**JIMMA UNIVERSITY**

**JIMMA INSTITUTE OF TECHNOLOGY**

**SCHOOL OF POST GRADUATE STUDIES**

**FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING**

# Corpus Based Word Sense Disambiguation For Geez Language

By

**Amlakie Aschale**

**A Thesis Submitted to School of Graduate Studies of Jimma University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering**

**March 2020**

**Jimma, Ethiopia**

**JIMMA UNIVERSITY**

**JIMMA INSTITUTE OF TECHNOLOGY**

**SCHOOL OF POST GRADUATE STUDIES**

**FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING**

# Corpus Based Word Sense Disambiguation For Geez Language

**By**

**Amlakie Aschale**

**Advisor:** **Dr. Kinde Anlay**

**Co-Advisor:** **Mr. Fetulhak Abdurahman**

**Submission Date:** **March 2020**

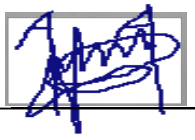
# Dedication

To my phenomenal Wife, Malefia, who knew the timely words of affirmation and motivation that kept me moving forward. I have no words to express your love, patience and motive. You made me happy again and again throughout my life. Without you this thesis would have not been completed one year earlier.

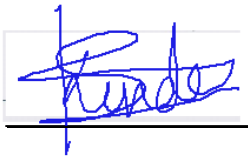
To my Hero brother, Hager, you are not only my brother you are my father, you are my friend, and you are my light. You showed me what a wonderful man mean for his family; I feel proud being your brother and always I wish to be with you to have a real joy.

# Declaration

I, declare that this thesis entitled as **Corpus Based Word Sense Disambiguation For Geez Language** is my original work and has not been worked and presented for any degree in this institution or any other universities, and all sources and references used for this thesis work have been appropriately duly acknowledged.

Student Name: Amlakie Aschale Signature: 

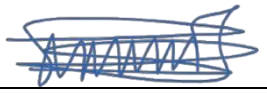
**A thesis has been submitted for examination with the approval of university advisers.**

Main Advisor: Dr.Kinde Anlay Signature: 

Co-advisor: Mr.Fetulhak Abdurhman Signature: _____

**This Thesis has been Approved by Board of Examiners**

Internal Examinor: Dr.SRINIVASAN.T.R. Signature: _____

External Examinor: Dr.Michael Melese Signature: 

Chair Person: Mr.Fetulhak Abdurahman Signature: _____

# *Acknowledgement*

First and foremost, praises and thanks to Almighty God, for His blessings throughout my research work to complete the research successfully. I am dedicated to thank Blessed Virgin Mary who has been my help and my solace throughout my entire life.

Then I would like to express gratefulness to my advisor **Dr. Kinde Anlay** for his encouragement, guidance, understanding and motivation throughout this thesis work. He has showed me how researches are produced and how to tackle a problem by investing his invaluable time. I really thanks! In addition, I would like to thank my co-advisor **Mr Fetulhak Abdurrahman** for providing me good guidance and feedback.

Further, I take the opportunity to express my very sincere thanks to my teachers, colleagues, family and friends. I will not be able to mention the name list here, but I had it in my memory. God bless you!

# Contents


Dedication . . . . . i

Declaration . . . . . i

Acknowledgements . . . . . ii

List of Figures . . . . . xiv

List of Tables . . . . . xv

List of Abbreviations . . . . . xvi

Abstract . . . . . xvii

1 Introduction . . . . . 1

1.1 Background . . . . . 1

1.2 Research Motivation . . . . . 3

1.3 Statement of the problem . . . . . 3

1.4 Objective of the study . . . . . 4

1.4.1 General Objective . . . . . 4

1.4.2 Specific Objectives . . . . . 5

1.5 Research Methodology . . . . . 5

1.6 Scope and Limitation of the Study . . . . . 6

1.7 Significance of the Study . . . . . 7

1.8 Thesis Organization . . . . . 7

2 Litrature Review . . . . . 9

2.1 Introduction . . . . . 9

2.2 Historical Background of Word Sense Disambiguation . . . . . 9

2.3 Basic Concepts of Word Sense Disambiguation . . . . . 11

2.4 Approaches to Word Sense Disambiguation . . . . . 12

2.4.1 Knowledge-Based Approach to WSD . . . . . 12

2.4.2 Hybrid Approaches to WSD . . . . . 13

2.4.3 Corpus-based approach To WSD . . . . . 14

2.4.3.1 Unsupervised Learning . . . . . 14

2.4.3.2 Supervised Learning . . . 17
2.4.3.3 Semi-supervised Learning . . . 18
2.5 Related Works . . . 20
2.5.1 WSD for Foreign Languages . . . 21
2.5.1.1 WSD to Persian Language . . . 21
2.5.1.2 WSD to Urdu Language . . . 21
2.5.1.3 WSD to English Language . . . 22
2.5.2 WSD for Local Languages . . . 22
2.5.2.1 WSD for Afaan Oromo Language . . . 22
2.5.2.2 WSD For Amharic Languag ɜ . . . 23
2.5.2.3 WSD for Tigrigna Language . . . 24
3 Ge'ez Language . . . 25
3.1 Overview of Ge'ez Language . . . 25
3.1.1 Geez Writting System . . . 25
3.1.2 Ge'ez Punctuation Marks . . . 26
3.1.3 Ge'ez Morphology . . . 27
3.2 Ambiguities in Ge'ez Language . . . 27
3.2.1 Orthographic Ambiguities . . . 28
3.2.2 Phonological ambiguity . . . 28
3.2.3 Lexical Ambiguity . . . 28
3.2.4 Semantic Ambiguity . . . 29
3.2.5 Structural Ambiguity . . . 29
3.2.6 Referential Ambiguity . . . 29
4 System Design And Architecture . . . 30
4.1 Introduction . . . 30
4.2 Data collection . . . 32
4.3 Design of corpus . . . 33
4.4 Proposed System Architecture . . . 33
4.5 Preparing Machine Readable Data . . . 38

4.6 Preparing Datasets . . . . . 39
4.7 Evaluation Techniques . . . . . 42
5 Experimental Result, Discussion and Evaluation . . . . . 45
5.1 Experimental Results and Discussion . . . . . 46
5.2 Comparative analysis of Geez WSD with other Languages . . . . . 56
6 Conclusion and Recommendation . . . . . 57
6.1 Conclusion . . . . . 57
6.2 Recommendation . . . . . 59
REFERENCES . . . . . 60

# List of Figures

Figure 2.1 Bootstrapping Classification (Adapted from, Yarowsky, 2008) . . . . . 20

Figure 4.1 Sample Searching result . . . . . 32

Figure 4.2 Corpus-Based Geez WSD system Architecture . . . . . 34

Figure 5.1 Average performance of the three-machine learning approaches . . . . . 50

Figure 5.2 Average values of each classification algorithm . . . . . 51

Figure 5.3 Average Accuracy of each window size using semi-supervised approach algorithms . . . . . 55

# List of Tables


Table 4.1 selected ambiguous words with their sense examples . . . . . 31

Table 4.2 Description of attributes used for this study . . . . . 39

Table 4.3 Example of WSD Dataset for Semi-Supervised Learning . . . . . 42

Table 4.4 Evaluation Metrics . . . . . 44

Table 5.1 Experimentation Results using unsupervised algorithms . . . . . 46

Table 5.2 Experimentation Results using supervised algorithms . . . . . 47

Table 5.3 Experimentation Results using Semi-supervised algorithms . . . . . 48

Table 5.4 performances of each window size using SMO algorithm . . . . . 52

Table 5.5 Average performances of each window size using ADtree algorithm . . . 53

Table 5.6 Average performances of each window size using AdaBoostM1 algorithm 54

Table 5.7 Comparison of word sense disambiguation on local languages with Geez Language . . . . . 56

# List of Abbreviations

| | |
|---|---|
| **ARFF** | **A**ttribute **R**elation **F**ile **F**ormat |
| **CSV** | **C**omma **S**eparated **V**alue |
| **EM** | **E**xpectation **M**aximization |
| **IR** | **I**nformation **R**etrieval |
| **IE** | **I**nformation **E**xtraction |
| **MNB** | **M**ultinomial **N**aïve **B**ayes |
| **MRD** | **M**achine **R**eadable **D**ictionaries |
| **MT** | **M**achine **T**ranslation |
| **NLP** | **N**atural **L**anguage **P**rocessing |
| **NLU** | **N**atural **L**anguage **U**nderstanding |
| **POS** | **P**art **O**f **S**peech |
| **SERA** | **S**ystem for **E**thiopic **R**epresentation **A**SCII |
| **SMO** | **S**equential **M**inimal **O**ptimization |
| **SR** | **S**peech **R**ecognition |
| **SVM** | **S**upport **V**ector **M**achines |
| **WEKA** | **W**aikato **E**nvironment **K**nowledge **A**nalysis |
| **WSD** | **W**ord **S**ense **D**isambiguation |

# *Abstract*

In 21$^{st}$ century,Improving several natural language applications could be used to make the communication between human beings and computers easy. Words in one language may have two or more meanings depending on the contexts that we use. Those words make the communication between computers and humans difficult because computers cannot differentiate or identify proper meanings of ambiguos words. So that, Word Sense Disambiguation(WSD) helps computers to identify the related meaning of the ambiguous words depending on the surrounding contexts. In this study, we tried to build WSD prototype model for Geez language because WSD is an intermediate task for other NLP tasks like information extraction, machine translation, speech recognition and information retrieval.There are three types of WSD approches; hybrid,knowldge based and corpus based approaches.From those approaches, we used a corpus-based approach to build the WSD model and it can be further classified as supervised, semi-supervised, and unsupervised machine learning methods. We conducted our experiments on six ambiguous words of Geez language by collecting a total of 2119 sentences or instances of the language.Those six ambiguous words of the language are:- ሀለፉ (Halafe), ቆመ (ḱome), ባረከ (bareke), አስተርዓየ(astaraya), ገብረ(gebira), ሰዓለ (Se'ale).We applied four clustering algo- rithms (EM, Simple K-Means, Farthest First, and Hierarchical Clusterer) and five classification algorithms (ADTree,AdaBoostM1,SMO, Bagging and Naïve Bayes) for clustering and classifi- cation purposes of the sentences. We compared the Corpus-based machine learning approachs, and we found that semi-supervised machine learning approach achieved the best performance. The proposed method achieved an average performance of 92.1%, 91.3%, 91% and 91.1% for Precision, Recall, F1-score and Accuracy using ADTree algorithm respectively. Window size of 4-4 has been the optimal window size to identify the meaning of the selected ambiguous words of Geez language using ADTree algorithm

***Key Words: Word Sense Disambiguation**, **Semi- supervised**, **AD tree**, **Geez Language***

# CHAPTER 1

# Introduction

## 1.1 Background

In $21^{st}$ century, the growth of information technology has lead the way for a large volume of information to be available for the society [1]. Discussion about importance of a language for using the available information is not far from obvious since it serves as a medium of communication among the races. Language has a potential to express a wide range of ideas and to convey complex thoughts. In particular, natural language is now being used to exchange information among humans and has now reached to the extent of being evolution criteria for the technology. In order to make available information useful for the society, an interest has emerged to make use of technology to process natural language. In response to such a need, Natural Language processing (NLP) has come up with a main focus of natural language computations [1].

NLP is a field of computer science that deals with the interaction among computers and humans using natural language that aims to enhance human-to-human communication and human to computer communication [2]. It is normally used to describe the function of software or hardware components in a computer system, which analyzes or synthesizes spoken or written language [3] . There are in fact two distinct focuses of NLP: language processing and language generation. The former one refers to the analysis of language for producing a meaningful representation, while the second refers to the production of language from a representation [4].

The field of NLP was originally referred to as Natural Language Understanding (NLU) in the early days of Artificial Intelligence. It is well agreed today that while the goal of NLP is true NLU, that goal has not yet been accomplished. A full NLU System would be able to paraphrase

an input text, translate the text into another language, answer questions about the contents of the text and draw inferences from the text [5]. The aim of NLP is studying problems in the automatic generation and understanding of natural languages. Natural language is understood as a tool that people use to express themselves and has specific properties that reduce the efficiency of textual information retrieval systems. These properties are linguistic variation and ambiguity. Natural language processing (NLP) is also a subfield of artificial intelligence and linguistics [6]. Ambiguity is one of the greatest challenges in NLP, is the term refers to understanding of something in two or more possible ways or something that has more than one meaning. It can appear in sentence (called structural or syntactic ambiguity) or at a word level (called lexical ambiguity) and phonological ambiguity. Ambiguity is a universally recognized linguistic phenomenon, which arises from the structure of the language and can be explained in terms of the analysis at different levels [7]. So that developing word sense disambiguation (WSD) is crucial for latter development of natural language applications such as information extraction, machine translation, information retrieval, question answering, text summarization and others.

In the field of computational linguistics, word sense disambiguation is defined as the problem of computationally determining which "sense" of a word is activated by the use of the word in a particular context. Lexical disambiguation in its broadest definition is nothing less than determining the meaning of every word in context, which appears to be a largely unconscious process in people. Due to the importance of WSD for understanding semantics and many real-world applications, researchers have been interestingly trying to tackle that problem. Therefore, the goal of this research is to design and develop a WSD prototype model for Ge'ez language using corpus-based Approach to extract training sets that minimizes amount of the required human intervention and to produce considerable improvement in learning accuracy of the data sets by preparing 2119 instances of fully annotated datasets.

## 1.2 Research Motivation

Currently, some natural language processing applications are being developed for Geez language.which includes morphological analysis of geez verbs and parsing of the language. Therefore, WSD task is a potential intermediate task for many other NLP systems, including mono and multilingual information retrieval, information extraction, machine translation or natural language understanding. Resolving the sense of ambiguity of words is essential for many Natural Language Understanding. However, due to large number of polysemous words in the language automatic processing of Geez text is facing difficulties. This motivates us to develop automatic word sense disambiguation system for Geez language.

## 1.3 Statement of the problem

Ge'ez is the classical language of Ethiopia and belongs to the Semitic language family. The other Semitic-Ethiopian languages are Tigrinya (known as North Ethiopic), Amharic (The working language of Federal Republic of Ethiopia and other regional state), Argobba, Harari, and Gurage (called South Ethiopic). The ancient philosophy, tradition, history and knowledge of Ethiopia was being written in Ge'ez and there are different books that are written in this language[8]. The resources can also be used as sources of philosophy, creativity, knowledge and civilization both to Ethiopia and the rest of the world. Moreover, if someone who wants to conduct a research on issues related to the classical custom, history, politics, tradition, and religion of Ethiopia, he/she need to explore the works handed down from the previous generations. Therefore, the researcher must investigate these literatures. To use those resources, one must know the language itself or else these literatures have to be translated into either of the currently spoken languages manually, which may take a long time.

To solve this problem, studying the linguistic nature of the language in a scientific approach with the help of Information Technology is necessary [8].Word ambiguity problem hinders to develop WSD based applications using this language i.e. highly polysemous words with Subtle

sense distinctions still pose major challenges for automatic systems[1]. To address these challenges and create background solution for natural language processing systems for the language, word sense disambiguation (WSD) is one of the techniques proposed to reduce or avoid word ambiguities. It helps them to overcome the problem of ambiguity. Absence of WSD is one of the most challenging tasks in Machine Translation system and other WSD based applications [9].

Since Ge'ez language has a number of ambiguous words; absence of automatic word sense disambiguation to Ge'ez language can be a challenge for the development of WSD based applications of Ge'ez language. Therefore, solving such kind of problem and the previous one for the language can help for the development of WSD using corpus-based Approach due to the unavailability or shortage of Ge'ez annotated data sets. Even though WSD is important for developing other natural language applications, there is no WSD system which has been developed for Ge'ez language. The research will answer the following questions:

- What would be the most effective algorithm from the selected algorithms that improve the performance of Ge'ez WSD model?
- What would be the optimal window size for Ge'ez WSD systems using corpus-based approach?

## 1.4 Objective of the study

### 1.4.1 General Objective

The general objective of this study is to develop word sense disambiguation model for Ge'ez language using corpus-based approach.

### 1.4.2 Specific Objectives

To achieve the general objective of this research the specific objectives are:-

- To develop mannualy annotated corpus for Geez ambiguous words
- To identify the most effective learning paradigm and the most effective algorithm from corpus based approaches
- To identify the context window size of Word sense disambiguation model;
- To design and develop a corpus-based WSD for Geez model;
- To evaluate the performance of a model for Word Sense Disambiguation to Ge'ez language;

## 1.5 Research Methodology

To conduct this research, the researcher has used tools, techniques, and procedures by using the following methods.

