ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION SCIENCE

UNSUPERVISED MACHINE LEARNING APPROACH FOR WORD SENSE DISAMBIGUATION TO AMHARIC WORDS

SOLOMON ASSEMU

JUNE, 2011

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION SCIENCE

UNSUPERVISED MACHINE LEARNING APPROACH FOR WORD SENSE DISAMBIGUATION TO AMHARIC WORDS

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Science

By
SOLOMON ASSEMU

JUNE, 2011

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

UNSUPERVISED MACHINE LEARNING APPROACH FOR WORD SENSE DISAMBIGUATION TO AMHARIC WORDS

By

SOLOMON ASSEMU

Name and signature of Members of the Examining Board

Name	**Title**	**Signature**	**Date**
_____	Chairperson	_____	_____
_____	Advisor(s),	_____	_____
_____	Advisor(s),	_____	_____
_____	Examiner,	_____	_____

ACKNOWLEDGMENT

I sincerely thank my thesis advisor Ato Ermias Abebe, for his critical comments on my work, for being my driving force throughout this thesis and his patience in helping me complete it within deadline. I am also grateful to our research group members Lakechew Yayeh and Gedefew Mehari for sharing different resources and ideas.

I owe a considerable debt to Ato Solomon Mekonnen for initiating the research idea, providing different research materials and giving me extremely useful technical assistance in finishing this research. My thank goes to my friends Tsehay Wasihun, Abduselam Ali, Hailemariam Abebe, Eshete Dereb, Zewdie Mossie and Temesgen Haile for their support and encouragement.

My foremost gratitude goes to my father, Ato Assemu W/agagnehu, who is my inspiration, and my mother, Wro Ejgayehu Alem, who is my idol. My greatest love and gratefulness goes to them for always being there for me, for loving me so unconditionally, and for having faith in me when I myself couldn't have it.

My special thanks goes to my brothers, Mastewal and Netsanet, my sister Aynadis, for making extremely supportive and encouraging in difficult times to become my life prettier, and also, for passing all those hardships I had to go through easier. No words seem to express my foremost gratitude for my brother, Mastewal, for his invaluable support, and above all, for being the greatest brother. I couldn't have been able to make it without him.

Finally, I thanks to Addis Abeba University for their financial support to my work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ACRONYMS

AI	Artificial Intelligence
BNC	British National Corpus
IR	Information Retrieval
IE	Information Extraction
MRD	Machine Readable Dictionary
MT	Machine Translation
NLP	Natural Language Processing
EM	Expectation Maximization
SL	Single Link
CL	Complete Link
AL	Average Link
ANNs	Artificial Neural Networks
OALD	Oxford Advanced Learner's Dictionary
SERA	System for Ethiopic Representation in ASCII
GHSOM	Growing Hierarchical Self-Organizing Map
IA	Inter-Annotation Agreement
WSD	Word Sense Disambiguation

ABSTRACT

Word Sense Disambiguation (WSD) in text is still a difficult problem as the best supervised methods require laborious and costly manual preparation of tagged training data. This work presents a corpus based approach to word sense disambiguation that only requires information that can be automatically extracted from untagged text. We use unsupervised techniques to address the problem of automatically deciding the correct sense of an ambiguous word based on its surrounding context. It was motivated by its use in many crucial applications such as Information Retrieval (IR), Information Extraction (IE), Machine Translation (MT), etc. For this study, we report experiments on five selected Amharic ambiguous words, these are *አጠና (eTena), መሳል (mesal), መሣሣት (me`sa`sat), መጥራት (metrat),* and *ቀረጸ (qereSe).*

For the purposes of this research, unsupervised machine learning technique was applied to a corpus of Amharic sentences so as to acquire disambiguation information automatically. 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.

We tested five clustering algorithms (simple k means, hierarchical agglomerative: Single, Average and complete link and Expectation Maximization algorithms) in the existing implementation of Weka 3.6.4 package. "Class to cluster" evaluation mode was selected to learn the selected algorithms in the preprocessed dataset. The achieved result was encouraging, because best clustering algorithms were close in terms of accuracy of supervised machine learning approaches on the same dataset, using the same features. But, further experiments for other ambiguous words and using different approaches will be needed for a better natural language understanding of Amharic language.

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

Word Sense Disambiguation (WSD) deals with contextual resolution of lexical ambiguity [1]. Most words in natural language have more than one lexical meaning or sense, but usually only one of them is active in a given context. Fundamentally, WSD deals with choosing the correct sense (i.e., meaning) of a word in a given text from a list of possible senses based on the content [2]. WSD is important problem for applications in domain of Natural Language Processing (NLP). Machine translation (MT) cannot work without some form of disambiguation, WSD can be helpful also for information retrieval (IR), information extraction (IE) and lexicography among others [2].

WSD is a hard problem. Most difficulties arise from the fact that the concept of a meaning is vague. Usually, there are no clear boundaries between one sense and the other [3]. Typically, the problem of defining meaning is tackled with using dictionaries, which are called sense inventory in a context of WSD, i.e., from the algorithmic point of view sense inventories are used to specify all the meanings that a given word has. Now, the goal of WSD can be stated as choosing appropriate sense from sense inventory in a given context of a word [2].

Over the years, there have been several robust, stand-alone WSD systems designed to operate with minimal assumptions about the type of information available from other processes [1]. Each of the systems has employed several common WSD approaches such as Artificial Intelligence (AI)-based[2], [3], [4], [5] knowledge-based[6], [7], and corpus-based, also known as Machine learning approach[1], [7], [8]to perform the word sense disambiguation task.

AI methods began to flourish in the early 1960's and began to attack the problem of language understanding. As a result, WSD in AI work was typically accomplished in the context of larger systems intended for full language understanding. In the spirit of the

times, such systems were almost always grounded in some theory of human language understanding which they attempted to model and often involved the use of detailed knowledge about syntax and semantics to perform their task, which was exploited for WSD.

With knowledge-based approaches, the machine readable dictionaries (MRD) provide both the means for constructing a sense tagger along with the necessary target senses that will be employed in the system [9]. Lesk [10] first implemented an approach in which all of the sense definitions of the word to be disambiguated were retrieved from the dictionary. Each of the senses was compared to the dictionary definitions of all the remaining words in the context.

There are supervised and unsupervised and bootstrapping or combination approaches to WSD based on machine learning [2]. Supervised learning focuses on the usage of manually disambiguated examples of text snippets containing ambiguous words. We need to choose an appropriate sense inventory in advance, at early stages of the construction of supervised WSD system. Some features are extracted from those text snippets or contexts and classifiers are trained using this manually labeled data.

Nevertheless, there is an issue in creation of resources used for automatic system performing WSD. This is especially evident in creation of corpora manually annotated or tagged with senses, which are used for training machine learning classifiers in a supervised setting. There are two important problems during manual sense tagging of a corpus: low interannotator agreement (IA) and high cost of annotation process. IA is a way of measuring how much annotations assigned by one annotator deferrers from annotations assigned by another annotator. IA is used for estimation on the performance on automatic WSD. Typically, it is not enough to give a value of percentage agreement, because agreements and disagreements may arise by chance. Artsteinand Poesio[41] is widely used in computational linguistic community for this purpose. The cost of annotation is high, because large effort is required during manual annotation. Mihalcea [8] estimated that a construction of a corpus with sufficient amount of data for supervised

classification algorithms for 20 000 ambiguous words would require 80 man-years of work.

On the other hand, unsupervised algorithms can be used. The amount of manual labor required is much lower in learning without supervision. Unsupervised approaches to WSD tend to use unlabeled data and automatically find sense distinctions. Usually those methods involve some form of clustering. Harris' distributional hypothesis [7] can be used as a theoretical foundation for unsupervised methods of WSD. It states that "meaning of words is related to the restrictions on combinations of these words relative to other words."

The Bootstrapping method; it is a combination of supervised and unsupervised methods that deals with far few resources. In essence, the initial classifier is constructed with a small amount of labeled instances using any of the supervised methods and then is employed to extract a larger training set from the unlabeled instances.

The major concern of this research was to investigate unsupervised machine learning approach to WSD for Amharic words, test the results in order to develop a bit further natural language understanding for Amharic word disambiguation and compare the results with supervised approach [23] because, both approach used the same ambiguous words, the same data set and the same feature set for disambiguation. Unlike that of supervised, unsupervised WSD system deals with grouping of contexts for given word that express the same meaning without providing explicit sense labels for each group (e.g., without using a dictionary) [37].

