semantic mapping (SM), semantic annotation (SA), and ontology learning (OL), information retrieval (IR), information extraction (IE), and speech recognition (SR) (Xiaohua & Hyoil , 2005).

According to the extent to which major words in text are sense tagged, WSD tasks fall into two types: tag all major words and tag some major words (Xiaohua & Hyoil , 2005). In all words task noun, verb, adjective and adverb word classes are considered. In tagging major word classes, only noun and verb word classes are considered. Most of the time it uses supervised model which is specific for each word. In all words task, a more general approach which can be used for all words is followed.

There are knowledge based and corpus based or the combination of the two approaches for WSD (Nyein, Khin, & Ni, 2011)(Rajani & Ravi, 2015). The knowledge based approach uses either structured or unstructured knowledge sources. The structured knowledge sources are Machine Readable Dictionaries (MRD), thesaurus, WordNet etc. These knowledge sources provide different relationships among words like synonym, antonym, and hyponymy to disambiguate. In this approach, all the senses of a word to be disambiguated are retrieved from the knowledge source. Each of this sense is then compared to the dictionary definitions of all the remaining words in context. The sense having the highest overlap with these context words is chosen as the correct sense. On the other hand, corpus based approach uses large sense-annotated or raw examples. Eneko and David(2001) indicated that large manually annotated corpus has robust results, but due to unavailability of it and the time it takes to prepare large corpus this approach is insignificant.

The reasons that WSD is challenging lie in two aspects (Xiaohua & Hyoil , 2005). First, dictionary-based word sense definitions are ambiguous. Even if trained linguists manually tag the word sense, the inter-agreement is not as high as it would be expected; i.e., different annotators may assign different senses to the same instance (Ng, 1999; Fellbaum and Palmer, 2001; cited in (Xiaohua & Hyoil , 2005)). Second, WSD involves much world knowledge or common sense, which is difficult to verbalize in dictionaries (Veronis, 2000; cited in (Xiaohua & Hyoil , 2005)).

Most of the popular algorithms like extended Lesk (Satanjeev & Ted, 2002) are developed for English. As such, the state of the art advancement on WSD is more on English. Alessandro et. al.(2017) made comparison on three supervised and three knowledge based all-words WSD systems using English test data sets from senseval. The result shown that WSD systems obtain low results for high level of ambiguities and supervised approach outperforms knowledge based system. However, the supervised once have knowledge acquisition problems. Thus, they proposed to enrich knowledge resources with semantic connections and use them will be better than using supervised approaches.

1.2. Motivation

It is difficult to know the sense of ambiguous words without understanding the surrounding context. WSD is the most challenging task in NLP because knowing the meaning of ambiguous words from the context is not easy for the machines unlike human. There is a need to investigate means to allow machine understand intelligently contextual meaning of words like human beings.

WSD used in the NLP techniques, increases the effectivness in selecting the accurate keywords that are used as features in the classification processes. The accuracy increases when WSD is used along with the semantics of the word, though WSD is considered as an open problem in NLP techniques (Swathy, 2017).The challenges and the effects of WSD on different NLP techniques are seen in different studies. A study done by Samrawit (2014) showed that WSD applied to query expansion in Amharic information retrieval increased the overall performance by 6%. Udaya & Subarna (2014) indicated that ambiguous words create a big problem in Machine Translatio. To translate the correct meaning of the polysemy word, the machine must first know the context in which the polysemy word has been used. Solomon (2010), citing (Yehenew, 2004) indicated that both lexical and structural ambiguities were challenges in his research on machine translation of English to Amharic. Solomon (2010) referring to (Yoseph, 2004) also noted that, in an attempt to design Amharic-English cross language information retrieval, he faced problems of synonym, polysemy and homonymy. The challenge has also been noticed as (Atelach *et al*., 2004); cited in (Solomon M. , 2010)) attempted to translate Amharic queries into

English “Bags-of-words”. They were required to perform manual disambiguation which misses domain specific senses that often contain rare senses and is time taking.

1.3. Statement of the Problem

If disambiguation is to be done by humans manually, it is expensive, tedious, prone to errors, and time consuming. Considering these facts automatic Amharic WSD is important. There are researches done for Amharic WSD especially using machine learning techniques. These are Solomon(2010) and Solomon (2011) explored Amharic WSD by using the same five ambiguous words and by using supervised and unsupervised approaches, respectively. Getahun (2014)also conducted a research by using semi- supervised approach. Later the work by Hagerie (2013) used Adaboost and Bagging ensemble classifiers. These works have limitations as they used small dataset containing limited ambiguous words and their senses, which makes performance evaluation difficult. In all of the above researches only ambiguous Amharic words from verb word class where considered but disambiguation have to take into account all open class words such as noun, verb, adjective and adverb. In addition, these works are fully based on corpus evidence which requires large amount of data and training. The performance of these kinds of approaches is greatly dependent on the amount of data we have. Moreover, this is very difficult to achieve for under resourced languages like Amharic, which is difficult to find sense tagged data.

When we looked at other works that did not use machine learning approaches we found a research done by Samrawit(2014), which applies WSD for query expansion in IR,based on semantic similarity measures by Lesk algorithm. WordNet was used to know appropriate meanings (senses) for queries having ambiguous words and the result shows 6% increase from the original query. The expansion would achieve better than this if the WordNet contains more information. However, the WordNet contains synset and gloss information only, other relationships were not considered. The Lesk algorithm has limitation when multiple words having multiple senses are considered at once (combinatorial explosion problem)(Satanjeev & Ted, 2002).

The work by Segid (2015)was based on WordNet which is constructed as a relational database by considering maximum of three senses for a word and implemented using Lesk algorithm. There are 2000 words included in the WordNet but how many of them are ambiguous is not indicated. Although Segid attempted to consider related words (the WordNet hierarchy) for small number of synsets (the number is not indicated) in constructing the WordNet, the actual disambiguation process and the experiments are only on the use of gloss of the word itself not related words. In addition to this as the objective is to disambiguate all open class wordsin the input sentence, the system is expected to disambiguate all open class words in a given sentence such as nouns, adjectives, adverbs and verbs. Nevertheless, it disambiguates only one frequently occurring target word in the WordNet for an input sentence.

Dureti(2017)proposed generic approach towards all Amharic word classes using corpus based Lesk algorithm. Still there is combinatorial explosion problem on Lesk algorithm. The information in the WordNet is limited, it contains only synsets and their glosses, the hierarchy of related words were not considered.However, WordNet should contain different relationships between those synonyms which makes it very important from other dictionaries.It was indicated to use well-constructed Amharic WordNet having richer informationin the future works. The developed prototype works by accepting the only one target word from the user one for a sentence. The experiment does not show disambiguation of more than one word and cannot identify that word automatically from a given input sentence.Dureti (2017)has achieved the highest result but this is because the experiments considered only one target word. To investigate WSD for Amharic more relationships between words have to be considered, ways to automatically identify the ambiguous word and other approaches have to be analyzed. Therefore, the main aim of this research is to design an approach for WSD which considers different relationships (hyponymy, hypernymy, meronymy, attribute, causes) between WordNet synsets to disambiguate more than one open class words from a given input sentenceat sentence level. To use these relationships the augmented semantic space and context-to-gloss overlap are implemented which also overcomes the combinatorial explosion problem of the Lesk algorithm.

To this end, the current study investigates and answers the following research questions.

- How to develop a WSD system that can disambiguate all ambiguous words in a given sentence?
- To what extent does the systemworks in disambiguating Amharic ambiguous words?

1.4. Objective of the Study

1.4.1. General Objective

The general objective of this studyis to designword sense disambiguation for Amharic sentences containing multiple all open class Amharic ambiguous wordsusing WordNet hierarchy.

1.4.2. Specific Objectives

To achieve the general objective of the study, the following specific objectives are attempted.

- ✓ To study features of Amharic language for word sense disambiguation.
- ✓ To prepare WordNet for the word sense disambiguation.
- ✓ To implement appropriate algorithm for the disambiguation.
- ✓ To develop a system for Amharic word sense disambiguation.
- ✓ To evaluate the performance of Amharic word sense disambiguationsystem.

1.5. Scope and Limitation of the Study

The research is focused on the disambiguation of Amharic ambiguous words at sentence level for all open class words.The main challenge for the study was getting and preparing the data for WordNet construction and testing. As Amharic is resource deficient language (Tessema, Meron, & Teshome, 2008) it is not possible to consider all ambiguous words in Amharic. Thus,17 words used by the previous researcher, which are from verb, noun, adjective and adverb word classes are considered. The WordNet containing about 250 synsetsis constructed with the help of language experts, but it is challenging in terms of time as it requires deep thinking and knowledge about the language. Because of this the WordNet does not include all relationships for single sense words in the WordNet and full WordNet hierarchy, we are only able to include those relationships for our ambiguous words in the WordNet.

Lexical and semantic ambiguities which are due to lexical elements are tried to be included. But ambiguities related to phonological order of speech (which are coused by placement of pouses) and due to structural arrangement of sentences are not considered because of time and data limitations.

1.6. Significance of the Study

Amharic WordNet having different relationships, which is developed here can be used in different works that can be done in Amharic NLP using WordNet. Also it can be a base for further extension of the WordNet. Recommendations and challenges of this study can be considered as aground for further studies on WSD.WSD is used in near about all kinds of linguistic researches(Alok & Diganta, 2015). It is an intermediate task which facilitates the performance of different NLP applications. Some of them are described below as discussed in (Samta & Monika, 2017)(Ravi, Mahesh, & Prashant, 2014)(Alok & Diganta, 2015)(Gerard, 2006)(Roberto, 2009).

Machine Translation (MT): This is the field in which the first attempt to perform WSD were carried out. WSD is required in MT for words that have different translations with different senses based on context. For instance, in Nyein, Khin, & Ni, (2011) research done on WSD using Naïve Bayes classifier using Myanmar-English Parallel Corpus, it is indicated that the result achieved shows the system improved the accuracy of Myanmar to English language translation system.

Information Retrieval (IR): IR systems need to resolve ambiguity of some queries to decide what information should be retrieved because of ambiguous words. An accurate disambiguation would allow it to eliminate documents containing the same words used with different meanings and to retrieve documents expressing the same meaning with different wordings.So WSD is prominent for query formulation and expansion.

Information Extraction (IE) and text mining: WSD plays an important role for accurate analysis of text information extraction in different research works as Bioinformatics research (which is investigating biological issues using mathematics, informatics, statistics and computer science) and Named Entity recognition system(subtask of information extraction aiming to locate names of known entities from unstructured text) (Alok & Diganta, 2015).Both of them includes the task of analyzing meanings, this is how they are related with WSD.

Speech Processing: WSD could be useful for the correct phonetisation of words in Speech Synthesis and for word segmentation and homophone discrimination in Speech Recognition.

Lexicography: lexicographers can be not only suppliers of NLP resources, but also customers of WSD systems (Kilgarri, 1997; cited in (Gerard, 2006)). WSD can help provide empirical sense groupings and statistically significant indicators of context for new or existing senses.

Acquisition of Lexical Knowledge: many approaches designed to automatically acquire large-scale NLP resources such as, selectional restrictions, sub-categorization verbal patterns (Briscoe and Carroll(1997); cited in (Gerard, 2006)), translation links (Atserias et al., 1997;cited in(Gerard, 2006)) have obtained limited success because of the use of limited WSD approaches.

1.7. Methodology

To achieve the aforementioned objectives of the research, different activities grouped in phase are done using different techniques. This needs a well-defined scientific methods.
This research formulates a new way of identification and assignment of senses for Amharic ambiguous sentences by preparing our own dataset. It is more about exploring a problem and making scientific investigations and experiments to address the problem. So in this study we followed design science researchapproach.According toPeffers et al.(2007), design science process model has six key steps. Those are problem identification and motivation, objectives of the solution, design and development, demonstration, evaluation and communication. This research takes these steps into consideration.

1.7.1. Problem Identification and Motivation

This step is the first step in design science which leads the researcher to understand the current state of the problem. To identify the problems we conducted literature review and discussion with experts on the area.For the successful completion of this study, different global and local researches were thoroughly reviewed from journals, conference proceeding and the Internet to have deep understanding on the area and to have detailed knowledge on the various techniques that are essential for WSD system.

1.7.2. Objectives of the Solution

The objectives of the research are driven from the problems identified in the first step. To define the objectives clearly the problems and the results of current solutions must be well identified and defined.This is done by reviewing literatures and state of the art solutions.

1.7.3. Design and Development

After the objectives are defined the next step is design and development of the study, which is based on the objectives. This includes architecture design, data set preparation and implementation of algorithms that solves the problem.

Approaches

The most widely used approaches for WSD are corpus based, knowledge based and hybrid approaches. In this research knowledge based approach is used, which uses WordNet as a knowledge base. The reason for using this approach is to take advantage of the rich information provided by the WordNet hierarchy and to tackle the data sparseness and training requirement problems of corpus based approach. The algorithms used for implementation are augmented semantic space and Context-to-Gloss overlap.

Tools

In this research we have used Python for development and testing, which is an open source scripting, cross-platform language used in wide range of NLP applications. After selecting algorithms, and manual development of WordNet, the application prototype is developed using Python including the user interface and performance evaluation. Python is selected for the development because it is readable and easy to learn as it has similarities with the usual English language , easy to find and correct errors, lets us to work more quickly and integrate our system more effectively and suitable for text processing applications(www.python.org/doc/, n.d.)(www.fullstackpython.com/why-use-python, n.d.).

Data collection and preparation

The data preparation process includes preparation of external knowledge source WordNet and test data set. The WordNet contains 17 ambiguous words from verb, adverb, noun and adjective word classes used by the previous researcher. Different relationships are added to the WordNet used by previous researchers. The related synsets of Amharic WordNet are translated from the English WordNet since we are unable to get Amharic WordNet which includes those relationships. This is done after getting synsets and glosses of our ambiguous words from Amharic-Amharic dictionary(አማርኛመዝገበቃላት, 1993 E.C). The related synsets for the synsets were collected from the English WordNet by translating the words to English. Hereafter the English synsets were translated back to Amharic. All of the data collection and test dataset preparation were performed with the help of Amharic Language experts.

1.7.4. Demonstration

The proposed design and architecture is demonstrated by deploying the prototype using python programming language. The system is demonstrated inseven experiments to show the word-wise and sentence-wise performance.The experiments are done on a total of 350 test sentences having one up to three ambiguous words. Also it is demonstrated by a graphical user interface which works for a sentence given by a user or for a file uploaded by the user.

Experimental setup

The system is developed and tested on a system with Intel Core i5 CPU of 2.5 GHZ speed, 8 GB RAM, 1 TB hard disk and Window 10 Operating system. We have used Python 3.5 programming language for implementing the algorithms, developing the user interface and our WordNet.

1.7.5. Evaluation

After preparing test set of Amharic sentences, automatic evaluation of the WSD system is done and reported by using precision, recall, accuracy and f1 measure. Which are discussed in chapter three in detail. This is to show how the implemented algorithms perform and the results of the demonstrations for the sentences in our test set.

1.7.6. Communication

After getting the results and findings the approaches followed, challenges, limitations and recommendations are reported here. After submission and presentation of this thesis report we have planned to publish in different scientific journals and conference proceedings. This will help the researcher for advancement of the academic journey for the future and will give the way forward for subsequent researchers for further improvements.To make the research paper general and understood by different people doesn't speak Amharic we append the English transliteration for all of the Amaric words in the examples.Also the Amharic alphabets and transliterations can be found on appendix III.

1.8. Organization of the Thesis

The remainder of the thesis is organized as follows: The second chapter presents reviews made on different literatures on WSD including overview of Amharic language and local researches done and their gaps. Chapter Three illustrates data preparation, techniques and algorithms which are applied here. The fourth chapter discusses the experimentation and discussion of the findings and results. Finally, Chapter Five deals with the conclusion and the way forward drawn from the findings of the study.

CHAPTER TWO

LITERATURE REVIEW

This chapter includes brief explanation about WSD.Different methods used and knowledge types required for the tasks are included.Recent related works that has been done by different scholars are also discussed.

2.1. Overview of Word Sense Disambiguation

Word Sense Disambiguation WSD is the task of assigning the correct sense of one or more words used in a sentence if they are ambiguous(Pierpaolo, Marco, Pasquale, & Giovanni, 2008). We say a word is ambiguous if it can be interpreted in more than one way having different meaning or sense. This happens in the vocabulary of any natural language and it is easy for humans to know the sense of the ambiguous words by understanding the context they come with. The objective of any WSD task is to enable machines understand the correct sense of this ambiguous words like humans do. The procedure of any WSD systemincludes applying a technique which makes use of one or more sources of knowledge to associate the most appropriate senses given a set of words with words in context(Swathy, 2017).

The problem of WSD has been described as Artificial Intelligence (AI)complete, i.e. The problem is ashard as any other problem in AI. NLP is the subset of AI and WSD belongs to NLP hence WSD is NLP-complete as well(Samhith, Arun, & Panda, 2016)(Devendra, 2014).According toDevendra(2014)this isbecause of different factors. One being the representation of word senses, senses can be represented at many levels of granularity. The main issue is to decide the refinement level to which the sense discrimination should be considered. The other important reason behind the complexity of the problem is that of heavy dependence on knowledge. Without knowledge, it will be impossible to disambiguate for machines and even for humans. WSD works bythe context of the word to be disambiguated and external knowledge sources, including lexical resources, as well as hand-devised knowledge sources, which provide data useful to associate words with senses(Pierpaolo, Marco, Pasquale, & Giovanni, 2008).

Main tasks of WSD includedetermining the different possible senses (or meanings) of each word, and, thentagging each word of a text with its appropriate sense with high accuracy and efficiency(Swathy, 2017).According toXiaohua & Hyoil(2005)and Devendra(2014),based on the extent to which major words in text are sense tagged, WSD tasks fall into two types.

Lexical sample task(Target word WSD**):** A restricted set of target words(usually nouns or verbs) is taken.The task focuses on disambiguation of this restricted set of words. Supervised systems are generally used for this task because system can be trained for each of the target word using manually tagged data.

All-words task: All word WSD expects the disambiguation of all the content words (verbs, nouns,adverbs and adjectives) in the given input. It is more challenging and has more practical applications than lexical sample task(Devendra & Ruslan, 2018).

2.2. WSD System Requirements

In generalWSD systems require four main parts (Roberto, 2009). Theseare the selection of word senses, the use of external knowledge sources, representation of context, and the selection of an automatic classification method.

2.2.1. Selection of Word Senses

Thefirst task before the disambiguation is to know possible senses for the target words to be disambiguated. Generally, in order to enable an objective evaluationand comparison of WSD systems, senses must be enumerated in a sense inventory. However, determining the sense inventory of a word is a key problem and expected senses may not be covered by the sense repository or it has too many senses unnecessarily for the algorithm. Also in identifying senses, we have to take care of semantic and lexical relations between senses and words(Nick, 2010).

The following are the four lexical relations between senses of words(Nick, 2010). The first one is Antonymy - is relationship between words which are opposite in meaning,their oppositeness may be gradable or non-gradable .Non-gradable antonyms are antonyms which doesn’t have mid points .For example ,girl-man have no mid points, if someone says girl/boy she/he is talking about necessarily girl/boy. If they are gradable, there are mid points between them for example hot-cold are gradable antonyms.

The second relation is Meronymy -is part-to-whole relationship when one word is part of another as foot is a part of leg. This relations are transitive where, if A is meronym of B and B is meronym of C then A is also meronym of C.

The other is Hyponymy-is sort/kind/type of relationship between words for example water is hyponym of drink. We say A is hyponym of B if all A is necessarily B but not vice versa.

The final one is Synonymy-words having similar meanings in any context. There are synonymy of senses and synonymy of words. Synonymy of senses is for words having similar meanings in some but not all of their senses whereas synonymy of words is for words sharing all their senses.

In addition the semantic relations between words include the following (Roberto, 2009)(Nick, 2010): Polysemy, Monosomy and Homonymy.

One of the semantic relations is Polysemy, which is for words having same phonological form but several semantically related meanings. For example, the Amharic word “አፍ”‹a*f› has senses “A mouthful of a tool that can hold fluid or other objects” and “oral opening of a human being” which are semantically related. Devendra (2014) noted that in WSD differentiating polysemy is difficult because it would be challenging to differentiate between closely related senses (meanings).

The other semantic relation is Monosomy, This is the opposite of polysemy in which a word have only one meaning (sense). Words with this characteristic do not require the application of WSD. The final one is Homonymy, which is for words having the same phonological form and orthography but different unrelated meanings (senses).

2.2.2. Use of External Knowledge Sources

Knowledge is a fundamental component of WSD. Knowledge sources provide data that are essential to associate senses with words. They can vary from corpora of texts, either unlabeled or annotated with word senses, to machine-readable dictionaries, thesauri, glossaries, ontologies, etc. which are described later in section2.4.

2.2.3. Representation of Context

The context is used to know the sense of the target words.But it has to be preprocessed as it is unstructured text to be used properly(Devendra, 2014).The preprocessing usually containsthe following tasks(Roberto, 2009):-

- Tokenization: it is the task ofsplitting up the text into a set of, tokens (usually, into a bag of words).
- Part-of-speech tagging: consisting in the assignment of a grammatical category to each word (e.g., “the/DT bar/NN was/VBD crowded/JJ,” where DT, NN, VBD and JJ are tags for determiners, nouns, verbs, and adjectives, respectively).
- Lemmatization: that is, the reduction of morphological variants to their base form (e.g. was → be, bars → bar).
- Chunking: which consists of dividing a text in syntactically correlated parts (e.g., [the bar] NP [was crowded] VP, respectively the noun phrase and the verb phrase of the example).
- Parsing: whose aim is to identify the syntactic structure of a sentence (usually involving the generation of a parse tree of the sentence structure as per the grammar rule of the language).