**Literature Review:-** Various literatures that are considered relevant for the research work were reviewed to get better understanding of the area and to have detailed knowledge on various techniques that are essential for WSD systems. Different literatures were reviewed in the following areas:-

- Researchers conducted on Word sense disambiguation for both local and global languages.
- Different machine learning approaches with their advantage and disadvantage.
- Ge'ez writing system, punctuation marks and its syntactic structure.
- Different documents which were written in Ge'ez have been reviewed.

**Tools and Techniques:-** Different tools and techniques have been used to achieve the goal of this research. Weka 3.9 and both python and java programming languages are used to develop word sense disambiguation. When we use Weka 3.9 as a development platform, we apply different classification and clustering algorithms to develop best WSD model, Python programming language is used to do preprocessing tasks like tokenization, normalization and java-programming language was used to stop word removal and stemming. We have choosen those tools for conducting this study because of their familiarity with the researcher.

**Data Source and Corpus Preparation:-** To achieve the objective of this study, corpus preparation is the major task. A text corpus is one of the resources required in natural language processing researches. A good-sized text can reasonably show the best performance of the model that we develop.

For the purpose of this study, the researcher collects sample texts or sentences of different disciplines that were collected from different sources such as Geez bible, Sinksar, Fithanegest, Gedile semaetat, and Bahir Dar University Geez Department teaching materials. Those sample texts or sentences contain ambiguous words of Geez language. Before using those collected data directly to our system, they have to be preprocessed. Finding large number of manually annotated datasets for Geez language is so difficult. The reason that makes difficult for finding large number of manually annotated datasets for Geez language is due to the absence of context-based database repository.

## 1.6 Scope and Limitation of the Study

Word Sense Disambiguation is a complex and recent research discipline that requires the effective analysis and processing of ambiguous words in its textual context. There are no publicly available linguistic resources for Ge'ez language like WordNet and Machine readable dictionaries(MRD). Due to unavailability of words context database for Ge'ez language, the study was limited to train and evaluate by using sentences that we will collect with only six ambiguous words and 2119 sense examples. Even if a word may have multiple senses, ambiguous words

having two senses are used in this study. However, the study will focus only on lexical ambiguity because it is difficult and time taking to get enough Ge'ez datasets which are referred to the other ambiguities. This study is focused in Textual information only which means the system accepts data in text form only not sound or voice form and Semantic level analysis only which means the system does not perform any kind of grammar and spelling correction or any other ambiguities in the language.

## 1.7 Significance of the Study

The results of the study are expected to produce experimental evidences that demonstrate the applicability of corpus-based approaches to word sense disambiguation of Ge'ez texts. The datasets that we will prepare can serve as Ge'ez dataset repository. It can also serve as the initiation for the preparation of part of speech tagged dataset repository for the development and evaluation of Ge'ez WSD by applying part of speech tagging. It can also have a contribution to future researches and development of other Natural Language Processing applications specifically machine translation, information retrieval, grammatical analysis as those areas require word sense disambiguation.

## 1.8 Thesis Organization

The thesis is organized into six chapters which comprises Introduction, Literature review, overview of Geez language, Methodology, Experimentation, Conclusion and Recommendations. The first chapter gives the general introduction of the study. The second chapter presents review made on different literatures regarding to the Word Sense Disambiguation together with its approaches and different machine learning techniques. The third chapter discusses the overview of Geez language. The fourth chapter discusses the methodology and presents the process of data preparation for the system to be developed for Geez language. It presents the algorithms

and dataset preparation based on the analysis of categories as reviewed. The fifth chapter discusses the experimentation and discusses of the findings. Chapter sixth presents the conclusion and recommendations as well as some directions to future works.

# CHAPTER 2

# Litrature Review

## 2.1 Introduction

The purpose of this chapter is to provide a brief review of word sense disambiguation (WSD). This is achieved by reviewing literature in the area of WSD to investigate WSD approaches, techniques and tools that were employed during the evolution of WSD researches since its inception. The comprehensive coverage of existing approaches is considered indispensible since it serves for understanding the central problems in WSD. The chapter also encompasses brief background and history, application areas and discussions on the three classes of machine learning methods that have been employed for WSD research and finaly it describes about overview of geez languages .

## 2.2 Historical Background of Word Sense Disambiguation

It was in the 1940's that Word Sense Disambiguation(WSD) was formulated as a separate task. In the 1950's, ideas for machine translation encouraged the first major research push in WSD, and although there was relatively little computing power around at the time, researchers thinking about WSD created algorithms and ideas that are still in use today. The inherent difficulty in the task of WSD was well appreciated further in the 1960s, at that time; researchers had already in mind essential ingredients of WSD, such as the context in which a target word occurs, statistical information about words and senses, knowledge resources. Very soon it became clear that WSD was a very difficult problem, also given the limited means available for computation.

Indeed, its acknowledged hardness was one of the main obstacles to the development of machine translation in the 1960s. WSD was then regenerated in the 1970s, within the research in Artificial Intelligence on complete natural language understanding[10].

The 1980s witnessed a turning point in WSD research, and large scale corpora and other lexical resources became available. Before the 1980s, much of WSD research depended on handcrafting of rules. Now it became possible to use knowledge extracted automatically from the resources. The research by came up with a simple yet seminal algorithm that used dictionary definitions from the Oxford Advanced Learner's Dictionary (OALD) and this marked the beginning of dictionary-based WSD. The study in[11] combined the information in Roget's thesaurus with co-occurrence data from large corpora in order to learn disambiguation rules for Roget's classes. The 1990s saw further improvements in the field, which can be categorized in three major groups WordNet, statistical NLP, and SensEval (later SemEval).

The corpus-based WSD in statistical MT first used in [10]. SensEval (later SemEval) made it possible for researchers to compare different systems with each other because of the fixed set of test words, annotators, sense inventories, and corpora. Before SensEval, the only common ground that WSD researchers had were a lower bound (calculated by either picking a random sense, or taking into account the most frequent senses) and an upper bound (derived from inter-tagger agreement). Now it became possible to develop different systems and evaluate them on the data sets provided by SensEval, thereby introducing scientific rigor and uniformity. SensEval eventually became the primary forum for all WSD evaluations. In SensEval-3, a conclusion was reached that WSD in itself has reached a performance level, and no significant rise in the results obtained already is possible. It has been since then, that people started thinking about new directions in which WSD research can go. In particular, in recent years there has been considerable growth in the areas of parallel bilingual corpora and unsupervised corpus-based WSD.

## 2.3 Basic Concepts of Word Sense Disambiguation

In natural languages, each word may have more than one meaning that is why a single word sometimes can have many senses. Disambiguation means to remove all ambiguity. WSD is a technique used in finding the meaning of a word in a sentence. A word can have multiple meanings and the exact meaning of the word is decided based upon context by humans. Computers also can follow similar technique i.e. decide meaning of an ambiguous word in a sentence using context information. Likewise, it is the process of identification to decide the appropriate meaning of an ambiguous word in a particular context. People decide the meaning of a word based on the characteristic senses of a discussion or situation using their own merits. Machines have no ability to decide such an ambiguous situation unless some practices have been planted into the machines' memory [12].

The word sense is extracted by human by connecting it to that specific context. But for corpus this function is performed by natural language running by the qualities of the term. It disambiguates ambiguity of an individual word that can be used (in different contexts) to express multiple meanings. The context is the only real way to recognize the significance of the homonym term. Generally, context can be used in the bag of words and relational information. Here the bag of words is the context that considered as words in some window surrounding the ambiguous word in terms of distance. The surrounding word of the ambiguous word decides correct choice of word. Since WSD as a process for finding the appropriate sense of a target, word is performed on a set of words. It can be thought as a classification task where each word can be classified in to its candidate senses[13]. The two major variations of WSD are:

**Lexical sample (or targeted WSD):-** where a system is required to disambiguate a restricted set of target words usually occurring one per sentence. Supervised systems are typically employed in this setting, as they can be trained using a number of hand-labeled instances (training set) and then applied to classify a set of unlabeled examples (test set).

**All-words WSD:-** where systems are expected to disambiguate all openclass words in a text

(i.e., nouns, verbs, adjectives, and adverbs). This task requires wide-coverage systems. Consequently, purely supervised systems can potentially suffer from the problem of data sparseness, as it is unlikely that a training set of adequate size is available which covers the full lexicon of the language of interest. But in this study we used the Lexical sample since we cover a restricted numbers of words and it is preferable from The two variations of WSD; because all-words task is very similar to part-of-speech tagging, except with a much larger set of tags, since each lemma has its own set. A consequence of this larger set of tag is a serious data sparseness problem; there is unlikely to be adequate training data for every word in the test set.

## 2.4 Approaches to Word Sense Disambiguation

All disambiguation work involves matching the context of the instance of the word to be disambiguated with either information from an external knowledge source (knowledge driven WSD), or information about the contexts of previously disambiguated instances of the word derived from corpora (data-driven or corpus-based WSD) [10]. Any of a variety of association methods are used to determine the best match between the current context and one of these sources of information, in order to assign a sense to each word, which are called knowledge acquisition bottleneck in WSD literatures. Different WSD approaches have been used through the evolution of WSD research. The major ones are Corpus-Based Approaches, Knowledge-based Approaches, and Hybrid Approaches. In this section, the survey of these approaches will be presented[14].

### 2.4.1 Knowledge-Based Approach to WSD

Corpus based approach requires considerable amount of work to create a classifier for each word in a language as discussed earlier. As a result, researchers tend to work for a few words. Knowledge-based approaches use an explicit lexicon like machine readable dictionaries (MRD), thesauri, computational lexicons such as WordNet or (handcrafted) knowledge bases as information source to resolve lexical ambiguities for many words. Thesaurus based disambiguation

makes use of the semantic categorization provided by a thesaurus or a dictionary with subject categories. The basic inference in thesaurus-based disambiguation is that semantic categories of the words in a context determine the semantic category of that context as a whole [15].

In addition, this category then determines the correct senses that are used. Similar to machine readable dictionaries, a thesaurus is a resource for humans, so there is not enough information about word relations. Computational Lexicons are a large electronic database containing useful lexical relations in linguistic psycholinguistic and computational research has led to a number of efforts to create large electronic databases of such relations [16]. Lexicon like WordNet is used for sense evaluation and for similarity measure in WSD. For example, a works of [17] created a knowledge base from WordNet's hierarchy and apply a semantic similarity function to accomplish disambiguation, also for the purposes of information retrieval.

### 2.4.2 Hybrid Approaches to WSD

Since they obtain disambiguation information from both corpora and explicit knowledge base, hybrid approaches do not fall into either knowledge or corpus-based. Hybrid systems aim to use the strengths of the both conquering specific limitations associated with a particular approach, to improve WSD accuracy. They are both on a 'knowledge driven, corpus-supported' theme, utilizing as much information as possible from different sources. the researcher in [17] used Bootstrapping approaches where initial data comes from an explicit knowledge source, which is then improved with information derived from corpora. He defines a small number of seed definitions for each of the senses of a word (the seeds can also be derived from dictionary definitions or lexicons such WordNet). Then the seed definitions are used to classify the obvious cases in the corpus.

### 2.4.3 Corpus-based approach To WSD

A major challenge facing on WSD research is the ability to acquire a large amount of words with their different contexts. Corpus-based approaches came up with alternate solution to the challenge by obtaining information necessary for WSD directly from textual data that is called a corpus. A corpus provides a bank of samples which enable the development of numerical language models, and thus the use of corpora goes hand-in hand with empirical methods . Corpus-based approaches can be categorized into three sub classes based on the form of training: Supervised learning, unsupervised learning and Semisupervised learning Approach to WSD. However, we used Corpus Based Approaches, which involves machine-learning techniques (supervised, unsupervised and semisupervised) techniques to induce models of word usage from which is larger than the previous works collections of text examples[16].

#### 2.4.3.1 Unsupervised Learning

Unsupervised learning is an independent process where no supervision is involved during the learning step. Unsupervised corpus based methods are knowledge-lean and do not rely on external knowledge sources such as MRD, concept hierarchies and sense tagged texts. Unsupervised approaches are mainly clustering approaches where words and contexts are clustered. During clustering, each cluster corresponds to a sense of a target word. The goal of clustering is to group together elements in a way which maximizes similarity between elements in one cluster and to minimize similarity between elements belonging to different clusters.

During unsupervised technique, feature vector representation of unlabeled instances are taken as input and are then grouped into clusters according to a similarity metric [15]. The following are algorithms of unsupervised approaches. Clustering algorithms are generally categorized as partitioned and hierarchical. The next section describes some common clustering algorithms. Here are general properties that characterize clustering algorithms.

**Partitioning clustering:-** Partitioning clustering is the method that require the number of clusters which will be determined by the user to rearrange each object place starting from the initial

partitioning. In this method of clustering, the process needs many times to achieve the better result by using all possible partitions. Because of this it is not easy to clusters the given large data sets. The most common type is the K-Means, Expectation maximization (EM) and they are discussed as the following.

**K-means algorithm :-** The most commonly used family of partitional algorithms is based on the K-means algorithm. K-means clustering is often used on large data sets since its complexity is linear in ”n”, the number of elements to be clustered. It creates a partitioning such that the intra-cluster similarity is high and the inter-cluster similarity is low. K-means uses the concept of a centroid where a centroid represents the center of a cluster. A centroid is usually not an element from the cluster. Rather, it is a pseudo-element that represents the center of all other elements.

Often the mean of the feature vectors of the elements within a cluster is used as that cluster”s” centroid. It is often difficult to define a centroid for categorical features. K-means iteratively assigns each element to one of K clusters according to the centroid closest to it and recomputed the centroid of each cluster as the average of the cluster”s” elements. The following steps outline the algorithm for generating a set of K clusters [18]:

1. Randomly select K elements as the initial centroids of the clusters;
2. Assign each element to a cluster according to the centroid closest to it;
3. Recomputed the centroid of each cluster as the average of the cluster’s elements;
4. Repeat Steps 2-3 for T iterations or until a criterion converges, where T is a predetermined constant.

**Expectation Maximization(EM) Algorithm:-** Expectation maximization (EM) is a clustering algorithm that works based on partitioning methods. This algorithm is a memory efficient and easy to implement algorithm, with a profound probabilistic background. EM is widely used iterative algorithms for estimating model parameters in the presence of missing data, in our case: the missing data are the senses of the ambiguous words. This method estimates missing parameters of probabilistic models. Generally, this is an optimization approach, which had given some initial approximation of the cluster parameters, iteratively performs two steps: first, the expectation step computes the values expected for the cluster probabilities, and second, the

maximization step computes the distribution parameters and their likelihood given the data. It iterates until the parameters being optimized reach a fix point or until the log-likelihood function, which measures the quality of clustering, reaches its maximum. The expectation maximization is an iterative estimation procedure in which a problem with missing data is present in a different form to make use of complete data estimation techniques [18].

**Hierarchical clustering:-** Hierarchical clustering algorithms produce a cluster hierarchy named a dendrogram (Berkhin, 2002). These algorithms can be categorized as divisive (top-down) and agglomerative (bottomup) (Jain, 1999) (Berkhin, 2002). We discuss these approaches in the following sub-sections.

**Divisive Hierarchical Clustering:-** Divisive algorithms start with one cluster of all documents and at each iteration split the most appropriate cluster until a stopping criterion such as a requested number k of clusters is achieved. A method to implement a divisive hierarchical algorithm is described by Kaufman and Rousseeuw[18]. In this technique in each step the cluster with the largest diameter is split, i.e. the cluster containing the most distant pair of documents. As we use document similarity instead of distance as a proximity measure, the cluster to be split is the one containing the least similar pair of documents.

Within this cluster the document with the least average similarity to the other documents is removed to form a new singleton cluster. The algorithm proceeds by iteratively assigning the documents in the cluster being split to the new cluster if they have greater average similarity to the documents in the new cluster (Kang, 2003).

**Agglomerative Hierarchical Clustering:-** Agglomerative clustering algorithms start with each document in a separate cluster and at each iteration merge the most similar clusters until the stopping criterion is met. They are mainly 32 categorized as single-link, complete-link and average-link depending on the method they define inter-cluster similarity.