As with the other languages, Amharic has many words that have multiple meanings, for example the Amharic word *"መሳል"* have three meanings in different contexts. It was translated into English as *"to sharpen"*, "to cough", or *"to vow"*. When we look up a word in any dictionary, it can be seen that a word may have many meanings some of which are very different from the other. Given these complications, it is important for Amharic WSD to correctly determine the meaning in which a word used.

1.2 STATEMENT OF THE PROBLEM

According to the latest census results, Amharic is a mother tongues of more that 21 million people (which is 29% of the total population). The language is also used as a second language for over 5 million people [11]. It has also been, for a long period, the principal literal language and medium of instruction and school subject in primary and secondary schools of the country. Moreover, it is the working language of the Ethiopian Federal Government and five of the regional states, all of which make the language to be predominantly used in word processing activities in different offices. Furthermore, there are also a large number of documents published and recorded in Amharic. Thus researches which are conducted in the language benefit a significant number of the language speakers.

Word sense ambiguity is a central problem for many established Human Language Technology applications (e.g., machine translation, information extraction, question answering, information retrieval, text classification, and text summarization) [1]. This is also the case for associated subtasks (e.g., reference resolution, acquisition of sub-categorization patterns, parsing, and, obviously, semantic interpretation). For this reason, many international research groups are working on WSD, using a wide range of approaches. However, current state-of-the-art accuracy is in the range 60–70%, WSD is one of the most important open problems in NLP[12] .

Ambiguities have been an issue in researches conducted in Amharic language. Yehenew [13] indicated that both lexical and structural ambiguities were challenges in research on machine translation of English to Amharic. Yoseph[14] has also faced problems of synonym, polysemy and homonymy in his research on Amharic-English cross language information retrieval. The challenge has also been noticed as Atelach, et al [15] attempted to translate Amharic queries into English “Bags-of-words”. They were required to perform manual disambiguation which misses domain specific senses and also time taking [16]. In addition, like any other manual system the process of disambiguation may result in error.

There are also researches that were conducted to deal with ambiguities in Amharic language. Atelach [17] and Daniel [18]tried to resolve structural ambiguity using statistical approaches for parsing. Wube [19] also attempted to resolve structural ambiguities using a rule based approach.

As discussed earlier, there are many uses for word sense disambiguation. The most common are application of WSD in machine translation, information retrieval, speech processing, text processing, grammatical analysis, content and thematic analysis. The absence of Automatic WSD would make the development of such NLP and IR applications difficult.

To the researcher's knowledge, Teshome [20] was the first research attempt in WSD for Amharic which tries to resolve lexical ambiguity. He demonstrated word sense disambiguation based on semantic vector analysis in order to improve the effectiveness of an Amharic Information Retrieval system.

However, Machine Learning approach has been used successfully for WSD in other language like English [1], [7], and [8], Chinese [21], Hebrew and German [22]. Solomon [23] was the first researcher that employ supervised machine learning approach for Amharic WSD, and the achieved accuracy was 70% to 83% in training and test set.

However, supervised ML approach has been the following limitations. First, among the techniques of machine learning approaches such as, supervised, unsupervised and bootstrapping, the researcher uses only Naive Bayes algorithm to build the model. Second, Owing to luck of sense annotated sentences and linguistic resources the study was limited to experiment five ambiguous words only. Finally, as supervised WSD requires manually labeled sense examples, which is time taking and so exhaustive when the number of corpus size increased.

Despite, unavailability of sense tagged corpora used for WSD research for Amharic language; manual sense-tagging is very difficult, time taking and limiting the number of sense tagged words to be used. To deal with this problem unsupervised approach to WSD technique has been proposed by [23], to a void knowledge acquisition bottleneck in

supervised approach. In unsupervised WSD system deals with grouping of contexts for given word that express the same meaning without providing explicit sense labels for each group e.g., without using a dictionary.

Therefore, the major concern of this research was to investigate unsupervised machine learning approach for Amharic WSD, test the results in order to develop a bit further natural language understanding and compare the results with supervised approach that were studied before[23]. More specifically, this study aims using unsupervised machine learning approach WSD for Amharic words.

1.3 OBJECTIVE OF THE STUDY

1.3.1 General objective

The general objective of this research was to investigate the application of unsupervised machine learning techniques to word sense disambiguation of Amharic texts:-

1.3.2 Specific objective

To achieve the general objective, the study attempts to address the following specific objectives:

- ✓ Study ambiguities in Amharic so as to understand WSD issues in the language;
- ✓ Acquire training and test data set (corpus);
- ✓ Compare the selected algorithms in the task of unsupervised Word Sense Disambiguation for Amharic words;
- ✓ Build and train WSD model using the selected unsupervised machine learning algorithms;
- ✓ Evaluate the performance of the model;
- ✓ Forward conclusion and recommendations.

1.4 METHODOLOGY

1.4.1 Literature review

Different literatures that are considered to be relevant for the research were reviewed for the accomplishment of every stages of the study.
The topics covered include:

- ✓ Word sense disambiguation in both local and other languages;
- ✓ Different Machine learning approaches, techniques and their advantage and disadvantage;
- ✓ Ambiguities in Amharic language, Amharic Writing system, punctuation marks and its syntactic structure;
- ✓ Different clustering algorithms and their application in machine learning technique;

1.4.2 Data collections

In this study, unsupervised learning approach was selected to develop a model. In this approach a significant number of sense examples are required to make training possible for the algorithms, which is difficult to get for Amharic. For other languages like English, German and French a standard sense annotated data are available and used for WSD research. After reviewing available literatures, approaches that use monolingual of another language to acquire sense examples is used for this study. The corpus used in this study was previously used in research on supervised WSD for Amharic [23]. According to Solomon [23] an English corpus, British National Corpus was used to acquire sense examples for Amharic ambiguous words and the examples are translated to Amharic.

Due to lack of Amharic ambiguous words acquiring, translation and annotation of sense examples for this research, the researcher uses the corpuses and ambiguous words selected by [23] and make an arrangement that fits this research direction. Solomon [23] used five ambiguous words selected by a linguistic expert from a list of Homonyms

collected by Girma [24]. The selected words are *ኦጠና (eTena), መሳል (mesal), መሣሣት (me`sa`sat), መጥራት (metrat),* and *ቀረጸ (qereSe).* In addition to the basic words their variations are also considered. Girma [24] compiled the senses of the selected words and it was included in Appendix A. The summary of the senses for each ambiguous word are presented in table 1.1.

Ambiguous Word	Senses		
	Sense1	**Sense2**	**Sense3**
eTena	strengthen	study	-
mesal	cough	sharp	vow
me`sa`sat	taking care	thin	-
metrat	call	clean	-
qereSe	record	shape	-

Table 1.1 Senses of selected ambiguous words

We acquired a total 1045 sentences for five ambiguous words from previous work [23], on average a total of 100 sentences were used for each sense of ambiguous word.

1.4.3 Tools and techniques

In building the WSD model, the researcher used five unsupervised algorithms that are found in the existing implementation Weka 3.6.4 package. But we tried to choose algorithms representing a few different approaches to the problem of clustering[25].

We started with simple k-means algorithms, which represent simple, hard and flat clustering methods. We choose agglomerative single, average and complete link algorithms for representative family of hierarchical clustering algorithms. Last but not least, we test also the Expectation Maximization algorithms also known as the EM which is probabilistic clustering algorithms.

Weka 3.6.4 machine learning tool was selected due to the familiarity of the researcher to the tool and because of its accessibility, processing capability and language independent features.

1.5 EXPERIMENTATION

1.5.1 Data processing

The source data, which are English sense examples was translated to Amharic, stemmed, transliterated to Latin script, in order to improve the result of the selected clustering algorithms. In the experiment there was no need to split the data into training and test sets for evaluation because of unsupervised nature of clustering algorithms.

1.5.2 Training and testing

The system was trained for the five ambiguous words using a set of unannotated instances of the ambiguous words to create a model. A total of four experiments were carried out using “Class to Cluster” evaluation technique with different features and its parameters to train the model. Finally, the performances of the clustering algorithm were evaluated using the maximum accuracy of their result.