2.2.4. Selection of an Automatic Classification Method

The last one is choosing the classification approach to be used. This approaches range from the field of machine learning to knowledge based depending on the amount and type of data as well as the knowledge type they use which is discussed later in section 2.5.

2.3. Knowledge Sources used in WSD

Since knowledge is the fundamental and the basic component of any WSD system, any WSD system uses one or a combination of more than one knowledge sources. Knowledge sources used in WSD can be learned world knowledge or Lexical knowledge. Most of the time unsupervised systems use lexical knowledge, while supervised once use both learned and world knowledge(Xiaohua & Hyoil , 2005).

Lexical knowledge is usually associated witha dictionary. It is used in knowledge based, supervised and also foundation of unsupervised approaches(Roshan & Manoj , 2015)(Xiaohua & Hyoil , 2005).There are components of Lexicalknowledge as described inXiaohua & Hyoil(2005), such as sense frequency, sense gloss, concept tree, selectional restriction, subject code and Part of Speech (POS).

Sense Frequencyassigns most frequently occurring sense of a word. This frequency is used as the benchmark for evaluating other WSD algorithms. Our WSD algorithm should have accuracy equal to the frequent sense or above.

Sense gloss includes the definitions (gloss) and examples for senses of a word. The correct sense of a word is identified by counting overlaps between the context of the definition and examples,

Concept threegives related concepts to the target word. The relationship is hierarchical including hypernym, hyponym, holonym, meronym and synonym which are described below in section 2.4.2.

Selectional restrictionsare restrictions on each sense of the word. For example the Amharic word “ጠጣ“‹*TeTa*›is restricted to a human subject and the object which is liquid.

Subject code is a code assigned to a sense based on general category in which it is mostly used.

Part of speech (POS) is used to disambiguate a word fully if all of its senses have different POS,partially if some of them have same POS,and completely not if all of them have same POS.

Learned world knowledge is common sense knowledge. As it is difficult to use common sense knowledge, it can be represented and acquired from training corpus or a dictionary like WordNet.

2.3.1. Components of Learned World Knowledge

The components of learned world knowledge are described below(Xiaohua & Hyoil , 2005).The first one is indicative words. Indicative words are words that come around the target word that indicates which sense of the word is used in that context. These words can be selected by using fixed size window around the target word.

The other is domain specific knowledge which is knowledge acquired from corpora about each sense of a word. The domain can be terrorism, health, education etc.(Xiaohua & Hyoil , 2005).

It is also possible to construct parallel corpora.This isbilingual corpora of two languages(Xiaohua & Hyoil , 2005). Major words between the two languages are aligned in which pair of aligned words has the same sense. This is used in disambiguation of major words in the primary language.

2.3.2. External Knowledge Sources

External knowledge sources used in WSD can be further classified into structured and unstructured resources (Eneko & David, 2001)(Roberto, 2009).

Structured Knowledge Sources

There are different structured knowledge sources that can be used for designing word sense disambiguation. The main lexical sources are discussed below.

The first one is MRD (Machine Readable Dictionary). This is a dictionary which is available to read electronically. The first machine readable dictionaries were Collins English dictionary, Oxford Dictionary of English and Longman Dictionary of Contemporary English (LDDCE). LDDCE is widely used and have additional information such as subject codes, subcategorization information, and basic selectional preferences(Eneko & David, 2001). Among this we can mention the Lesk algorithm(Satanjeev & Ted, 2002)using LDOCE and the foundation of many different researches using LDOCE and WordNet as a knowledge base.

Thesaurusisalso another electronic resource like MRD containing information about relationships between words arranged in semantic categories representing different senses of words(Gerard, 2006).The most widely used thesaurus is Roget's international thesaurus. But it doesn't best fit for WSD because it has no rich information about word relations.

The third one is Ontologies, which are specifications of conceptualizations of specific domains of interest usually having different semantic relations(Roberto, 2009).The most widely used ontology is WordNet. WordNet is a lexical database which is different from traditional dictionaries and thesaurus developed by George Miller at the cognitive science laboratory of Princeton University (Samta & Monika, 2017)(Samhith, Arun, & Panda, 2016).After the development of English WordNet, other WordNets in Spanish, Italian, and Hindi were built(Udaya & Subarna, 2014). The reason which makes it different from traditional dictionaries and the thesaurus is that, it is arranged semantically rather than alphabetically (Satanjeev & Ted, 2002). Words are arranged in synonym sets called synsets. Words in the same synset(synonyms) have similar meaning and can be used inter changeably without changing the meaning of a sentence. Each synset also contains gloss and examples for the concept(Devendra & Ruslan, 2018). In addition to the synonymy relationship it contains different semantic relationships between the synsets. Most of these relationships are between synsets having the same POS(Devendra & Ruslan, 2018). These relations are discussed below as described by Satanjeev & Ted(2002).

Hyponymy and hypernymy are relationships for noun synsets where a synset A is kind of another synset B. We say A is hyponym of B and B is a hypernym of A. For example, the synset containing the Amharic word “መኪና”‹*mekina*›is hyponym of the synset containing “*ተሽከርካሪ*”‹*texkerkari*› and “*ተሽከርካሪ*”‹*texkerkar*›is the hypernym of “*መኪና*”‹*mekina*›.

Holonymy and meronymy are also relationships for noun synsets when a synset B has synset A as a part and A is a part of B. We say B is holonym of A and A is meronym of B. For example, the synset containing the Amharic word“*መኪና*”‹*mekina*›is holonym of the synset containing “*ሞተር*” ‹*moter*› and “*ሞተር*”‹*moter*› is meronym of “*መኪና*”‹*mekina*›.

Hypernymy and troponymyare relationships for verbs when synset B is one way to A.We say A is hypernym of B and B is troponym of A.It can be viewed as hypernym and hyponym relation of nouns.

Attribute is the only cross POS relationship which is between noun and adjective. When an adjective synset B is a value of a noun synset A we say B is an attribute of A. For example the adjective synset containing the Amharic word “*ቆንጆ*”‹*qonjo*› is value of noun synset containing the word “*ውበት*”‹*wbt*›.

Unstructured Knowledge Sources

The unstructured knowledge source for WSD is corpus, whether it is labeled or unlabeled (raw). When we discuss about raw corpus the widely used and mentioned is the brown corpus which is a million word balanced collection of texts published in United States (Roberto, 2009). There are other widely known corpus, such as British National Corpus (BNC), Wall Street Journal (WSJ) corpus, American National corpus and the Gigoword corpus (Roberto, 2009).These unlabeled corpus areused for developing unsupervised WSD.

There are also sense annotated corpora in which the examples are labeled with the senses the most widely used sense around 234,000 sense annotations. It contains all the open class words annotated with POS tags, lemmas, and word senses from the WordNet inventory(Roberto, 2009).

2.4. Approaches to WSD

Commonly WSD systems classified based on the knowledge type they use. There are three different approaches for WSD;such as knowledge based,corpus based and hybrid approach which is the combination of both corpus based and knowledge based approaches. These approaches use different procedures, knowledge sources and algorithms.

2.4.1. Knowledge Based Approaches

Knowledge based approaches for WSD involve methods that use explicit lexicon such as MDR(Machine Readable Dictionary), thesauri,ontologies,collocations etc. to extract knowledge from word definitions and relation among word senses(Ravi, Mahesh, & Prashant, 2014)(Roberto, 2009)(Rajani & Ravi, 2015).The works done earlier on WSD were theoretically interesting but practically in limited domains until 1980s and 1990s when these lexical resources become widely available (Rajani & Ravi, 2015).The first knowledge based approaches to WSD date back to 1970s but the lack of these large scale computational resources prevented a proper evaluation, comparison and exploitation of these methods(Roberto, 2009).

Knowledge based approaches trust only knowledge sources mentioned above without using any corpus evidence. This makes these systems a powerful alternative to supervised systems which are heavily relying on large amount sense annotated data (Rajani & Ravi, 2015)(Pierpaolo, Marco, Pasquale, & Giovanni, 2008).This makes knowledge based systems ready to use and scalable but they reach lower precision than supervised corpus based methods when training data is available(Pierpaolo, Marco, Pasquale, & Giovanni, 2008)(Rajani & Ravi, 2015).This poor performance is because of their complete dependence on dictionary defined senses whose readiness must be guaranteed and lack of world knowledge (Rajani & Ravi, 2015).

There are four main types of knowledge based methods. These are overlaps based approach (Lesk algorithm), selectional preferences (restrictions), semantic similarity and heuristic(Ravi, Mahesh, & Prashant, 2014)(Roberto, 2009)(Devendra, 2014).

Overlap Based Approaches

Overlap based approaches are based on MRD (Machine Readable Dictionary) which works by calculating the overlap between sense bag or context bag of the two or more target words(Roberto, 2009)(Rajani & Ravi, 2015).Sense bags are features of different senses of an ambiguous word whereas context bag is feature of the word in its context. The features can be sense definitions, example sentences or hypernyms. Then the objective is to select a sense with highest overlap (Rajani & Ravi, 2015).This approach is heavily dependent on dictionaries which also have some restrictions over acquiring the common sense knowledge. It is based on the first MRD based algorithm called Lesk(Alok & Diganta, 2015).But there are other variants of Lesk algorithm like simplified Lesk, adaptedLesk and simulated annealing which are discussed below.

The first MRD based algorithm proposed in 1986uses overlap of word definitions from Oxford Advanced Learners Dictionary (OALD) to disambiguate the word senses(Udaya & Subarna, 2014).The algorithm works for ambiguous words in short phrases by comparing each sense of the ambiguous word with the glosses of every other word in the phrase.The ambiguous word will be given the sense whose gloss shares greatest number of words with the glosses of words in the phrase(Satanjeev & Ted, 2002)(Sudip & Sivaji, 2007).The algorithm was demonstrated by the English words pine and cone and a precision of 50–70% was observed (Satanjeev & Ted, 2002).

The drawback of the algorithm is its dependency on glosses of traditional dictionaries.These dictionaries often do not have enough words for this algorithm to work well which can be overcome by using WordNet (which includes different types of relationships between words).In addition Lesk algorithm works for short phrases,meaning it uses local approach by disambiguating each word separately and does not utilize sense previously assigned(Sudip & Sivaji, 2007)(Satanjeev & Ted, 2002). Also there is combinatorial explosion problem,the problem arises when there are more than one ambiguous (open class) words in the input text (Satanjeev & Ted, 2002)(Phiip & Eneko, 2007). For example: if we have nine open class words with the following number of senses: 26, 11, 4, 8, 5, 4, 10, 8, 3 then the number of sense combinations is 43,929,600(Phiip &

Eneko, 2007). It is huge number and hence practically difficult to figure out the optimal combination using definition overlaps is not a tractable approach(Satanjeev & Ted, 2002)(Phiip & Eneko, 2007).The other variants of Lesk algorithm are proposed to fill these limitations.

One of the solutions proposed for limitation of Lesk algorithm was Simulated Annealing, which was proposed by Cowie *et al*.(1992)to tackle the combinatorial explosion problem ,exists when more than one ambiguous words appears and allows all senses to be identified once. Selection of word senses is based on a function E (see equation 2.1) that they seek to minimize indicating higher redundancy. Redundancy is computed by giving a stemmed word from which appears n times a score of n-1 and adding up the scores and

$$E = \frac{1}{1 + R} \tag{2.1}$$

Another variant of Lesk algorithm is augmented semantic space proposed by Banerje and Paderson(2002) which uses WordNet as a sense inventory. The Lesk algorithm considers only the gloss of the word itself and very sensitive to exact wording but this definitions are very short and insufficient which greatly reduces performance. In adapted Lesk gloss of the word itself and glosses of related words are considered in the disambiguation. These related words are based on the WordNet hierarchy(hypernym, hyponym, meronym…).They also introduced a scoring mechanism that gives highest scores for long sequence of matches. When this algorithm was evaluated on English senseval-2 lexical sample data it shows accuracy twice of the Lesk algorithm.

Selectional Preferences (restrictions)

The aim of this approach is to constrain the possible meaning of word. This is by imposing restrictions on the semantic type that a word sense imposes on the words with which it combines in sentences usually through grammatical relations(Rajani & Ravi, 2015)(Roberto, 2009). EAT-FOOD, DRINK-LIQUID, are examples of such semantic constraints, which can be used to rule out incorrect wordmeanings and select only those senses that are in harmony with common sense rules(Phiip & Eneko, 2007). Selectional restrictions rule out senses that violate the constraint, whereas selectional preferences tend to select those senses which better satisfy the requirements(Roberto, 2009).

As Hindle and Rooth, (1993); cited in(Roberto, 2009) pointed out, the easiest way to learn selectional preferences is to determine the semantic appropriateness of the association provided by a word-to-word relation. The ways to measure the semantic appropriateness to measure word-to-word relations are frequency count and conditional probability.Frequency count is the easiest way which counts how many times this kind of word pair appears in the corpus with syntactic relation. The later (conditional probability) estimates semantic appropriateness by calculating conditional probability of a word given other word and the relation. In general selectional preferences (restrictions) have not been found to perform as well as Lesk based and most frequent sense heuristic(Roberto, 2009).

Measure of Semantic Similarity

These are methods for finding the semantic distance between concepts. Appropriate sense of a word is a sense having smallest semantic distance from the given context. Since the early 1990's, when WordNet was introduced a number of measures of semantic similarity are introduced at different times(Roberto, 2009). These similarity measures, as noted by (Jason, 2005)and ((Pederson *et al*., 2005) as; cited in(Verena & Erhard, 2012)can be grouped into path-based and information content based. These are discussed below with different similarity measure algorithms proposed at different times.

Path-based measures

Path measure(Jason, 2005)(Verena & Erhard, 2012)is a simple measure that uses the path length between two concepts to know their relatedness. This method uses WordNet and the concepts to be measured are synsets in the WordNet. The distance between two synsets is measured using node counting (Jason, 2005).The drawback of this node counting measure is that links in taxonomy like WordNet can represent different distances between synsets.

Some links may represent a large difference in meaning, while others may represent only a small difference in meaning (Jason, 2005).

As cited in Verena & Erhard(2012), (Wu and Palmer (1994) and also Leacock and Chadorow(1998) as cited in (Jason, 2005)) proposed path based similarity measures. The first one which is proposed by Wu & Palmer(1994) present conceptual density based on the distance between each of the concepts and their LCS (Lowest Common Subsumer) as well as distance between LCS and the root of the taxonomy in which the synsets reside. The similarity measure is computed as the shortest path length normalized by the depth of their LCS. The similarity measure of Leacock and Chadorow (1998) is also based on distance and depth between root of the taxonomy and the synset. Similarity between concepts is computed as negative logarithm of the length of the shortest path between the concepts over the path length of overall depth of the taxonomy or WordNet. Similarity between synset s1 and synset s2is computed using equation 2.2 given below (Jason, 2005):

$$\mathrm{Simpath}(s1;\ s2) = \frac{1}{\mathrm{distnode}(s1;\ s2)} \qquad (2.2)$$

Such that distnode(s1; s2) is the distance between synset s1 and synset s2 using node counting.

Information Content(IC)-based measures

This measure depends on relative frequency of concept (synset).IC is inversely proportional with frequency occurrence of concepts, common concepts have low IC and rare senses have high IC. The probability of a concept (synset) is calculated as frequency of the concept divided by number of concepts occurring in a corpus. IC is negative logarithm of probability of the concept. Then high IC means that the concept conveys a lot of meaning when it occurs in a text (Verena & Erhard, 2012)(Jason, 2005).

Resnic(1995) and Lin(1998); cited in (Verena & Erhard, 2012)and(Jason, 2005)) introduced IC based similarity measures. In the first one similarity is computed as IC of their LCS (Lowest Common Subsumer) of two synsets (concepts).When there are more than one subsume of two synsets,LCS is defined as common subsumer with greatest IC.

All pairs of synsets with the same LCS will have the same similarity score. And maximum similarity value for the Resnik measure occurs when the frequency of a LCS is one. Lin's similarity measure is based on three assumptions. Firstly, the more similar two concepts are, the more they will have in common. Secondly, the less two concepts have in common, the less similar they are. Thirdly, maximum similarity occurs when two concepts are identical. Similarity between two concepts is measured as the IC (multiplied by two) of their LCS over the sum of the ICs of the concepts. The Lin measure is similar to the measure of Wu and Palmer, except that depth is replaced with information content (Jason, 2005). Mathematically, the information content of a concept is:

$$IC(c) = -\log P(c),$$

Where P(c) is the probability of the concept c. In semantic similarity measures, a concept is a synset, and the probability of a concept is the frequency of the concept divided by the number of concepts occurring in a corpus:

P(c) = frequency(c) =N,such that N is the number of concepts in the corpus from which the frequency counts were extracted.

Heuristic

Heuristic is a method which assigns senses based on three assumptions most frequent sense, one sense per discourse and one sense per collocation(Ravi, Mahesh, & Prashant, 2014). Most frequent sense is based on the idea that there are senses appearing most frequently than others. One sense per discourse based on the assumption that a word preserves its sense in all of its occurrences of a given document. And one sense per collocation assumes that a word preserves its meaning when collocated with the same nearby words and these collocated words greatly affect its sense.

2.4.2. Corpus Based Approaches

This system represents context on the form of feature vectors. These features may be word collocations, POS labels, domain information, grammatical relationships etc. These approaches is then used in an automatic learning process. But are highly dependent on human intervention, type of training data needed, the nature of linguistic knowledge used and the output produced (Rajani & Ravi, 2015).

Corpus based algorithms perform better than knowledge based systems when large training data are available. But this requires large amount of data and introduces knowledge acquisition bottle neck problem especially when supervised WSD is used(Pierpaolo, Marco, Pasquale, & Giovanni, 2008).

Supervised Corpus Based Approach

This methods uses corpus consisting of sense annotated training data to train machine learning algorithms(Ravi, Mahesh, & Prashant, 2014). This methods achieve high results than others(Ravi, Mahesh, & Prashant, 2014)(Alok & Diganta, 2015)(Devendra, 2014)but it requires large amount of annotated examples. This is called “knowledge acquisition bottleneck”(Gole *et al.*, 1993). Ng (19976); cited in(Gerard, 2006) estimated that “to obtain highest result at least 3700 words should be tagged with about 1000 occurrences each” and the necessary effort to move thus is estimated to be 16 person – years. Some of the machine learning (ML) techniques applied in supervised learning are discussed here.

Decision list

Decision list is an ordered set of weighted if-then-else rules learned from tagged training set.The rules are created in the form feature, sense, and weight. Feature is the condition for a particular sense of a word and the weight is the score (likelihood) to be that sense of a word for the given feature (condition) (Abhishek & Manoj B., 2013)(Devendra, 2014).For a test case this rules are checked in decreasing order and sense having the highest score is selected as the correct sense. As in(Devendra, 2014) if we consider a test sentence containing a word w the decision list is checked and a sense having the highest score will be selected among the list using the following formula.

$$S = argmax_{Si \in SenseD(w)^{Score(Si)}} \tag{2.3}$$

The score of each sense *Si* is calculated as:

$$Score(Si) = max_f log\left(\frac{P(Si\backslash f)}{\sum_{i\neq j} P(Sj\backslash f)}\right) \quad (2.4)$$

Naïve Bayes

Naïve Bayes classification (Abhishek & Manoj B., 2013)(Alok & Diganta, 2015)(Nyein, Khin, & Ni, 2011)is a probabilistic method which assigns class for a sample based on Bayes theorem.To determine sense for a word w sense having maximum probability p(w=Si/f1,f2…..fn) is selected using each sense Si and features that comes with the context of w is chosen(Nyein, Khin, & Ni, 2011).As shown inAlok & Diganta(2015) the most appropriate sense is calculated by the following formula:

$$\hat{S} = \underset{S_i \in Sense_D(w)}{argmax} P\big((S_i|f_{1,\dots}f_m)S_i\big) = \underset{S_i \in Sense_D(w)}{argmax} \frac{\big(P(f_{1,\dots,}f_m|S_i)P(S_i)}{P(f_{1,\dots,}f_m)}$$

$$= \underset{S_i \in Sense_D(w)}{argmax} P(S_i)\prod_{j}^{m} P(f_i|S_i) \quad (2.5)$$

Where m represent number of features (Si) is probability calculated from the co-occurrence frequency in training set of sense and P$\big(f_j|S_i\big)$ is calculated from the feature in the presence of the sense. The probabilities are determined using maximum-likelihood estimation as follows(Nyein, Khin, & Ni, 2011).

$P(S_i) = C(S_i)/N$

$\mathrm{P}\big(f_j\backslash \mathrm{w} = S_i\big) = \mathrm{C}(f_j, S_i)/\mathrm{C}(S_i)$

C (Si) and C (fi,Si) are number of frequency counts of Si and number of counts of feature fi in sense Si, respectively.

Pederson(2000)in his research appliedensembles of Naïve Bayesian classifiers on WSD on widely studied nouns, showed best result than previously reported results. Each classifier was based on co-occurrence feature extracted from different window size. Also Rezapour, Fakhrahmad, & Sadreddini(2011) reported that the ensemble of Naïve Bayes outperforms decision list, Naïve Bayesian, Nearest Neighbor, Transformation based learning and boosting with respect to the data set used.

Decision tree

Decision tree is prediction based model which uses rules to partition the training dataset and to decide senses(Abhishek & Manoj B., 2013). Moreover it uses the same logic as decision list where feature vector used is the same as feature vector of decision list the rules are in the form of yes-no. The test is applied on each internal node of the tree which is the feature value and each branch represents the output of the test.When the traversal reaches on the leaf node the sense of an ambiguous word is decided(Alok & Diganta, 2015).For example, for the noun sense of the word “Bank” in the sentence “I will be at the bank of Narmada River in the afternoon” decision tree is traversed to select the correct sense bank/RIVER in this context.