#### 2.4.3.2 Supervised Learning

Supervised is the use of algorithms that is from externally supplied instances (training set) to form classes to differentiate new data. The goal of supervised learning is to build a model of the distribution of class labels in terms of predictor features. In order to build the model it involves training and testing phases. During the training phase a sense-annotated training corpus is required, from which syntactic and semantic features are extracted to build a classifier using machine learning techniques and in testing phase the classifier tries to find out the appropriate sense for the word based on surrounding words present in the sentence.

Supervised methods used for WSD problems are probabilistic based, similarity based, discriminating rule based and linear classification based. Probabilistic methods involve estimate of set of parameter such as conditional or joint probability distribution and context where the parameter is then used for assigning category to new sample which maximizes the conditional probability. In similarity based methods the categorization is done by comparing the features of new sample with the features of trained sample and assign sense of most similar pattern (features). In case of methods involving discriminating rules, one or more selected rules associated with each word sense are involved to classify new sample and then assignment of sense will be followed based on their predictions. Defining of word senses using supervised approaches equals construction of a semantic lexicon and manual annotation of word senses in text which is difficult and slow process [12].

Supervised WSD approaches yield very high accuracy at the expense of the costly resources required to be incurred in terms of time and manual efforts for sense tagging. Despite the higher accuracy for these approaches, the impracticability of creating corpora for all languages in all domains becomes a hindrance. In addition, ensemble methods which function by combining the different classification algorithms were also seen to yield better performance scores than using the classification algorithms separately [10].

**Naïve Bayes:-** Naïve Bayes is a statistical algorithm based on Bayes' theorem. In this model, each feature is considered independent and the conditional probabilities of occurrence of words

are computed. In text classification, the Bayes' Theorem calculates the probability of each label for a given text and then output the label with the highest one (Pawar and Gawande, 2012). Among the Naïve Bayes family of algorithms, Multinomial Naïve Bayes (MNB) is the one which focuses on a multinomial distribution of features. MNB is a probabilistic model that computes class probabilities for a given dataset using Bayes' rule. Assume there are N vocabularies and C set of classes[13].

**Bagging :-** Bagging (Breiman, 1996) creates an ensemble of classifiers in order to increase the accuracy by stabilizing the base learning algorithm, or in other words decrease its variance. This is done by generating a set of "new" training sets using bootstrapping and applying the base learning algorithm on these data sets. Prediction is determined by the majority vote of the ensemble. The success of bagging depends heavily on the properties of the base learning algorithm. It should be unstable, meaning that it is sensitive to small changes in the training set, so that its variance can be decreased.

**SMO :-** SMO, proposed by John Platt, is a sequential minimum optimization algorithm for training support vector machines (SVM). The algorithm finds the maximum margin hyperplane represented as a set of vectors known as support vectors. In order to solve non-linear problems with this linear classifier, the instance space is transformed using a non-linear kernel function. We chose to use the default polynomial kernel (Stig-Erland, 2007).

#### 2.4.3.3 Semi-supervised Learning

Semi-supervised techniques involve training information like in supervised but the information given at initial training phase is less. Semi-supervised or minimally supervised methods are gaining popularity because of their ability to get by with only a small amount of anno- tated reference data while often outperforming totally unsupervised methods on large data sets. There are a host of diverse methods and approaches, which learn important characteristics from auxiliary data and cluster or annotated data using acquired information. it uses combined information of labeled and unlabeled examples for achieving better classification performance that could be obtained by discarding the unlabeled data and doing the supervised or unsupervised

by discarding the labels. Semi-supervised algorithms involve both classification and clustering assumptions while employed in WSD. In one sense these algorithms are used when supervised learning is fed with more and more unlabeled data for the purpose of classification of the unlabeled data to a specific class and in the other case they are used when an unsupervised algorithm is provided with more and more labeled data for the sake of clustering [19].

**Bootstrapping:** Semi-supervised methods for WSD are characterized in terms of exploiting unlabeled data in learning procedure with the requirement of predefined sense inventory for target words. There are two main approaches of semi-supervised learning methods in WSD: co-training and self-training [1]. Bootstrapping algorithms are commonly used semi-supervised learning methods for WSD. Bootstrapping sense tagged seed examples are used to overcome the bottleneck of acquisition of large sense-tagged data. It works by iteratively classifying unlabeled examples and adding confidently classified examples into labeled dataset using a model learned from augmented labeled dataset in previous iteration. It can be found that the affinity information among unlabeled examples is not fully explored in this bootstrapping process. Bootstrapping is based on a local consistency assumption: examples close to labeled examples within same class will have same labels, which is also the assumption underlying many supervised learning algorithms. The aim of bootstrapping is to build a sense classifier with a little training data, and thus overcome the main problems of supervision: the lack of annotated data and the data sparsity problem. Bootstrapping usually starts from few annotated data A, a large corpus of un-annotated data , and a set of one or more basic classifiers. As a result of iterative applications of a bootstrapping algorithm, the annotated corpus A grows increasingly and the untagged data set ”?” shrinks until some threshold is reached for the remaining examples in A and B according to the following figure [20].

**ADTree:** ADTree (Freund, 1999) creates what is known as an alternative decision tree by using boosting to add the different branches. An alternative decision tree is simply a set of interconnected decision stumps with numerical leaves, where each leaf may be connected to a set of other stumps. The tree is used to classify unlabeled instances with binary classes by summing all the numerical nodes encountered while following the different paths of the tree applicable

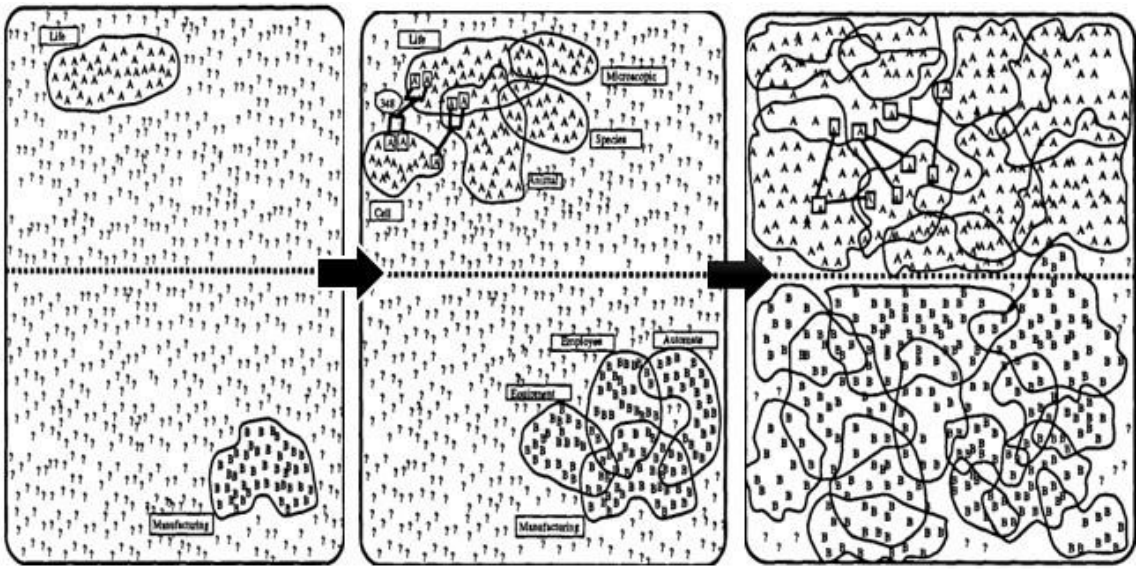



**Figure 2.1:** Bootstrapping Classification (Adapted from, Yarowsky, 2008)

for the instances. The sign of this value determines the predicted class.

**AdaBoost.M1:** AdaBoost.M1 (Freund, 1996) is a boosting algorithm that builds an ensemble of classifiers by forcing the base learning algorithm to focus on the instances that the previous classifiers had problems classifying correctly. This is done by accompanying every training instance with a weight representing the severity of misclassification such that the error rate is calculated as the sum of the weights of the misclassified instances divided by the sum of all the weights. Initially, each instance has equal weights, but after a new classifier is induced, the weights are updated so that misclassified instances increase in weight while the other instances decrease.

## 2.5 Related Works

This gives brief description of the various solutions that are done before for word sense disambiguation. The task of disambiguation system is to resolve the lexical ambiguity of a word in a given context. To put it more precisely, the term lexical ambiguity refers to two different concepts homonymy and polysemy. The distinction between chair (authorized person) and chair (sitting material made from wood or metal) has been used as an example of homonym,

and rust (verb) and rust (noun) for polysemy. Reflecting the rapid growth in utilization of machinereadable texts, word sense disambiguation techniques have been explored variously in the context of CorpusBased and Knowledgebased approaches. This section surveys past research associated with CorpusBased (supervised, unsupervised and semisupervised learning) of word sense disambiguation for local and foreign languages.

### 2.5.1 WSD for Foreign Languages

#### 2.5.1.1 WSD to Persian Language

The Authors in [21] used semi-supervised machine learning approach with minimal supervision. The applied method tries to disambiguate word senses by considering defined features of target words and applying collaborative learning method. Extracted corpus from published news that news agencies are used as the reference corpus. Evaluating the program by the available corpus on three considered ambiguous words, the implemented method was able to properly identify the meaning of 5368 documents with 88% recall, 95% precision and 93% accuracy rate

#### 2.5.1.2 WSD to Urdu Language

The Authors in [22] used a statistical approach that means only Bayesian classification is applied for resolving this peculiar type of lexical ambiguity for some Urdu words. In this work, the accuracy of the system depend on the window that means as the window size increases from 3x3 to 7x7. As more and more context is entertained while determining the desired sense, the accuracy of classifying the correct sense has increased. The precision of the system increased from 92.38% to 98.35%, when window size increased from 3x3 to 7x7. It is also true for recall; it increased from 85.58% to 92.17%.

#### 2.5.1.3 WSD to English Language

The Authors in [23] used a hybrid approach to word sense disambiguation by combining both supervised and unsupervised learning. In this work the proposed approach focuses on overcoming the limitation using learning set which is enriched in dynamic way maintaining new data. Trivial filtering method is utilized to achieve appropriate training data. They introduce a mixed methodology having Modified Lesk approach and Bag-of-Words having enriched bags using learning methods. Their approach establishes the superiority over individual Modified Lesk and Bag-of-Words approaches based on experimentation. The disambiguation accuracy is improved based on the enrichment of datasets having populated by new data. They have achieved better precision value, recall value, and F-Measure through extensive experimentation.

### 2.5.2 WSD for Local Languages

#### 2.5.2.1 WSD for Afaan Oromo Language

The work reported in [24] for Word Sense Disambiguation of Afan Oromo using unsupervised corpus-based approach. Five clustering algorithms (simple k means, hierarchical agglomerative: Single, Average and complete link and Expectation Maximization algorithms) have been tested on seven ambiguous words which contains 1501 sentences in the existing implementation of Weka 3.6.11 package. To make the sense example sentences ready for experimentation, the researcher has performed activities like tokenization, stop word removal and stemming. He used contextual co-occurrence feature that indicate word occurrence within some number of words to the left or right of the ambiguous word. In general, the system performs with average accuracy of 81.9% for simple K Means, 79.8% for EM and 76.1% for complete linkage respectively.

The research reported in [12] tried to solve the problem of word sense disambiguation (WSD) for Afaan Oromo language. He used a corpus based approach to disambiguation where supervised machine learning techniques are applied to a corpus of Afaan Oromo language, to acquire

disambiguation information automatically. He developed a WSD algorithm and he tried to cross check his systems with the linguists and an iterative improvement has been made on the system. The system uses information gathered from training corpus to assign senses to unseen examples. Hence, he developed Afaan Oromo corpus from the scratch. The corpus contains 1240 sentences and he evaluated for 5 Afaan Oromo ambiguous words. Finally, for the first experiment he has achieved 79% accuracy of the model and for the second experiment he has found that four-word window on each side of the ambiguous word is enough for Afaan Oromo WSD.

#### 2.5.2.2 WSD For Amharic Language

The work reported [10] apply an unsupervised corpus-based approach WSD to acquire disambiguation information automatically. He used five selected unsupervised algorithms such as Simple k means, EM and agglomerative single, average and complete link clustering algorithms. The researcher achieved an accuracy within the range of 65.1 to 79.4 % for Simple k means, 67.9% to 76.9% for EM and 54.4% to 71.1% for Complete Link clustering algorithms for the five ambiguous words with A total of 1045 English sense examples for the five ambiguous words were collected from British National Corpus (BNC). The sense examples were translated to Amharic using the Amharic-English dictionary and preprocessed to make it ready for experimentation.

The Author in [1] has conducted a research with the main objective of the research was to design a word sense disambiguation prototype model for Amharic words using semi-supervised learning method for five ambiguous words with a total of 1031 sense examples and to extract training sets which minimizes the amount of the required human intervention and produce considerable improvement in learning accuracy. The researcher incorporated unlabeled data by combining some seeds instances directly into the data representation (features), so that unsupervised algorithms can be directly applied for clustering based on instances similarity. He used a python program, which he made it in line with his preprocessing tasks. The experiment of semi-supervised methods using bootstrapping algorithms was conducted on the five Amharic WSD

datasets following semi-supervised clustering assumption. The experiment showed that the average performance results of Adaboost, bagging and ADtree algorithms are 84.90%, 81.25% and 88.45% . In addition, the researcher concluded that semi-supervised learning using bootstrapping algorithm performs better in his study and it is more adaptive on WSD for Amharic.

#### 2.5.2.3 WSD for Tigrigna Language

In the research done by [18] used unsupervised machine learning method to develop WSD prototype model for Tigrigna language. To do this, the researcher used unsupervised machine learning method that requires information that can be automatically extracted from untagged texts. He reported his experiment on four ambiguous words of Tigrigna language with a total 631 sense examples. Those datasets were collected from different online Tigrigna websites, newspapers, and Tigrigna bible. To disambiguate the selected four ambiguous words of the language; the researcher used five clustering algorithms which are available in Weka 3.8.1 package. Clustering algorithms selected by the researcher for this study were Simple K-means, Single Link, Average Link, Complete Link, and Expectation Maximization.

# CHAPTER 3

# Ge'ez Language

## 3.1 Overview of Ge'ez Language

Ethiopic writing system is used to represent Semitic languages. Some of the semitic language that are represented using Ethiopic writing system are: - Ge'ez, Amharic, Gurage and Tigrigna. These languages are confined to Ethiopia. However, no more mother tongue of any person in Ge'ez language, and still has a very significant role in the traditional language of literature and religion. Amharic and Ge'ez are closely related to each other. Ethiopic writing system is written from left to right. It does not make any distinction between upper and lower-case letters and has no conventional cursive form. There are no systematic variations in the form of the symbol according to its position in the word[25].

### 3.1.1 Geez Writting System

A study of the Ge"ez writing systems is essential to understanding the history of Ethiopia and the evolution and modern usage of the language. The only language in Ethiopia which has its own alphabets is Geez. Other languages like Amharic and Tigrigna adopted these alphabets fully from Ge"ez. An alphabet or fidel of a language represents its sound. They are also called hohyat (ሆህያት in Amharic). So as to that the total number of Ge"ez alphabets were twenty six each with vowel sound ሀ(e),ሁ(u), ሂ(i), ሃ(a), ሄ(e), ህ(i) and ሆ(o). However, every consonant was combined with seven vowel sound alphabets to produce seven derived alphabets [8]. For example, the combination of the original alphabet ሀ(ha) with the six vowel sound

alphabets yields six more derived alphabets as:ሁ(hu), ሂ(hi), ሃ(ha),ሄ(he), ህ(hi) and ሆ(ho) respectively[26].

Geez language is different in syntactic structure from the other languages. For example Amharic language sentence structure Subject Object Verb (SOV) but sentence structure for Ge'ez Subject Verb Object (SVO, "እግዚአብሔር ነበቦ ለሙሴ"), Verb Subject Object (VSO,"ወነበቦ እግዚአብሔር ለሙሴ") and Object Verb Subject (OVS,"ለሙሴ ነበቦ እግዚአብሔር"). In which ever types syn- tax it is in Amharic it follows SOV, which means"እግዚአብሔር ሙሴን ተናገር".The position of an adverb and adjective in Amharic is before the verb and noun respectively, while in Ge'ez it is used in both before or after verb and noun[27]. Generally, Ge'ez has essentially 26 main syllographs or alphabets, all consonants, and each with six more derivations while the rest are essentially those with additional strokes and modifications added on to the main forms to indi- cate a vowel sound associated with it or to make aural adjustments in the basic consonant sound [25]. Hence, it has a total of 182 syllographs, with twenty six alphabets and seven vowels (26×7
= 182 alphabets).