1.5.3 Evaluation technique

We evaluated our method (unsupervised machine learning approach) using sources of sense-tagged corpus. In supervised learning, sense-tagged corpus was used to induce a classifier and then applied to classify test data. Our approach, however, was purely unsupervised and the sense tagged corpus was used to carry out an evaluation of the discovered sense groups. The way the tool used for processing clustering depends on the cluster mode one selects. For this study “class to cluster” evaluation mode was selected. In this mode Weka first ignores the class attribute and generates the clustering and during the test phase it assigns classes to the clusters based on the majority value of the class attribute within each cluster. Based on the above technique its prediction accuracy was used to measure how well it has been able to generalize the clustering result[26].

1.6 SIGNIFICANCE OF THE STUDY

The results of this study was expected to produce experimental evidences that demonstrate different application areas of unsupervised machine learning technique to word sense disambiguation of Amharic texts. It also contribute to future researches and development in the area of Natural Language Processing specifically in machine translation, speech processing, text processing, Information Retrieval, grammatical analysis, content and thematic analysis as those areas require word sense disambiguation as complement.

1.7 SCOPE AND LIMITATION OF THE STUDY

There is supervised, unsupervised and bootstrapping machine learning techniques for WSD, due to time constraint to train, test and analyze the results, only five unsupervised machine learning algorithm were used to build and evaluate the WSD model. Because of unavailability of sense annotated data and linguistic resources; the study was limited to the experimentation of five ambiguous words.

1.8 ORGANIZATION OF THE THESIS

The thesis was organized into six chapters comprising Introduction, Literature review, the Amharic Language, Methodology, Experimentation and Discussion and Conclusion and Recommendations. The first chapter gives the general introduction of the thesis. The second chapter presents reviews made on different literatures regarding Word Sense Disambiguation together with its approaches and different machine learning techniques. The third chapter reviews the Amharic writing system and ambiguities in the language. The fourth chapter discusses the methodology, which is composed of corpus preparation, system architecture and clustering and evaluation technique. The fifth chapter discusses the experimentation and discussion of the findings. Finally, chapter Six deals with the conclusion and the recommendations drawn from the findings of the study.

CHAPTER TWO

LITERATURE SURVEY

In this chapter literature in the field of Word Sense Disambiguation (WSD) is reviewed and discussed. The chapter covers brief background and history, application areas and discussion on major approaches that have been employed for WSD research with special focus on a machine learning approach or corpus based approach which is used in this study. Moreover, machine learning algorithms that are tested to perform well for WSD research including unsupervised algorithms which is going to be employed in this research and others are discussed. The discussion on different approaches and algorithms would help the understanding of the central problem in WSD research and also facilitates the comparison of existing approaches to the specific solutions that are employed in this study.

2.1 INTRODUCTION

Many words in many natural languages including Amharic have multiple meanings or senses. For Example, an Amharic word "*መኳኳት*" can mean *"to take care"* or *"to be thin"* in different contexts which lead to ambiguity. Humans resolve such ambiguity by understanding the context of each ambiguous word in a document and its sound. But for a machine, it will be difficult to determine the meaning of ambiguous words. For machines, there is a need for Word sense disambiguation which is the task of automatically determining the meaning of an ambiguous word from its context. So in our example above, given the ambiguous word "*መኳኳት*", WSD involves interpreting the surrounding context of the word and analyzing the properties exhibited by the context to determine the right sense of "*መኳኳት*".

According to Ide [27] WSD involves two steps. The first step is to determine all the different senses for every word relevant to the text or discourse under consideration, i.e., to choose a sense inventory from the lists of senses in everyday dictionary, from the synonyms in a thesaurus, or from the translations in a translation dictionary. The second step involves a means to assign the appropriate sense to each occurrence of a word in

context. All disambiguation work involves matching the context of an instance of the word to be disambiguated either by using information from external knowledge sources or with contexts of previously disambiguated instances of the word. For both of these sources, we need preprocessing or knowledge-extraction procedures representing the information as context features. However, it is useful to recognize that another step is also involved here: the computer needs to learn how to associate a word sense with a word in context using either machine learning or manual creation of rules or metrics. Unless the associations between word senses and context features are given explicitly in the form of rules by a human being, the computer will need to use machine learning techniques to infer the associations from some training material.

WSD is an “intermediate task”, which is not an end in itself, but rather is necessary at one level or another to accomplish many natural language processing tasks such as Information Retrieval(IR) [28], Machine translation(MT)[29] and Question and answering(QA) [30]. For MT, WSD is important when it comes to selecting the appropriate target language word for an ambiguous source language word. For IR, sense disambiguation would prevent the retrieval of irrelevant documents that contain query words of a different sense, while use of semantic tags could help in solving the prepositional phrase attachment problem. In question answering systems, WSD is used to retrieve the appropriate answer from document collection for a given query containing ambiguous words.

2.2 HISTORY OF WSD

WSD is one of the oldest problems in computational linguistics. It was in the 1940s that WSD was first formulated as a separate task [31]. 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. [31]] discussed a 'window size' around the target word, taking every word within a distance N before and after the target word, He found that using a window size of only 2 words either side of the target word

offered no substantial difference in disambiguation accuracy than using the whole sentence.

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, etc. Very soon it became clear that WSD was a very difficult problem, also given the limited means available for computation. Indeed, its acknowledged hardness [32]was one of the main obstacles to the development of MT in the 1960s. WSD was then resurrected in the 1970s, within the research in artificial intelligence on complete natural language understanding. Wilks's preference semantics, as mentioned in[33], was one of the first systems to explicitly account for WSD.

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 [10].

King [34] 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. Yarowsky[35] 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). WordNet made it possible for all the researchers to have easy and free access to a standardized inventory using which to compare their work. Its hierarchical structure, synsets, and other such features made it the most used general sense inventory in WSD research.

The typical approach in WSD so far has been supervised learning, where systems are trained on manually tagged corpora. Statistical approaches in supervised learning were used by [33] and several others which was a foresight to the so-called statistical

revolution" in the 1990s. Brown[36] was the first to use corpus-based WSD in statistical MT.

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, there by introducing scientific rigor and uniformity. SensEval eventually became the primary forum for all WSD evaluations.

In SensEval-3 (2004), 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 is 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. This thesis employs unsupervised WSD and attempts to draw upon the idea that unsupervised WSD is the way to go in future.

2.3 BASIC METHODOLOGICAL APPROACHES TO WSD

Different approaches have been used through the evolution of WSD research. Many approaches have been proposed for assigning senses to words in context, although early attempts only served as models for toy systems. Currently, there are three main methodological approaches in this area: knowledge-based, corpus-based and hybrid approach [27]. In this section the survey of these approaches can be presented.

2.3.1 Knowledge-based Approaches

Under this approach disambiguation is carried out using information from an explicit lexicon or knowledge base. Since corpus based approaches require considerable amount of work to create a classifier for each word in a language, as a result researchers tend to work on few words. Knowledge-based approaches use an explicit lexon like, Machine

Readable Dictionaries (MRD), thesauri, computational lexicons such as WordNet or (hand-crafted) knowledge bases as information source to resolve lexical ambiguities for many words [29].

Lesk [10]created knowledge bases which associate each sense in a dictionary with a signature composed of the list of words appearing in the definition of that sense. Disambiguation was accomplished by selecting the sense of the target word whose signature contained the greatest number of overlaps with the signatures of neighboring words in its context. Because of the fact that dictionaries are created for human use, not for computers, there are some inconsistencies [10]. Although they provide detailed information at the lexical level, they lack pragmatic information used for sense determination.

Thesauri provide information about relationships among words, most notably synonymy [37]. 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 [27]. And 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[12]. Lexicon like WordNet is used for sense evaluation and for similarity measure in WSD. For example [38]created a knowledge base from WordNet's hierarchy and apply a semantic similarity function to accomplish disambiguation, also for the purposes of information retrieval.

2.3.2 Corpus-Based Approaches

A major challenge facing WSD research is the ability to acquire a large number 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 which 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 [27].Corpus based approaches can be categorized into three sub classes based on the form of training : Supervised WSD, unsupervised WSD and Bootstrapping Approach to WSD[27].

2.3.2.1 Supervised Word Sense Disambiguation

Supervised Word Sense Disambiguation use machine-learning techniques to learn a classifier from labeled training sets, that is, sets of examples encoded in terms of a number of features together with their appropriate sense label (or class)[12]. The systems in the supervised learning approach category are trained to develop a classifier that can be used to assign a yet unseen example to one of a fixed number of senses. That means, there is trained corpus, where the system learns to classify and a test corpus which the system must annotate. So, supervised learning can be considered as a classification task.