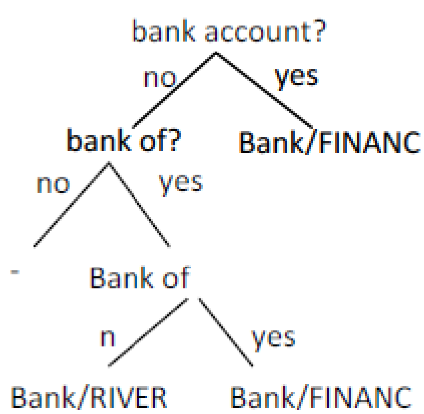


Figure 2. 1Sample decision tree for the given example(Alok & Diganta, 2015)

Exemplar-Based approach(K-NN)

Exemplar-Based approach was first introduced for WSD by (Ng and Lee(1996);cited in (Phiip & Eneko, 2007)).This approach works by selecting K nearest(most similar) neighbor example senses for an ambiguous word which is measure of smallest hamming distance. It is memory based learning because training examples are kept in memory during the training phase(Gerard, Lluis, & German, 2000)(Rezapour, Fakhrahmad, & Sadreddini, 2011).The value of K is known experimentally and if it is one the test instance it will be assigned to its nearest neighbors(Rezapour, Fakhrahmad, & Sadreddini, 2011).As it is indicated before, the similar vector means the nearest neighbor,so most of the time this distance is calculated as Euclidean distance between the instances to be classified to each of the examples in the training set.

Gerard, Lluis, & German, (2000), citing(Ng,1997a)and (Daelemans *et al*., 1999), noted that exemplar based learning is the best and superior learning in WSD and other language processing applications because as they are memory based they do not forget exceptions. Also Gerard, Llu´ıs, & German(2000, p. 39) discussed that exemplar based learning algorithms outperform naïve Bayesian when they are extended with example attribute weighting. This was tested by Rezapour *et al*.(2011)in their research of applying a feature weighting strategy and achieved promising improvements.

Support Vector Machine(SVM)

SVM is a binary classifier in which a hyper plane is learned from the training examples. The hyper plane separates positive examples from negative examples by maximizing the distance between closest positive and closest negative examples which are known as support vectors (Abhishek & Manoj B., 2013)(Devendra, 2014)(Alok & Diganta, 2015).So the main goal is to maximize this distance and minimize classification error.As we have said SVM is a binary classifier it classifies the samples into two, but in WSD a word can have more than two senses.So to make SVM fit to WSD problem,each sense of a word is considered as one class and the other remaining senses will be considered as members of the same class(Abhishek & Manoj B., 2013).This has to bedone for each sense versus all other classes.

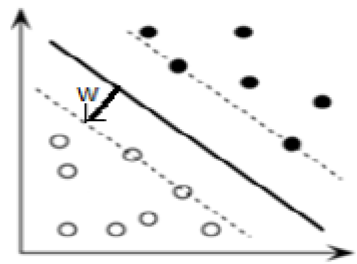


Figure 2.2Hyperplane constructed by SVM(Abhishek & Manoj B., 2013)

SVM is based on weight vector w perpendicular to the hyper plane and bias b which determines the offset of the hyper plane from the origin as shown in Figure 2.2(Abhishek & Manoj B., 2013)(Devendra, 2014).

Ensemble methods

These methods are based on combining different classifiers for constructing different models that work together so as to get improved result. There are different strategies to combine the classifiers(Alok & Diganta, 2015), like majority voting, probability mixture, rank based combination and adaboost.

In majority voting, the vote is given to each sense by the classifiers, and then the sense getting the majority vote will be selected as the correct sense. When we come to probability mixture, at the beginning confidence score for each sense is evaluated for the classifiers. Then the sense with the highest probability score will be chosen forthe disambiguation task. The third one is rank based combination in which the first order classifiers gives rank for each sense then, this rank will be summed up to assign the correct sense. The last one is adaboost where different weak classifiers are combined to give a strong one. The classifiers are given equal weight at the beginning from the weighted training set. For each classifier iteration is performed and weight for classifier for incorrect classification is increased. The other classifiers focus on disambiguating those incorrect examples having highest weight.

Semi-supervised Corpus Based Approach

Semi-supervised approach is in between supervised and unsupervised approaches which require both labeled and unlabeled data. These methods are gaining popularity because it reduces large amount of tagged data required for supervised learning (knowledge acquisition bottleneck) and out performs totally unsupervised learning(Ankita, 2013).Also Lokesh & Kalyani(2015) pointed that, when unlabeled data is used with small quantity of labeled data increases the machine learning algorithm efficiency and gives improved performance with less effort. The common semi supervised learning methods are discussed below.

Yarowsky Bootstrapping method

Bootstrapping algorithm proposed by Yarowsky was the first most successful algorithm which relies on small amount of labeled instances(Ankita, 2013)(Lokesh & Kalyani, 2015). In this algorithm initially classification model is learned from the instances using one of supervised algorithms, in case of Yarowsky the algorithm was decision lists. Then unlabeled instances are classified iteratively by the learned model and added to labeled dataset. The training setsused in each iteration will be classified with confidence above a certain threshold will be used to further classify untagged sets for the future. The main advantage is the ability to increase training examplesinevery iterations from small amount of initial training data(Lokesh & Kalyani, 2015).This iteration stops when no change is observed from the previous iteration. Bootstrapping works based on local consistency assumption, that examples close to the labeled examples will have the same class which is like K-nearest neighbor algorithm(Ankita, 2013).

Ankita,(2013), citing (Martinez and Agirre(2000), pointed out that, a far less predictive power of the one sense per discourse and one sense per collocation heuristics (which are the real domains of this algorithm) was observed when tested on a real domain with highly polysemous words.

Bilingual bootstrapping methods

Bilingual bootstrapping methodsworks on words to be translated to other language using small amount of labeled and large amount of unlabeled data for both target and source languages(Ankita, 2013). This is the only thing which makes it different from Yarowsky's bootstrapping. At every step a classifier is constructed for both languages and unclassified data will be added to the classified data for both languages. In constructing the classifier we can use one of the languages because words in one language have translations to the other(Ankita, 2013).When evaluated on word translation disambiguation it outperforms monolingual bootstrapping(Lokesh & Kalyani, 2015).

Label propagation algorithm

As discussed inLokesh & Kalyani(2015) label propagation algorithm is graph based algorithm with labeled and unlabeled examples are vertices and edges as label information. The unlabeled examples are labeled by using the labeled vertices by propagating through the edges. If two vertices (examples) are closer they will have similar labels. But this is not determined only by labeled examples; closer labeled examples can also determine it(Ankita, 2013).This algorithm works by global consistency assumption which makes it different from bootstrapping algorithms which are based on local consistency assumption (Ankita, 2013).Local consistency assumption is based on the assumption that, examples lose to labeled examples within the same class,whereas global assumption assumes that similar examples should have similar labels.

Unsupervised Corpus Based Approach

Although supervised approaches have highest accuracy and performance it is difficult and time consuming to get the required resource(Devendra, 2014). Even if we have enough resource it is not easily scalable for use in other languages and the disambiguation is constrained on fixed number of senses found in the repository. One of the efforts done for those problems is supervised learning. But unsupervised learning is data driven and language independent which can easily be scalable to other languages and new domain(Devendra, 2014)(Ted, 2007).

Unsupervised WSD performs grouping (clustering) of senses rather than classification. Although it overcomes different problems of unsupervised learning it has also some shortcomings(Lokesh & Kalyani, 2015).The first one is lower performance than other approaches as it is fully unsupervised. In addition number of clusters may differ from actual number of senses, so this makes performance evaluation difficult. Consequently to check the quality of clusters human must involve looking for the relationships between the members of clusters(Roberto, 2009)(Lokesh & Kalyani, 2015).

There are different methods, approaches and algorithms under unsupervised WSD. According to Pederson(2007, p. 42) the two main approaches are distributional and translational equivalent approaches in which he called them knowledge-lean approaches due to the fact that they do not use knowledge other than un-annotated corpora. Distributional approaches works on discrimination and are based on words that co-occur in similar context have similar meaning. On the other hand, translational equivalent approaches use parallel bilingual corpora of two languages which can be used for automatic construction of sense inventory.

Under the mentioned approaches, there are two important methods; token based and type based methods(Ted, 2007)(Roberto, 2009)(Alok & Diganta, 2015)(Lokesh & Kalyani, 2015). In case of distributional approach, token based method works by clustering (grouping) contexts in which the target word occurs with the same sense to one group. On the other hand type based method identify and cluster words that are to be co-occurring together with similar contexts. Whereas in translational equivalent, type based method produces set of related words in the source language (bilingual dictionary) and token based method produces sense tagged text by giving appropriate translation for a sense of target word for each of its occurrences.

Let us see the most common methods and algorithms that are used for unsupervised WSD.

Context clustering

Unsupervised approaches based on context clustering have basically two steps which are representing the target word as context vector (which is a word space vector meaning that its dimensions are words) then clustering the vectors(Roberto, 2009).The context vector is constructed for each occurrence of a word in a corpus. A vector includes all senses of a word and it is represented by average agglomerative clustering(Lokesh & Kalyani, 2015). This set of vectors for a word is used to construct co-occurrence matrix which is going to be used for similarity calculation between words(Roberto, 2009)(Lokesh & Kalyani, 2015). In constructing the matrix if we are dealing with large number of dimensions Latent Semantic Analysis (LSA) is used to reduce it, then similarity between words is calculated using cosine similarity. Finally, clustering is done based on the context similarity.

Word clustering

In word clustering clusters of words are formed rather than contexts.First words which are similar to the target word are identified(Roberto, 2009)(Ted, 2007).The similarity is based on Information Content(IC) based on single feature, where having highest IC is more similar and less similar if the IC is low.The listed words represent different senses of the word so clustering of these words is performed at the last to classify them into senses.To do the clustering there are Latent Semantic Analysis (LSA), Hyperspace Analogue to Language (HAL) and clustering By Committee (CBC) algorithm(Ted, 2007).

Translational equivalence

In this method word aligned parallel corpora of two languages is required. Also it is sentence aligned, but it is indicated that word alignment is an open problem. For a target word its lexical or syntactic features as well as its translation to the target language are used to create training context. Then the features will be used to indicate the appropriate translation of the target word(Ted, 2007).

2.4.3. Hybrid Approach

Hybrid approach is a combination of two or more corpus based approaches or mostly the combination of corpus based and knowledge based approaches.The main aim of this approach is to get advantage of having more knowledge sources and the strength of different approaches (Rajani & Ravi, 2015)(Mark & Yorick, 2001)(Pierpaolo, Marco, Pasquale, & Giovanni, 2008).As discussed earlier both corpus based and knowledge based approaches have their own limitations.Knowledge based approaches can help corpus based approaches when there is lack of training data and corpus based methods helps in achieving high performance results(Pierpaolo, Marco, Pasquale, & Giovanni, 2008).

Different works before have shown better results using hybrid approaches for WSD.Here we can mention the work by Pierpaolo *et al*.(2008) which uses K-NN classifier and WordNet for Italian all-words disambiguation.The experiments were done on Italian all-words task dataset and have shown that applying the knowledge based approach after K-NN shows better performance In addition Roshan and Manoj (2015) proposed a hybrid approach to evaluate performance of their system with and without learned knowledge. They used WordNet, Semcor and POS as knowledge sources. Naïve Bayes classifier was applied to get senses with highest probability. Finally they have concluded that WordNet (without world knowledge) is better to identify ambiguous words from the input because world knowledge (corpus) contains unnecessary information about a word which is more than enough to identify it is ambiguous.

2.5. Senseval Evaluation Exercises for WSD

Sensevalis a series of evaluation exercise on WSD computer programs(www.senseval.org, n.d.). The aim of these exercises is to compare the strengths of different WSD programs and to make this programs and datasets available for later uses. The first senseval started on September 2, 1998 for English, French and Italian lexical sample task at Hurst Monteux castle, Sussex, England(Roberto, 2009). There were three best performing programs, the first one is supervised algorithm based on hierarchies of decision lists. The second and the third are hybrid approaches which uses a hybrid of frequency of senses in training data, manually crafted clue words and contextual similarity measures as knowledge sources and memory based learning, respectively.

Then senseval-2 took place on July 5-6,2001 in France(www.senseval.org, n.d.). Different from senseval-1 12 languages were included with three different tasks; all-word,lexical sample and translation(lexica sample task in which word sense is defined according to translation distinction).The knowledge sources used are lexicon of word sense mapping(the first time WordNet is used in senseval),manually tagged corpus and sense hierarchy to be used in scoring. The best performing system from lexical sample task was the one using ensembles of cosine similarity, Bayesian models and decision lists and in all-words task was based on pattern learning from few examples(Roberto, 2009).

After 3 years senseval-3 (www.senseval.org, n.d.)took place in Barcelona on seven languages and total of 14 tasks in addition to lexical and all-words task. In this competition (Roberto, 2009)lexical sample task founded to be less interesting and in all-words task the best supervised method which uses semcor, the previous senseval corpora and usage examples in WordNet was the best method. But the performance for all-words was less than sensevsl-2 because of more difficult input texts(Alok & Diganta, 2015).

After senseval-3 (www.senseval.org, n.d.),different editions of senseval took place on the interval of three years by renaming it as semval which includes tasks on semantic analysis not only related to WSD.

2.6. AmharicLanguage

Amharic is a Semitic branch of the Afro-Asiatic language family, which is an advanced form of widely spoken language in Ethiopia and the working language of the Federal Government of Ethiopia(Eldward, 1973).The origins of the language and its people are traced back to the first millennium B.C.

Amharic is not only spoken in Ethiopia, there are also speakers in Canada, the USA, Eritrea and Sweden. This makes researches done on Amharic to have significant benefits. Besides, it is the second most spoken Semitic language next to Arabic in the world.

2.6.1. Amharic Writing System

Amharic is a syllabic language which uses a script, that originated from the Ge‘ez alphabet. Amharic language consists of 33 basic characters; each basic character has seven different orders. Therefore, there are 231 core characters in Amharic (Hayward, Katrina, & Richard, 1999)containing a set of 38 phones: 7 vowels and 31 consonants(Bender, Bowen, Cooper, & Ferguson, 1976).

2.6.2. Ambiguities in Amharic

As discussed by Getahun(2001),in Amharic language there are five types of ambiguities: Phonological Ambiguity, Lexical Ambiguity, Structural Ambiguity, Referential Ambiguity, Semantic Ambiguity, and Orthographic ambiguity.These are discussed below one by one with examples:

Phonological Ambiguity

Phonological ambiguity caused due to placement of pause on structures. The difference in placement of pauses and their absence causes categorical and meaning difference. For example,

a. [dägg + säw] näbbär
b. [däggtsäw] näbbär
c. [däggts -äw] näbbär

Sentence (a) interpreted as “He was a kind man” when there is a pause (+) in [dägg + säw] but if there is no pause in [däggtsäw] as in sentence (b) the sentence is interpreted as “They had made a preparation for a banquet”. Other than pauses, in sentence (c) [däggts -äw] appears as a verbal predicate which is interpreted as “They had prepared”.

Lexical Ambiguity

This is caused by lexical elements. Under lexical ambiguity there are different factors which are discussed below:

Categorical ambiguity

This is caused by lexical elements having the same phonological form but different word classes. For example, akrma saťť-ačč-ňň can be interpreted as:

a) She gave me akrima(a kind of grass)
b) She gave me something after delaying it for some time.

The source of ambiguity is /akrima/ which have a noun meaning ‘a kind of grasses’ in the first sentence and a nominal or verbal meaning ‘delaying’ in the second sentence.

Homonymy

This type of ambiguity is caused due to lexical items having the same phonological form but different meanings. For example,

a. bä - wär -e al - 1- t- fátta –mm interpreted as “'I will not be released in a month”
b. bä - wäre al - 1 -t-fatta –mm interpreted as “I will not get frustrated by any rumour”

The source of ambiguity is / bäwäre / which can be /bä-wär-e / in which /-e/ is first person possessive suffix to /wär / means ‘month’ and /bä-wäre / which means ‘by rumor’.

Homophonous affixes

This is caused by affixes having the same phonological form but gives different meanings when added to the same stem word.

For example, bet-u färräsä, can be interpreted as:

a. The house is destroyed
b. His house is destroyed

The source of the ambiguity is the suffix /-u/ which may mean “the house” or “his house” when added to the word /bet/ .It is serving as definite article or as a third person masculine marker.

Structural Ambiguity

Structural ambiguity is the most common type of ambiguity in Amharic which is caused because of sentences having more than one possible arrangements or syntax.

For example, yä-gojjam gäbs t'älla can be interpreted as:

a. beer made of barley from Gojjam

b. beer of barley from Gojjam

The ambiguity is because of / gäbs / ‘barely’ which can occur as head of the genitive noun phrase / yä-gojjam gäbs / or as complement of / t'älla / ‘beer’ in / gäbs t'älla / ‘barley beer’.

Referential Ambiguity

Referential ambiguity arises because of pronouns having many possible antecedents having different meanings.

For example, kasa stlä - tä - märräq -ä tädässät –ä can be interpreted as:

a. Kassa was pleased because he graduate
b. Kassa was pleased because he graduated
c. He was pleased because Kassa graduated

As we can see from the interpretations ‘he’ can refer to Kassa or another person.

Semantic Ambiguity

Semantic ambiguity is a type of ambiguity caused by words having different related and/or unrelated meanings. This are ambiguities caused by polysemic, idiomatic and metaphorical constitutes.

Polysemic constitutes: a word is polysemy if it has different senses but related in meaning.

For example, .mäbrat-u t'äff-a can be interpreted as:

a. The light went off
b. Mebratu (a person) disappeared

The ambiguity is because of / mäbratu/ which may refer to ‘the light’ or a male person with the name ‘Mebratu’.

Idiomaticconstitutes: when there are words that can be interpreted in different way from the literal meaning.

For example, bäre wâllãd-ã can be interpreted as:

a. unheard - of or impossible to happen
b. An ox gave birth to a calf

The ambiguity is because of / mäbratu/ which may refer to ‘the light’ or a male person with the name ‘Mebratu’.

Metaphoricalconstitutes: are due to usage of words to compare or symbolizeun-related subjects to represent some situation.

For example, aras näb+r can be interpreted as:

a. Irascible hot tempered
b. leopard with new-born cubs

Orthographic Ambiguity

This type of ambiguity is caused by lexical units having same orthography since the system doesn’t show distinctions between geminate and non-geminate sounds.

For example,መኪናውይሰራል‹mekinaw yseral›can be interpreted as:

a. The car works.
b. The car will be repaired.

The ambiguity is because of the word ‘ይሰራል’ ‹yseral›having the same orthographic form for both the active and passive voice.

2.7. Related Work

There are different related works done before on WSD for Amharic language. This section provides description of problems solved, approaches followed, results achieved and the way forward given by the studies.

Solomon(2010),carried out a research on WSDusing corpus based supervised machine learning approach to disambiguate five selected ambiguous Amharic words. Naive Bayes classifier was applied to classify a word to its correct sense on Weka 3.62 package. An English monolingual text corpus was used for acquisition of sense examples. A total of 1045 English sense examples for the five ambiguous words were collected from British National Corpus (BNC) and the sense examples are translated back to Amharic using dictionary. With total of 100 sentences acquired for each senses of ambiguous words the accuracy achieved was within the range of 70% to 83%.The challenges in this research were lack of Amharic language resources and lack of sense annotated data used for testing and training. And it is suggested that the development of linguistic resources like thesaurus and WordNet and standard sense annotated data could be better. And it is recommended to test unsupervised,knowledge based and other supervised approaches for the future.

Solomon(2011), further used corpus based unsupervised approach to disambiguate five similar words and data set that are used by Solomon(2010). The problem in Solomon(2011) is tried to be solved here by using unsupervised approach which doesn't require labeling of data.Five selected unsupervised clustering algorithms such as simple k-means, EM and agglomerative single, average and complete link clustering algorithms were evaluated on WEKA 3.7.9.The results showed that accuracy 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 where achieved. On the two works senses for ambiguous word in Amharic is translated to English and after acquiring English sentences for each sense it will be translated to Amharic. They have used this approach to minimize resource bottleneck of Amharic corpus but would be better if there was MT system but it is not available. So it consumes much time for preprocessing and translation.The main challenges faced here are the same as Solomon(2010).To overcome the lack of data it is recommended to test bootstrapping approach which requires little training data. And it is recommended to increase the number of ambiguous words covered andtest other approaches such as: knowledge based and hybrid.

Hagere(2013) used supervised machine learning approach as Solomon(2010) by adding three target words.Adaboost and Bagging ensemble classifiers algorithms that enable to create ensemble of mini-models which would involve in building the final model where employed. A total of 1770 sense examples were used and experiments were conducted on Weka 3.7.9 using five decision tree algorithms: DecisionStump, J48, RandomForest, RandomTree and REPTree as base classifiers. The performance of the ensemble algorithms is reported to increase when window size is set to two, i.e. 79.70 % for AdaBoost and 80.46% for Bagging and it is proved that RandomForest is the most effective base classifier. In this study all ambiguous words are considered to have only two senses and all word classes are not considered.The challenges faced here are the same as the above works.The forwarded recommendations are: to apply other ensemble and base classifiers, increase the size of the corpus, to consider more than two senses for a word as well as the number of ambiguous words to test.