### 3.1.2 Ge'ez Punctuation Marks

Identifying punctuation marks is very important to know word demarcation for natural language processing. The Ge'ez writing system uses punctuation marks. Some of the most commonly used Ge'ez punctuation marks are: - **The word separator mark (:)** is used in the old literature to separate one word from other words. In the current literature, it is rarely used. As, a result a single space is used to separate words instead of this punctuation marks. **The end of sentence mark**(።)is used to shows when an idea is finished. **The sentence connector mark**(፤)is used to connect two sentences in to one sentence. **The list separator mark**(፣) is used to list things, separate parts of a sentence, and indicate a pause in a sentence or question. Nevertheless, the removal of punctuation marks increases the effectiveness and efficiency of natural language processing systems as morphological analyzer and stop word removal does [20].

### 3.1.3 Ge'ez Morphology

The main verbs in Ge"ez are perfect and imperfect. Perfect is usually past or completed action. It includes past perfect, past continuous, past participle with the relative pronouns ዘ (of). The imperfect one is usually present, continuous and future action. The end of all perfect verbs is the first order while all imperfect verbs have in the end the six orders. This is under the pronoun ውእቱ (he). The other main verbs are subjunctive (with ከመ, ኀበ or without), imperative, infinitive and gerundive [26].

To illustrate the importance of the verb some scholars [8] say that the verbs are the most important part of the Ge"ez language which serves as a basis for almost all other POS categories.. Ge"ez is studied by local and foreign scholars. Among local scholars Aleka Kidane Wold Kifle and Memhr Dessie Keleb are mentioned. The analysis of these scholars is used for this study to understande the language. Nowaday, the language doesn't have native speakers modern institutions, and the main resources are books. Moreover, the linguistic work of [28] is also used to understand the way of classification of Ge"ez verbs in the traditional schools of ቅኔ (ḳine) found under the Ethiopian Orthodox Tewahido Church.

## 3.2 Ambiguities in Ge'ez Language

Ambiguity is an attribute of any concept, idea, and statement or claim whose meaning, intention or interpretation cannot be definitively resolved according to a rule or process consisting of a finite number of steps. In ambiguity, specific and distinct interpretations are permitted , whereas with information that is vague, it is difficult to form any interpretation at the desired level of specificity. The ambiguity words in a language raised by speakers or writers can be disambiguated for understandability by listeners or readers. As a reasrcher stated in [27] there are six types of ambiguity in Amharic and those ambiguities are also present in Ge'ez in addition to character ambiguities. Those are: Phonological, Lexical, Structural, Referential, Semantic and Orthographic ambiguities. now We are summarized each type of ambiguity as follows.

### 3.2.1 Orthographic Ambiguities

Orthographic Ambiguity is resulted from geminate and nongeminate sounds.. The sentence is ambiguous between the following meanings. When we geminate the word, its meaning is “She ordered me to give service” and when we use nongeminate sounds, its meaning is “She asked me to bring a dish”.

### 3.2.2 Phonological ambiguity

It is a result due to the sound used for the word from the placement of pause with in a structure which occurs in speech. When the sentence is pronounced with pause it, its meaning is “He is misled or deceive person” but the meaning differs if it is pronounced without pause. It will mean “He is boring or tedious person”.

### 3.2.3 Lexical Ambiguity

Word meanings in more than one sense can lead to different interpretations by different individuals. The scope of lexical ambiguity is on individual words or wordlevel understanding. Lexical ambiguity refers to a case in which either a lexical unit belongs to different partofspeech categories with different senses, or to a lexical unit for which there is more than one sense, while these different senses fall into the same partofspeech category. There are three factors that can cause lexical ambiguity which are: categorical ambiguity, Homonymy and Homophonous Affixes and synonymy . **Categorical ambiguity**it refers an ambiguity which results from lexical elements which have the same phonological and homographic form but belongs to different word class . **Homonymy:** Homonymy is the state of a given word’s having the same phonological form (and possibly with same spelling) however with different meanings which will cause ambiguity . **Homophonous Affixes:** This ambiguity is resulted when affixes serve as different word classes. The word can be morphologically analyzed into three separate morphemes: the

prefix, the rootand suffix. Since the meaning of each morpheme remains the same across words, it creates similar words that are difficult to understand .

### 3.2.4 Semantic Ambiguity

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

### 3.2.5 Structural Ambiguity

Structural ambiguity results when word order becomes in unorganized manner and holds more than one possible position in the grammatical structure of the sentence. Syntactic disambiguation of words that can function as multiple partsofspeech is accomplished by reorganizing the order at the syntactic level.

### 3.2.6 Referential Ambiguity

Referential ambiguity is a type of ambiguity which arises when a pronoun has more than one possible antecedent.The following sentence is an example of such ambiguity. 1.Abebe was pleased because he graduated. 2.Somebody was pleased because Abebe graduated.

# CHAPTER 4

# System Design And Architecture

## 4.1 Introduction

This chapter describes the design of WSD system for Geez language. It mainly focuses on how the corpus is prepared, word selection, architecture of Geez WSD model, document preprocessing techniques, preparing machine readable datasets, and evaluation techniques of the model.

**Word selection:** According to different scholars, words that we want to disambiguate could be selected by the researchers from WordNet, which is available on the web or online, or from different sources of the language or documents of the language, which is annotated manually.
**WordNet:** WordNet is a lexical database, it provides a large repository of some languages lexical items, which is available online. The WordNet was designed to establish relations between the main four types of Parts of Speech (POS): noun, verb, adjective and adverb. WordNet defines the relations between synsets and relations between word senses. A specific meaning of one word under one type of POS is called a sense and synset represents the smallest unit in WordNet, which describes a specific meaning of a word. It includes the word itself, explanation and the synonyms of its meaning. The difference is that lexical relations are relations between members of two different synsets, however semantic relations are relations between two whole synsets[29]. The researchers in [29],[30] use multiple numbers of ambiguous words and those are 31 words for Amharic and 26 words for Afaan Oromo by using WordNet of each language.
**Words that are selected from language documents:** According to [14] the construction of the sense tagged corpus needs a great amount of time and cost. Due to this reason, we have stayed on the small number of ambiguous words. The corpora, which have sense information of all

words, have been built recently,but they are not large enough to provide sufficient disambiguation information of the all words. Therefore, the methods based on the sense tagged corpora have difficulties in disambiguating senses of all words.so as to that the selection of ambiguous words that were used in this study was based on the number of senses that a single word have. In [31] there are words that have multiple sense from two sense up to sixteen sense but in this study,we select the words that have two senses. Those ambiguous words are selected from ግስ (gis) which is found in ቅኔ (kinie) school in Ethiopian Orthodox Tewahido church. WSD performance can be affected by the distribution of training data for each sense that means number of sense examples are equal as much as possible and a balanced distribution of training data has been employed to maximize performance in the work of [10] then the Selected ambiguous words have the same number of sense examples for each ambiguous word as shown below in table 4.1

**Table 4.1:** selected ambiguous words with their sense examples

| Ambiguous words | Senses | No. of Sense examples | Total |
|---|---|---|---|
| ሀለፈ(halafe) | Died | 205 | 409 |
| | Pass | 204 | |
| ቆመ (ḱome) | Intercede | 182 | 364 |
| | cleave | 182 | |
| ባረከ (bareke) | curse | 171 | 342 |
| | bless | 171 | |
| አስተርአየ(astaraye) | Appear | 170 | 340 |
| | dipict | 170 | |
| ገብረ(gebira) | fabricate | 160 | 320 |
| | ensure | 160 | |
| ሰአለ(seala) | adjure | 172 | 344 |
| | cough | 172 | |
| **Total** | | | **2119** |

## 4.2 Data collection

In this research, we have used corpus-based approach. It is challenging to acquire sense annotated corpus for WSD studies due to lack of standard sense annotated corpus or context-based repository (Wordnets) for Geez language. Due to this reason, data collection becomes first rather than corpus preparation. By considering this, the researcher collected data from different sources such as Geez bible, Sinksar, Fithanegest, Gedile semaetat, and Bahir Dar university Geez Department learning and teaching materials. Here we were first collected a huge data that contains around 193,000 sentences or instances. To retrieve the sentences that contain the selected ambiguous words; we develop a simple algorithm and This simple algorithm accepts a word which is an ambiguous words from the user and then displays the sentences that contain the given ambiguous word. As explained in his study selecting sentences of ambiguous words from a variety of domains is very important to build efficient and reliable WSD prototype since similar domains usually restrict words to one sense[10]. Therefore, in order to build efficient and reliable WSD prototype for Geez language, we collected data from different domain areas. The following Figure shows how the corpus is searched from the collected data

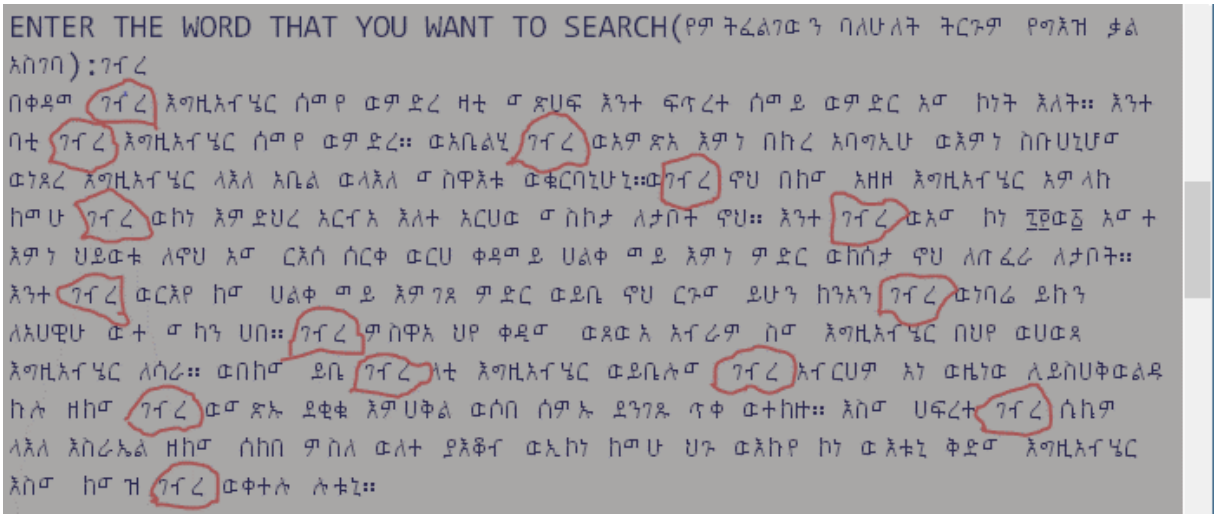



**Figure 4.1:** Sample Searching result

As it shown in Figure 4.1, the algorithm asks us to enter the word that we want to search from

the whole sentences that we have collected. For example, the word *"ገበረ"* is one of the ambiguous word of the language that we select in our study which means we need to have more numbers of sentences or instances which contain this ambiguous word. After searching, the sentences that contains all ambiguous words then the instances must be categorized in to their senses manually by language experts.In This study three annotators are involved(two of them are teachers of ቅኔ (kinie) and one of them is a teacher of ድጓ (digua)), who were communicate like native speakers of Geez language but they are not learn in modern institutions rather they are experts in the language in traditional schools of Ethiopian Orthodox Tewahido church.

## 4.3 Design of corpus

A corpus is a collection of texts used for linguistic analyses, which stores in an electronic database so that the data can be accessed easily by means of a computer. Corpus texts usually consist of thousands or millions of words and are not made up of the linguists or a native speakers invented examples but on naturally occurring spoken and written language. WSD systems need well organized datasets for training to make their accuracy attractive. Therefore, corpus preparation becomes must after data collection; by then we prepared a corpus from the documents that we collected from different sources. Because of the absence of standard datasets for Geez language, our corpus was limited to a total of 2119 Geez sentences or instances which contains 6 ambiguous words of Geez language to develop the proposed model.

## 4.4 Proposed System Architecture

The flow of activities that are used to develop the proposed WSD is given in Figure 4.2. The proposed system architecture contains different steps. The first step is accepting sentences that contains ambiguous words. The next step is applying preprocessing activities like normalization, tokenization, stop word removal, stemming and transliteration. Unsupervised learning shows that unlabeld datasets are given to the selected clustering algorithms to build WSD prototype

model of Geez language. Whereas, Supervised learning shows labeled datasets are given to the selected classification algorithms to build WSD prototype model of Geez language. Semisupervised shows that few number of labeled seed examples together with large number of unlabeled datatasets are given to the selected clustering algorithms inorder to have fully labeled datasets based on labeled seed examples. Those fully labeled datasets are given to the selected classification algorithms to build WSD prototype model of the language.

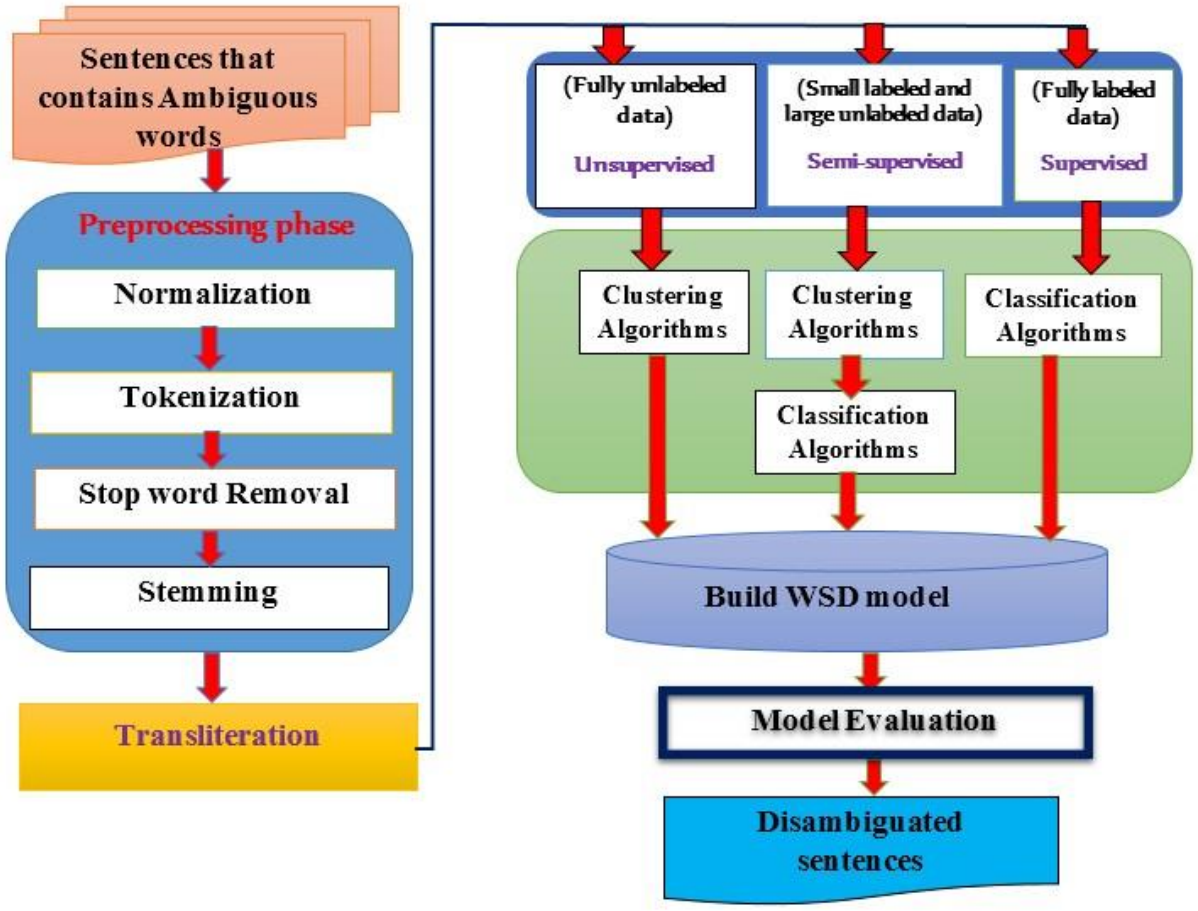



**Figure 4.2:** Corpus-Based Geez WSD system Architecture

**Preprocessing phase:** preprocessing describes any type of processing performed on raw data to prepare it for another processing procedure. Hence, preprocessing is the preliminary step which transforms the data into a format that will be more easily and effectively processed. Preprocessing must ensure that the source text be presented to NLP in a form usable for it. In this

study, preprocessing is a primary step to make our data sets compatible with the machine learning tool that was used in our study called Weka. In the preprocessing stage this study addresses tokenization, stemming, stop word removal, transliteration and normalization.