Supervised learning requires labeled training data where every instance in the training data is associated with an output value or label that can be thought of as a special attribute or feature for each instance. For WSD, every instance in the training data should be assigned a label that corresponds to the correct sense of the ambiguous word that the instance contains or represents. Machine learning algorithms make use of the instance attributes or features in the training data and generate a model to predict the label of any given instance. This model can be applied to unseen instances to predict their labels. Algorithms that can learn to predict discrete valued labels are called classification algorithms or classifiers, whereas the algorithms that can learn to predict continuous valued labels are called regression algorithms. As the task of WSD only involves discrete valued labels for word senses, we use only classification algorithms.

The main problem associated with supervised approach is the need for a large sense-tagged training set. Despite the availability of large corpora in some language, manual sense-tagging of a corpus is very difficult limiting the number of sense tagged words to be used and very few sense-tagged data are available now. To deal with this problem a variety of unsupervised WSD methods, which use a machine readable dictionary or thesaurus in addition to a corpus, have also been proposed [31], [35], [1]. Bilingual

parallel corpora, in which the senses of words in the text of one language are indicated by their counterparts in the text of another language, have also been used in order to avoid manually sense-tagging training data [36]. In this method, bilingual corpora are used since different senses of some words often translate differently in another language. Parallel corpora, especially accurately aligned parallel corpora are rare, although attempts have been made to mine them from the Web [39]. In[21] it is proposed to use Chinese monolingual corpora and Chinese-English bilingual dictionaries to automatically acquire sense examples for English ambiguous words and is reported that the result exceed previous state-of-the-art comparable systems. Their approach does not rely on scarce resources such as aligned parallel corpora or accurate parsers. In [22] the use of monolingual corpora of English for Hebrew and German language WSD is also tested and found the approach very useful for disambiguation.

2.3.2.2 Unsupervised Word Sense Disambiguation

Unsupervised Word Sense Disambiguation methods are based on unlabeled corpora, and do not exploit any manually sense-tagged corpus to provide a sense choice for a word in context unlike supervised method [12]. Unsupervised methods have the potential to overcome the knowledge acquisition bottleneck [9], that is, the lack of large-scale resources manually annotated with word senses. These approaches to WSD are based on the idea that the same sense of a word will have similar neighboring words. They are able to induce word senses from input text by clustering word occurrences, and then classifying new occurrences into the induced clusters. They do not rely on labeled training text and, in their purest version, do not make use of any machine–readable resources like dictionaries, thesauri, ontology, etc. However, the main disadvantage of fully unsupervised systems is that, as they do not exploit any dictionary, they cannot rely on a shared reference inventory of senses [37].

Most of the time, supervised approaches are superior to unsupervised in terms of accuracy of automatic disambiguation when used on the same type of texts that the systems were trained on [27].

Nevertheless, there is another issue connected with the problem of the definition of a meaning, i.e., an issue of creation of other resources used for automatic system performing WSD. This is especially evident in creation of corpora that is manually annotated (tagged) with senses, which are used for training machine learning classifiers in a supervised setting. There are two important problems during manual sense tagging of a corpus: low interannotator agreement (IA) and high cost of annotation process. IA is a way of measuring how much an annotation assigned by one annotator differs from annotations assigned by another annotator. IA is used for estimation of an upper bound on performance on automatic WSD but there is also another measure [40].

Mihalcea [6] identified that the cost of annotation preparing corpuses for supervised classification algorithm is high, because large effort is required during manual annotation. He also estimated that a construction of a corpus with sufficient amount of data for supervised classification algorithms for 20,000 English ambiguous words would require 80 man-years of work."

Like the supervised learning, even the unsupervised WSD methods strive from the data sparseness problem, since enormous amounts of text are needed to ensure that all senses of a polysemous word are represented in the corpus.

2.3.2.3 Bootstrapping Approach

The bootstrapping approach is situated between the supervised and unsupervised approach of WSD. The aim of bootstrapping is to build a sense classifier with little training data, and thus overcome the main problems of supervision: the data scarcity problem specially lack of annotated data. The bootstrapping methods use a small number of contexts labeled with senses having a high degree of confidence. This could be accomplished by hand tagging with senses the contexts of an ambiguous word w for which the sense of w is clear because some seed collocations [27]occur in these contexts. These labeled contexts are used as seeds to train an initial classifier. This is then used to extract a larger training set from the remaining untagged contexts. Repeating this process the number of training contexts grows and the number of untagged contexts reduces. We

will stop when the remaining unannotated corpus is empty or any new context can't be annotated.

2.3.3 Hybrid Approaches

Since they obtain disambiguation information from both corpora and explicit knowledge-bases, 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 base both on a 'knowledge-driven, corpus-supported' theme, utilizing as much information as possible from different sources. Yarowsky [1]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 a corpus.

2.4 WSD FOR AMHARIC

Though there is clearly a need for WSD for Amharic, to the researcher's knowledge the first research attempt was done by Teshome [20]. He has studied the use of WSD based on semantic vector for improving the precision and recall measurements of information retrieval for Amharic legal texts. The Ethiopian postal code which consisted of 865 articles was used as a corpus in the study. He developed his own algorithm based on distributional hypothesis stating that words with similar meanings tend to occur in similar contexts. For disambiguation of a given word, he computed the context vector of each occurrence of the words. The context vector was derived from the sum of the thesaurus vectors of the context words. He constructed the thesaurus by associating each word with its nearest neighbors.

For evaluating WSD, he used pseudo words which are artificial words rather than real sense tagged words reasoning that it is costly to prepare sense annotation data. He compared his algorithm with Lucene algorithm and reported that the algorithm is superior over the Lucene's one. The approach used in [20] and in this study is similar in the way that a corpus is used as a source of information for disambiguation.

The second attempted work was by Solomon [23]. He used corpus based, supervised machine learning approach using Naive Bayes algorithm for Amharic WSD, to check standard optimal context window size which refers to the number of surrounding words sufficient for extracting useful disambiguation. Based on Native Bayes algorithms, experiment found that three-word window on other side of the ambiguous word is enough for disambiguation.

He used a monolingual corpus of English language to acquire sense examples and the sense examples are translated back to Amharic which is one approach of tackling the knowledge acquisition bottleneck.

Based on Naive Bayes algorithm, experiments were conducted on Weka 3.6.2 package concluded that, Naive Bayes methods achieve higher accuracy on the task of WSD for selected ambiguous word, provided that the quality of the labeled data set. He achieved accuracy within the range of 70% to 83% for all classifiers. This is an impressive accuracy for supervised WSD but it suffers from knowledge acquisition bottleneck.

The difference between Teshome [20] and Solomon [23] was, the latter used a Naive-Bayes algorithm which is a machine learning techniques, where as Teshome [20] employ an algorithm based on semantic vector developed by him. In addition, [23] used for his study a real sense annotated data for ambiguous words, unlike pseudo words which is used in [20]. Finally the domain for[20] was a specific legal text where as the corpus used by [23] was domain independent.

2.5 MACHINE LEARNING

Hutchins and Sommers [41] define learning in terms of having the ability to memorize something, learn facts through observation, improve cognitive skill through practicing, and organize new knowledge in to an effective representation. The authors added that, the most common kinds of learning are the acquisition of information with the goal of making prediction about the future. Hutchins and Sommers [41] Classified machine learning techniques in to:

- **Supervised learning,** which includes finding a rule to predict the output of a new input using a set of example input/output pairs.
- **Clustering,** which includes grouping a set of given examples (with no associated labeling) into natural clusters, this is the so called unsupervised learning.
- **Reinforcement learning,** which includes observing the real world and learning to take action in such a way to obtain a lot of rewards; some other machine learning researchers would say this is just a special case of supervised learning, adapting an initial model.

According to Nello et al. [42] learning can be also defined as; improving one's performance on a given task with the aid of prior experience. One way of making computers learn involves training machine learning algorithms with the help of an initial set of training data. The experience that the machine learning algorithms gain from the training data can then be applied to make predictions about previously unseen data. One can train a machine learning algorithm such as partitional clustering algorithm to cluster disambiguate occurrences of the ambiguous a given word. Such a trained algorithm can then takes as input previously unseen sentences containing the word, and predict the correct sense of word in sentences.