Getahun(2014)used semi-supervised approach to disambiguate five ambiguous words that are different from words used by the previous researchers having a corpus of 1,031 sentences. Two clustering algorithms, expected maximization and k-means were employed to cluster sentences in to senses. And five classification algorithms (Adaboost, Bagging, ADtree, SMO and Naïve Bayes) were then experimented using python. The result showed that the average performance results of Adaboost, Bagging and ADtree algorithms are 84.90%, 81.25% and 88.45%.In this study the highest accuracy has achieved than the previous researches.The knowledge acquisition problems faced in the previous works are also challenges here. Finally, it is suggested to test other semi- supervised algorithms and other approaches by adding the number of ambiguous words to test.

Yehuwalashet (2016) used hybrid of unsupervised and rule based approaches for disambiguation of 20 most frequent Oromo words having a corpus containing thousands of Oromo sentences. He implemented partition and hierarchical clustering algorithms and manually crafted rules. Agglomerative and complete link algorithms from hierarchical and k-means and expectation maximization from partition clustering were implemented using Weka 3.7.9. The experiment had shown that the optimal window size is with two words. Unsupervised approach achieved 76.05% and hybrid achieved 89.47% accuracies. Also it is shown that expectation maximization and k-means partition clustering are better than hierarchical clustering. The main limitation of this work is the manual development of the rules which is prone to error, time taking, and difficult to cover many number words. Also investigation on hybrid of corpus and knowledge based approaches to have better result is recommended.

Segid(2015)attempted for knowledge based approach for disambiguation of Amharic all-words task having a WordNet containing thousands of Amharic words with their related words and glosses. He implemented overlap based algorithm and preprocessing tasks using python and Java programming. Two main experiments were done; the first was to evaluate the effect of WordNet with or without morphological analysis and achieved a performance of 55.5% and 80% respectively. And the second one was on determining the optimal window size and two-word window founded to be the optimal window. Although the use of the WordNet hierarchy is discussed how it is used in not shown in the experiments. And also there is no explanation about the word classes used and the system is expected to disambiguate all open class words in the running text since it is all-words task, but the experiments show that only one word is disambiguated.The challenges reported in the study were lack of linguistic resources like WordNet and lack of word searching software to collect texts with ambiguous words for testing.As a future work the development of thesaurus and MRD to test on WSD is recommended. In addition it is recommended to construct WordNet and ontology for Amharic to enhance performance of WSD and other NLP applications.

Recently, Dureti (2017)attempted to design a generic approach which is based on similarity measures towards WSD for all words. In this research 100 tagged example sentences for each sense of ambiguous words and WordNet composed of 17 ambiguous words from noun, verb, adjective and adverb word classes with their synonyms and gloss definition was developed and used as information source for the disambiguation. Cosine similarity and Jaccard Coefficient similarity measures were evaluated to measure similarity between the input sentence and tagged example sentences. To extract information from WordNet Lesk algorithm was employed with python. Experiments done to show the performance when the two knowledge sources are combined and the performance of two similarity measures combined with Lesk algorithm. Cosine similarity with Lesk resulted 86.69% and Jaccard Coefficient with Lesk resulted 89.83% which is the highest. The highest accuracy than the former researches is gained in this research and more gaps are filled. But this research considers only synonyms relationships between words and still number of words in the WordNetare limited. The disambiguation system can disambiguate only a word using only the glosses of the target word.The main challenges in this study were getting example sentences and manually tag the corpus. It is recommended to test other variants of Lesk algorithm which access a dictionary with senses arranged in a hierarchical order, considering not only glosses of the synset but also meanings of related words. To implement this it is recommended to fully construct WordNet containing different relationships between synsets. And it is forwarded to have a way to identify the ambiguous words and disambiguate more than one ambiguous words in a sentence rather than for a single word in a sentence.

Instead of using large amount of corpus which is time taking for preparation, training and testing,it is better to use knowledge based approach which requires less data and training compared to corpus based approaches.Also the knowledge sources used are general and standard. When we see Dureti's and Segid's work, which are knowledge based, both are claiming that they are using WordNet. But to say WordNet is used the use of synsets and other relations between synsets must be used, this is how WordNet differs from using a dictionary. In addition their task is all-words task but the system is able to disambiguate only one word from a sentence. This indicates that the disambiguation is limited on word level disambiguation and the sense selection is dependent only on the words in the WordNet, other words in the context of the sentence are not considered. Based on these gaps, in this study a WordNet based disambiguation system which uses sysnsets and different relationships between synsets is developed. The system automatically identifies all ambiguous words in a sentence and disambiguates them simultaneously at sentence level using glosses and synsets from the WordNet as well as the words in the context.

CHAPTER THREE

DESIGN OF THESTUDY

In this chapter the proposed architecture is discussed in brief. It describes how the data required for our source of knowledge, WordNet is collected, how this data is organized and processed to make suitable for our algorithm and our python program. The details of the algorithms implemented under each system component are discussed with examples. Finally the system evaluation measures used are presented.

3.1. System Architecture

In the current study we proposed architecture for designing word sense disambiguation for Amharic sentences containing all open class Amharic ambiguous words using WordNet.The proposed architecture is depicted in figure 3.1 below.

As shown in figure 3.1, first the system accepts input sentences from the test datasetcontaining ambiguous words. After the tokenization, normalization and stemming are done on theinput sentence respectively; the words which are ambiguous are automatically identified from the WordNet. Also single sense words which are available in the WordNet are identified and their glosses and synsets retrieved and used for simultaneous disambiguation of ambiguous words using augmented semantic space and context-gloss similarity measures.Then related synsets and context words are identified. The related synsets for the ambiguous words are retrieved with their definitions/glosses and synsets from the WordNet. For the context-to-gloss similarity the context words are all words in the context except the target words.In case of augmented semantic space context words are words in the context which exists in the WordNet but not ambiguous.The context-to-gloss overlap counts the frequency of context words from the combination of synsets and glosses of the target words and their related synsets. Then the sense combination having highest frequency count is chosen and senses of each word in that combination are selected as the right sense. But if the context-to-gloss fails to assign any sense the sentence is passed to augmented semantic space.We say a word is not disambiguated if it is assigned equal scores for all of its senses and a sentence is not disambiguated if all of the target words are not assigned sense.The augmented semantic space chooses sense by counting overlaps between the combination of senses containing glosses of the target words and their related synsets.

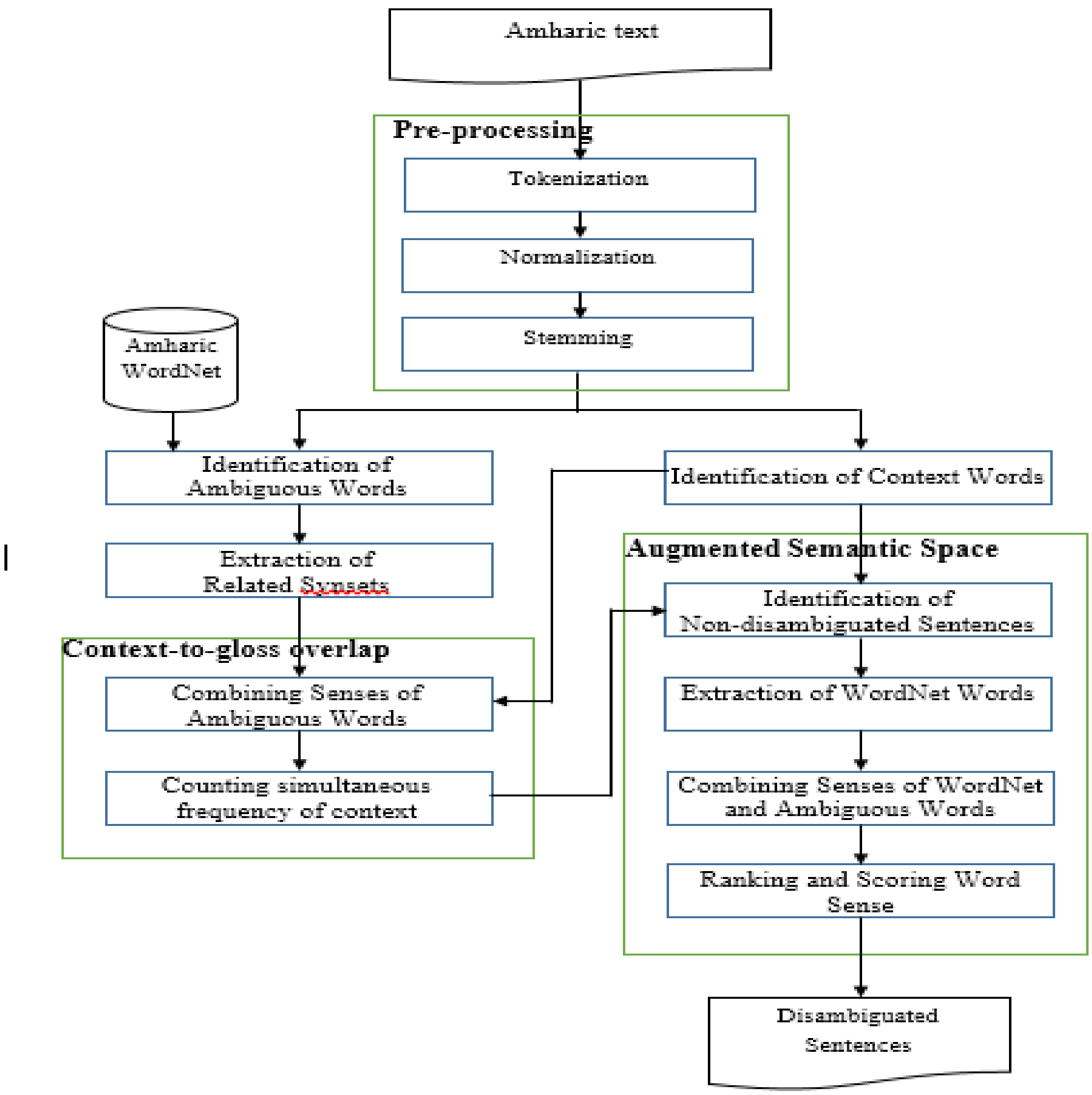


Figure 3. 1The Architecture for Amharic WSD

3.2. Preprocessing

To make our data suitable for experiment some text preprocessing tasks are done.These aretokenization,normalization and stemming.

3.2.1. Tokenization

The first step is tokenization, to split the given input to list of words or tokens. This is done on both the input sentences and the WordNet entries. It includes separation of individual words from the text and ignoring punctuation marks and return list of words. The tokenizer removes the Amharic punctuation marks like 'huletneTb' (:), 'aratneTb'(: :), 'deribserez' (፤), 'netelaserez'(፣), exclamation mark '!' and question mark'?' because they are not relevant for our disambiguation process. Tokenization is not implemented separately;it is includedwith the normalization.

3.2.2. Normalization

Normalization is making homophonous characters having the same pronunciation but different orthography, consistent .In Amharic the use of those characters interchangeably doesn't change the meaning. The characters are ኣandዐ, ጸandፀ, ሰand ሠandሀ, ሐ, and ኀthis includes first order to seventh order Amharic characters (e.g. ኣ and ዐ,ኡ and ዑ,ኢ and ዒ, ኣ and ዓ, ኤ and ዔ, እ and ዕ, ኦ and ዖ). For example, the Amharic word can be written asሃብል‹habl›, ሀብል ‹hebl›, ሐብል‹Hebl ›andኀብል‹ 'hebl› in which all are to mean "neckless".

1) *Open normalization. text file*
2) *Read each word w in the input text*
3) *nor =read each line in normalization.txt*
4) *for i in w:*
5) *for j in nor:*
6) *x=j.split( )*
7) *if iequals to_(x[0]):*
8) *Replace i by x[1]*

Algorithm 3. 1Amharic Variant characters Normalization

3.2.3. Stemming

Stemming is done to reduce different variants of a word having the same root form. Stemming helps our algorithm for not missing match of words having the same root but which are inflected. This includes prefix and suffix removal algorithms which works by first removing prefix, and then the suffix and reducing to root word, as shown in algorithm 3.2 below.

1) *Get input text*
2) *Open suffix.txt and prefix.txt*
3) *For word in input text*
4) *If word starts with prefix*
5) *Remove prefix*
6) *If word ends with suffix & not in exception list*
7) *Remove suffix*
8) *If word ends with sab'I*
9) *Replace by sadis*
10) *Return stemmedinput text*

Algorithm 3. 2 Stemming Amharic suffix and prefix

As shown in algorithm 3.2 first the prefix is removed using the rule based prefix removal algorithm suggested by Solomon (2010). Then the same rule based suffix removal is done using predefined suffixes.

The algorithm is working not only on removing the suffixes but after the suffix is removed from a word normalization is done based on a condition. For example: for the Amharic word "ሰዎች" ‹sewoc›after the suffix"ች"‹*c*› is removed"ሰዎ" ‹*sewo*› will be the root word, but the correct root is "ሰው"‹*sew*›.The algorithm handles this by changing the last character of the word from sab'I (seventh order) to sadis(six order) Amharic character if it is ending with seventh order after suffix removal. The infixes and some exceptional errors are done and corrected manually.

3.3. Identification of Ambiguous Words

A word is ambiguous if it is found in more than one synsets, each synset representing different meanings. The number of senses for a word represents the number of synsets in which it appears. From an input sentence the ambiguous words are identified by looking for multiple occurrences of the words from the WordNet (see algorithm 3.3). All words appearing more than one in the WordNet are considered as an ambiguous word and they are identified with their IDs in the WordNet.

For example, if we have a sentence *“ጨዋታእናየአካልብቃትእንቅስቃሴየልጆችንአእምሮእድገትያፋጥናል”‹Cewata Ina yea*kal bqat InqsqasE yeljocn a*Imro Idget yafaTnal›*after the sentence is preprocessed the ambigouse words are identified.*አካል‹a*kal›*,*ልጅ ‹lj›*,*እድገት‹Idget›*appears three, two and two times in the wordnet respectively.Therefore, their ids are rerived from the WordNet.*አካል ‹a*kal›*(11,12,13), *ልጅ‹lj›*(7,8) ,*እድገት‹Idget›* (3,4).Which shows *አካል‹a*kal›* has three senses, *ልጅ‹lj›*has two senses and *እድገት‹Idget›* has two senses.

```
1) ambW←array to store ambiguous words
2) ContextW← array to store words in the WordNet
3) For each word in a sentence
4)   If word in WordNet
5)      Store in ContextW
6) For each word in ContextW
7)      If word has [no unique id]in WordNet
8)     Append to ambW
9) Return ambW
```

Algorithm 3. 3Ambiguous words identifier

3.4. Simultaneous Disambiguation

From an input sentence all words having more than one meaning are selected and disambiguated simultaneously by collecting their synsets and related synsets. After the ambiguous words are identified with their associated senses all of them are disambiguated together at the same time by using combination of glosses. For each ambiguous word having multiple senses each sense is combined with each sense of other ambiguous words. Then sense is assigned to each ambiguous word by selecting the best candidate combination having highest score. Non ambiguous words from the input sentence which exists in the WordNet are addedin the combination to help in selecting the best sense in augmented semantic space, because as they have only one sense there is no need to assign sense for them.For example, if we have “ዘር”‹zer› and “ድካም” ‹dkam›in a sentence their ids are extracted from the WordNet and combined. Then their score is calculated simultaneously in each combination.

Id(ዘር) = [16, 17]

Id(ድካም) = [1, 2]

Product of ids= ([16, 1], [16, 2], [17, 1], [17, 2])

1) *Get list of ambiguous words ambW*
2) *IdS ← set of ids of ambiguous words*
3) *SenseComb ← array to store sense combinations*
4) *for word in ambW*
5) *Append each id(all senses for a word) from WordNet to set IdS*
6) *Calculate product of set IdS (create unique combinations of ids from different sets)*
7) *Append to SenseComb*
8) *Return SenseComb*

Algorithm 3. 4Simultaneous disambiguation

3.5. Sense Selection

Selection of senses for an ambiguous word is dependent on selection of senses for other ambiguous words in an input sentence.In augmented semantic spacethe glosses of non-ambiguous words have also impact on selection of senses for the ambiguous words. Even though there is no need to assign sense for them, their glosses are used to help the disambiguation process.

3.6. Related Synsets Extraction

The related synsets to take are dependent on the POS of the word. Even though there are a number of relationships in WordNet our WordNet includes some selected semantic relationships for each POS and lexical relationships are leaved. Semantic relationships are relationships between synsets but lexical relationships are between words, so as our algorithm is based on synsets we consider only semantic relationships. And also we have considered those relationships in a one level hierarchy and by assuming a synset to have 0 up to four related synsets to the upper and to the lower level per a relationship. This is because the inclusion of all the relationships takes much time and development of full WordNet by itself is not possible unless it is developed as a full project. We have selected the relationships which have more coverage in the English WordNet and used by different researches in order to make our WordNet information rich.The included relation namesfor each POS are summarized in table 3.1 below with examples.

Table 3. 1Relationships between synsets in the WordNet

Relationships	POS		Examples
Synonymy	All POSs	Similar meaning relationship	*ልፋት*‹lfat ›and *ጥረት‹Tret›* are synonyms
Hypernymy/ Hyponymy Hypernymy/ Troponymy	Noun Verb	Kind of relationship between nouns or verbs	/*ትግል*/‹*tgl*›is Hypernym of /*ልፋት፣ጥረት*/‹lfat,*Tret*› ; /*ሙከራ*/‹mukera›is Hyponym of /*ልፋት፣ጥረት*/‹lfat,*Tret* ›
Attribute	Noun, adjective	If an adjective synset is a value of a noun synset	/*ትኩስ፣ሙቅ*/‹tkus,muq›is attribute of / *ሙቀት፣ የሙቀትመጠን*/‹muqet,yemuqet meTen›and vice versa

Holonymy/meronymy	Noun	A noun synset is a part of another noun synset (holonym) and/or vice versa (meronym)	/እጅ/ ‹Ij›is holonym of / አካል፣ሰውነት፣ገላ/ ‹ a*kal, sewnet, gela›and /አካል፣ሰውነት፣ገላ/‹a*kal, säwənätə , gäla› is meronym of /እጅ/‹Ij›
Entailments	verb	Some action follows if some action happened	/ወሰነ፣ቆረጠ/ ‹wesene, qoreTe› and /አሰበ/ ‹a*sebe›
Causes	verb	If something causes another one	/ጠቀመ/ ‹Teqeme›and /ደረሰ፣በቃ/‹derese, beqa ›
Similar to	adjective	similar in meaning, but not close enough to be put together in the same synset	/ያልተፈተሸ፣ያልተሞከረ/‹yaltefetexe,yaltemokere› and /ትኩስ፣አዲስ፣ለጋ/‹tkus,a*dis ,lega›
Also_sees	adjective	a relation of relatedness between adjectives	/ወቅታዊ/ ‹wqtawi ›and /ትኩስ፣አዲስ፣ለጋ/‹tkus,a*dis ,lega›

Algorithm 3.5 below presents how our implementation attempts to identify and extract related synsets from the Amharic WordNet.

1) *relatedSyn* ← *nested array to store related synsets*
2) *ambIds* ← *array to store ids of ambigouse words*
3) *For each ambiguous word*
4) *Get ids from WordNet*
5) *Store ids in array ambIds*
6) *For each id in ambIds*
7) *While (hypernym,hyponym,troponym,meronym,holonym,attribute,causes,entailments,similar to [not empty])*
8) *Get synsets and definition*
9) *Store synsets and definitions in relatedSyn*
10) *Return relatedSyn*

Algorithm 3. 5Related synsets identification

3.7. Augmented Semantic Space

Augmented semantic space is one of the similarity measuring algorithms proposed by Satanjeev & Ted(2002) to solve the limitations of dictionary based Lesk algorithm and doubles the performance, which is implemented in this thesis. The algorithm uses our Amharic WordNet as a knowledge base,It uses the WordNet to identify the ambiguous words from the input sentence and disambiguate all of them simultaneously using their sense information and related synsets from the WordNet.

From the given input sentence all the WordNet words are considered for the disambiguation of one or more than one words. WordNet words means words appearing in our WordNet in one or more synsets. Algorithm 3.6uses the gloss, related synsets (hypernym/hyponym) and synonyms of those words to disambiguate all words having multiple senses in the WordNet.

Therefore,all the WordNet tokens from the input except the target words are considered as context window. After all the synsets and related synsets with their glosses in the WordNet containing words from the input sentence are retrieved the algorithm begins by counting number of overlaps between the glosses. Number of matches between two glosses is numbers of words in common. This is done for each sense of all multi sense words from the input sentence at the same time. Then after the overlap count (number of matches) is done, the best sense having maximum overlap is assigned for all ambiguous words.

```
1)  ambW ← Get array of ambiguous words from Amharic sentence
2)  ContextW← array to store context words
3)  ContextId ← array to store ids of context words
4)  AmbId ← nested array to store ids of ambiguous words
5)  For word in the sentence
6)      If word exist in WordNet and have [unique] id in WordNet
7)          Store to ContextW
8)  For word in ContextW
9)      Get id from WordNet
10)     Store in ContextId
11) For each word in ambW
12)     Get array of ids/senses/ from WordNet and append to AmbId
13) Make unique combinations of AmbId with ContextId
14) For each ids in a combination
15)     Get gloss, glosses of related synsets from WordNet
16)     Definitions=Concatenate(gloss, glosses of related synsets from WordNet)
17) For each combination
18)     CountOverlap between definitions
19)     If the length of words in an overlap [>] 1
20)         Add the square of the length to the overlap count
21) Assign the score to each combination in AmbId
22) Select combination with highest score
23) Return the definition/senses/ of ambiguous words with in the selected combination
```

Algorithm 3. 6Augmented semantic space

3.7.1. Ranking and Scoring word senses

Our algorithms works by giving ranks for senses and selecting the one having the top rank. The top ranked sense means the sense having highest score. As proposed in (Satanjeev & Ted, 2002)our ranking scheme uses Zipf's law which says rank is inversely proportional to the frequency of an event. Numbers of overlaps (matches between two glosses) are counted to give the score for a sense. But we have to give more credit for long sequence of matches as this occurs rarely and which shows high relatedness between two concepts. In such cases the score is given by the square of the length of matches in augmented semantic space. For example, if we get a single word match the score will be one but if we get consecutive match of n words we give a score of n^2.