**Normalization:** In this study normalizing the characters is proceed because it is challenged for the next preprocessing stages. In Geez writing system characters with the same sound have different symbols. These different symbols must be considered as similar even if they have effect on meaning. As a result, in this study, some symbols of the same sound were converted to one common form. Thus, for example, if the character was one of ዐ,ኣ,ዓ (with the sound a) were changed to their equivalent respective orders of አ.Because in this study we use the word ሰአለ with meanings of ለመነ(adjure) and ሳለ(cough) but there may be the word ሰዐለ in our corpus with a meaning of ሳለ(draw).that is why we normalize the characters in such a way in order to use the exact word ሰአለ because ሰዐለ do not have the meaning ለመነ(adjure) and ሳለ(cough) rather its meaning is only ሳለ(draw). ሐ,ሓ,ሗ,ኀ and ኃ (all of them with a similar sound , h) then it was converted to ሀ to make ሀለፈ. By the same token, all orders of ሠ (with the sound s) were changed to their equivalent respective orders of ሰ to make ሰአለ.Generally we normalize the characters based on the words that we select in our study but not for all Geez language.

**Algorithm 1: Normalization**

Read corpus file from directory
For each token in file
If token is ሐ or ሓ or ሗ or ኀ or ኃ then token
is ሀ
elseifሠ
token is ሰ
elseif ዐ or ኣ or ዓ
token is አ . . .
End if
Endfor

**Tokenization:** tokenization is very important to this study. It is the process of breaking for the posted documents for the sentences into words or tokens. The corpus, which is a set of sentences first tokenized into words. Tokenization is done by identifying with the white spaces, comma (,) and special symbols between the words. All punctuation marks, numbers and special characters are removed from the text before the data is processed. Hence, These punctuation marks don't

have any relevance to identify the meaning of ambigous words using WSD. Therefore except '።' which is used to detect the end of the sentence, all other punctuations are detached from words in tokenization process. if a sequence of characters is followed by space, that sequence is identified as a word. A consecutive sequence of valid characters was recognized as a word in the tokenization process. Tokenization is used to get context words for disambiguation purpose. For instance, if we have a sentence like ወተርፈ ፡ አዳም ፡ ውስተ ፡ ምድረ ፡ ኤዶም ፡ ወቃየንሰ ፡ ሀለፈ ፡ ወሀደረ ፡ ታህተ ፡ ምስራቀ ፡ ኤዶም።Here if we tokenize this sentence it looks like ወተርፈ, አዳም, ውስተ, ምድረ, ኤዶም, ወቃየንሰ, ሀለፈ, ወሀደረ, ታህተ, ምስራቀ, ኤዶም and።

**Algorithm 2: Tokenization**

Read input corpus from file directory line by line
For each line split them using Geez sentence demarcation either of(፤,፡, ?, !, ።, ፣)
Add lines on ArrayList of string
For each lines in list of lines
    Split tokens using space delimiter
    Return tokens
Endfor

**Stop word removal:** After tokenization take place, we have removed Geez language stop words, hence it has no effect on meaning of the words. In this study, Stop word removal is used to remove stop words from the corpus because the absence or presence of these words has no contribution to identify appropriate sense. Not all tokenized words are necessary for this work hence, one word carry the meaning than other words and other words that have no own meaning.for this study we collected stop words which are conjunctions, prepositions and articles of the language because of absence of standard stop words. For instance, words such as (‘ባህቱ’, ‘እንተ’, ‘ከማሁ’, ‘ኩሉ’, ‘እምዚ’,’እስመ’, ‘ድህረ’, ‘እምነ’). Since stop words do not have significant discriminating powers in the meaning of ambiguous words; we filtered stop-words list to ensure only content bearing words are included Stop words are the words that is carried as the important meaning of the keywords. Nevertheless, stop words like ‘ሀበ’ and ‘መንገለ’did not remove from the corpus because they have a significant role on the word‘ሀለፈ’.

**Algorithm 3: Stopword removal**

```
Open a file that contains stop word list
Read a sentence from the corpus
        For each word in the sentence
                If a word is in stop word list then
                        Remove a word
                End if
        End for
Returntheremainingwords
```

**Stemming:** Stemming is the process of reducing morphological variants of words into base or root form. In morphologically rich languages like Geez, a stemmer will lead to significant improvements in WSD systems. In Geez language, there are different terms that are generated from the same root word due to their grammatical use. To create different derivational and inflectional word forms Geez language makes use of prefixes, suffixes, and infixes. Therefore, those extra words or characters that change the root word to different forms are stemmed from the corpus using the stemming algorithm developed by ours selves which is suitable for the language. This algorithm removes both prefixes and suffixes only since we developed affix removal of the stemmer. Therefore, to get the common form of the ambiguous word we tried to normalize infixes the root word manually. For Example, let's take the ambiguous word *'ገበረ'* may become like *'ገበሩ', 'ይገበር', 'ይግበር'*, etc after removing both prefixes and suffixes. To make it suitable for machine learning algorithms, we inspect manually all those words into one word which is*'ገበረ'*. The same thing was applied for other ambiguous words that are used in this study but not for the context words. The reason behind not applying normalization after stemming for the context words is due time taken to normalize all the context words that are used in this study.

**Algorithm 4: Stemming**

```
Read corpus file from directory
Read prefix list
Read suffix list
Assign the first 1, 2, 3,… character(s) of the token to prefix
Assign the last 1, 2, 3,… character(s) of the token to suffix
For each token in file
        If token prefix is in prefix list then
                Remove prefix from token
        End if
        If token suffix is in suffix list then
                Remove suffix from token
        End if
Endfor
```

**Transliteration:** after the above preprocessing tasks are done for Geez documents that we collect; transliteration was accomplished for Geez language documents. It is the representation of the characters of one language by corresponding characters of another language.In this study, the transliteration was accomplished from Geez characters into Latin characters. [1] noted in his study; doing transliteration makes documents compatible with the machine learning tool called Weka. Hence, it was used to only compatibility of the language with WEKA tool. Since we selected a machine learning tool (called Weka) for conducting our experiment; we applied transliteration because of WEKA platform uses ARFF or CSV and those file format can applied after translitration have been performed in order to compatible with the WEKA tool. The transliteration of the Geez corpus was conducted by using System for Ethiopic representation in ASCII (SERA).

## 4.5 Preparing Machine Readable Data

After preprocessing tasks and transliteration were applied for the corpus, preparing machine readable data became important for applying clustering and classifying activities. First, we prepared our corpus in the form text files. But for experimentation of this study, we used a machine learning tool which is Weka as a development tool for the proposed WSD prototype

model. Therefore, preparing corpora in the form of text file is not compatible (or readable) with Weka. To make it readable using Weka, datasets must be prepared in the form of CSV or ARFF file formats. Therefore, we prepared our datasets in the form of ARFF file formats. When we prepared our datasets, the target word (or ambiguous word) must be aligned in the middle column and context words were aligned to the right and left of the target word as shown in the following table.

**Table 4.2:** Description of attributes used for this study

| No. | Attribute | Description | Value |
|---|---|---|---|
| 1 | LContexts(i) | Used to hold the ith left word | Any word in the corpus |
| 1 | RContexts(i) | Used to hold the ith right word | Any word in the corpus |
| 3 | Target word | Holds the ambiguous word | Ambiguous word |
| 4 | Class | Holds the label of the target word | different sense of the word |

## 4.6 Preparing Datasets

In this study, we used corpus based approach. Since we used this machine learning method for this study, we prepared a combination of labeled datasets for supervised learning ,unlabeled datasets for unsupervised learning and semi-labeled datasets semi-supervised for training. Because, corpus-based approach uses both labeled and unlabeled datasets for training and testing. In unsupervised learning technique, that clusters similar items together. This approach finds inherent clusters in the unlabeled data. For clustering purpose, we used four clustering algorithms which are available in Weka. The clustering algorithms used in our study are: Expected Maximization (EM), Simple K-Means, Farthest First, Hierarchical Clusterer. We select those clustering algorithms because they have better efficiency than the other clustering algorithms for our datasets that we prepared for this study.but in supervised learning we must labeled our datasets.

According to [21] noted in their study, more number of sentences need not be annotated manually in semi-supervised machine learning approach and the representative labeled and unlabeled data size distribution for training set is typically 85-98% unlabeled datasets; and the rest

are for labeled datasets. By considering this, we prepared 12% of labeled datasets and 88% of unlabeled datasets for each of the six chosen ambiguous words from the total datasets before clustering.That means A word 'ገበረ' has 160 instances so from this 12% are labeled and 88% are unlableled which becomes 20 instances are labeled and 140 instances are unlabled from 160 instances. When we labeled seed examples automatically, we applied some techniques. Those techniques will be described in the following section.

**Seed selection Techniques:** In this section, the techniques that were applied in this study will be presented. This research was conducted by using corpus based approach. Therefore, both labeled and unlabeled documents must be available in this study because using semi-supervised approach requires both labeled and unlabeled training datasets. To do this, we used important techniques. Those techniques were taken from the research that was conducted by [1]. Techniques that were used in this study consist of four steps. Those steps are: -

- Selecting representative seed examples for each class or sense of the ambiguous words.
- Clustering both labeled and unlabeled seed examples using classes to clusters evaluation mode.
- Perform feature selection on fully labeled dataset and feature extraction.
- Build the final classifier.

**Step 1: Selecting representative seed examples for each class or sense of the ambiguous words:** Selecting representative seed examples for each class is effective; Those selected seed words are used to label unlabeled documents. As [1] described in his study, selecting seed words to select representative seed examples for semi-supervised approach is challenging task. That means if we select improper seed words, the selected seed examples would be improper. Then the result of selecting improper seed examples will be poor in performance. Improper seed examples can be selected when we tagged (labeled) our datasets randomly by humans. When humans select seed examples randomly, they may select improper seed words which means the selected seed words cannot differentiate the senses (meanings) of the ambiguous word.

To minimize manual selection of limited number of seed examples from the total datasets, we used tree algorithms because tree algorithms represent the related concept of the target word starting from the node. As[10] discussed, tree algorithms used information gain as lexical knowledge and information gain can minimize subjectivity problem in manual selection of seed examples. By considering this concept, we used ADTree algorithm to select seed words which are used to select seed examples from our datasets. We classified our datasets by using ADTree algorithm for each ambiguous words; then tree visualization would be taking place. After tree visualization we took seed words scoring high information gain in the tree structure. Those seed words were used for discrimination purpose of each sense of the ambiguous word. After all those processes were taking place, then we were selected instances which contain those seed words and assigned their senses depending on the selected seed words.

**Step 2: Clustering both labeled and unlabeled seed examples using "classes to clusters evaluation" mode:** Here, we could not use the resulting clusters for classification. We only use them to identify the cluster of missing instances based on Labeled seed examples. After clustering have been done using EM algorithm which is best performed in clustering algorithms for this study, we can see the effect of semi-supervised learning method in our work. The clustering result may show that the class labels of some seed examples were misclassified. However, automatic clustering suggests such label changes, the change was not required because those seeds were labeled with their sense class as the promise they are chosen by experts intentionally.

**Step 3: Feature Selection and feature extraction Feature Selection:** The success of machine learning requires instances to be represented using an effective set of features that are correlated with the categories of word senses. For this study, Feature selection was performed by preparing a Eight-Eight window size. Because in our Datasets the highest window size is Eight. Instances with missing values were also removed from the feature sets. Therefore, feature selection is a data reduction mechanism.

**Feature Extraction:** feature vector which represents words for each instance of a target word, i.e. files of comma-separated values, a line in WEKA with an extension of. arff or .csv. These vectors should represent a text window surrounding ambiguous word of a Eight-Eight words in our work case.

**Step 4: build the classifier:** Before classifying our datasets using the selected classification algorithms, we labeled our datasets manually depending on the selected seed words. Because manually labeled seed examples being used as cluster labels during clustering on both labeled and unlabeled documents (datasets). Knowing cluster label of each instance become important for differentiating the class of missed instances by taking each cluster as a distinctive class. This helps us to label unlabeled instances with their classes. We prepared our datasets with 18

**Table 4.3:** Example of WSD Dataset for Semi-Supervised Learning

| LContext3 | LContext2 | LContext1 | Target word | RContext1 | RContext2 | RContext3 | Class |
|---|---|---|---|---|---|---|---|
| ? | ? | emeze | Halefe | kaeba | bahere | horu | pass |
| ? | tanesio | emeheya | Halefe | halafa | behera | tirose | pass |
| halifo | kaeba | tirose | Halefe | wosidona | galila | maekala | pass |
| ? | tanesio | emeheya | Halefe | haba | behera | yehuda | ? |
| reeyo | soba | maseya | Halefe | haba | bitaneya | aseretu | ? |
| ahadu | aseretu | keleetu | Halefe | haba | liqana | kahenate | died |
| maseyo | halafa | aseretu | Halefe | aseretu | keleetu | aredaihu | died |
| ? | ? | sanita | Halefe | hagara | nayene | horu | ? |
| ? | ? | emeze | Halefe | soba | baseha | gize | ? |
| ahadu | beesi | kebure | Halefe | behera | rehuqa | yenesae | ? |

attributes which includes the target word and class. The rest 16 attributes are context words that are used to determine the meaning of the ambiguous word which means 8 words to the left and 8 words to the right of the target word. Attributes that exceeds from this size was removed because the researcher was focused on 8-8 window sizes.When we use eight words to the left and eight words to the right, there might be missing values. Those missing values were replaced by question marks, because question mark is compatible with Weka. Reducing dimensionality of datasets can improve the performance of WSD; because instances having redundancy and missing values problems will be reduced.[10]

## 4.7 Evaluation Techniques

To evaluate the performance, we depend on the objectives of our study and we selected the related metrics to our objectives. To evaluate performance of clustering and classification algorithms, four different modes are available in WEKA. Those are using training sets, supplied test

set, percentage split, classes to clusters evaluation mode and cross validation for both clustering and classification algorithms. However, for this study we used ‘classes to clusters evaluation’ mode for clustering algorithm and cross validation for classification algorithm.

When we use ‘classes to cluster evaluation’ mode, WEKA shows the clustering result as error rate using ‘classes to clusters evaluation’ mode. Therefore, accuracy of clustering algorithms was obtained after subtracting the error rate from hundred. This accuracy would be used to measure how well it has been able to generalize the clustering results.For this study 10-fold cross-validation evaluation technique is used in our experiment. In this technique, first the total data set is divided into 10 mutually disjoint folds approximately of equal size using stratified sampling mechanism. In stratified sampling, the folds are stratified so that the class distribution of the tuples in each fold is approximately the same as that in the initial data.

We have a total of 2119 manually tagged sense examples which is divided into 10 approximately of equal size. As a result of this each fold of a data set contains 212 sense examples with balanced distribution number of senses per fold.After identifin and sparated the training set and testing set from the total datasets,we remove manually tagged sense examples from test set.During this process nine fold were used for training to develop the system whereas the remaining tenth fold was used for testing the system that was trained on the previous nine folds. The process was repeated ten times by taking other nine as training and tenth one as testing. After each training phase, the system was tested on average of 212 Geez sentence. Each of the corresponding training set contains an average of 1907 sentences. The performance of classification algorithms is usually measured by parameters such as accuracy, recall, precision and F-measure. These performance parameters are the functions of the numbers of correctly and incorrectly classified instances which are obtained on the confusion matrix of WEKA output. Confusion matrix analyzes how well an algorithm classifies instances in their corresponding classes. performance measures are available for the effectiveness evaluation for WSD, the definition of almost all measures is based on the same 2×2 contingency table model that is constructed as shown in the following Table 4.4 [32]

**Table 4.4:** Evaluation Metrics

| Classified as | A | B |
|---|---|---|
| A | TP (True-Positive) | FN (False-Negative) |
| B | FP (False-Positive) | TN (True-Negative) |

Where TP refers to actual positive instances, which were correctly labeled, and FP refers to actual positive instances that were incorrectly labeled and FN refers to actual negative instances, which were correctly labeled.TN refers to actual negative instances that were correctly labeled.