On the other hand learning is further categorized as supervised or unsupervised. In supervised learning, a trainer provides the correct labels or outputs for the training data. In unsupervised learning, there is no trainer involved; the correct label for the training data instances is not available. The advantage of supervised learning is that high accuracy can be obtained on unseen instances given that a sufficient amount of manually labeled training data is provided to generate a good model. The drawback of the supervised learning approach is that manually labeled data is highly expensive to generate in terms of time as well as money. Unsupervised methods benefit from the fact that they do not require manually labeled data.

2.6 MACHINE LEARNING ALGORITHMS

Clustering algorithms are generally categorized as partitional and hierarchical. The next section describes some common clustering algorithms. Here are general properties that characterize clustering algorithms [43].

Agglomerative vs. Divisive: In agglomerative algorithms (bottom-up approach), each element is initially its own cluster and then the most similar clusters are iteratively merged until we are left with one large cluster containing all elements or until a stopping condition is met. Conversely, divisive algorithms (top-down approach) initially begin with a single all-encompassing cluster and iteratively split the clusters until each element belongs to its own cluster or until a stopping condition is met.

Hard vs. Soft: Hard clustering algorithms assign each element to exactly one cluster whereas soft (fuzzy) algorithms may assign an element to multiple clusters. In soft clustering, a membership degree is associated with each element's assignment to a cluster.

Deterministic vs. Stochastic: These types of searches mostly apply to partitional algorithms that optimize some clustering function. Stochastic algorithms use random searches of the feature space while deterministic algorithms do not.

Throughout the next section, we use ***n*** to represent the number of elements that are to be clustered. When the number of clusters must be fixed by an input parameter, like in many partitional clustering algorithms, we refer to this number by *K*.

2.6.1 Hierarchical algorithms

Hierarchical algorithms produce a nested partitioning of the data elements by merging or splitting clusters. Agglomerative algorithms iteratively merge clusters until an all-encompassing cluster is formed [45], while divisive algorithms iteratively split clusters until each element belongs to its own cluster. The merge and split decisions are based on the similarity metric. The resulting decomposition (tree of clusters) is called a dendrogram.

Figure 2.1 shows a possible dendrogram produced by an agglomerative hierarchical algorithm. At the topmost level of the dendrogram, we have a single cluster containing all elements. Using a similarity threshold, we can extract a clustering of the data by cutting the dendrogram according to this threshold.

Then, each connected component of the dendrogram forms a cluster. For example, assuming that the best clustering in the 2-dimensional space of Figure 2.1 consists of small tight clusters, the dotted line in (b) gives a good threshold for this data resulting in three clusters: The problem with any threshold is that on some data sets, a particular threshold will be good but on another data set, it will fail. For example, in Figure 2.1, if the similarity threshold was just a little higher, we would have five clusters with elements *C* and *D* in separate clusters.

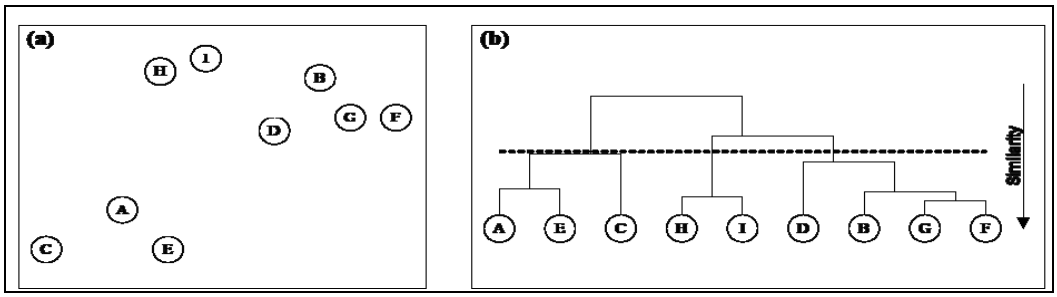

Figure 2.1 – Dendrogram visualization of a hierarchical clustering result.

(a) Nine data points in 2-dimensional space; (b) the dendrogram produced by a hierarchical agglomerative clustering algorithm (the dotted line indicates a possible similarity threshold for selecting the final clustering).

The dendrogram provides a visualization of how the algorithm produced its output. For example, if a particular output cluster is bad, the dendrogram provides a method of verifying how this bad cluster was formed. Hierarchical algorithms rigidly make merge and split decisions. If a particular decision is wrong, the algorithm will never go back and undo the decision. This makes the algorithm more efficient than performing a combinatorial search of all possible decisions but it can never correct itself.

2.6.1.1 Agglomerative

1. *Initially start with n clusters each containing a different element;*
2. *Merge the two most similar clusters (repeat n – 1 time).*

In the final step, an all-encompassing cluster is created and the result is a dendrogram like the one in Figure 2.1. The different versions of agglomerative clustering differ in how they compute cluster similarity. The most common versions of agglomerative clustering algorithm are single-link, complete-link and average-link clustering. The complexity of these algorithms is $O(n^2 \log n)$ [1].

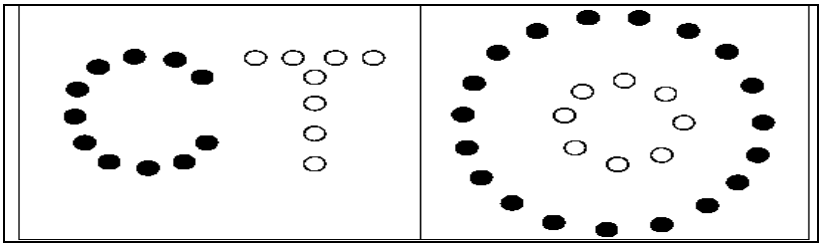

Figure 2.2 Clusters discoverable using single-link clustering.

Complete-link and average-link cannot discover these two clustering

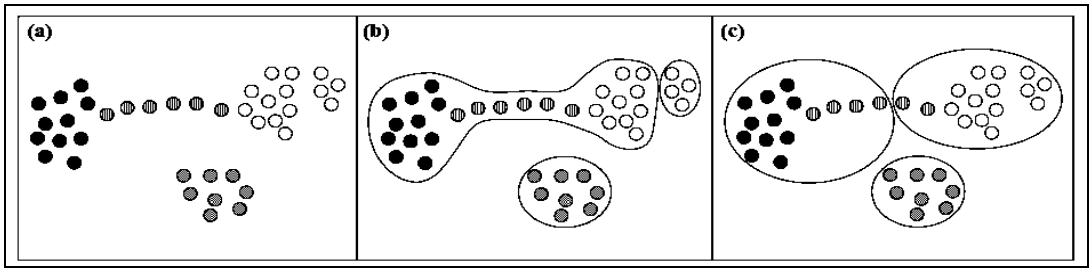

Figure 2.3 – The chaining effect in single-link clustering.

(a) Data points in 2-dimentional space; (b) the clustering produced by single-link clustering; (c) the clustering produced by complete-link clustering. The proximity measure is the Euclidean distance.

Single-link clustering: In single-link clustering the similarity between two clusters is the similarity between their most similar members (e.g. using the Euclidean distance) [44]. It

is capable of discovering clusters of varying shapes like the clusters of Figure 2.3. However, single-link is not practical because it suffers from the chaining effect [45]. For example, in Figure 2.3 (b), single-link clustering generates an elongated cluster because of a bridge of elements connecting two clusters.

Complete-link clustering: In complete-link clustering, the similarity between two clusters is the similarity between their least similar members (e.g. using the Euclidean distance) [34]. Although complete-link clustering is not capable of discovering clusters like the two in Figure 2.3, it does not suffer from the chaining effect. Rather than producing straggly elongated clusters like single-link, complete-link generates compact clusters. Figure 2.3 (c) shows an example. Complete-link generates better clustering's than single-link in many applications [46]. Figure 2.4 illustrates the different computations for cluster similarity between single-link and complete-link.

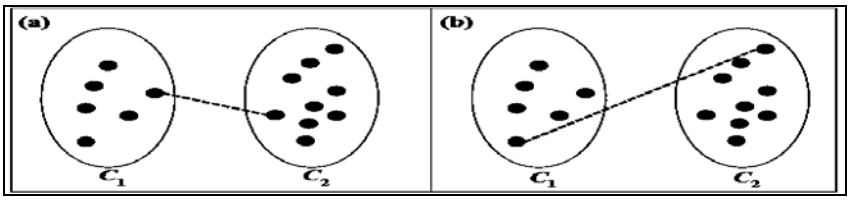

Figure 2.4 – Single-link vs. complete-link cluster similarity.