For example, for the text*“ምላስአካል”‹mlasa*kal›,”*አካል”*‹a*kal›*having three senses and five related synsets and “*ምላሾ*’*‹mlas›*having only one sense:

*“አካል”‹a*kal›*has three senses

- *ሰውነት፣ገላ:እያንዳንዱ ሰውነት ክፍል; ‹sewnet , gela:Iyandandu sewnet kfl ›*
- *ክፍል:አንድንተግባርየሚፈፅም ቡድንአባል; ‹ kfl:a*ndntegbaryemife'Smbudn a*bal›*
- *ማህበር፣ጭፍራ: የሆነየአንድነገርክፍል*;*‹mahbr,Cfra:yehone ye a*nd neger kfl›*

*“ምላስ”‹məlas›*has one sense

- *ምላስ: ለመናገር ለመቅመስ ለመላስ የሚያገለግል የሰው እና የእንስሳት አፍ ውስጥ የሚገኝ የሰውነት አካል; ‹mlas: lemenager lemeqmes lemelas yemiyagelegl yesew Ina yeInssat a*f wst yemigeN yesewnet a*kal;›*

Related synsets for*“አካል”‹a*kal›*sense one፡

- **Hypernym** *፡ፍጡር:በምድር ላይ የሚኖር ሰው ወይም እንስሳ;‹fTur:bemdr lay yeminorsew weym Inssa›*
- **Hyponym** *፡ቅርጽ፣አቋም፣ስጋ:በህይወት ያለ የሰው ወይም እንስሳ የሰውነት አካል;‹qrS,a*qwam,sga:behywet yal yesew weym Inssa yesewnet a*kal›*
- **Meronym1** *፡እጅ:ከእጅ መዳፍ እስከ ጣት ድረስ ያለው የሰውነት ክፍል; ‹Ij:keIj medaf Iske Tat dres yalew yesewnet kfl›*
- **Meronym2** *፡እግር:ለመሄጃ ለመራመጃ ለመቆሚያ የሚያገለግል የሰውነት ክፍል;‹Igr:lemeheja lemerameja lemeqomiya yemiyagelegl yesewnet kfl›*
- **Meronym3** *፡ጭንቅላት: የሰው እና የእንስሳት ሰውነት የላይኛው ክፍል ጭንቅላት አናት አንጎልን የያዘው ክፍለ አካል;‹Cnqlat: yesew Ina yeInssat sewnet yelayNaw kfl Cnqlat a*nat a*ngoln yeyazew kfle a*kal›*

Table 3. 2Example for scoring

Sense1 (አካል‹	Gloss (አካል	Gloss (Hypernym)	Gloss (Hyponym)	Gloss (Meronym1)	Gloss (Meronym2)	Gloss (Meronym3)	Tot

*a*kal*›)	‹*a*kal*›)						
Gloss (ምላስ‹ *mlas*›)	1	4	4	1	1	6	17

The table shows comparisons of the gloss overlap between "ምላሽ"‹*mlas*› and the first sense of "አካል"‹*a*kal*›having one hypernym, one hyponym and three meronyms. The algorithm counts number of matches between glosses and squaring the number of exacts sequences of matches. For example in the glosses of "ምላሽ"‹*mlas*›and meronym3 of "አካል"‹*a*kal*› there are four common words, but the words "ሰው"‹*sew*›and *"እንስሳት"*‹*Inssat*›appears exactly the same sequence in the two glosses after the stop words are removed. Because of this the score will be the square of the length of the match(2^2=4) which is added with the two single word matches "ሰውነት"‹*sewnet*›and "አካል"‹*a*kal*›to give six. The total score for the comparisons is seventeen which is then to be compared with the scores of other senses.

Table 3. 3Example for sense selection

	Sense1(አካል‹*a*kal*›)	Sense2(አካል‹*a*kal*›)	Sense3(አካል‹*a*kal*›)
Gloss (ምላስ‹*mlas*›)	17	3	2

Table 3.4 shows the overlap scores for the three senses of "አካል"‹*a*kal*›with"ምላሽ"‹*mlas*›.As we can see from the score the correct sense of "አካል"‹*a*kal*›in the given context is sense 1 having the highest overlap. The scoring shows that using the squares of longest sequence of matches have highest importance showing the two concepts are highly related by having much difference between the correct sense and other senses.

3.8. Context-gloss Overlap

This algorithm makes use of the context in which the ambiguous word appears in the input text .The idea is based on concepts which co-occur together are related. From the input text words other than the ambiguous words excluding the stop words are considered as the context words. This algorithm works by finding the frequency of those words in the glosses and synsets of all related synsets of all the senses of the ambiguous word. Then the sense having higher number of frequency of context words is considered as the best score.

```
1)  ambW ←  Get array of ambiguous words from Amharic sentences
2)  ContextW← array to store context words
3)  AmbId←nested array to store ids of ambiguous words
4)  For each word in sentence
5)      If word not exist in ambW
6)      Append to ContextW
7)  For each word in ambW
8)      Get ids from WordNet and store in AmbId
9)  For each array of ids in AmbId
10)     Make unique combinations
11) For each id in combinations
12)     Get synsets,gloss, related synsets from wordNet
13)      Concatenate(synsets,gloss, related synsets)
14) For all ids within one combination
15)     Concatenate(synsets,gloss, related synsets) into one
16) For each concatenated definitions and synsets in each combination
17)     For word in ContextW
18)         If word exists in the combination
19)             Count++
20)     Weight=count
21) Select the combination with highest weight
22) Assign senses for ambiguous words from the combination having the highest weight
```

Algorithm 3. 7Context-to-gloss overlap

CHAPTER FOUR

EVALUATION AND DISCUSSION OF RESULTS

In this chapter the performance evaluations of the proposed architecture are discussed in brief. The results of the algorithms implemented under each system component are discussed under seven experiments. Finally the results are analyzed based on precision, recall, accuracy and F1-measure.

4.1. Test Datasets

At present time there is no standard sense tagged Amharic test dataset for WSD or related researches. The set of 305 sentences containing 17 ambiguous words from the previous researcher (the 17 ambiguous words can be found in Appendix I) for the experimentations are prepared manually with the help of linguistic experts.

The data set is partitioned into three datasets based on the number of ambiguous words in the sentences, containing one ambiguous word,two ambiguous words and three ambiguous words for evaluation. The number of ambiguous words considered is limited to three words per a sentence,since we have only 17 words and the probability of getting those words together is low as we have limited test sentences. But if we could find sentences having more than three ambiguous words the system can handle it.

There are a total of 170 sentences for the first partition containing 5 sentences for each ambiguous word.In the second partition there are 105 sentences which include at least 3 sentences for each sense of the ambiguous word. And the third partition contains 30 sentences each containing three ambiguous words. The number of ambiguous words in the second and third partitions is lower than the first because it is difficult to find the combinations of the selected words in one sentence.

The sentences are first preprocessed with the steps, tokenization, stop word detection to reduce the search space of context words, normalization and stemming discussed in chapter three. The preprocessed text is manually sense tagged to make performance evaluation simple and automatic. Sample test sentences are shown in table 4.1 below.

Table 4. 1Sample list of test sets ofAmharic sentences

Number of ambiguous words	Sentences
One ambiguous word	***ሀሰተኛ****የዲግሪእናዲፕሎማየትምህርትማስረጃየያዙተማሪዎችለህግቀረቡ* *‹**haseteNa** yedigri Ina diploma yetmhrt masreja yeyazu temariwoc lehg qerebu›*
	*በቅርብጊዜየተመረተውንየፋብሪካምርት**ወደፊት**ለገቢያለማቅረብእቅድተይዟል* *‹beqrb gize yetemeretewn yefabrika mrt**wedefit** legebeya lemaqrb Iqd teyzwal›*
	*በስብሰባውየቀረበውመፍትሄከጥቅሙጉዳቱስላመዘነለመተው**ተቆረጠ*** *‹besbsebaw yeqerebew mefthe ketqmu gudatu slamezene lemetew**teqoreTe**›*
	*ህዝቡየትንሳኤበአልንባህላዊእሴቱንጠብቆ**አከበረ*** *‹hzbu yetenesa bea*ln bahlawi IsEtun tebqo a***kebere**›*
	*ተፈላጊውንመስፈርትበማሟላቱከፍተኛደረጃላይ**ደረሰ*** *‹tefelagiwn mesfert bemamwalatu kefteNa dereja lay**derese**›*
Two ambiguous words	*የደረጃ**እድገት**መስፈርቱንየሚያሟሉተወዳዳሪዎችየፅሁፍእናየተግባር**ፈተና**ተፈተኑ* *‹yedereja I**dget** mesfertun yemiyamwalu tewedadariwoc ye’ShufIna yetegbarfetena tefetenu›*
	*የንግድሚኒስቴርባለስልጣንአዲስየቀረጥ**ዋጋቆረጠ*** *‹yengd ministEr balesltan a*dis yeqeret**wagaqoreTe**›*
	*የስንዴ**ዘርእድገት**አራትደረጃዎችአሉት* *‹yesndE**zer Idget** a*rat derejawoc a*lut›*
	*የሀገሪቱኢኮኖሚ**እድገት**እናልማት**በቀስታ**በማደግላይይገኛል* *‹yehageritu ikonomi Idget Ina lmat beqesta bemadeg lay ygeNal›*
	*ምርትበባህላዊ**መንገድ**በማምረትጥቅምላይለማዋልያደረጉትጥረትእና**ድካም**ውጤትአገኘ*

	*‹mrt bebahlawi **menged** bemamәret tqm lay lemawal yaderegut Tret Ina **dkam**wTEt a*geNe›*
Three ambiguous words	*በምርጫው**ወደፊት**የሚመሩመሪዎች**በድምፅ**ቆጠራተቆረጠ* *‹bemrcaw**wedefit** yemimeru meriwoc**bedm'S** qot'era teqorete›*
	*የሰንዴ**ዘር**ጥራትደረጃከፍተኛ**እድገት**በማሳየቱ**ዋጋው**ተወዳዳሪለመሆንደረሰ* *‹yesәnәdE**zer**Trat derejakefteNa**Idget**bemasayetu wagaw tewedadari lemehon derese›*
	ልጅነቱ**በሰጠው**ትኩስ**የወጣትነትጉልበትለህብረተሰቡአገልግሎትበመስጠቱለትልቅደረጃ**ደረሰ *‹**ljnetu** beseTw**tkus** yeweTatnet gulbet lehbretesebu a*gelglot bemesTetu letlq dereja **derese**›*
	*ጨዋታእና**የአካል**ብቃትእንቅስቃሴ**የልጆችን**አእምሮ**እድገት**ያፋጥናል* *‹Cewata Ina **yea*kal** bqat InqsqasE**yeljocn** a*Imro **Idget** yafaTnal›*
	*ህዝቡ**ለደረሰበትፈተና**እናችግርስልጣንየተሰጠው**አካል**መፍትሄለመስጠትእየሰራነው* *‹hzbu **lederesebetfetena** Ina cgr slTan yeteseTew a*kal mefthe lemesTet Iyesera new›*

4.2. Data Collection for Amharic WordNet

Our knowledge base for the disambiguation is WordNet which is manually developed in this research using the idea of Princeton English WordNet(Fellbaum, 1998). The aim of using WordNet is to make use of its hierarchical structure which shows relationship between synsets of different POSs. But for Amharic we couldn't get such well-constructed WordNet and any dictionary including semantic relations required by our algorithm. We are able to get an Amharic dictionary having synsets and their glosses(አማርኛመዝገበቃላት, 1993 E.C).

Though the previous researchers Samrawit(2014)Dureti(2017)Segid(2015)tried to construct WordNet they didn't include those relationships other than synonymy and we couldn't find such linguistic resources. So as it is challenging to get those relationships for our construction of WordNet, we used an approach which is used to construct WordNet for low resource languages from existing WordNets.

There are two approaches that are used to construct WordNet for low resource languages called extended and merge approaches (Nurril, Suerya, & Francis, 2011). Extended approach translates the synsets in the Princeton WordNet to the target language, take over the relations from Princeton and revise.And the merge approach defines synsets and relations in the target language and then aligns with the Princeton WordNet using equivalence relations. In the extended approach the Princeton WordNet is used on each step from the beginning but in the merge approach it is checked after an independent WordNet is created in other language.

In (Tessema, Meron, & Teshome, 2008) it is recommendedto use extended approach for under resourced language like Amharic because it minimizes time and cost to construct it from scratch and it will be easy to incorporate it with other WordNets. Accordingly, in this study we follow the extended approach.

As Princeton WordNet is the benchmark for the development of WordNet in other languages, we used it to manually construct our Amharic WordNet using the extended approach.

Our WordNet construction method includes the following steps in short. All the steps are done with the help of linguistic experts and lexicool online dictionary containing three Amharic-English dictionaries and Amharic-English Google translate.

1. Identify the Amharic ambiguous words to be included with their synonyms, senses and glosses for each sense from አማርኛመዝገበቃላት(1993 E.C).
2. For each synset translate its synonyms and gloss to English and search for the gloss and sysnset in the English WordNet, If we have more than one English translations for a word , among the synsets select the one in which its gloss is related in meaning with its Amharic gloss.
3. After we identify which synset to take, get the related synsets with that particular POS.
4. For each translated synsets translate the synonyms (the words in the synset) to Amharic and also the glosses.

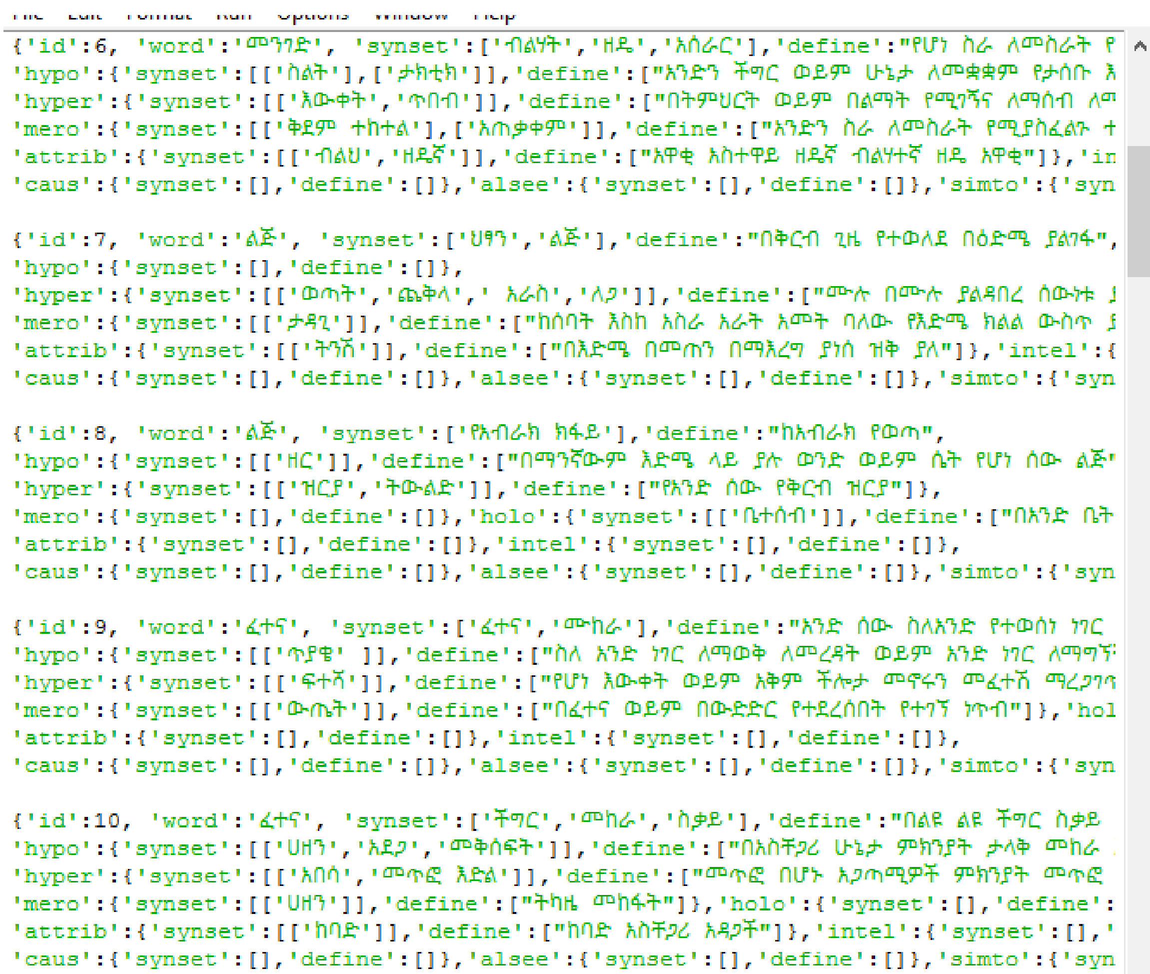
For example, for the Amharic noun “*ድካም*”‹dkam›we considered two senses:”*አንድንነገርለማግኘት፣ለአላማመድረስመስራት*” ‹a*ndn neger lemagNet lea*lama medres mesrat›with synset /*ልፋት፣ጥረት፣መታገል*/‹lfat, Tret, metagel›and “*በስራመውጣትመውረድ፣ዕረፍትማጣት*” ‹besra mewTat mewred, ʾIreft maTat› with synset /*ሀይልማጣት*/ ‹hayl maTat›, respectively.First, “*ድካም*”‹dkam›has five translations in English. From the translations we select “effort” and “powerlessness” which fits to our first and second sense respectively. Then when we look for the synsets containing “effort” in the English WordNet we get 4 synsets which shows the English word by itself is ambiguous. So after checking glosses of each synset, by translating to Amharic we select which synset to consider for each sense. Then we take the Amharic translations of synonyms, glosses and related synsets of those synsets.

WordNet structure

The WordNet follows the structure of Princeton WordNet by using synsets as the main building blocks and their semantic relationships. Words are organized in set of synonyms called synsets containing words having the same meanings, even can used interchangeably without changing the meaning of a sentence, For example, the words /*ተመን፣መጠን፣ልክ፣ዋጋ*/‹temen,meTen, lk, waga›are in the same synset having gloss “*አንድነገርለተገዛበትለተሸጠበትለተሰራስራለተሰጠአገልግሎትየሚከፈልየገንዘብመጠኝ*” ‹a*ndneger letegezabet letexeTebet letesera sra leteseTe a*gelglot mikefel yegenzeb meTen›.Each synset has one gloss and different related synsets connected by different types of relationships specific for each POS. The above synset in the example is a noun synset having five related synsets,which are hyponym, hypernym, meronym, holonym and attributes.These related synsets have their own synonym sets and glosses. For verbs hypernym/hyponym, entailment and cause relationships are included. As well as for adverbs also-sees relationship is included, and for adjectives attributes relation which is the only cross POS relationship between adjective and noun synsets, also-sees and similar to relationships are included.

The WordNet is organized in such a simple way for our python program to access. After the file is preprocessed, it is stored in a python file containing nested python dictionaries and lists.Every synset has an identifier (ID), a representative word, synonym set, gloss and related synsets with their glosses. For one particular relationship a synset may not have any related synset or may have more than one related synsets.In the first case,the entry for that relationship is left empty,while in the second case nested dictionaries of synsets and glosses are used. The figure 4.1 shows some sample data from the WordNet before it is preprocessed.