**Precission:** is the most performance measurement in both traditional information retrieval and the WSD system and which measures the performance of the system that provides the information with the relatedness of the information retrieved. So in our cases precision in WSD is percentage of words that were labeled correctly, out of the words addressed by the system

$$Precission = \frac{TP}{TP + FP} \quad \text{.....(4.1)}$$

**Recall(R):**is another measurement of WSD to measure the performance of many system and it measures how much items are correctly classified.in our cases WSD is percentage of words that are tagged correctly, out of all words in the test set.

$$Recall = \frac{TP}{TP + FN} \quad \text{.....(4.2)}$$

**F1-Score(F1):** is the harmonic mean value of both precision and recall results as it is formula is shown below

$$F1 - Score = \frac{2TP}{2TP + FN + FP} \quad \text{.....(4.3)}$$

**Accuracy:**accuracy in WSD is percentage of correctly classified instances.

$$Accuracy = \frac{TN + TP}{TP + FN + FP + TN} \quad \text{.....(4.4)}$$

However; in this study, we used all evaluation metrics listed in the above in order to check with in different evaluation metrics, because they are common way of reporting the performance of machine learning methods.

# CHAPTER 5

# Experimental Result, Discussion and Evaluation

This chapter presents the implemented model and the experiment in this study with the result obtained for evaluating the performance of the study regarding the problems we studied and comparing different machine learning approaches. To do so, we need to have analysis of our data set that we used for implementing our model. The result generated by the implementation are discussed and the evaluation of the result obtained are done by using the metrics we used according to our problem dimensions. To achieve our specific objectives the following experiments or scenarios are conducted which are applied on our prepared datasets.

- Comparison of corpus based approaches which are supervised, semi-supervised and unsupervised with different modes ;
- Investigating the most effective approach and effective algorithm from the selected algorithms that improve the performance of Ge'ez WSD model;
- Experimenting with different context window size of ambiguous words.

## 5.1 Experimental Results and Discussion

**Experiment 1: Comparison of corpus based approaches, which are unsupervised, supervised and semi-supervised**

To compare results of unsupervised, supervised, and semi-supervised machine learning approaches; we used the same datasets of the language, and the classification algorithms for both semi-supervised and supervised approaches but for unsupervised approach we used the selected clustering algorithms which are Expectation maximization, Simple K-Means, Farthest First, Hierarchical Clusterer for clustering purpose of our datasets. We used clustering algorithms for unsupervised machine learning approach because clustering is unsupervised technique. Therefore, comparison of those three machine learning approaches was conducted on the same datasets. In addition to this comparison of semi-supervised and supervised approaches was conducted on the same datasets by using the same classification algorithms which are SMO, Naïve Bayes and Bagging for supervised and Adaboost and ADtree are used in and semi-supervised approach.

**Unsupervised Learning:** Unsupervised learning is an independent process where no supervision is involved during the learning step. Unsupervised corpus based methods do not rely on external knowledge sources such as MRD, concept hierarchies and sense tagged texts. Those approaches are mainly clustering approaches where words and contexts are clustered. During clustering, each cluster corresponds to a sense of a target word. The goal of clustering is to group together elements in a way which maximizes similarity between elements in one cluster and to minimize similarity between elements belonging to different clusters.

**Table 5.1:** Experimentation Results using unsupervised algorithms

| | **Accuracy of ambiguous words** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Algorithms** | halafe | ḱome | bareke | astaraya | gebire | seala | Average |
| Simple k-means | 52.41 % | 57.21 % | 50.76 % | 64.26 % | 51.25 % | 53 % | 54.82 % |
| Farthest First | 50.49 % | 54.89 % | 50.76 % | 51.21 % | 50.32 % | 51.5 % | 51.52 % |
| Expectation maximization | 66.82 % | 66.97 % | 67.33 % | 55.55 % | 51.25 % | 63.5 % | **61.90 %** |
| HierarchicalClusterer | 50.97 % | 50.24 % | 52.27 % | 55.08 % | 50.63 % | 52 % | 51.86 % |

The performance achieved by using the selected four clustering algorithms was in the range of 51.52 %-61.90 % accuracy. The results shown in the above table are the optimal results for each ambiguous word using the selected clustering algorithms. The performance that was achieved by using the selected clustering algorithms may be low because occurrences of common words on the selected datasets may be high. Occurring more common words on the selected datasets may lead to misclassification of their senses. Therefore, this difficulty results low performance for clustering algorithms. Generally, Farthest First algorithm results better in average accuracy (i.e 61.90 %) for our datasets when we compared with the other three clustering algorithms. Still the average accuracy that we obtained is not attractive and really not applicable for Geez WSD prototype model. Therefore, to improve the performance score of clustering algorithms other experiment is important by using other machine learning approaches. Then we conducted our experiment using supervised and semi-supervised machine learning methods as follows.

**Supervised Learning:** Supervised is the use of algorithms that reason from externally supplied instances (training set) to form classes to differentiate new data. The goal of supervised is to build a model of the distribution of class labels in terms of predictor features. In order to supervised to build the model it involves training and testing phases. During the training phase a sense-annotated training corpus is required, from which syntactic and semantic features are extracted to build a classifier using machine learning techniques and in testing phase the classifier tries to find out the appropriate sense for the word based on surrounding words present in the instances.

**Table 5.2:** Experimentation Results using supervised algorithms

| | **SMO** | | | | **Naïve Bayes** | | | | **Bagging** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **ambiguous words** | Precision | Recall | F1-score | Accuracy | Precision | Recall | F1-score | Accuracy | Precision | Recall | F1-score | Accuracy |
| halafe | 69.9 % | 68.8 % | 68.1 % | 68.7 % | 75.2 % | 75 % | 75 % | 75 % | 55.9 % | 55.8 % | 54.5 % | 55.7 % |
| kome | 83.8 % | 83.7 % | 83.7 % | 83.7 % | 85.2 % | 84.7 % | 84.6 % | 84.6 % | 73.9 % | 71.2 % | 70 % | 71.1 % |
| bareke | 76.4 % | 76.4 % | 76.4 % | 76.3 % | 78.9 % | 78.9 % | 78.9 % | 78.8 % | 71 % | 70.4 % | 70 % | 70.3 % |
| astaraya | 84.4 % | 81.2 % | 80.8 % | 81.1 % | 84.1 % | 84.1 % | 84.1 % | 84 % | 63.5 % | 61.4 % | 59 % | 61.3 % |
| gebire | 84.2 % | 84.4 % | 83.6 % | 84.3 % | 71 % | 69.7 % | 70.2 % | 69.6 % | 69.9 % | 72.2 % | 65.8 % | 72.1 % |
| seala | 67.8 % | 67.5 % | 67.4 % | 67.5 % | 64.6 % | 64 % | 63.6 % | 64 % | 59.1 % | 58.5 % | 57.8 % | 58.5 % |
| **Average** | **77.7 %** | **77 %** | **76.6 %** | **76.9 %** | **76.5 %** | **76.6 %** | **76.6 %** | **76 %** | **65.5 %** | **64.9 %** | **62.8 %** | **64.8 %** |

The results shown in the above table 5.2 are the performances achieved using supervised machine learning methods using cross-validation test options. Here, we gave fully labeled datasets to the machine learning tool which is Weka for experimentation purpose. After that, the results shown in the above was achieved. We achieved the best result using SMO algorithm when we compare to the other classification algorithms which are Bagging and Naïve Bayes for our datasets. The performance achieved using SMO algorithm was 77.7 %,77 %,76.6 % and 76.9 % in average for Precesion,Recall,F1-score and Accuracy. But the performance achieved using Bagging were 65.5 %, 64.9 %, 62.8 % and 64.8 % and using Naïve Bayes were 76.5 % ,76.6 %,76.6 % and 76 % for Precesion,Recall,F1-score and Accuracy respectively. From this end we can say that, SMO algorithm performs better on fully labeled datasets of Geez ambiguous words or using supervised learning methods with cross-validation.

**Semi-Supervised Learning** Semi-supervised techniques involve training information like in supervised but the information given at initial training phase is less. Semi-supervised or minimally supervised methods are gaining popularity because of their ability to get by with only a small amount of annotated reference data while often outperforming totally unsupervised methods on large data sets. There are a host of diverse methods and approaches, which learn important characteristics from auxiliary data and cluster or annotated data using acquired information.

**Table 5.3:** Experimentation Results using Semi-supervised algorithms

| | SMO | | | | AdaboostM1 | | | | ADtree | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **ambiguous words** | Precision | Recall | F1-score | Accuracy | Precision | Recall | F1-score | Accuracy | Precision | Recall | F1-score | Accuracy |
| halafe | 87.6 % | 87.5 % | 87.4 % | 87.5 % | 89.7 % | 89.6 % | 89.6 % | 89.5 % | 89.7 % | 89.6 % | 89.6 % | 89.5 % |
| ḱome | 93.7 % | 92.9 % | 92.8 % | 92.8 % | 93.8 % | 92.9 % | 92.8 % | 98.8 % | 98.6 % | 98.6 % | 98.6 % | 98.5 % |
| bareke | 94 % | 93.2 % | 93.1 % | 93.2 % | 92.9 % | 91.9 % | 91.7 % | 91.8 % | 92.9 % | 91.9 % | 91.7 % | 91.5 % |
| astaraya | 83.2 % | 76.3 % | 73.7 % | 76.3 % | 70.8 % | 71.1 % | 70.5 % | 71 % | 88.5 % | 84.2 % | 84.2 % | 84.2 % |
| gebire | 84.2 % | 84.4 % | 83.6 % | 84.3 % | 73.7 % | 76.3 % | 70.8 % | 71.1 % | 86.6 % | 86.6 % | 85.9 % | 86.5 % |
| seala | 85.7 % | 80 % | 79.2 % | 80 % | 85.7 % | 80 % | 79.2 % | 80 % | 96.7 % | 97 % | 96.5 % | 96.8 % |
| **Average** | **88 %** | **85.7 %** | **84.9 %** | **85.6 %** | **84.4 %** | **83.6 %** | **82.4 %** | **82.7 %** | **92.1 %** | **91.3 %** | **91 %** | **91.1 %** |

As it is shown in table 5.3 semi-supervised machine learning method using semi-supervised algorithms records better performance when we compared with unsupervised and supervised machine learning methods. It results really applicable and attractive results using most of

the classification algorithms specially using semi-supervised algorithms.because we achieved 88 %,85.7 %,84.9 %and 85.6 % for Precesion,Recall,F1-score and Accuracy respectively for SMO, 84.4 %, 83.6 %, 82.45 % and 82.7 % for Precesion,Recall,F1-score and Accuracy respectively for AdaBoostM1 and 92.1 %, 91.3 %, 91 % and 91.1 % for Precesion,Recall,F1-score and Accuracy respectively for ADTree algorithm. There fore For Geez ambiguous words, we obtained an average 92.1 %, 91.3 %, 91 % and 91.1 % for Precision, Recall, F1-score and Accuracy using ADTree algorithm. In addition to this when we compare semi-supervised with unsupervised and supervised machine learning methods, it records really applicable and attractive results using most of the classification algorithms specially using semi-supervised algorithms. Most of the selected classification algorithm improves their accuracy when we compared to results achieved in Table 5.2 With average performance for Bagging, SMO, and Naive Bayes using Precision, Recall, F1-score and Accuracy to supervised learning method and using semi-supervised learning method, we used SMO, AdaBoostM1and ADTree algorithms using Precision, Recall, F1-score and Accuracy. For Geez ambiguous words, we obtained 92.1 %, 91.3 %, 91 % and 91.1 %for Precision, Recall, F1-score and Accuracy using ADTree algorithm. For comparison purpose, We can take the maximum average performance or accuracy of the three machine learning approaches by using their best performing algorithms since they record best accuracy for all unsupervised, supervised and semi-supervised methods.The result is shown in figure 5.1 below.

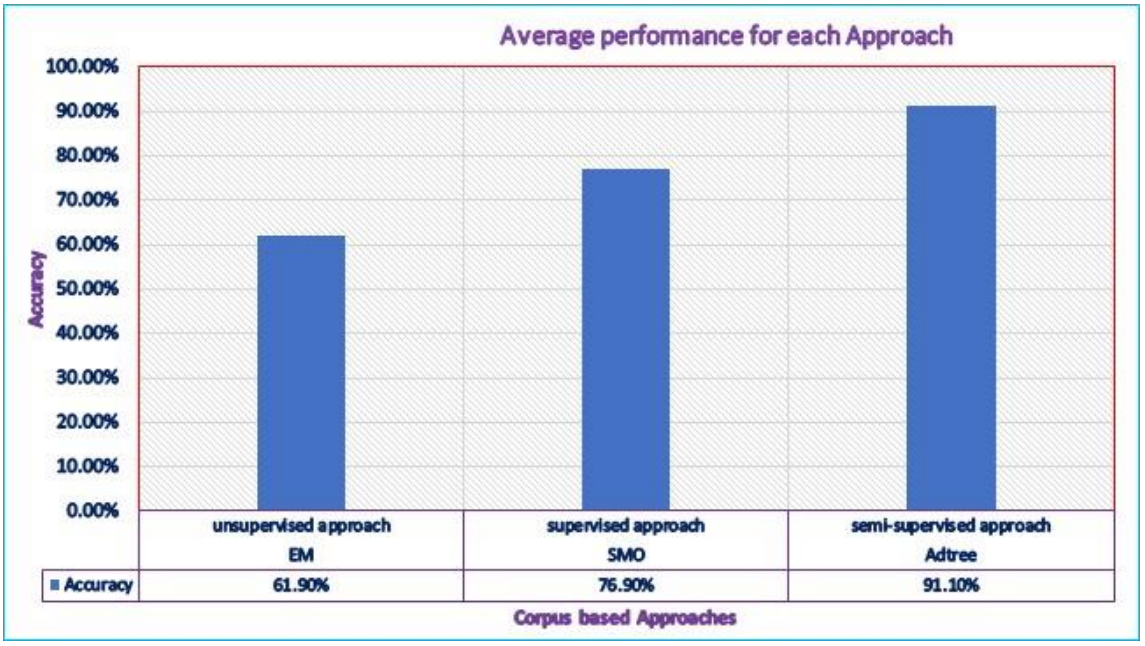



**Figure 5.1:** Average performance of the three-machine learning approaches

From the above figure 5.1 we can say that, semi-supervised machine learning improves the accuracy of WSD prototype model. By using both labeled and unlabeled datasets, the performance of WSD prototype model was improved compared with other approaches. This is because unlabeled datasets are clustered using manually labeled datasets during clustering. From this we can say that semi-supervised machine learning is the most important machine learning methods for the development of Geez WSD prototype model than supervised and unsupervised machine learning methods using bootstrapping ,which means using ADTree, AdaBoostM1 and SMO algorithms. We achieved the best performance of the classifier for Geez ambiguous words using semi-supervised corpus based approach, because seed words which obtained high information gain using ADTree algorithm were used for the selection of representative seed examples of this study.