C_1 and C_2 are two clusters in 2-dimensional space where their similarity is the similarity between the two elements joined by a dotted line for (a) the single-link algorithm and (b) the complete-link algorithm.

Average-link clustering: Average-link clustering produces similar clusters to complete-link clustering except that it is less susceptible to outliers [29]. It computes the similarity between two clusters as the average similarity between all pairs of elements across clusters (e.g. using the Euclidean distance). Figure 2.5 shows snapshots of merge decisions comparing the three linkage algorithms on a 2-dimensional data set.

2.6.1.2. Divisive Clustering

although it is not as common as agglomerative clustering [47]. Divisive clustering algorithms start with a single cluster containing all elements. Considering all possible

splits of the cluster into two clusters gives $2^{(2n-1)} - 1$ possibilities. Using a splitting heuristic to iteratively split the largest cluster, Divisive clustering algorithms has worst-case time complexity $O(n^2 \log n)$.

Let the diameter of a cluster *c* be the similarity between the two least similar elements in *c*. The algorithm is as follows:

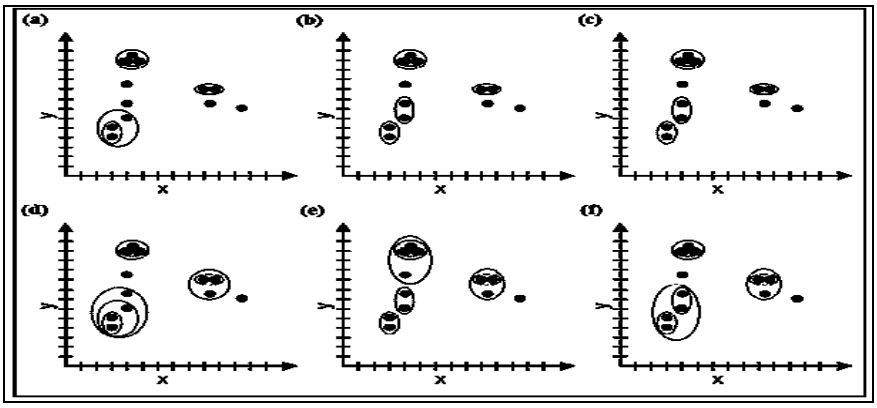

Figure 2.5 – Single-link, complete-link and average-link clustering.

Dotted ellipses denote previously merged clusters and solid ellipses denote newly merged clusters. (a), (b) and (c) illustrate the fifth merge decisions for single-link, complete-link and average-link respectively while (d), (e) and (f) illustrate the seventh merge decisions.

1. *Initially start with a single cluster encompassing all elements;*
2. *Select l, the largest cluster or the cluster with highest diameter;*
3. *Find the element e in l that has the lowest average similarity to the other elements in l;*
4. *e is the first element added to the splinter group while the other elements in l remain in the original group;*
5. *Find the element f in the original group that has highest average similarity with the splinter group;*
6. *If the average similarity of f with the splinter group is higher than its average similarity with the original group then assign f to the splinter group and go to Step 5; otherwise do nothing;*
7. *Repeats step 2-6 until each element belongs to its own cluster.*

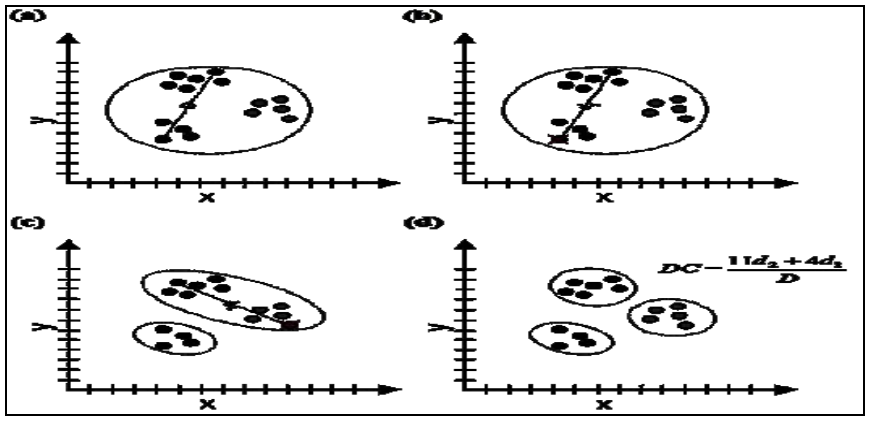

Figure 2.6 divisive clustering.

(a) the initial all-encompassing cluster with diameter D; (b) the first splinter group defined by the cross (d1 is the diameter of l from Step 2); (c) the result of the reassignment of elements to the splinter group after the first iteration - the new splinter group is defined by the cross (d2 is the diameter of the new l from Step 2); (d) the result of the reassignment of elements to the splinter group after the second iteration and the DC measure assuming that this is the final partitioning.

After completion, each element will belong in its own cluster (i.e. there will be *n* clusters). Divisive clustering provides a measure of the strength of the clustering structure called the divisive coefficient, *DC*:

$$DC = \frac{\left(\sum_{e \in D} d(e)\right)}{d}$$

eq.2.1

Where *D* is the set containing all elements to be clustered, *d* (*e*) is the diameter of the last cluster to which element *e* belonged before being split to a single-element cluster, and *d* is the diameter of *D*. The higher the *DC*, the stronger becomes the clustering structure. *DC* will be lowest when each element's before-last split results in a very tight cluster. However, the union of before-last-splits of the elements is rarely the desired clustering. When using the hierarchy given by a hierarchical clustering algorithm, one usually obtains a partitioning by applying a similarity threshold on the hierarchy. Defining *d*(*e*) as the diameter of the cluster to which element *e* belonged before being split into the cluster in which it resides in the final partitioning (by applying some threshold on the hierarchy) gives a better indication of the strength of the clustering structure. Here, an element lowers the *DC* if its last split before the final partitioning was unnecessary.

Figure 2.6 shows an example of clustering using divisive clustering. The different shadings represent a possible target clustering. In (a), the all-encompassing cluster is shown as well as the diameter *D* of the data set, which is used in computing the divisive coefficient. The element with the cross in (b) is the element with the lowest average similarity to all other elements and it defines the first splinter group. The small cluster in (c) shows all the elements that were added to the splinter group (Step 5 and Step 6 of the algorithm). The larger cluster is then selected as the next cluster to split since it has the largest diameter, shown by *d*2. The element with the cross in (c) represents the next splinter group. The resulting reassignment of elements to the splinter group is shown in (d) as well as the divisive coefficient assuming that this is the selected partitioning.

2.6.2 Partitional algorithms

Partitional algorithms do not produce a nested series of partitions. Instead, they generate a single partitioning, often of predefined size *K*, by optimizing some criterion. A combinatorial search of all possible clustering's to find the optimal solution is clearly intractable. The algorithms are then typically run multiple times with different starting points. Partitional algorithms are not as versatile as hierarchical algorithms but they often offer more efficient running time [43].

2.6.2.1 K-means

The most commonly used family of partitional algorithms is based on the *K*-means algorithm [48]. *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.

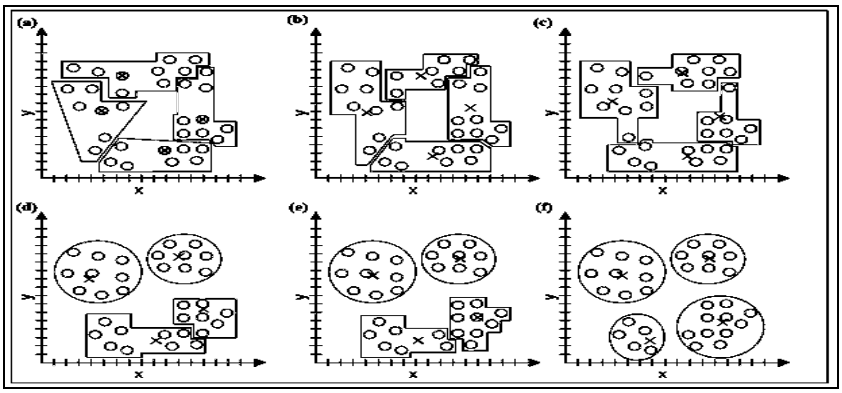

Figure 2.7 – K-means clustering.