Following normalization stop wordsremoval is applied on the WordNet. Stop words are words which occur frequently in a language but are less or even have no relevance on the meaning of the text. This has to be filtered out before doing any processing and text analysis. Most of the time stop words in Amharic are conjunctions, articles and prepositions. Our algorithm filters them from the WordNet by using the predefined stop words list. The lists of sample stop words are presented in appendix II.



```
{'id':6, 'word':'መንገድ', 'synset':['ብልሃት','ዘዴ','አሰራር'],'define':"የሆነ ስራ ለመስራት የ
'hypo':{'synset':[['ስልት'],['ታክቲክ']],'define':["አንድን ችግር ወይም ሁኔታ ለመቋቋም የታሰቡ እ
'hyper':{'synset':[['እውቀት','ጥበብ']],'define':["በትምህርት ወይም በልማት የሚገኝና ለማሰብ ለ
'mero':{'synset':[['ቅደም ተከተል'],['አጠቃቀም']],'define':["አንድን ስራ ለመስራት የሚያስፈልጉ ተ
'attrib':{'synset':[['ብልሀ','ዘዴኛ']],'define':["አዋቂ አስተዋይ ዘዴኛ ብልሃተኛ ዘዴ አዋቂ"]},'in
'caus':{'synset':[],'define':[]},'alsee':{'synset':[],'define':[]},'simto':{'syn

{'id':7, 'word':'ልጅ', 'synset':['ህፃን','ልጅ'],'define':"በቅርብ ጊዜ የተወለደ በዕድሜ ያልገፋ",
'hypo':{'synset':[],'define':[]},
'hyper':{'synset':[['ወጣት','ጨቅላ',' አራስ','ለጋ']],'define':["ሙሉ በሙሉ ያልዳበረ ሰውነቱ
'mero':{'synset':[['ታዳጊ']],'define':["ከሰባት እስከ አስራ አራት አመት ባለው የእድሜ ክልል ውስጥ
'attrib':{'synset':[['ትንሽ']],'define':["በእድሜ በመጠን በማእረግ ያነሰ ዝቅ ያለ"]},'intel':{
'caus':{'synset':[],'define':[]},'alsee':{'synset':[],'define':[]},'simto':{'syn

{'id':8, 'word':'ልጅ', 'synset':['የአብራክ ክፋይ'],'define':"ከአብራክ የወጣ",
'hypo':{'synset':[['ዘር']],'define':["በማንኛውም እድሜ ላይ ያሉ ወንድ ወይም ሴት የሆነ ሰው ልጅ'
'hyper':{'synset':[['ዝርያ','ትውልድ']],'define':["የአንድ ሰው የቅርብ ዝርያ"]},
'mero':{'synset':[],'define':[]},'holo':{'synset':[['ቤተሰብ']],'define':["በአንድ ቤት
'attrib':{'synset':[],'define':[]},'intel':{'synset':[],'define':[]},
'caus':{'synset':[],'define':[]},'alsee':{'synset':[],'define':[]},'simto':{'syn

{'id':9, 'word':'ፈተና', 'synset':['ፈተና','ሙከራ'],'define':"አንድ ሰው ስለአንድ የተወሰነ ነገር
'hypo':{'synset':[['ጥያቄ' ]],'define':["ስለ አንድ ነገር ለማወቅ ለመረዳት ወይም አንድ ነገር ለማግኘ
'hyper':{'synset':[['ፍተሻ']],'define':["የሆነ እውቀት ወይም አቅም ችሎታ መኖሩን መፈተሽ ማረጋገ
'mero':{'synset':[['ውጤት']],'define':["በፈተና ወይም በውድድር የተደረሰበት የተገኘ ነጥብ"]},'hol
'attrib':{'synset':[],'define':[]},'intel':{'synset':[],'define':[]},
'caus':{'synset':[],'define':[]},'alsee':{'synset':[],'define':[]},'simto':{'syn

{'id':10, 'word':'ፈተና', 'synset':['ችግር','መከራ','ስቃይ'],'define':"በልዩ ልዩ ችግር ስቃይ
'hypo':{'synset':[['ሀዘን','አደጋ','መቅሰፍት']],'define':["በአስቸጋሪ ሁኔታ ምክንያት ታላቅ መከራ
'hyper':{'synset':[['አበሳ','መጥፎ እድል']],'define':["መጥፎ በሆኑ አጋጣሚዎች ምክንያት መጥፎ
'mero':{'synset':[['ሀዘን']],'define':["ትካዜ መከፋት"]},'holo':{'synset':[],'define':
'attrib':{'synset':[['ከባድ']],'define':["ከባድ አስቸጋሪ አዳጋች"]},'intel':{'synset':[],'
'caus':{'synset':[],'define':[]},'alsee':{'synset':[],'define':[]},'simto':{'syn
```

Figure 4. 1Sample WordNet file entries

4.1. Implementation

The algorithms are implemented using Python 3.6.5.To make the system open for user interaction a graphical user interface prototype is developed and integrated with the implemented algorithms. Using the interface, user can insert a sentence containing one or more ambiguous words. Also can upload a file containing ambiguous sentences using the file upload button on the top left side of the interface. Then the system identifies the ambiguous words from the input after preprocessing steps are done. Finally the system displays the disambiguated sentences and for a file uploaded from the user it writes the disambiguated sentences to a file selected by the user.The user interface and its output for a given sentence is depicted in Figure 4.1 below.

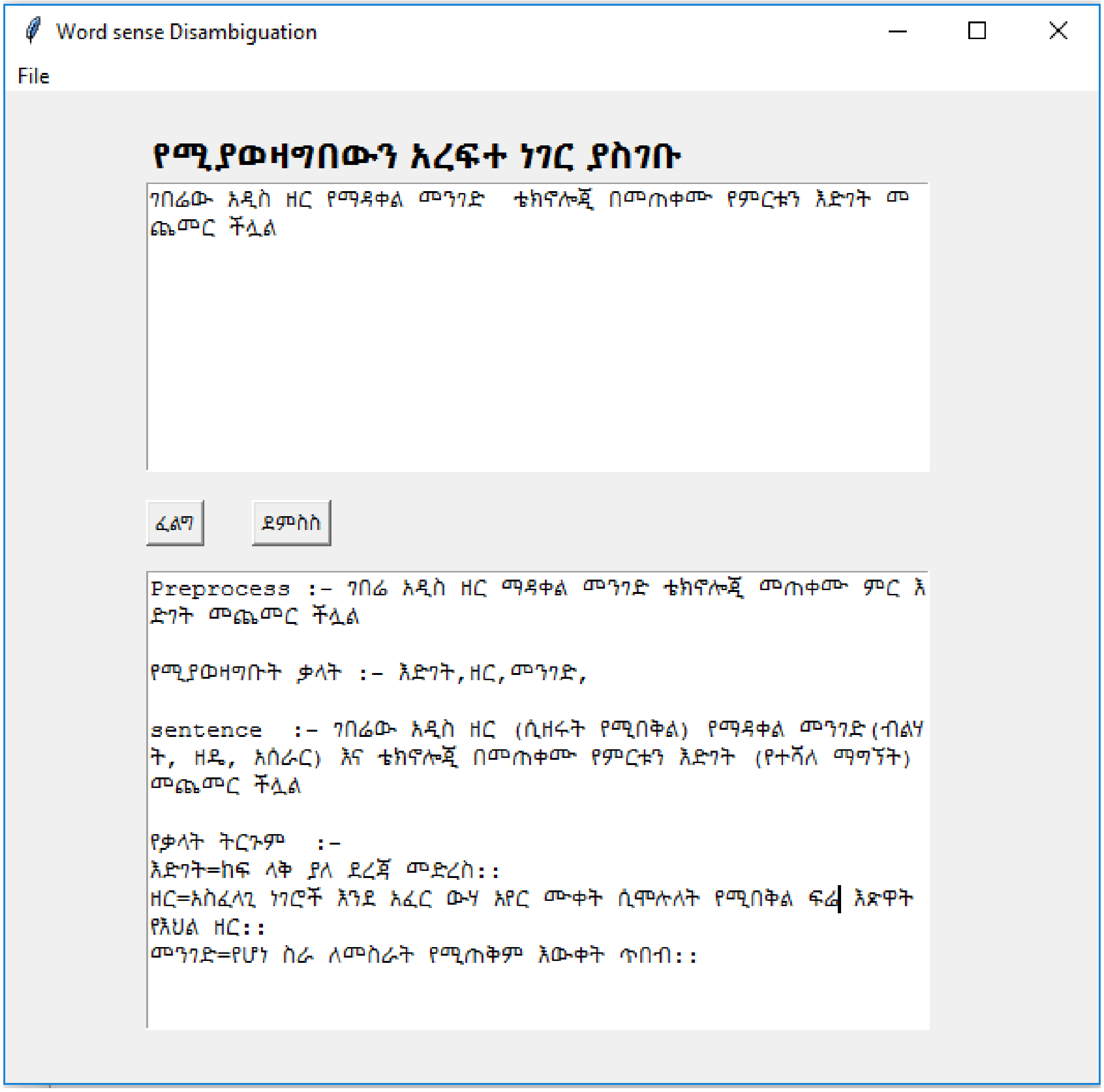


Figure 4. 2. Graphical User Interface

4.2. Evaluation of the Amharic WSD

Atotal of seven experiments were done to evaluate the performance of the developed system. The main objective of the evaluations is to measure how the implemented algorithms perform as the numbers of ambiguous words in a sentence are increasing. The performance is measured both for word-wise and sentence-wise disambiguation. Word-wise means how many ambiguous words in a sentence are disambiguated correctly and sentence-wise means the disambiguation of all of the ambiguous words (check if all of the ambiguous words are disambiguated) in a sentence as a whole.

4.3. PerformanceEvaluation

The major objective of WSD is enhancing the performance of other NLP applications like IR, MT, QA, etc. But as discussed in Trevor(2003)the evaluation metrics should fulfill some criteria. The first one is, when disambiguating between three or more sentences the negative predictions should penalized based on the distance from the correct sense (not only are the positive predictions needed).And they shall allow performance ranking between two or more classifiers. Their performance also should be able to be compared to the base line performance (a method which predicts the most frequent sense) then; finally the results obtained should be interpretable.

The widely used evaluation measures for WSD are precision and recall which are taken from the field of information retrieval (IR)(Roberto, 2009). Even if they have some limitations they are widely used to make comparisons between different WSD systems.

Precision, P is a measure of the percentage of correct answers given by the system, which is calculated as:

$$\mathrm{P} = \frac{\text{Number of correct answers provided}}{\text{Number of answers provided}} \quad (4.1)$$

Recall, R measures the number of correct answers given by the system over the total number of answers expected to be given,

$$\mathrm{R} = \frac{\text{Number of correct answers provided}}{\text{Number of total answers to provided}} \quad (4.2)$$

In this study accuracy is with the same value as recall. Because, the aim of this study is to know how many of the sentences in the test set are correctly disambiguated.

After computing recall and precision, F1-measure (or balanced F-score)is computed to determine the weighted harmonic mean of precision and recall and is calculated as follows:

$$\mathrm{F1} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}} \tag{4.3}$$

The evaluations are performed to test how the performance of disambiguation changes as the number of ambiguous words in the input sentence is increasing. The performance is checked using the implemented algorithms individually and using the combination of them one after the other. Also the performance is checked using their combination together at the same time. All evaluationsare based on precision and recall. Our performance evaluation is done automatically using python script. Python program checks both word-wise and sentence-wise performance of the algorithms from each input sentence and report the result of all test sentences individually and the average of all at once.

The results for the experiments are reported based on both word-wise and sentence-wise precision, recall, accuracy and F1 measure of one, two and three ambiguous wordsin a sentence containing a mixture of ambiguous words from different word classes.For sentences, containing one ambiguous word the performance for both word-wise and sentence-wise is the same in all experiments. This is because as we have only one target word whether the sentence is correctly disambiguated or not is determined by the disambiguation of the target word applied to.

Experiment 1: Applying context to gloss overlap without stemming

This experiment is done to check the performance of context to gloss overlap without stemming words. We show hereunder howit works with example. For example, to disambiguate a sentence "*የደረጃእድገትመስፈርቶችንያሟሉተወዳዳሪዎችየጽሁፍእና የተግባርፈተናተፈተኑ*"*‹yederejaIdgetmesfertocnyamwalutewedadariwocyeShufInayetegbarfetenatefetenu›.*

First the sentence passes through the preprocessing steps to generate the context words excluding the ambiguous words are extracted.{*የደረጃ,መስፈርቶችን,ያሟሉ,ተወዳዳሪዎችየፅሁፍ*

,እና,የተግባር,ተፈተኑ}‹ {*yedereja, mesfertocn, yamwalu, tewedadariwoc, ye'Shuf,Ina, yetegbar, tefetenu*} ›.

The disambiguation starts by getting IDs of the words in the WordNet (IDs represent the identifier of a synset, a word is identified as ambiguous if it exist in synsets having more than one ID) .After extracting the words having more than one IDs ,combinations of the IDs is computed to retrieve definitions associated with those IDs for frequency count. In our example we have four different combinations of senses.

For the ambiguous words:"*እድገት*"‹*Idget*›and"*ፈተና*"‹*fetena*›, the meaning of them and related synsets extracted from the WordNet as follows:

The following are the different senses of the word "እድገት"‹*Idget*›***:***

Sense 1("*እድገት*"‹*Idget*›):*የአካልክፍልመጨመር ‹yea*kal kfl meCemer›*

- Definition:"*የሰውነትክፍልቀስበቀስከትንሽወደትልቅመለወጥማደግጥንካሬ*" ‹*yesewnet kfl qes beqes ketnx wede tlq melewet madeg Tnkare*›
- Hyponym(definition):"*የአጥንትየአእምሮየጡንቻቁመትየጥርስመጠንመጨመር*"‹*ye*aTnt yea*əImro yeTunca qumet yeTrs meTen meCemer*›
    - Hyponym(synset):{መጨመር‹*meCemer*› }
- Hypernym(definition):"*ሀይወትባላቸውፍጥረታትላይየሚከሰትሂደት*"‹*hywet balacew fTretat lay yemikeset hidet*›
    - Hypernym (synset):{የተፈጥሮሂደት‹*yetefeTro hidet*›}
- Attribute(definition):"*በመጠንበቁመትበእርዝመትበእድሜከፍያለ*"‹*bemeTen bequmet beIrzmet beIdmE kef yale*›
    - Attribute (synset):{ትልቅ‹*tlq*›}

Sense 2("*እድገት*"‹*Idget*›): *የተሻለማግኘት*‹*yetexalemagNet*›

- Definition: "*ከፍ ላቅ ያለ ደረጃ መድረስ*"‹ *kef laq yale dereja medres*›
- Hyponym(definition):"*አዲስየተለየየተሻለሁኔታአሰራርልዩነትለውጥ*" ‹*a*dis yeteleye yatexale huneta a*serar lyunet lewT*›

- Hyponym(synset):{ለውጥ‹*lewT*› }

- Hypernym(definition*):"ኢኮኖሚያዊእናማህበራዊእድገትብልፅግናግንባታ*"‹*ikonomiyawi Ina mahberawi Idget bl'Sgna gnbata*›
    - Hypernym (synset):{'ልማት','ብልፅግና' ‹*'lmat'*,'*bl'Sgna* '›}
- Holonym(definition):"*እድገትዘመናዊነትከፍተኛየእውቀትወይምየመሻሻልደረጃ*"‹*Idget zemenawinet kefteNa yeIwqet weym yeme'Sa'Sal dereja*›
    - Holonym (synset):{ስልጣኔ‹*slTanE*› }

In the same way, the different senses of the word "*ፈተና*"‹*fetena*› are extracted from the WordNet:

Sense 1("*ፈተና*"‹*fetena*›):*ሙከራ*‹*mukera*›

- Definition: "*አንድ ሰው ስለአንድ የተወሰነ ነገር ማወቅ አለማወቁን ለመገንዘብ በተግባር በፅሁፍ ወይም በቃል እንዲያሳይ የሚቀርብ ጥያቄ*"‹*a*nd sew slea*nd yetewesene neger maweq a*lemawequn lemegenzeb betegbar be'Shuf weym beqal Indiyasay yemiqerb TyaqE*›
- Hyponym (definition):"*ስለአንድነገርለማወቅለመረዳትወይምአንድነገርለማግኘትበመፈለግበቃልወይምበተግባርየሚቀርብመልስየሚሻሃሣብ*' ‹*sle a*nd neger lemaweq lemeredat weyma*nd neger lemagNet bemefeleg beqal weym betgbar yemiqerb melsyemixa hasab*›
    - Hyponym(synset):{ጥያቄ‹*TyaqE*›}
- Hypernym (definition):"*የሆነእውቀትወይምአቅምችሎታመኖሩንመፈተሽማረጋገጥ*"‹*yehone IwqeT weym a*qm clota menorun mefetex maregageT*›
    - Hypernym (synset):{ፍተሻ‹*ftexa*›}
- Meronym (definition):"*በፈተናወይምበውድድርየተደረሰበትየተገኘነጥብ*'‹befetena weym bewddr yetederesebet yetegeNe neTb›
    - Meronym (synset):{ውጤት‹*wTEt*›}

Sense 2("*ፈተና*"‹*fetena*›*):ችግር*‹*cgr*›

- Definition: *"በልዩልዩችግርስቃይሲያጋጥም"*‹*belyu lyu cgr sqay siyagaTm*›

- Hyponym(definition):"*በአስቸጋሪሁኔታምክንያትታላቅመከራእናጭንቀትነው*" ‹*bea*scegari huneta mknyat talaq mekera Ina cnqet new*›
    - Hyponym(synset):{'ሀዘን','አደጋ','*መቅሰፍት*'‹ '*hazen*','*a*dega*,'*meqseft*'›}
- Hypernym(definition):"*መጥፎበሆኑአጋጣሚዎችምክንያትመጥፎአጋጣሚዎችሲመጡሲከሰት*"‹*me Tfobehonua*gaTamiwocmknyat meTfoa*gaTamiwocsimeTusikeset*›
    - Hypernym (synset):{'አበሳ','*መጥፎእድል*'‹'a*besa','*meTfoIdl*'›}
- Meronym(definition):"*ትካዜ መከፋት*" ‹*tkazE mekefat*›
    - Meronym (synset):{ሀዘን‹*hazen*›}
- Attribute(definition):"*ከባድአስቸጋሪአዳጋች*"‹*kebad a*scegari a*dagac*›
    - Attribute (synset):{ከባድ‹*kebad*›}

Then frequencies of the words in each combination of the glosses and synsets of the ambiguous words and their related synsets are calculated.

- Context words= {*የደረጃ*‹yedereja›,*መስፈርቶችን*‹mesfertocn›,,*ያሟሉ*‹yamwalu›, *ተወዳዳሪዎች*‹*tewedadariwoc*›, *የፅሁፍ*‹*ye'Shuf*›, *ተወዳዳሪ*‹*tewedadari*›,*ተፈተኑ* ‹*tefetenu*›}
- frequency(context words,c(sense2(*ኢድገት*‹*Idget*›,), sense2(*ፈተና*‹fätäna›)))=0
- frequency(context words,c(sense1(*ኢድገት*‹*Idget*›), sense1(*ፈተና*‹fätäna›)))=0
- frequency(context words,c(sense2(*ኢድገት*‹*Idget*›), sense1(*ፈተና*‹fätäna›)))=0
- frequency(context words,c(sense1(*ኢድገት*‹*Idget*›), sense2(*ፈተና*‹fätäna›)))=0

As we can see all combinations have equal score, 0; because the algorithm is unable to find any of the context words in the glosses and synsets of the ambiguous words and their related synsets .So the two words are not assigned any sense.

Table 4. 2Performance of context to gloss overlap without stemmer algorithm

	One ambiguous word	Twoambiguous words	Three ambiguous words	Average

	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise
Precision	0.57	0.57	0.51	0.19	0.52	0.11	0.53	0.29
Recall	0.47	0.47	0.33	0.12	0.19	0.04	0.33	0.21
Accuracy	47%	47%	33%	12%	19%	4%	33%	21%
F1 measure	0.57	0.57	0.4	0.15	0.27	0.05		

From the above example and the result in table 4.2 we can conclude that we have to apply stemming on the WordNet and on the test sentences. This will increase the probability of getting matches between texts by reducing variants of a word to its stem. In our example above "*በፅሁፍ*"‹*be'Shuf*›and *"የፅሁፍ"*‹*ye'Shuf*›are taken as different words but if stemming is applied they will be reduced to the same root "*ፅሁፍ*"‹*'Shuf*›.Because of this we decided to apply stemming on the WordNet and on the test sentences for the rest of our experiments. The next experiment shows the application of stemmer on context gloss overlap.

Experiment 2: Applying context to gloss overlap with stemmer

This experiment is done to evaluate the performance of context to gloss overlap with stemmer algorithm. It works by calculating frequency of words in the context of the sentence from the gloss definition of the ambiguous words, glosses and synsets of the ambiguous word and its related synsets.

For the above example in Experiment 1, the context words excluding the ambiguous words are extracted and preprocessing is done including stemming. Then frequencies of the words in each combination of the glosses and synsets of the ambiguous words are calculated.

- Context words= {*የደረጃ*‹yedereja›, *መስፈርቶችን*‹mesfertocn›, *ያሟሉ*‹yamwalu›, *ተወዳዳሪዎች*‹*tewedadariwoc*›,*የፅሁፍ*‹ye'Shuf›,*ተወዳዳሪ* ‹*tewedadari*›,*ተፈተኑ* ‹*tefetenu*›}
- frequency(context words,c(sense2(*እድገት*‹*Idget*›), sense2(*ፈተና*‹*fetena*›)))=2
- frequency(context words,c(sense1(*እድገት*‹*Idget*›), sense1(*ፈተና*‹*fetena*›)))=2

- frequency(context words,c(sense2(*ኢድገት‹Idget›*), sense1(*ፈተና‹fetena›*)))=4
- frequency(context words,c(sense1(*ኢድገት‹Idget›*), sense2(*ፈተና‹fetena›*)))=0

As we can see the sense combination having the highest score is the combination containing second sense of “*ኢድገች*”*‹Idget›*and first sense of “*ፈተና*”*‹fetena›*. The two words are assigned the sense having the highest score.

Experimental result of context to gloss overlap for one, two and three ambiguous words in a sentence are summarized in table 4.3 below.

Table 4. 3Performance of context to gloss overlap with stemmer algorithm

	One ambiguous word		Two ambiguous words		Three ambiguous words		Average	
	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise
Precision	0.89	0.89	0.93	0.84	1.0	1.0	0.94	0.91
Recall	0.70	0.70	0.50	0.47	0.31	0.31	0.5	0.49
Accuracy	70%	70%	50%	47%	31%	31%	50%	49%
F1 measure	0.78	0.78	0.65	0.6	0.5	0.5		

The result shows that the precision for the word-wise is increasing as the number of target words are increasing. But the sentence-wise precision is decreasing because if one of the target words in a sentence assigned wrong sense the whole sentence is assigned wrong sense.

Both the word-wise and sentence-wise recall are decreasing and the gap between recall and precision is increasing. This shows that the system is covering very small number of ambiguous words in sentences but giving the correct answers for those it is able to give answers there by precision is increasing. This method is based on context words excluding the target words. So, as the number of target words is increasing the context words in which the frequencies that are checked has been less in number which makes the word wise and sentence wise recall low for three and two target words.

This algorithm is unable to disambiguate some of the sentences, because it is dependent on the number of context words. As we can see its performance is decreasing for sentences having more ambiguous words which indicate less number of context words.