**Experiment 2: Comparision of classification algorithm for Geez datasets**

For investigating the best performing classification and clustering algorithm for Geez WSD prototype model, we applied three approaches namely unsupervised, supervised and semi-supervised approaches for the selected six ambiguous words of Geez language. so we used the result achieved by using semi-supervised methods because the performances achieved by using those machine learning method were the most preferable when we compared with the performances

achieved by unsupervised and supervised learning method. For investigation purpose of those selected semi-supervised algorithms, we took average accuracy,precision, recall and F1-score obtained in the above using three machine learning algorithms. The comparison of those three classifying algorithms was based on the achieved performance to classify ambiguous words of Geez language. Those selected classification algorithms were taking place on the same Geez datasets. The result is shown in the following fig 5.2

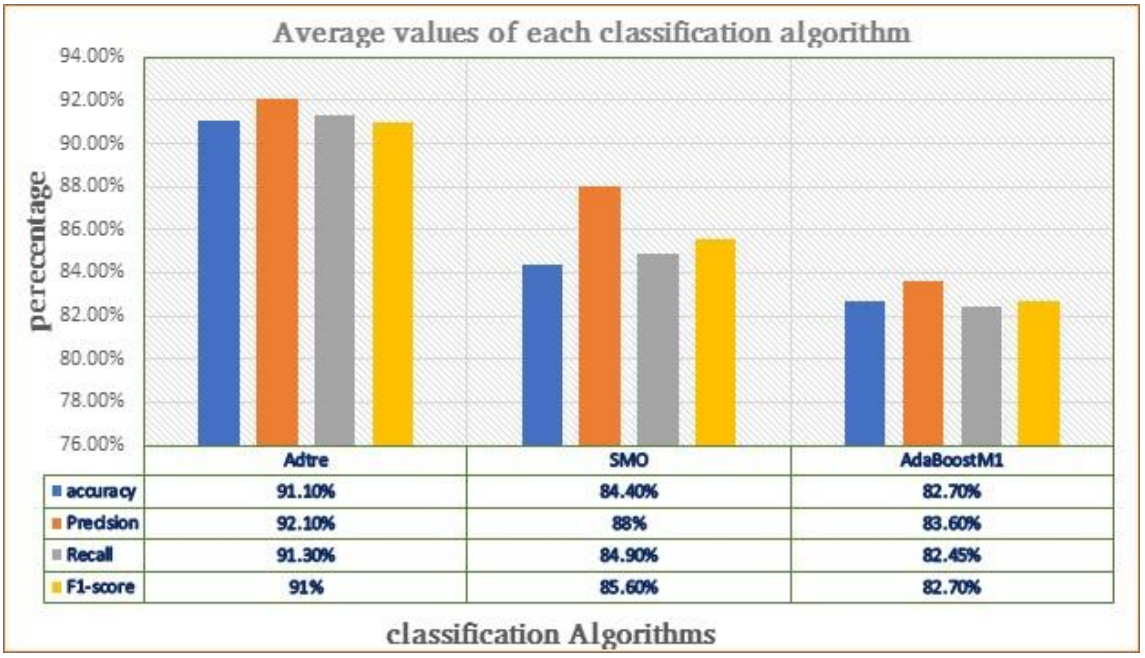



**Figure 5.2:** Average values of each classification algorithm

From fig, 5.2 we employed that ADTree algorithm is achieved an effective result for our datasets. We achieved an average Precision, Recall, F1-score and Accuracy of 92.1 %, 91.3 %, 91 % and 91.1 % respectivly. This algorithm became an effective for Geez WSD prototype model compared to the selected classification algorithms. Its efficiency was also better than AdaBoostM1 and SMO algorithm. Lastly, we can say that ADTree algorithm is the best performer algorithm for Geez WSD using semi-supervised learning method. AdaBoostM1 and SMO algorithms were also performed comparable result to each other for Geez WSD prototype model.

**Experiment 3: Determining the optimal context window size of the Language**

To find the optimal context window size, different studies have conducted using different WSD approaches for different languages. WSD researches were conducted for Amharic language by

different researchers starting from one-one to ten-ten window sizes to find out the optimal context window size for this language using different approaches. In study [11] was conducted a research for Amharic language using supervised machine learning method for five ambiguous words of the language (mesasat, meTrat, qereSe, Atena and mesal) and he was advised that window size 3-3 is an effective by using Naïve Bayes algorithm. According to [10] was also conducted a research for Amharic language using unsupervised machine learning approach and he was advised an optimal context window size of 3-3 or 2-2 is enough for disambiguation purpose depending on the algorithms he was used. As he discussed in his study, window size of 3-3 became optimal window size for Simple K means and EM clustering algorithms, and 2-2 window size became optimal window size for agglomerative SL and CL clustering algorithms on the same five ambiguous words of Amharic language (mesasat, meTrat, qereSe, ATena and mesal). The researcher in [1] was also another researcher who have done a research on Amharic WSD using semi-supervised machine learning method and he advised that the optimal window size is 2-2 or 3-3 window size using five classification algorithms for of the selected five ambiguous words of the language (ATena, derese, tenesa, ale, bela). The optimal window size 3-3 was effective for three of the bootstrapping and SVM algorithms (ADTree, AdaBoostM1, bagging, and SMO), and 2-2 window size became effective using Naïve Bayes algorithm. Due to those reason we used semi-supervised algorithms to determine the size of context window because Semi-supervised approach scores the highest accuracy than the others approaches.

**Table 5.4:** performances of each window size using SMO algorithm

| | **Ambiguous words and Accuracy result of them** | | | | | | |
|---|---|---|---|---|---|---|---|
| **window sizes** | halafe | kome | bareke | astaraya | gebire | seala | Average |
| One-One | 70.1 % | 96.7 % | 93.4 % | 95.1 % | 72.5 % | 83.5 % | 85.2 % |
| Two-Two | 75 % | 96.2 % | 91.4 % | 94.6 % | 74.3 % | 82 % | 85.5 % |
| Three-Three | 79.3 % | 95.2 % | 90.9 % | 93.7 % | 83.4 % | 82 % | 87.4 % |
| Four-Four | 80.7 % | 95.3 % | 90.9 % | 93.7 % | 83.7 % | 82 % | **87.7 %** |
| Five-Five | 82.2 % | 94.8 % | 90.4 % | 93.2 % | 60.3 % | 82 % | 83.8 % |
| Six-Six | 84.1 % | 94.4 % | 88.9 % | 93.2 % | 60.3 % | 82 % | 83.8 % |
| Seven-Seven | 84.6 % | 94.4 % | 87.9 % | 92.7 % | 60.3 % | 82 % | 83.6 % |
| Eight-Eight | 87.5 % | 92.8 % | 93.2 % | 76.3 % | 84.3 % | 80 % | 85.6 % |

As it is shown in the above Table 5.4, window size 8-8 became enough for the datasets of "halafe" and 1-1 for "ḱome" by achieving an accuracy of 87.5% and 96.7% respectively. Window size of 1-1, became the optimal window sizes for the datasets of "bareke", "seala" and astaraya by achieving an accuracy of 93.4%, 83.5% and 95.1% respectively. Window size of 4-4 become optimal for gebire. Here, different window sizes became the optimum window size for different datasets as. This was the difficulty to determine which window size became enough to differentiate the senses of the selected ambiguous words using SMO algorithm. Therefore,when we tried to calculate the average accuracy of each window size for determining the optimal window size. The window size which scores the highest average accuracy became the optimal window size for differentiating the senses of the selected datasets using SMO algorithm. However, from Table 5.4, window size of 4-4 scores the highest performance when we compared to other window sizes. Window size that scores highest performance became enough to differentiate the meaning of the selected datasets. Since window size of 4-4 scores highest performance in average, we concluded that window size of 4-4 became the optimal window size for the selected Geez datasets using SMO algorithm by achieving an average accuracy of 87.7%.

**Table 5.5:** Average performances of each window size using ADtree algorithm

| | **Ambiguous words and Accuracy result of them** | | | | | | |
|---|---|---|---|---|---|---|---|
| **window sizes** | halafe | ḱome | bareke | astaraya | gebire | seala | Average |
| One-One | 63 % | 97.3 % | 91.9 % | 94.6 % | 78.12 % | 81.5 % | 84.40 % |
| Two-Two | 73 % | 97.2 % | 91.9 % | 95.6 % | 71.87 % | 81.5 % | 85.1 % |
| Three-Three | 76.9 % | 97.2 % | 90.9 % | 95.6 % | 90.6 % | 81.5 % | 88.7 % |
| Four-Four | 81.8 % | 97.7 % | 93.5 % | 98.9 % | 92.72 % | 85.5 % | **91.6 %** |
| Five-Five | 81.2 % | 96.7 % | 93.5 % | 96.6 % | 60 % | 83.5 % | 85.2 % |
| Six-Six | 78.3 % | 96.7 % | 93.4 % | 95.6 % | 50.6 % | 83.5 % | 83 % |
| Seven-Seven | 76.9 % | 96.7 % | 92.4 % | 95.6 % | 50.6 % | 83.5 % | 82.6 % |
| Eight-Eight | 89.5 % | 98.5 % | 91.5 % | 84.2 % | 86.5 % | 96.8 % | 91.1 % |

As it could be seen in the above table 5.5, window size of 8-8 became enough for ambiguous word Halefe and Kome. Since, it achieved the accuracy of 89.5 % and 98.5 %. Window size of 4-4 became the optimal window sizes for the datasets of astaraye, gəbire and seala . Those window sizes became enough for those datasets, because they achieved the performance of 98.9 %,92.72 % and 85.5 %.From this we can understand that, finding the optimal window size

for all of the datasets using different algorithm became difficult; because all of the datasets have different optimal window sizes. Therefore, to determine the optimal window size for all of the selected datasets using classification algorithms. Hence, the above Table 5.5, window size of 4-4 became the optimal window size for determination of the meaning of the selected ambiguous words using ADTree algorithm by achieving an average accuracy of 91.6 %, which was the highest result. Therefore, window size of 4-4 became the optimal window size for identifying the meaning of the the selected ambiguous words. However we used the following table to find the optimal window size for the selected ambiguous words of the language using the rest of the selected classification algorithms.

**Table 5.6:** Average performances of each window size using AdaBoostM1 algorithm

| | **Ambiguous words and Accuracy result of them** | | | | | | |
|---|---|---|---|---|---|---|---|
| **window sizes** | halafe | ḳome | bareke | astaraya | gebire | seala | Average |
| One-One | 64.4 % | 93.4 % | 89.4 % | 90.3 % | 78.1 % | 82 % | 82.9 % |
| Two-Two | 68.2 % | 93.9 % | 89.4 % | 90.3 % | 75 % | 82 % | 83.1 % |
| Three-Three | 68.2 % | 93.9 % | 89.9 % | 90.3 % | 90.62 % | 82 % | 85.8 % |
| Four-Four | 68.2 % | 93.9 % | 89.9 % | 90.3 % | 90.6 % | 84 % | **86.1 %** |
| Five-Five | 66.8 % | 93.9 % | 89.9 % | 90.3 % | 68.7 % | 84 % | 82.2 % |
| Six-Six | 63.9 % | 93.9 % | 89.9 % | 90.3 % | 46.8 % | 84 % | 78.1 % |
| Seven-Seven | 63.4 % | 93.9 % | 89.4 % | 90.3 % | 46.8 % | 84 % | 77.9 % |
| Eight-Eight | 89.5 % | 92.8 % | 91.8 % | 71 % | 71.1 % | 80 % | 82.7 % |

As it is shown in the above Table 5.6, the optimal window size for “halafe” and bareke, is 8-8 and for gebire and “seala” became 4-4, by achieving the performance of 89.5 %, 91.8 % for“halafe” and bareke, and 90.64 %and 84.12 % for gebire and “seala” by using AdaBoostM1 algorithm respectively. Here, all of the Geez datasets have different optimal window sizes. This differentiation makes us difficult to find out the optimal window size for all of the Geez datasets using AdaBoostM1 algorithm. Then the window size that the highest performance can be the optimal window size for all of the selected Geez datasets. Therefore from above Table 5.6, window size of 4-4 became enough for sense determination of all the selected Geez datasets by achieving an average accuracy of 86.1 % using AdaBoostM1 algorithm. We said that window size of 4-4 became the optimal window size in order to differentiate the meaning of the selected Geez ambiguous words using AdaBoostM1 algorithm for our study. The reason that we have

said that window size 4-4 became the optimal window size because the highest performance was achieved on this window size.

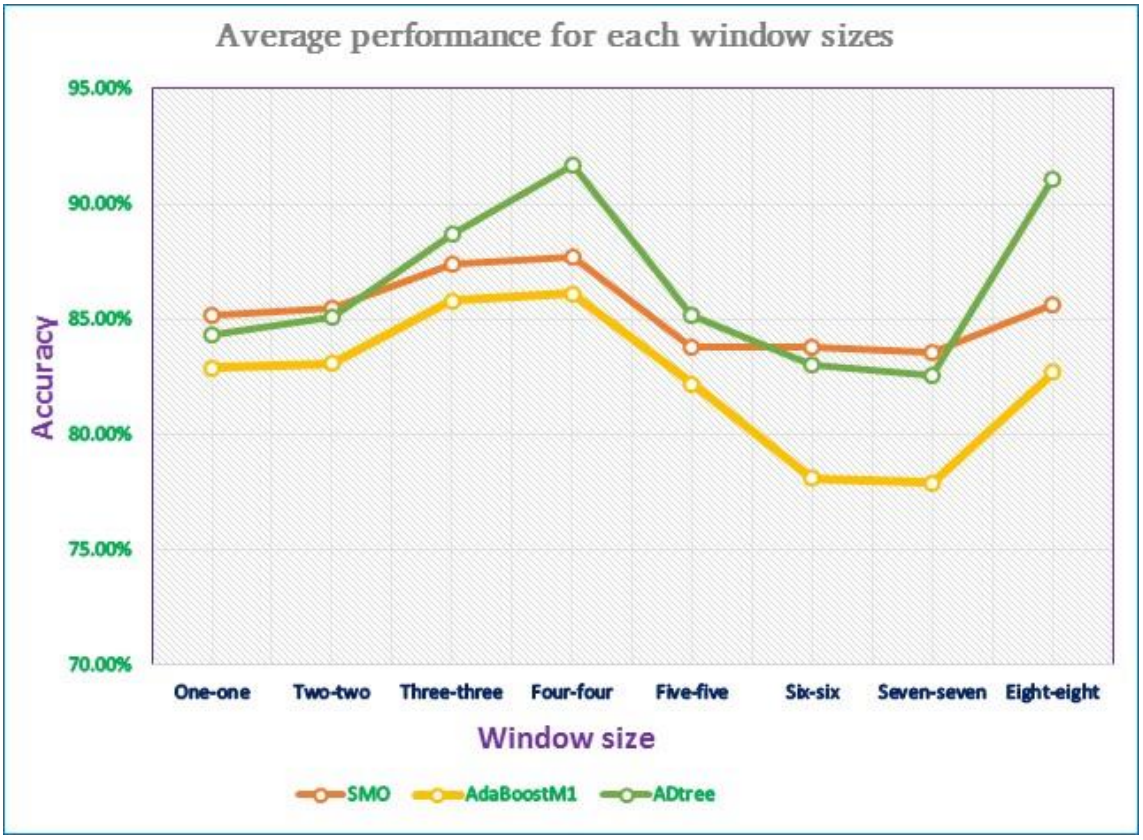



**Figure 5.3:** Average Accuracy of each window size using semi-supervised approach algorithms

From the above figure 5.3 we concluded that, semi-supervised algorithms perform much better than the other algorithms which means bootstrapping algorithms (ADTree, and AdaBoostM1) SVM algorithm (SMO). We know that ADTree, AdaBoostM1 and SMO achieved best performance on semi-supervised Datasets. However, our focus is which window size best performer to Geez language. Then all algorithms that are ADtree, AdaBoostM1 and SMO scores best accuracy on window size of 4-4.Therefore, window size 4-4 were best window size using ADTree algorithm for our Geez datasets. SMO was also the best performer algorithm next to ADTree algorithm for window size of 4-4 using semi-supervised learning method. Lastly, we can concluded that window size of 4-4 becomes best performer using ADTree algorithm for WSD prototype model of our Geez datasets depending on our experiments.

## 5.2 Comparative analysis of Geez WSD with other Languages

As described in chapter two, to the best knowledge of the researcher, there is no related works on word sense Disambiguation for Geez language. The current study tried to model and test machine learning perspective of Geez morphological analysis which includes all category verb forms[8] and stemming of geez language [32]. Even if we cannot compare on different data size and different language we compare our study with other local languages that uses the same Geez Fidel and one local language which uses English letter but those languages are different with Geez language in subject verb agreement as well as sentences structure. Those languages are Tigrigna and Amharic that was done before ours.