The crosses represent cluster centroids and K=4. (a) The initial randomly selected centroids and the first cluster assignment; (b) – (f) the second to sixth

iterations of K-means. After the sixth iteration, the element assignments do not change and the algorithm terminates.

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:

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.*

The most commonly used criterion is the squared-error criterion, *E*:

$$E = \sum_{i=1}^{K} \sum_{e \in c_i} |e - m_i|^2$$

Eq. 2.2

Where e is an element in cluster c_i and m_i is the centroid of c_i. Figure 2.8 illustrates the operation of *K* means on 2-dimensional elements with *K*=4. In the initialization of *K*-means, four elements are chosen as the initial centroids (represented by crosses). After the sixth iteration of the algorithm, shown in (*f*), the element assignments to clusters will no longer change and the algorithm terminates.

K-means has complexity $O(K \times T \times n)$ and is efficient for many clustering tasks since the parameters *K* and *T* are usually small fixed constants. Because the initial centroids are randomly selected, the resulting clusters vary in quality. Some sets of initial centroids lead to poor convergence rates or poor cluster quality.

2.6.2.2 Bisecting K-means

Bisecting *K*-means [44], a divisive variation of *K*-means, begins with a set containing one all-encompassing cluster consisting of every element and iteratively picks the largest cluster in the set, splits it into two clusters and replaces it by the split clusters. Splitting a cluster consists of applying the *K*-means algorithm α times with *K*=2 and keeping the split that has the highest average element-centroid similarity. Note here that $\alpha \neq T$. It is the whole *K*-means algorithm that is repeated α times. Each instantiation of *K*-means will have *T* iterations.

2.6.2.3 K-medoids

The centroids constructed by *K*-means are sensitive to outliers, if there are many of them, since each element has a direct influence on the construction of the centroids. *K*-medoids [49] is a family of algorithms that addresses this shortcoming. Instead of representing a cluster by its centroid, *K*-medoids uses one of the elements of the cluster as its representative. The algorithm is very similar to *K*-means. Initially, *K* random elements are chosen as the initial representative of the *K* clusters. In its iteration, the algorithm a representative element is replaced by a randomly chosen on representative element if the criterion (e.g. squared-error criterion) is improved. The elements are then reassigned to their closest cluster. Examples of *K*-medoids algorithms include PAM [49] and CLARA [49].

2.6.3 Expectation Maximization Algorithm

Expectation maximization (EM) is a well-known algorithm used for clustering in the context of mixture models. EM was proposed by [73]. 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. In our work, the sense of an ambiguous word is represented by a feature whose value is missing.

In order to use the EM algorithm, the parametric form of the model representing the data must be known. In these experiments, we assume that the model structure is the Naive Bayes [50]. In this model, all features are conditionally independent given the value of the classification feature, i.e., the sense of the ambiguous word. This assumption is based on the success of the Naive Bayes model when applied supervised word-sense disambiguation (e.g. [10], [72], [73], [16]).

There are two potential problems when using the EM algorithm. First, it is computationally expensive and convergence can be slow for problems with large numbers of model parameters. To solve the above problem we used small data set for this study. Second, if the likelihood function is very unbalanced it may always converge to a local maximum and not find the global maximum.

To simplify the discussion, we first briefly describe the EM algorithm. The algorithm is similar to the K-means procedure in that a set of parameters are re-computed until a desired convergence value is achieved. The parameters are re-computed until a desired convergence value is achieved. The finite mixtures model assumes all attributes to be independent random variables.

A mixture is a set of N probability distributions where each distribution represents a cluster. An individual instance is assigned a probability that it would have a certain set of attribute values given it was a member of a specific cluster. In the simplest case $N=2$, the probability distributes are assumed to be normal and data instances consist of a single real-valued attribute. Using the scenario, the job of the algorithm is to determine the value of five parameters, specifically:

1. The mean and standard deviation for cluster 1

2. The mean and standard deviation for cluster 2

3. The sampling probability *P* for cluster 1 (the probability for cluster 2 is *1-P*)

And the general procedure states as follow:

1. Guess initial values for the five parameters.

2. Use the probability density function for a normal distribution to compute the cluster probability for each instance. In the case of a single independent variable with mean μ and standard deviation σ, the formula is:

$$f(x) = \frac{1}{(\sqrt{2\pi}\sigma)e^{\frac{-(x-\mu)^2}{2\sigma^2}}} \qquad \text{Q.1}$$

In the two-cluster case, we will have the two probability distribution formulas each having differing mean and standard deviation values.

3. Use the probability scores to re-estimate the five parameters.

4. Return to Step 2.

The algorithm terminates when a formula that measures cluster quality no longer shows significant increases. One measure of cluster quality is the likelihood that the data came from the dataset determined by the clustering. The likelihood computation is simply the multiplication of the sum of the probabilities for each of the instances. With two clusters *A* and *B* containing instances *x1, x2, x3,…,*x_n where $P_A=P_B=0.5$ the computation is:

$$[.5P\ (x_1|A) + .5\ (x_1|B)][.5P\ (x_2\ |A) + .5\ (x_2\ |B)]... [.5P\ (x_n|\ A) + .5(x_n\ |B)] \qquad \text{Q.2}$$

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.

2.6.4 Hybrid algorithms

Hybrid clustering algorithms are characterized as multi-phase algorithms that combine hierarchical and partitional techniques. In this section, we present five algorithms: Buckshot, BIRCH, CURE, Rock and Chameleon.

2.6.4.1 Buckshot

According to Cutting, [51] Buckshot addresses the problem of randomly selecting initial centroids in K-means by combining it with average-link clustering. Buckshot first applies average-link to a random sample of n elements to generate K clusters. It then uses the centroids of the clusters as the initial K centroids of K-means clustering.

As the random sample-size approaches K, Buckshot degenerates to the K-means algorithm. The strict definition of the sample size makes Buckshot unsuitable for some situations. Suppose one wish to cluster 100,000 documents into 1000 newsgroup topics. Buckshot could generate at most 100,000 ≈ 316 initial centroids. The sample size counterbalances the quadratic running time of average-link to make Buckshot efficient: $O(K \times T \times n + n\log n)$. However, the algorithm can be run with any sample size as long as the speed of clustering is acceptable.

2.6.4.2 BIRCH

BIRCH, Balanced Iterative Reducing and Clustering using Hierarchies [32] is a two phase algorithm that uses a structure called a *CF*-tree to abstract the data yielding an efficient algorithm. A *CF*-tree is a compression of the data elements that attempts to preserve the inherent structure of the data. The two phases are:

> *1) Construct a CF-tree by scanning through each element to be clustered;*
> *2) Apply any clustering algorithm to cluster the leaf nodes of the CF-tree.*

A *CF*-tree is a hierarchy of sets of clustering features. Given a sub cluster whose elements are represented by *m*-dimensional feature vectors, a clustering feature, *CF*, summarizes the information contained in the elements:

$$CF = \left(N, \vec{LS}, \vec{SS} \right)$$ Eq. 2.3

Where N is the number of elements in the sub cluster, $\vec{LS} = \sum_{i=1}^{N} \vec{x}_i$ and $\vec{SS} = \sum_{i=1}^{N} \vec{x}_i^2$

The first step of BIRCH has time complexity $O(n)$. As long as the chosen algorithm for step 2 is also linear (e.g. a partitional algorithm like K-means), BIRCH has overall time complexity $O(n)$, which is more efficient than Agglomerative and Divisive clustering algorithms. Because BIRCH uses a diameter parameter, it is not very good for discovering clusters that are not spherical. Another problem with BIRCH is that it is sensitive to the order in which the elements are scanned in Step 1 of the algorithm.

2.6.4.3 CURE

Single-link clustering has the advantage of being able to discover clusters of various shapes and sizes but it is not robust in the presence of outliers (i.e. the chaining effect). CURE, Clustering Using representatives [52], is similar in operation to single-link clustering but is more robust to outliers. Clusters are represented by a set of initially well-scattered points that are shrunk towards the center of gravity of the cluster.

Given a set of elements X to cluster, CURE initially selects a random sample of size s from X. The random sample is then partitioned into p partitions each with size s/p and then the partitions are partially clustered using an agglomerative hierarchical algorithm. Setting a high similarity threshold in aggregative analysis gives many small clusters. Clusters that grow too slowly are tagged as outliers and are eliminated. At this point, we have several small tight clusters and each is represented by the mean of its constituting elements (a centroid).