Experiment 3: Applying augmented semantic space

In this experiment, we have evaluated the use augmented semantic space which works by comparing the gloss definitions of the ambiguous words and their related synsets with glosses of the context words found in the WordNet. For example, to disambiguate the sentence in experiment one and two "*የደረጃእድገትመስፈርቶችንያሟሉተወዳዳሪዎችየጽሁፍእና የተግባርፈተናተፈተኑ*"*‹yededejaIdget mesfertocn yamwalu tewedadariwoc yeShuf Ina yetegbarfetena tefetenu›*

First the sentence passes through the preprocessing steps to generate bag of words:*"ደረጃእድገትመስፈርትያሟላተወዳዳሪፅሁፍተግባርፈተናፈተነ"‹derejaIdgetmesfert yamwala tewedadari 'Shuf tegbar fetena feten›*

The ambiguous words are identified in the same way discussed in experiment 1.For the ambiguous words:"*እድገት*"*‹Idget›* and"*ፈተና*"*‹fetena›*, the meaning of them and related synsets extracted from the WordNet as it is shown in experiment one. In addition, the glosses of WordNet words in the context are extracted as follows:

WordNet words (context words): *"ደረጃ"‹dereja› and "መስፈርት"‹mesfert›*

- *"ደረጃ"‹dereja›: "አቅምችሎታየሶስተኛውአለምህዝብየኑሮደረጃዝቅተኛነው"‹a*qm clota yesosteNaw a*lem hzb yenuro dereja zəqətäña näwə›*
- *"መስፈርት"‹mesfert›: "መሟላትያለበትአስፈላጊነጥብመለኪያእዚህመስሪያቤትለመቀጠርየምታሟላቸ ውመስፈርቶችአሉ"‹memwalat yalebet a*sfelagi neTb melekiya Izih mesriya bEt lemeqeTer yemtamwalacew mesfertoc a*lu›*

Combinations of the IDs are computed to retrieve definitions associated with those IDs for overlap detection. In our example we have four different combinations of senses which includes definitions of the non-ambiguous words found in the WordNet.

C (sense2*(እድገት‹Idget›),* definition*(ደረጃ‹dereja›),*definition*(መስፈርት‹mesfert›),*sense1*(ፈተና‹fetena›))*

C (sense1*(እድገት‹Idget›),*definition*(ደረጃ‹dereja›),*definition*(መስፈርት‹mesfert›),*sense2*(ፈተና‹fetena›))*

C (sense1*(እድገት‹Idget›),*definition(*ደረጃ‹dereja›*),definition*(መስፈርት‹mesfert›),*sense1 *(ፈተና‹fetena›))*

C (sense2*(እድገት‹Idget›)*,definition*(ደረጃ‹dereja›),*definition*(መስፈርት‹mesfert›),*sense2*(ፈተና‹fetena›))*

From each combination of senses mentioned above, we get six different combinations which are intersected and the number of overlaps added to get the overall score under each combination. From the first combination intersection between glosses is calculated and added:

- sense2*(እድገት‹Idget›)*∩sense1*(ፈተና‹fetena›)*

where

sense1*(እድገት‹Idget›)=(*hyponym*(እድገት‹Idget›)U*hypernym*(እድገትIdget›)U*attribute*(እድገት‹Idget›),*and

sense1(*ፈተና‹fetena›*)=(hyponym*(ፈተና‹fetena›)U*hypernym*(ፈተና‹fetena›)U*Meronym *(ፈተና‹fetena›)*

- sense2*(እድገት‹Idget›)*∩definition*(ደረጃ‹dereja›)*
- sense2*(እድገት‹Idget›)*∩definition*(መስፈርት‹mesfert›)*

- sense1*(ፈተና‹fetena›)*∩definition*(ደረጃ‹dereja›)*
- sense1*(ፈተና‹fetena›)*∩definition*(መስፈርት‹mesfert›)*

The same is done for the remaining combinations and compared. For this case the first one gets the highest score which is six. The second, third and fourth combinations gets total score of 0, 0, 3 respectively. The senses in that combination for *ፈተና‹fetena›*(sense1 *(ፈተና‹fetena›))*and for *እድገት‹Idget›*(sense2*(እድገት‹Idget›))* are selected. The other words in the combination are not assigned any sense as they have only one sense in the WordNet but they are used to facilitate the disambiguation of other words.

The result of the experiment using augmented semantic space that compare the gloss definitions of the ambiguous words, their related synsets and WordNet words issummarized in table 4.4 below.

Table 4. 4Performance of augmented semantic space

	One ambiguous word		Two ambiguous words		Three ambiguous words		Average	
	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise
Precision	0.81	0.81	0.83	0.65	0.88	0.7	0.84	0.72
Recall	0.57	0.57	0.68	0.53	0.82	0.65	0.69	0.58
Accuracy	57%	57%	68%	53%	82%	65%	69%	58%
F1 measure	0.67	0.67	0.75	0.58	0.85	0.67		

As presented in table 4.4, the word-wise precision and recall are increasing as we go from one target word to three words. The sense decision for a word in one target word is dependent only on the definitions and synsets of the surrounding words, in addition to information (its definition and related synsets) about the word itself. But in three and two target words the disambiguation of a word is dependent not only on the target word and its related synsets, but also the information of other target words is used, which are rich of synsets, definitions and related synsets. This increases the number of disambiguated words for sentences having more than one target words.

As we can see the sentence-wise precision of three target words is higher than two words. Per a sentence which is correctly disambiguated indicates that three words and two words are correctly disambiguated for three target words and for two target words respectively. So the sentence-wise result for two and three target words is dependent on the word-wise result. The difference between the sentence-wise and word-wise recall is higher as the number of target words increase. This is because a sentence is classified as correctly disambiguated if and only if all the target words in it are correctly disambiguated. In addition the precision and recall values are becoming closer as the number of target words increased. This shows that the WSD system covers most of the sentences and they are correctly disambiguated.

The reason that the augmented semantic space failed to disambiguate some of the words is the limited number of synsets in the WordNet. This algorithm is dependent on counting overlaps between glosses of different related synsets to the ambiguous word. This makes its performance dependent on the number of context words which exists in the WordNet.

Experiment 4: Applying context to gloss overlap then augmented semantic space

This experiment is done to evaluate the performance of applying context to gloss overlap first, then applying augmented semantic space to disambiguate a sentence. The aim is to give senses for words in which the first algorithm failed to assign any sense and unable to disambiguate the sentence.The hypothesis is as the two algorithms uses different methods and information to disambiguate words, it is more advantageous if they work together during the disambiguation process. During the experiment, first the context to gloss overlap is applied on a sentence to be disambiguated as discussed in experiment 2, then if it is failed to assign any sense the sentence is given to the augmented semantic space.

For example, for a sentence *"ስፖርታዊእንቅስቃሴለአካልእድገትእናጥንካሬጠቃሚ ነው"‹sportawi InqsqasE lea*kal Idget Ina Tnkare Teqami new›*

- Ambiguous words=*አካል‹a*kal›*with 3 senses),*እድገት‹Idget›*(with 2 senses)

- Context words= {*ስፖርት‹sport›,እንቅስቃሴ‹InqsqasE›, ጥንካሬ‹TnkarE›,ጠቃሚ ‹Teqami›*}

After the above steps explained in experiment two, the final highest score comes to be one for three of the combination of senses and zero for the remaining combinations for our example. So the algorithm cannot assign senses as there are more than one combination of senses having equal highest scores. Then the sentence is given to the augmented semantic space. When the sentence is disambiguated using augmented semantic space using the steps in experiment three, sense one for "*አካል*"*‹a*kal›*and sense one for "*እድገት*"*‹Idget›*are assigned which are the correct senses.

The performance ofapplying firstcontext to gloss overlap, followed by augmented semantic spaceis summarized in table 4.5 below.

Table 4. 5performance of applying context to gloss overlap then augmented semantic space

	One ambiguous word		Two ambiguous words		Three ambiguous words		Average	
	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise
Precision	0.88	0.88	0.86	0.76	0.92	0.81	0.89	0.82
Recall	0.78	0.78	0.76	0.68	0.87	0.79	0.8	0.75
Accuracy	78%	78%	76%	68%	87%	79%	80%	75%
F1 measure	0.83	0.83	0.81	0.72	0.88	0.79		

As we can see from the result, the result for this experiment is dependent on the result of the preceding two experiments. The precision for both word-wise and sentence-wise is higher than the third experiment and lower than the second experiment. The loss in precision from the second experiment in which context-to-gloss overlap is alone applied, is the low precision of augmented semantic space applied after it.

The recall is increased when compared to the third and second experiments. This is because, sentences not covered by the first algorithm are disambiguated by the senses suggested by the second, which increases the coverage. The increase in recall shows that the combination of augmented semantic space following context-to-gloss overlap assigned correct senses for most ambiguities as compared to their individual usage.

Here the performance for three and two ambiguous words is improved. But some sentences are not disambiguated correctly. This is because of wrong senses assigned by context-gloss overlap which is applied first. Since the augmented semantic space assign senses only for sentences not given any sense by the first algorithm, the wrong senses assigned by the context-gloss overlap are taken as they are. Especially the performance is greatly improved for three words, because the augmented semantic space performs well when the number of ambiguous words increases and as we have seen in the second experiment the context-to-gloss overlap is performing very low for three words, even couldn't assign any sense for much of the sentences.

Experiment 5: Applying augmented semantic space then context to gloss overlap

This experiment is done to evaluate the performance of applying augmented semantic space then context to gloss overlap after that. The aim is to give senses for words in which the first algorithm failed to disambiguate the sentence with the same hypothesis in the fourth experiment. The procedure is the same as the fourth experiment other than the augmented semantic space is applied first.

The result of applying augmented semantic space followed by context to gloss overlapis summarized in table 4.6 below.

Table 4. 6 performance of applying augmented semantic spacethen context to gloss overlap

	One ambiguous word		Two ambiguous words		Three ambiguous words		Average	
	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise
Precision	0.81	0.81	0.80	0.65	0.88	0.7	0.83	0.72
Recall	0.72	0.72	0.71	0.57	0.82	0.65	0.75	0.65
Accuracy	72%	72%	71%	57%	82%	65%	75%	65%
F1 measure	0.76	0.76	0.75	0.60	0.84	0.67		

From the result in table 4.6 there is some improvement in precision than applying augmented semantic space alone in the third experiment and lower than the second experiment. The increase in precision from the first (applying augmented semantic space) is because of high precision of context-to-gloss overlap applied on sentences in which the first method failed to assign any sense (to give answer).

The recall for one target word is showing greatest increase from the third experiment and the gap between precision and recall values is lowered. This shows that in the third experiment even if the precision was high the method does not give answers for most of the test sentences and the sense assigned for majority of them that the algorithm able to assign sense are correct. This shows that majority of the sentences that are not covered by the first method and which are given sense by the second method are assigned the correct sense. The improvement on recall for three target words and two target words is very small as the precision, which indicates that the method failed to give answer for the sentences that the first failed to give answer (assign sense).

From the result we can conclude that applying context-to-gloss overlap followed by augmented semantic space is better than applying the algorithms in the reverse order.

Experiment 6: Augmented semantic space and context to gloss overlap

This experiment is done to evaluate the performance of applying augmented semantic space and context to gloss overlap at the same time. The aim is to give scores for senses by using the two algorithms. A sense is selected for words if it is assigned the highest sum of scores by the two algorithms.The result for this experiment is summarized in table 4.7 below.

Table 4. 7 performance of augmented semantic space and context to gloss overlap at the same time

	One ambiguous word		Two ambiguous words		Three ambiguous words		Average	
	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise
Precision	0.86	0.86	0.82	0.70	0.90	0.78	0.86	0.78
Recall	0.76	0.76	0.73	0.62	0.84	0.72	0.78	0.7
Accuracy	76%	76%	73%	62%	84%	72%	78%	70%
F1 measure	0.81	0.81	0.77	0.66	0.87	0.75		

As shown in the table 4.7 the sentence-wise and word-wise recall is increased when we compare it with the results of applying the two methods alone. This shows that the number of sentences and words that the system is able to give answer as well as the number of correct disambiguation are increased here. The augmented semantic space gives credit for longer sequences of matches by squaring the length of overlaps which makes the score larger. But the context-to-gloss overlap only uses the frequency count, so the large score given by the augmented semantic space dominates the score given by the context-to-gloss. Due to this the performance of their combination is greatly dependent on the result given by the augmented semantic space. For example, to disambiguate a sentence "ለሃገርእድገትእናልማትሁሉምአካልሃላፊነትአለበት" ‹*lehagerIdget Ina lmat hulum a*kal halafinet a*lebet*› the two ambiguous words "እድገት"‹*Idget*› and "አካል"‹*a*kal*› have two and three senses respectively. The following are their senses,and the right sense for "እድገት"‹*Idget*›in our example is its second sense and second sense for "አካል"‹*a*kal*› also.

*Sense 1("እድገት"‹Idget›) ፡ የአካል ክፍል መጨመር‹yea*kal kfl meCemer›*

Sense 2("እድገት"‹Idget›) ፡ የተሻለ ማግኘት‹yetexale magNet›

*Sense 1("አካል"‹a*kal›) ፡ሰውነት‹sewnet›*

*Sense 2("አካል"‹a*kal›) ፡ክፍል‹kfl›*

*Sense 3("አካል"‹a*kal›) ፡ማህበር ጭፍራ‹mahbercfra›*

First the sentence passes through the preprocessing steps to generate bag of words:*"ሀገርእድገትልማሁሉሀገሪአካልሀላፊ"‹hagerIdget lma hulu hageri a*kal halafi›*

For the ambiguous words"*እድገት*"‹*Idget*›and"*አካል*"‹*a*kal*›, the glosses of their different sensesand their related synsets are extracted from the WordNet as shown in experiment 3.Also the meanings of context words "*ሀገር*"‹*hager*› and"*ልማት*" ‹*lmat*›are extracted from the WordNet.

The context words excluding the ambiguous words are extracted as explained in experiment 2.Also the words in the synsets of the ambiguous words as well as their related synsets are extracted with the meanings.

The score is calculated for each combination of senses following the steps in experiment 3:

C (sense2 *(እድገት‹Idget›),* definition *(ሀገር‹hager›),* definition *(ልማት‹lmat›),*sense1 *(አካል‹a*kal›)) =5*

C (sense1 *(እድገት‹Idget›),* definition *(ሀገር‹hager›),* definition *(ልማት‹lmat›),*sense2 *(አካል‹a*kal›)) =5*

C (sense2 *(እድገት‹Idget›),* definition *(ሀገር‹hager›),* definition *(ልማት‹lmat›),*sense3 *(አካል‹a*kal›)) =5*

C (sense1 *(እድገት‹Idget›),* definition *(ሀገር‹hager›),* definition *(ልማት‹lmat›),*sense3 *(አካል‹a*kal›)) =10*

C (sense1 *(እድገት‹Idget›),* definition (*ሀገር‹hager›*), definition *(ልማት‹lmat›),*sense1 *(አካል‹a*kal›)) =21*

C (sense2 *(እድገት‹Idget›)*,definition *(ሀገር‹hager›),* definition *(ልማት‹lmat›),*sense2 *(አካል‹a*kal›)) =7*

As we can see the highest score is given to the first sensesfor both "*አካል"‹a*kal*›and "*እድገኾ*"‹*Idget*›with score 21.Side by side the score for combination of senses using steps in experiment 2 is calculated,and the combination containing second senses of both"*አካል"‹a*kal*› and "*እድገት"* ‹*Idget*› with score 3 is the highest which is the right sense.

- frequency(*ሀገርልማሁሉሀገሪሀላፊ‹hager lma hulu hageri halafi*›,c(sense2(*እድገት‹Idget*›),sense2(*አካል‹a*kal*›)))=3
- frequency(*ሀገርልማሁሉሀገሪሀላፊ‹hager lma hulu hageri halafi*›,c(sense1(*እድገት‹Idget*›),sense1(*አካል‹a*kal*›)))=0
- frequency(*ሀገርልማሁሉሀገሪሀላፊ‹hager lma hulu hageri halafi*›,c(sense2(*እድገት‹Idget*›),sense1(*አካል‹a*kal*›))=1
- frequency(*ሀገርልማሁሉሀገሪሀላፊ‹hager lma hulu hageri halafi*›,c(sense1(*እድገት‹Idget*›),sense2(*አካል‹a*kal*›)))=2
- frequency(*ሀገርልማሁሉሀገሪሀላፊ‹hager lma hulu hageri halafi*›,c(sense2(*እድገት‹Idget*›), sense3(*አካል‹a*kal*›)))=1
- frequency(*ሀገርልማሁሉሀገሪሀላፊ‹hager lma hulu hageri halafi*›,c(sense1(*እድገት‹Idget*›), sense3(*አካል‹a*kal*›)))=0

Then score given by the two methods are added together for each combination of senses of ambiguouswords,and the sense with the highest score are selected. The scores given are shown below:

- c(sense2(እድገት‹*Idget*›),sense2(አካል‹*a*kal*›)))=10
- c(sense1(እድገት‹*Idget*›),sense1(አካል‹*a*kal*›)))=21
- c(sense2(እድገት‹*Idget*›),sense1(አካል‹*a*kal*›))=6
- c(sense1(እድገት‹*Idget*›),sense2(አካል‹*a*kal*›)))=7
- c(sense2(እድገት‹*Idget*›), sense3(አካል‹*a*kal*›)))=6
- c(sense1(እድገት‹*Idget*›), sense3(አካል‹*a*kal*›)))=10

Then the highest score which is 23 is selected but this is wrong sense selection. This shows that the highest score given by the augmented semantic space has dominated the score given by the context to gloss overlap which leads to wrong sense assignment.If the context to gloss overlap was applied alone, the senses assigned would be correct.

Experiment 7: Majority voting

Up to now we have seen the performance of the algorithms in different ways and we have seen the draw backs of each of them .Now let's see the performance by applying majority voting. The aim is to examine the performance of majority voting of the senses assigned by augmented semantic space and context to gloss applied individually with the approaches in experiment four, five and six respectively . Also the voting is done between the approaches in experiment four, five and six.But the result of applying voting on the results of experiment two, three and the results of applying them one after the other(augmented semantic space then context to gloss overlap and context to gloss overlap then augmented semantic space) and in combination gives the same performance results in experiment four, five and six. But when we test voting between the sense selections of context to gloss then augmented semantic space , augmented semantic space then context-to-gloss , and the combination of them at once(as in experiment 6) we got a 1% improvement on the results of experiment six. The result for this experiment is summarized in table 4.8 below.

Table 4. 8Result of majority voting

	One ambiguous word		Two ambiguous words		Three ambiguous words		Average	
	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise	Word wise	Sentence wise
Precision	0.87	0.87	0.83	0.71	0.91	0.79	0.87	0.79
Recall	0.77	0.77	0.74	0.63	0.85	0.73	0.79	0.71
Accuracy	77%	77%	74%	63%	84%	73%	79%	71%
F1 measure	0.82	0.82	0.78	0.67	0.88	0.76		

4.4. Discussion of Results

Generally experimental results shows that the number of context words and the amount of data used for the disambiguation of words and sentences greatly affects the performance. Also the number of target words to be disambiguated and the number of surrounding WordNet words are factors for the disambiguation performance. When the number of target words is high, the performance of context to gloss overlap decreases and augmented semantic space increases;whereas for small number of target words and longer sentences, the performance of augmented semantic space decrease and context gloss overlap increases.

The combination of augmented semantic space and context-to-gloss overlap gives better result than applying them individually.The method which gives the highest performance result is applying augmented semantic space after context-to-gloss. The word-wise recallachieved for one target word, two target words and three target words are 0.78, 0.76 and 0.87 respectively. And the sentence-wise recall achieved for one target word, two target words and three target words are 0.78, 0.68 and 0.79 respectively. Here we used recall/accuracy as the main performance metric because it shows for how many percent of the total test sentences does the system is able to give the correct answer.Therefore we propose the use of context-to-gloss followed by augmented semantic space for WSD of Amharic sentences based on the individual accuracies and its highest average accuracy 80% and 75% we got for word-wise and sentence-wise respectively.

However, the proposed approach has some sentences which are not correctly disambiguated. This case happens if augmented space applied after context-to-gloss overlap couldn’t assign any sense or if it assigns wrong sense. For example, “መንግስትአዲስለተመረተውየምርጥዘርምርትዋጋቆረጠ” ‹*mengst a*dis letemeretew yemrT zer mrt waga qoreTe*›have three ambiguous words (”ዘር”‹*zer*›,”ዋጋ”‹*waga*›,”ቆረጠ” ‹*qoreTe*›) having 2 senses each .The context-to-gloss overlap failed to assign any sense, because the score for all combinations of senses comes equal(0).So the sentence is given to augmented semantic space,but it gives wrong sense for”ዘር”‹*zer*›.This makes the sentence to be disambiguated wrongly as a whole.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1. Conclusion

This study attempts to develop an Amharic WSD system which uses Amharic WordNet as knowledge base. Seventeen ambiguous words used by Dureti(2017) are used here. In this study different experiments have been done using augmented semantic space and context-to-gloss individually and in combination.

The system identifies ambiguous words automatically from an input sentence. After preprocessing, ambiguous words identification, sense ranking, scoring and selection are done step by step, the final result is the disambiguated word meanings.The words are disambiguated simultaneously using augmented semantic space and context-to-gloss similarity which results in disambiguated sentence. Augmented semantic space uses synsets, definitions, and related synsets of the target words as well as the WordNet words around the target words and definitions of related synsets. Whereas the context-to-gloss uses frequency count of the words in the context with definitions, synsets, related synsets, related synsets' definitions of the target words.