**Table 5.7:** Comparison of word sense disambiguation on local languages with Geez Language

| Languages | Approaches | No of words | Data size | Window size | performance |
|---|---|---|---|---|---|
| Tigrigna[18] | Unsupervised | 4 | 631 | 1-1 | Accuracy=67% to 83.3 % |
| Afaan Oromo[13] | Unsupervised | 5 | 1240 | 4-4 | Accuracy=79 % |
| Amharic[11] | Supervised | 5 | 1045 | 3-3 | Accuracy=65.1% to 79.4% |
| Amharic[1] | Semi-supervised | 5 | 1031 | 3-3 | Accuracy=81.25% to 88.45 % |
| Our work Geez | Three Approaches | 6 | 2119 | 4-4 | Precision=92.1%,Recall=91.3%,F1-score=91%,Accuracy=91.1% |

# CHAPTER 6

# Conclusion and Recommendation

## 6.1 Conclusion

There are so many words with more than one meaning in natural language and the meaning is determined by its context. The automated process of recognizing word senses in context is known as Word Sense Disambiguation (WSD). It is believed to helpfull increase the performance of many applications such as information retrieval (IR), information extraction (IE), speech recognition (SR) and others. There are three main approaches used for assigning senses to words in a context. These are knowledge-based approaches, corpus-based approaches, and hybrid approaches. From those approaches, we select corpus-based approaches for this study. Corpus-based approach is further divided as a supervised, unsupervised, and semi-supervised approach.

This research work is the first attempt to develop a word sense disambiguation system for Geez Language using corpus-based approaches. As there is no large size corpus which is already prepared for Geez language for WSD purpose, we prepared Geez corpus manually. Although the Geez language has many ambiguous words, we selected only six ambiguous words to build the Geez WSD prototype model. These words are:- ሀለፉ (Halafe), ቆመ (ḳome), ባረከ (bareke), አስተርዓየ(astaraya), ገብረ(gebira), ሰዓለ (Se'ale) For those selected ambiguous words of Geez language, a separate dataset has been prepared. The datasets were composed of a total of 2119 sentences which were collected from different domain areas.

To make comfortable for the selected machine learning tool which is Weka, the ambiguous word was written in the middle of sentences with sets of surrounding contexts to the left and right for

disambiguation purpose. Eight to the left and Eight to the right was the maximum window sizes used for this work. In this study, three experiments have been conducted using different classification and clustering algorithms. The first experiment was a comparison of the results obtained using the three machine learning methods which means unsupervised, semi-supervised, and supervised learning methods. In this experiment, a semi-supervised approach has performed better compared to the other machine learning methods. Since the semi-supervised learning method was employed in this work, we used the final fully labeled dataset which is obtained from unlabeled data sets. Those unlabeled datasets were labeled after clustering using the clustering assumption. But for clustering purposes, we used the EM algorithm because the EM algorithm performs better compared to the other selected clustering algorithms. The second experiment was determining the best performing algorithm for the selected Geez datasets.

To do this experiment, first we used seed words those selected by using the ADTree algorithm. which means representative seed examples were selected depending on the seed words that have high information gain using the ADTree algorithm. from this end the best performing algorithm for the selected Geez datasets was the ADTree algorithm compared to both clustering and classification algorithms that were selected for this study. The last but not least experiment that was conducted in this study was investigating the optimal window size for determining the senses of each ambiguous word. Using all of the selected algorithms window size Four-Four became enough for differentiating the meaning of the selected Geez datasets.

Therefore, we conclud that the window size of 4-4 can be considerd as optimal window size for Geez WSD systems. In general, we conclud that semi-supervised learning using bootstrapping algorithms and SVM are a potential learning methods that perform better in our study. They are more adaptive to WSD for the Geez datasets. Specifically, ADTree, AdaBoostM1, and SMO are potential algorithms to be applied for Geez WSD systems using semi-supervised learning corpus-based approaches and ADtree is the best performing Algorithm for this language from those semi-supervised algorithms.

## 6.2 Recommendation

Word sense disambiguation researches require a variety of linguistic resources like thesaurus, WordNet, Machine-Readable Dictionaries and effective Geez language stemmer and there is no standard stop word of the language in which we faced a significant challenge as Geez lacks those resources. Lack of sense annotated data for the language was also another challenge that makes us to limit our study on six ambiguous words of the language. And this makes us to limit our dataset on 2119 sentences or instances only. Therefore, we have the following recommendations which include the development of resources and future research directions for WSD of Geez texts:

Researches in WSD for other languages used linguistic resources like Thesaurus, WordNet, machine-readable dictionaries and effective Geez language stemmer. Due to the unavailability of those resources for Geez language, researchers could not research the WSD of Geez language. The following points are recommended for future work:

- This study tries to addres words that have two senses so the researcher would be add the words that have more than two senses.
- This study has only concentrated on modeling WSD to tackle lexical ambiguity which is at word level. Further researches would be recommended to address other types of ambiguities in Geez language like Character and structural ambiguity (sentsnces level).
- Researchers should investigate the effect of applying part-of-speech tagging for WSD prototype model of Geez language by using the data that we have used in this study.
- In addition to the corpus-based approach there are also knowledge-based and hybrid approaches that were used for WSD of other languages.Therefore, we recommend that these approaches need to be investigated for Geez language as well.
- This paper didn't use sound and voice and uses only textual information so for the future the reasercher use such kinds of data in addition to textual information.

# REFERENCES


[1] Getahun Wassie, Babu P Ramesh, Solomon Teferra, and Million Meshesha. A word sense disambiguation model for amharic words using semi-supervised learning paradigm. *Science, Technology and Arts Research Journal*, 3(3):147–155, 2014.

[2] Constituency Parsing. *Speech and language processing*. 2009.

[3] JW Perry. *In Encyclopedia of library and information science*. New York: Marcel Dekker, Inc, 1977.

[4] Jisha P Jayan, RR Rajeev, S Rajendran, et al. Morphological analyser and morphological generator for malayalam-tamil machine translation. *International Journal of Computer Applications*, 13(8):0975–8887, 2011.

[5] Jumi Sarmah and Shikhar Kumar Sarma. Survey on word sense disambiguation: an initiative towards an indo-aryan language. *IJEM*, 6(3):37–52, 2016.

[6] Nancy Ide and Jean Véronis. *Introduction to the special issue on word sense disambiguation: the state of the art*, volume 24. MIT Press, 1998.

[7] Pranjal Protim Borah, Gitimoni Talukdar, and Arup Baruah. Approaches for word sense disambiguation–a survey. *International Journal of Recent Technology and Engineering*, 3(1):35–38, 2014.

[8] Yitayal Abate. Morphological analysis of ge'ez verbs using memory based learning, 2014.

[9] Guido Minnen, John Carroll, and Darren Pearce. Applied morphological processing of english. *Natural Language Engineering*, 7(3):207–223, 2001.

[10] Solomon Assemu. Unsupervised machine learning approach for word sense disambiguation to amharic words, 2011.

[11] Solomon Mekonnen. Word sense disambiguation for amharic text: A machine learning approach, 2010.

[12] Workineh Tesema, Debela Tesfaye, and Teferi Kibebew. Towards the sense disambiguation of afan oromo words using hybrid approach (unsupervised machine learning and rule based), 2016.

[13] Tesfa Kebede. Word sense disambiguation for afaan oromo language, 2013.

[14] Teklay Muruts. Word sense disambiguation for tigrigna language using semi-supervised machine learning approach, 2018.

[15] Piroska Lendvai, Antal van den Bosch, and Emiel Krahmer. Memory-based disfluency chunking. In *ISCA Tutorial and Research Workshop on Disfluency in Spontaneous Speech*, 2003.

[16] Antal van den Bosch, Sander Canisius, Walter Daelemans, Iris Hendrickx, and Erik Tjong Kim Sang. Memory-based semantic role labeling: Optimizing features, algorithm, and output. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 102–105, 2004.

[17] Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. A feature-relevance heuristic for indexing and compressing large case bases. In *Poster Papers of the Ninth European Conference on Machine Learing*, pages 29–38, 1997.

[18] Meresa Mebrahtu Reda. Unsupervised machine learning approach for tigrigna word sense disambiguation. *philosophy*, 9(6), 2018.

[19] Animut Belay Asres. A semi-supervised approach for amharic news classification. 2012.

[20] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.

[21] Mohamadreza Mahmoodvand and Maryam Hourali. Semi-supervised approach for persian word sense disambiguation. In *2017 7th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 104–110. IEEE, 2017.

[22] Asma Naseer and Sarmad Hussain. Supervised word sense disambiguation for urdu using bayesian classification. *Center for Research in Urdu Language Processing, Lahore, Pakistan*, 2009.

[23] Alok Ranjan Pal, Anirban Kundu, Abhay Singh, Raj Shekhar, and Kunal Sinha. A hybrid approach to word sense disambiguation combining supervised and unsupervised learning. *arXiv preprint arXiv:1611.01083*, 2015.

[24] Feyisa Gemechu Shoga. Unsupervised corpus based approach for word sense disambiguation to afaan oromo words. 2015.

[25] Halefom Tekle Weldegebriel, JinXiu Chen, and Defu Zhang. Deep learning for ethiopian ge'ez script optical character recognision. pages 540–545, 2018.

[26] ደሴ ቀለብ (መምህር). ትንሣኤ ግእዝ. በኢትዮጵያ ኦርቶድክስ ተዋሕድ ቤተክርስተያን በሰንበትትምህርት ቤቶች ማደራጃ መምሪያ ማኅበረቅ ዱሳን, 2010.

[27] Tadesse Kassa. Morpheme-based bi-directional ge'ez-amharic machine translation, 2018.

[28] Leuel Tesfasilassie, Abiy Yigzaw, and Muluken Andualem. The focus of teaching geʿ ez noun and pronoun words in google newspaper.

[29] Seid Hassen Yesuf and Yaregal Assabie. Amharic word sense disambiguation using wordnet. 2015.

[30] Shibiru Olika. Word sense disambiguation for afaan oromo: Using knowledge base. 2018.

[31] Leykun E Berhanu. *Contemporary Challenges in the Ministry of the Ethiopian Orthodox Church*. PhD thesis, Howard University, 2005.

[32] Sofia Visa, Brian Ramsay, Anca L Ralescu, and Esther Van Der Knaap. Confusion matrix-based feature selection. *MAICS*, 710:120–127, 2011.

[33] Abebe Belay Adege and Yibeltal Chanie Mannie. Designing a stemmer for ge'ez text using rule based approach. LAP LAMBERT Academic Publishing, 2017.

## Appendix A: Sample Geez Stop words adopted from [33]

ኩሎ
ዛ
ኮነ
ወ
ኮኑ
ኩሎ
መኑ
እም
እም
ቦ
ነያ
ናሁ
ከመ
ዘ
አው
ሀበ
ዝ
ባህቱ
አፍአ
ነገረ
እንተ
ከማሁ
ላእል
ታህት
ላእለ
ኩለሄ
እምዛ
እስመ
መኑመ
በከመ
ድህረ
ብዙህ
መካን
እምነ
ታህት
ውእቱ
እስከ
እምዝ
አኮኑ
ዳእሙ
እምዛ
አብይ
ውስተ
ውሳጤ
ደጊም
ምስለ
መንገለ
ንስቲት
ከዊኖት
ከዊኖመ
እምከመ
ከዊኖቶ
ዘእንበለ
ማእከል
ወእምነ
ወእንዘ
ዘይትረከቡ
ገቢሮት
እምዛመ
እንዘይከውን
ህየንተ
እምድህረ
እመይከውን
ባህቲቱ
ብዝህት
ትማልም
ሀለወትቦ
ሀለውቦ
ነይውሙ
እምዝንቱ
እምታህት
እምውስተ
እምምስለ
እምቅድም
ሀበቅድም
እንበለ
ዘታህት
ዘምስለ

## Appendix B: Sample instances of Geez language

- አስተርአየ እግዚአብሄር ለፈርኦን ለዘሀለይወ ዘይገብሮ
- እግዚአብሄር ጠየቆ ብሂሎ አስተርአየኒ ዘረከ
- በይነ ንጽህናሁ ሀበ ካህኑ ድህረ አስተርአየ ደቄሁ ሀበ ስጋሁ ድበ እንዘ ሰፋ አመአዲ ይቀርብ ሀበ ካህኑ
- ድበ ኪሩብኤል ነቢሮ ዘሮጸ ወእም ነፋሲሂ ከዊኖ አስተርአየ ክንፈ
- ይቤለኒ እግዚ አብሄር ባርኮ ለወልድከ ዘተረክበ እምፍሬ ከርስከ
- ባረከት ሜልኮል እለተ ልደታ በእንተ አውሰበት ዳዊትሀ ንጉሰ እስራኤል
- ባረከቶ ኤልሳቤጥ ለወልደ ዮሀንስ አመ ነበረት ውስተ ሀቅል ምስሌሁ
- ባረከወ እስራኤል ለሙሴ በይነ አሀለፎሙ ሳጢቆ ባህረ ኤርትራ
- ወአቤልሂ ገብረ ወአምጽአ እምነ በኩረ አባግኢሁ ወእምነ ስቡሀኒሆሙ ወነጸረ እግዚአብሄር ላእለ አቤል
- ወገብረ ኖህ በከመ አዘዞ እግዚአብሄር አምላክ ከማሁ ገብረ
- እንተ ፍጥረተ ሰማይ ወምድር አመ ኮነት እለት እንተ ባቲ ገብረ እግዚአብሄር ሰማየ ወምድረ
- ዉእተ መካን ሀበ ገብረ ምስዋአ ህየ ቀዲሙ ወጸዉአ አብርሀም ስመ እግዚአብሄር
- ወካእበ ከረዮን ይስሀቅ ለአዘቃተ ማይ እለ ከረዩ አግብርተ አቡሁ አብርሀም ዘደፈኑ ሰብአ ፍልስጥኤም እምድህረ ሀለፈ
- ወረሲኦ ሀለፈ ወወደይዎ ሀበ አዝማዲሁ ረሲአ ወፈጺሞ መዋእሊሁ ቀበርዎ ኤሳዉ ወያእቆብ ደቂቁ ወሀሎ አሀዱ ሀበ አቡነ ዘይንእስ ወካልኡሰ ሀለፈ
- ወተረፈ አዳም ዉስተ ምድረ ኤዶም ወቃየንሰ ሀለፈ ወሀደረ ታህተ ምስራቀ ኤዶም
- ወዉእቱሰ ቅድሜሆሙ ሀለፈ ወሰገደ ዉስተ ምድረ ሰብእ እሰከ ይበጽህ ሀበ እሁሁ
- ኤርምያስ ሰአለ በእንተ አቤሜሌክ ኢትዮጵያዊ

- መርዶኪወስ ሰአለ አስቴርሀ ከመ ታድህኖሙ እሞት
- ይስሀቅ ለዘይበልአ ምግብ ፕፕሬ ህየንተ በዝሀ ቦቱ ሰአለ
- ማእከለ ክልኤ ቀላያት ለዘረከቦ ነምር ፍሪሆ ህየንተ ሮጸ ሰአለ
- ጸዉአሙ ወልድ ለአህዉየ ከመ ይቁሙ ምስለ አቡሆሙ
- ተጽህቀ ወልድ ህየንተ ዘህጥአ ዘይቀዉሞ ምስለ አቡሁ ወእሙ
- ሰራዊተ ሰማያት ኩሉ በጸጋሙ ወበየማኑ ቀዊሞሙ ርኢኩ
- አንሰአ ኤልያስ ለዘወድቆ መንጦላእት ተመይጦ በዮርዳኖስ ቆመ

**Appendix C: Stemmer coding for Geez texts**

```
package Amlakie geez stemmer ;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
public class sam
public static char b[], l;
public static int k, i, j, i-end, i-beg;
public static String str2 = ” ”;
public static int x = 0;
private static final int INC = 50;
public sam()
b = new char[INC];
i = 0;
i-end = 0;
l=0;
public static void add(char ch)
if (i == b.length)
char[] new-b = new char[i + INC];
for (int c = 0; c < i; c++)
new-b[c] = b[c];
b = new-b;
b[i++] = ch;
public void add(char[] w, int wLen)
if (i + wLen >= b.length)
char[] new-b = new char[i + wLen + INC];
for (int c = 0; c < i; c++)
new-b[c] = b[c];
```

```
b = new-b;
for (int c = 0; c < wLen; c++)
b[i++] = w[c];
public static void main(String[] args)
b = new char[100000];
try (FileReader reader = new FileReader(”Stopsera.txt”))
int ch;
while ((ch = reader.read())!= -1)
if ((char)ch = ’ ’)
b[x] = (char) ch;
x += 1;
else
i = x;
stem();
for (int p = l; p < k+1; p++)
str2 += b[p];
b = new char[100000];
str2 += ’ ’;
x = 0;
catch (IOException e)
e.printStackTrace();
try
FileWriter writer = new FileWriter(”C:/Users/Abiy/Desktop/Stemsera.txt”, true);
writer.write(str2);
writer.close();
catch (IOException e)
e.printStackTrace();
```