To check the extent that thesystem works, it is evaluated in seven experiments by comparing the word-wise and sentence-wise performance for one up to three target words in Amharic sentences. Although the performance is reported based on precision, recall and F1-measure, our result analysis focuses on recall (accuracy). So the highest recall we got is 78% for one target word, 76% word-wise and 68% sentence-wise for two target words and 87% word-wise and 79% sentence-wise for three target words. The sentence-wise recall for two target words and three target words is lower than the word-wise because when at least one word from a sentence is disambiguated incorrectly, the sentence will be counted as incorrectly disambiguated. Those highest performance results are achieved by applying context-to-gloss then augmented semantic space after that.

The main challenge in this study was unavailability of lexical resources (WordNet). To the best of the researchers' knowledge there is no Amharic WordNet prepared before for commercial or research purpose. Even there is no an Amharic dictionary which contains the relationships that are needed for constructing a WordNet. So the WordNet is developed manually taking much amount of time for data collection, which makes as to consider limited number of words and relationships. Getting test sentences was also time taking because there is no any sense tagged Amharic corpus prepared for WSD and related works. Hence our experiment is limited to testing sentences having three target ambiguous words despite our system can work for n number of target words within a sentence.

In addition, the stemmer algorithm used in the preprocessing does not cover all exceptions and have limitations in returning the root word. It cannot identify the infixes and some exceptions that result from removal of suffixes and prefixes are corrected manually.

When we come to the algorithms implemented, theyhave their own limitations. The augmented semantic space is highly dependent on the number of words which exists in the WordNet. Also it depends on number of context words in the sentence which also exists in the WordNet. It works by counting overlaps between glosses which makes it dependent on the length of glosses, exact wording between glosses and number of related synsets. The limitation of the stemmer also has effect on the number of overlaps for the augmented semantic space. When we see the context-to-gloss overlap, it is limited to perform well for short sentences. Short sentences have less number of context words which makes the frequency count smaller. Also the limitation of the stemmer algorithm has effect on the frequency count.

5.2. Contribution of the Study

In this study an attempt has been made to show the way to identify more than one ambiguous words and disambiguating them simultaneously at sentence level using WordNet. The disambiguation uses related synsets of target words which are included in the WordNet through different relationships specific to each POS. And it is the first attempt to use those relations and use the context words that comes with the target words in addition to the WordNet words in simultaneous disambiguation of morethan one target word in a sentence at a time.We have used the extendedapproach which is one of the approaches used to construct WordNet for under resourced languages using existing WordNet for English language. This can be a base for further Amharic WordNet construction.

5.3. Recommendation

Based on the experiments conducted and discussion of result, the following future research directions are set as a way forward for the coming researches.

- As it is time taking to get the data for our WordNet, the relationships included in the WordNet other than synonymy are limited to only the ambiguous words, so fully constructed WordNet for Amharic in collaboration with linguists is very essential, not only for WSD but also for other NLP applications.
- Amharic is morphologically complex and our stemmer algorithm has limitations on covering all morphological variants, so there is a need to design a better,more effective stemmer algorithm or morphological analyzerfor the language.
- The augmented semantic space can be improved if all relationships between synsets are considered and more single sense words are added to the WordNet. Also the context-to-gloss can be improved by increasing the number of related synsets in the WordNet. This will increase the search space for the frequency count of the context words.In addition,the combination of the two approaches can be improved if a normalization scheme is integrated to the score given by augmented semantic space.
- Constructing generic ontology for Amharic words will be very help full for disambiguation if integrated with the proposed approach by giving additional information to disambiguate a word. So it would be better if the WSD system is integrated with ontology.
- One of the challenges for this research was getting test sentences for the experiments because there is no sense tagged Amharic corpus prepared for WSD or other applications. In other languages like English there are a number of sense tagged corpora like semcor which are open to be used in different researches. We are not able to clearly compare and contrast our result with previous researchers'

result because the test sentences are different in number and collected in different ways. So there is a need to prepare experimental test set as a test bed to compare and contrast the advancement done by different scholars.

References

Abhishek, F., & Manoj B., C. (2013, December). A Survey on Supervised Learning for Word Sense Disambiguation. *International Journal of Advanced Research in Computer and Communication Engineering, 2*(12), 4667-4670.

Alessandro, R., Jose, C.-C., & Roberto, N. (2017). Word Sense Disambiguation:A Unified Evaluation Framework and Empirical Comparison. *15th Conference of the European Chapter of the Association for Computational Linguistics: Long Papers (EACL)*, *1*, pp. 99-110. Valencia,Spain. doi:10.18653/v1/E17-1010

Alok, R., & Diganta, S. (2015, July). WORD SENSE DISAMBIGUATION: A SURVEY. *International Journal of Control Theory and Computer Modeling (IJCTCM), 5*(3), 1-16.

Ankita, S. (2013). Review: Semi-Supervised Learning Methods for Word Sense Disambiguation. *IOSR Journal of Computer Engineering (IOSR-JCE), 12*(4), 63-68.

Bender, Bowen, Cooper, & Ferguson. (1976). *Language in Ethiopia .* London: Oxford University Press.

Cowie, J., Joe, G., & Louise, G. (1992). Lexical Disambiguation using Simulated Annealing. *Computational Linguistics (COLING)*, (pp. 359-365). Nantes, France.

Daniel, Y., & Yitna, F. (1997). The Ethiopian script in ASCII.

Devendra, S. ,. (2014). *Literature Survey on UnsupervisedWord Sense Disambiguation.* Indian Institute of Technology, Computer Science and Engineering. Bombay: Indian Institute of Technology.

Devendra, S. C., & Ruslan, S. (2018). Knowledge-basedWord Sense Disambiguation using Topic Models. *AAAI Conference on Artificial Intelligence (AAAI-18)* (p. arXivpreprint arXiv:180101900). New Orleans, USA: 32nd.

Dureti, S. (2017). *A Generic Approach towards All Words Amharic Word Sense Disambiguation.* Addis Ababa, Ethiopia: Unpublished Masters thesis Addis ababa University .

Eldward, U. (1973). *The Ethiopians: An Introduction to the Country and People", October..* (3 ed.). London: Oxford University Press.

Eneko, A., & David, M. (2001). Knowledge Sources for Word Sense Disambiguation. *4th international conference on text speach and dialogue* (pp. 1-10). Berlin,Heidenburg: Springer.

Fellbaum, C. (1998). WordNet: An Electronic Lexical Database. (MA, Ed.) Cambridge.

Gerard, E. (2006). *Machine Learning Techniques for Word Sense Disambiguation.* PhD desetation, Universitat Politµecnica de Catalunya, Barcelona.

Gerard, E., Llu´ıs, M., & German, R. (2000). Naive Bayes and Exemplar-Based approaches to Word Sense Disambiguation Revisited. *14th international conference on Artificial Intelligence(ECAI).* ECAI-2000. Berlin, Germany. 2000.

Gerard, E., Lluis, M., & German, R. (2000). A Comparison between Supervised Learning Algorithms for Word Sense Disambiguation . *4th conference on Computational natural language learning.* Lisbon,Portugal.

Getahun, A. (2001, December). Towards the Analysis of Ambiguity in Amharic. *Journal of Ethiopian Studies, 34*(2), 35-56.

Getahun, W. (2014). *A word sense Disambiguation model for Amharic Words using Semi supervised learning paradigm.* Addis Ababa, Ethiopia: Unpublished Masters thesis Addis ababa University.

Hagerie, W. (2013). *ENSEMBLE CLASSIFIERS APPLIED TO AMHARIC WORD SENSE DISAMBIGUATION.* Addis Ababa, Ethiopia: Unpublished Masters thesis Addis ababa University.

Hayward, Katrina, & Richard, J. (1999). *Amharic. In Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet.* Cambridge: The University Press.

Jason, M. (2005). *Semantic Relatedness Applied to All Words Sense Disambiguation.* MINNESOTA: Unpublished MASTER THESIS.

Lokesh, N., & Kalyani, M. (2015, February). Supervised, Semi-Supervised and Unsupervised WSD Approaches: An Overview. *International Journal of Science and Research (IJSR), 4*(2), 1684-1688.

Mark, S., & Yorick, W. (2001). The Interaction of Knowledge Sources in Word Sense Disambiguation. *Computational Linguistics, 27*(3), 321-349 .

Nick, R. (2010). *Introducing Semantics* (7 ed.). New York, United States of America: Cambridge University Press.

Nurril, H., Suerya, S., & Francis, B. (2011). Creating the Open Wordnet Bahasa. *Pcific Asia Conference on Language,Information and Computation*, *25*, pp. 255-264.

Nyein, T. ,., Khin, M. S., & Ni, L. (2011, September). A Word Sense Disambiguation System Using Naïve Bayesian Algorithm for Myanmar Language. *International Journal of Scientific & Engineering Research, 2*(9).

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007, 12 01). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems, 24*(3), 45-77. doi:10.2753/MIS0742-1222240302

Phiip, E., & Eneko, A. (2007). *Word Sense Disambiguation Algorithms and Applications(Text, Speech and Language Technology)* (Vol. 33). Verlag Berlin, Heidelberg: Springer.

Pierpaolo, B., Marco, d. G., Pasquale, L., & Giovanni, S. (2008). Combining Knowledge-based Methods and Supervised Learning for Effective Italian Word Sense Disambiguation. *Semantics in Text Processing.8*, pp. 5-16. Venice,Italy: Assosiation for Computational Linguistics.

Rajani, M. ,., & Ravi, B. S. (2015). Performance Comparison of Word Sense Disambiguation Approaches for Indian Languages. *International Advance Computing Conference (IACC)* (pp. 166-169). IEEE.

Ravi, M., Mahesh, K., & Prashant, C. (2014). A Review Of Literature On Word Sense Disambiguation. *International Journal of Computer Science and Information Technology, 5*(2), 1475-1477.

Rezapour, A., Fakhrahmad, S., & Sadreddini, M. (2011). Applying Weighted KNN to Word Sense Disambiguation. *World Congress on Engineering*, *III.* London, U.K.

Roberto, N. (2009, February). Word Sense Disambiguation: A Survey. 41, 2, Article 10 (, p. 69 pages, . *ACM Computing Surveys, 41*(2), pp. 10-69. doi:10.1145/1459352.1459355

Roshan , K., & Manoj , C. (2015, April). AN HYBRID APPROACH TO WORD SENSE DISAMBIGUATION WITH AND WITHOUT LEARNED KNOWLEDGE. *International Journal on Natural Language Computing (IJNLC), 4*(2), 31-42.

Samhith, k., Arun, T., & Panda, G. (2016). Word Sense Disambiguation using WordNet Lexical categories. *International conference on Signal Processing, Communication, Power and Embedded System(SCOPES).* Paralakhemundi,Odisha.

Samrawit, Z. (2014). *Word Sense Disambiguation using Semantic Similarity for Query Expansion in Information Retrieval.* Addis Ababa,Ethiopia: Unpublished Masters thesis Addis ababa University .

Samta, T., & Monika, K. (2017). A Survey Paper on Approaches of Natural Language Processing(NLP). *International Journal of Advanced Research,Ideas and Innovations in Technology, 3*(3).

Satanjeev, B., & Ted, P. (2002). An adapted Lesk algorithm for word sense disambiguation using WordNet. *Computational linguistics and intelligent text processing*, 136–45.

Segid, H. (2015). *Amharic Word Sense Disambiguation Using WordNet.* Addis Ababa, Ethiopia: Unpublished Masters thesis Addis ababa University.

Solomon, A. (2011). *Unsupervised Machine Learning Approach for Word Sense Disambiguation to Amharic.* Addis Ababa,Ethiopia: Unpublished Masters thesis Addis ababa University.

Solomon, M. (2010). *Word Sense Disambiguation for Amharic Text: A machine Learning Approach.* Addis Ababa, Ethiopia: Unpublished Masters thesis Addis ababa University.

Sudip, K. N., & Sivaji, B. (2007). Word Sense Disambiguation Using Extended WordNet. *International Conference on Computing: Theory and Applications (ICCTA'07).* IEEE.

Swathy, R. (2017). A Survey on Word Sense Disambiguation Word Sense Disambiguation Used In NLP. *International Journal of Innovative Research in Computer and Communication Engineering, 5*(3).

Ted, P. (2000, May 7). A Simple Approach to Building Ensembles of Naive Bayesian Classifiers for Word Sense Disambiguation. *1st North American chapter of the Assosiation for Computational Linguistics* , (pp. 63-69). Seatle,Washington.

Ted, P. (2007). Unsupervised Corpus-Based Methods for WSD. In A. Eneko, & E. Philip (Eds.), *Word Sense Disambiguation: Algorithms and Applications* (pp. 133–166). Springer.

Tessema, M., Meron, S., & Teshome, K. (2008). The Need for Amharic WordNet.

Trevor, C. (2003). Performance metrics for word sense disambiguation. *Australasian Language Technology Workshop.* Melbourne, Australia,.

Udaya, R., & Subarna, S. (2014, Augest). WORD SENSE DISAMBIGUATION USING WSD SPECIFIC WORDNET OF POLYSEMY WORDS . *International Journal on Natural Language Computng(IJNLC), 3*.

Verena, H., & Erhard, H. (2012). A Comparative Evaluation of Word Sense Disambiguation Algorithms for German. *8th international conference on Language resources and evaluation(LREC)*, (pp. 576-583).

www.fullstackpython.com/why-use-python. (n.d.). Retrieved 11 27, 2017, from python.org.

www.python.org/doc/. (n.d.). (p. s. foundation, Producer) Retrieved 11 28, 2017, from python.org.

www.senseval.org. (n.d.). Retrieved 09 10, 2018

Xiaohua, Z., & Hyoil , H. (2005). Survey of Word Sense Disambiguation Approaches. *18th FLAIRS.* Clearwater Beach, Florida: American Association for Artificial Intelligence.

Yehualashet, B. (2016). *Hybrid Word Sense Disambiguation Approach for Afaan Oromo words.* ADDIS ABABA UNIVERSITY, COLLEGE OF NATURLAL AND COMPUTATIONAL SCIENCE SCHOOL OF INFORMATION SCIENCE. Addis Ababa: Unpublished masters thesis.

Zeynep, A. (n.d.). *WORD SENSE DISAMBIGUATION: METHODS AND APPLICATIONS.* İSTANBUL: Maltepe University.

አማርኛመዝገበቃላት (1 ed.). (1993 E.C). Addis Ababa, Addis ababa University, Ethiopia: Ethiopian Languages Research center.

APPENDICES

Appendix I. Ambiguous words and their senses

Word classes	Words	Senses		
	አካል‹*a*kal*›	ሰውነት‹*sewnet*›,ገላ‹*gela*›	ክፍል‹*kfl*›	ማህበር‹*mahber*›
	ዘር‹*zer*›	ትውልድ‹*twld*›,ዝርያ‹*zrya*›	ከፍሬ የሚገኝ‹*kfrEye migeN*›	
	እድገት‹*Idget*›	የአካል ክፍል መጨመር‹*ya*kal kfl meCemer*›	የተሻለ ማግኘት‹*yetexa lemagNet*›	
	ድካም‹*dkam*›	ዝለት‹*zlet*›	ጥረት‹*Tret*›	
	መንገድ‹*menged*›	ብልሃት‹*blhat*›,ዘዴ‹*zedE*›,አሰራር‹*a*serar*›	ጎዳና‹*godana*›	
	ፈተና‹*fetena*›	ሙከራ‹*mukera*›,ፍተሻ‹*ftexa*›, ጥያቄ‹*TyaqE*›	ችግር‹*cgr*›,ሙከራ‹*muker*›,ስቃይ‹*sqay*›	
Noun	ልጅ‹*lej*›	የአብራክ ክፋይ‹*yea*brak kfay*›	ህጻን‹*hSan*›	
	ድምፅ‹*dm'S*›	ጩህት‹*Cuhet*›	ምርጫ‹*mrCa*›, ውሳኔንማሳወቅ‹*wsanEnmasaweq*›	
	ዋጋ‹*waga*›	ፋይዳ‹*fayda*›,ጥቅም‹*Tqm*›	ተመን‹*tmen*›	
Verb	ደረሰ‹*derese*›	ተፈጸመሆነ‹*tefeSemehone*›	በቃ‹*beqa*›	
	ቆረጠ‹*qoreTe*›	ገመደ‹*gemede*›	ወሰነ‹*wesene*›	
	አከበረ‹*a*kebere*›	በአልአከበረ‹*bea*lea*kebere*›	ከፍአደረገ‹*kefa*derege*›	

Adjectives	ሃሰተኛ‹*haseteNa*›	ውሸትየሚናገር‹*wxetyeminager*›	ትክክለኛያልሆነ‹*tkkleNayalhone*›	
	ደማቅ‹*demaq*›	የጎላ‹*yegola*›	የሞቀ‹*yemoqe*›	
	ትኩስ‹*tkus*›	ሙቅ‹*muq*›	አዲስ‹*a*dis*›,ለጋ‹*lega*›	
Adverbs	በቀስታ‹*beqesta*›	በዝግታበርጋታ‹*bezgtabergata*›	በዝቅተኛድምፅ‹*bezqteNadm'S*›	
	ወደፊት ‹*wedefit*›	ለሚቀጥለው ጊዜ‹*lemiqeTlewgize*›	ከፊት አቅጣጫ‹*kefita*qTaCa*›	

Appendix II. Sample list of stop words

ነገር‹neger›	መሆኑ‹mehonu›	እያንዳንድ‹Iyandand›
አንድ‹a*nd›	ማለት‹malet›	በሆነ‹behone›
አንድን‹a*ndn›	ማለቱ‹maletu›	ከዚህ‹kezih›
እና‹Ina›	የሚገኝ‹yemigeN›	ከላይ‹kelay›
ና‹na›	የሚገኙ‹yemigeNu›	ከመሀል‹kemehal›
ወይም‹weym›	ማድረግ‹madreg›	ከመካከል‹kemekakel›
ሆኑ‹honu›	ማን‹man›	ከጋራ‹kegara›
ሆኖም‹honom›	ማንም‹manm›	ጋራ‹gara›
ነው‹new›	ሲሆን‹sihon›	ወዘተ‹wezete ›
ናቸው‹nacew›	ሲል‹silə›	ወደ‹wede›
ሁሉንም‹hulunm›	እዚህ‹Izih›	ያለ‹yale›
ላይ‹lay›	እንጂ‹Inji›	ሲሉ‹silu›
ሌላ‹lEla›	በኩል‹bekul›	በተመለከተ‹betemelekete›
ሌሎች‹lEloc›	በውስጥ‹bewsT›	በተመሳሳይ‹betemesasay›
ስለ‹sle›	በጣም‹beTam›	ያሉ‹yalu›
ቢሆን‹bihon›	ይህን‹yhn›	የኋላ‹yehuwala›
ብቻ‹bca›	በተለይ‹beteley›	የሰሞኑ ‹yesemonu›

Appendix III. The Amharic alphabet (‘fidel’) adopted from Daniel &Yitna(1997)

giz	ka`Ib	sals	rab`I	hams	sads	sab`I					
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
he	hu	hi	ha	hE	h	ho					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ				ሏ	
le	lu	li	la	lE	l	lo				lWa	
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ				ሗ	
He	Hu	Hi	Ha	HE	H	Ho				HWa	
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ				ሟ	
me	mu	mi	ma	mE	m	mo				mWa	
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ				ሧ	
`se	`su	`si	`sa	`sE	`s	`so				`sWa	
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ				ሯ	
re	ru	ri	ra	rE	r	ro				rWa	
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ				ሷ	
se	su	si	sa	sE	s	so				sWa	
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ				ሿ	
xe	xu	xi	xa	xE	x	xo				xWa	
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቍ	ቊ	ቋ	ቌ
qe	qu	qi	qa	qE	q	qo	qWe	qWu	qWi	qWa	qWE
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ				ቧ	
be	bu	bi	ba	bE	b	bo				bWa	
ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ				ቯ	
ve	vu	vi	va	vE	v	vo				vWa	
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ				ቷ	
te	tu	ti	ta	tE	t	to				tWa	
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ				ቿ	
ce	cu	ci	ca	cE	c	co				cWa	
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኍ	ኊ	ኋ	ኌ
`he	`hu	`hi	`ha	`hE	`h	`ho	hWe	hWu	hWi	hWa	Hwe
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ				ኗ	
ne	nu	ni	na	nE	n	no				nWa	
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ				ኟ	
Ne	Nu	Ni	Na	NE	N	No				NWa	
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ				ኧ	
e/a*	u	i	a	E	I	o				ea	
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	ኵ	ኲ	ኳ	ኴ

ke	ku	ki	ka	kE	k	ko	kWe	kWu	kWi	kWa	kWE
ኸ			ኻ		ኽ	ኾ					
`ke	`ku	`ki	`ka	`kE	`k	`ko					
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
we	wu	wi	wa	wE	w	wo					
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
`e	`u	`i	`a	`E	`I	`o					
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ				ዟ	
ze	zu	zi	za	zE	z	zo				zWa	
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ				ዧ	
Ze	Zu	Zi	Za	ZE	Z	Zo				ZWa	
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ					
ye	yu	yi	ya	yE	y	yo					
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ				ዷ	
de	du	di	da	dE	d	do				dWa	
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ				ጇ	
je	ju	ji	ja	jE	j	jo				jWa	
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጕ	ጒ	ጓ	ጔ
ge	gu	gi	ga	gE	g	go	gWe	gWu	gWi	gWa	gWE
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ				ጧ	
Te	Tu	Ti	Ta	TE	T	To				TWa	
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ				ጯ	
Ce	Cu	Ci	Ca	CE	C	Co				CWa	
ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ					
Pe	Pu	Pi	Pa	PE	P	Po					
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ				ጿ	
Se	Su	Si	Sa	SE	S	So				SWa	
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
`Se	`Su	`Si	`Sa	`SE	`S	`So					
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ				ፏ	
fe	fu	fi	fa	fE	f	fo				fWa	
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ				ፗ	
pe	pu	pi	pa	pE	p	po				pWa	