The time complexity of CURE is $O(n)$, making it efficient for large data sets. However, the algorithm is very sensitive to its input parameters: the shrinking factor α and the random sample size.

2.6.4.4 Rock

ROCK, Robust Clustering using links [53], is an algorithm for clustering binary and categorical data. Previous clustering methods that use a distance measure, such as the Euclidean distance between elements, are not suitable for binary and nominal data.

ROCK has worst-case time complexity of O ($n2 + nmmma + n2logn$) where mm is the maximum number of neighbors and ma is the average number of neighbors. ROCK is a good algorithm for categorical data but its complexity makes it inefficient for large data sets.

2.5.4.5 Chameleon

CURE ignores the aggregate interconnectivity between two clusters while ROCK ignores the average closeness between clusters. Chameleon [54] combines the advantages of CURE and ROCK while employing dynamic modeling of clusters to improve clustering quality. Clusters are merged in Chameleon if they have high interconnectivity and closeness relative to each cluster's internal interconnectivity and closeness. Chameleon has been shown to produce higher quality clusters than CURE but it suffers from a worst case time complexity of O ($n2$).

2.5.5 Other algorithms

There are several other well known families of algorithms. Density-based methods such as DBSCAN [55] and OPTICS [56] discover clusters of dense elements that are separated by low density regions. Grid-based multi-resolution algorithms typically collect statistical information in grid cells and perform all clustering operations on these grids. CLIQUE [57] is a hybrid of grid and density methods. It is capable of handling high-dimensional data because all clustering operations are performed on the quantized space of the grid.

2.7 SUMMARY

A basic introduction to the field of WSD has been presented in this chapter. A survey of the major approaches to WSD has discussed, emphasizing the key WSD research problems that should be addressed by any type of solution. The field of Machine learning with associated algorithms that are used for WSD research has also been discussed. For this study a corpus based approach specifically unsupervised WSD is adopted. The classifier will be modeled using the selected clustering algorithms.

CHAPTER THREE

THE AMHARIC LANGUAGE

Ethiopia is a linguistically diverse country where more than 80 languages are used in day-to-day communication. Although many languages are spoken in Ethiopia, Amharic is dominant in that it is spoken as a mother tongue by a substantial segment of the population and it is the most commonly learned second language throughout the country [58]. The language is the working language of the Federal Government of the country. It is the second most spoken Semitic language in the World (after Arabic) and today probably one of the five largest on the African continent (albeit difficult to determine, given the dramatic population size changes in many African countries in recent years). [59]. Manuscripts in Amharic are known from the 14th century and the language has been used as a general medium for literature, journalism, education, and so on[60]. A wide variety of Amharic literatures including books, religious writings, fiction, poetry, plays, and magazines are available both in printed and machine readable format.

3.1 THE AMHARIC WRITING SYSTEM

According to Bender et al. [58], three writing systems are in use in Ethiopia, the Ethiopic (Ge'ez) syllabary, the Roman alphabet, and Arabic script. The widely used Ethiopic syllabry, which is derived from the writing system of ancient South Arabian inscriptions, is used for Ge'ez, Amharic, Tigrigna and other semantic languages. The writing system has a similarity with some Semitic languages like Arabic in having vowel marks added to basically consonant letters. The present writing system of Amharic is taken from Ge'ez. Ge'ez in turn took its script from the South Arabian mainly attested in inscriptions in the Sabean dialect [58]. The original Sabaean alphabet is said to have had 29 symbols. When Ge'ez became the spoken and written language in common use in northern Ethiopia, it took only 24 of the 29 Sabaean symbols, modify most of them and add two new symbols to represent sounds of Greek and Latin loanwords not found in Ge'ez, these symbols are ጸ and ፕ. The style of the writing was also modified to left to right. By the time Ge'ez ceased to be a living spoken and written language and replaced by Amharic and other languages, further changes took place. Amharic did not discriminate in adopting the

Ge’ez fidel; it took all of the symbols [61] and added some new ones that represent sounds not found in Ge’ez. These added alphabetic characters are ቾ, ጨ, ጇ, ኘ, ቨ, ሸ, ኸ, and ዠ.

Currently, the language’s writing system contains 34 base characters each of which occur in a basic form and six other forms known as orders. The seven orders represent syllable combinations consisting of a consonant following vowel. This is why the Amharic writing system is often called syllabic rather than alphabetic, even if there is some opposition[61]. The 34 basic characters and their orders give 238 distinct symbols. In addition, there are forty others that contain a special feature usually representing labialization e.g. ቿ, ቋ. In Amharic there is no Capital-Lower case distinction. There are also punctuation marks and numeration system.

3.2 TYPICAL CHARACTERISTICS OF AMHARIC LANGUAGE

There is a process of change in any language in many of its aspects: change of meaning, change of syntax, phonetic change, etc[61]. The case for Amharic is not different; especially the script underwent changes when it was borrowed from Ge’ez. Through the adaptation process and other factors the Amharic writing system got some problems.

The first problem is the presence of “unnecessary” alphabets (fidels) in the language’s writing system. These fidels (alphabets) have the same pronunciation but different symbols. These different fidels can be used interchangeably without meaning change. The fidels are ኦ and ዐ, ጸ and ፀ, ሰ and ሠ, and ሀ, ሐ, and ኀ. For example, the word “sun” can be written as, ጸሀይ, ጸሃይ, ፀኃይ, ፀሃይ etc … all meaning the same thing, although written differently.

The other problem is in the formation of compound words. Compound words are sometimes written as two separate words and sometimes as a single word. For example, the word “kitchen” can be written as “ወጥቤት” or “ወጥ ቤት”.

Amharic is morphologically rich language where up to 120 words can be conflated to a single stem[62]. An Amharic root is a sequence of base characters. A collection of

phonemes or sounds creates a word, which can be as simple as a single morpheme or contain several of them.

In Amharic language, it is common to write some words in shorter form using "/" (forward slash) or "." (dot). The short form of words can be expanded as single or a combination of words. አ/አ, can be expanded as አዲስ አበባ (means Addis Ababa), is an example for the latter. መ/ር can be a short form of the single word መምህር (means teacher).

Another problem of the language is, there are different ways of writing a single word due to different reasons. One reason for this can be regional dialects that can impact word formation in the basic level where the words are more likely to be written following their spoken form; "ሂጂ" vs. "ሂጅ", "አይዶለም" vs. "አይደለም", "ዓጤ" vs. "ዓፄ" , etc [59]. Another one is, in Amharic there are many ways of writing loan words, i.e words that are taken from foreign languages. For example, the word Computer can be written as ኮምፒዩተር, ኮምፒውተር, ኮምፒዉተር, etc.

3.3 AMHARIC PUNCTUATION MARKS

Analysis of Amharic texts reveals that different Amharic Punctuations marks are used for different purposes. In [63] it is indicated that there are about 17 punctuation marks of which only a few of them are commonly used and have representations in Amharic software.

The Amharic writing system uses some indigenous and foreign punctuation marks (signs) in addition to the Amharic characters [64]. However, only few of them are practically used, especially in computer-written text. The word-separator (*hulet neTb*), two square dots arranged like colon (:), and sentence-separator (*arat neTb*), four square dots arranged in a square pattern (: :), are the basic punctuation marks in Amharic writing system that are used consistently. Today, the use of *Hulet Neteb* is not seen in modern typesetting. In typesetting its place is almost completely taken over by space. Lists in Amharic text are separated by an equivalent of comma, *'netela serez*(፣) followed by ASCII space and *'derib sereze'* (፤), which is the equivalent of semi-colon. The use of '**...**' for question mark is not used rather a '*?*' which is borrowed from English is used. Table 3.1 lists the

most commonly used Amharic punctuation with their equivalent in English which is adopted from [59].

Amharic	English
፡	White Space
።	.
፤	;
፣	,
፧	?

Table3.1 most commonly used Amharic punctuation marks with their English corresponding marks

3.4 SYNTACTIC STRUCTURE OF AMHARIC

The syntactic structure is formed by combining different words. Since Amharic word formation follows its own structure, the syntax of the language also exhibit a unique structure. The syntactic structure of Amharic is generally SOV (Subject-Object-Verb).The modifiers in such structure generally precede the word or the phrases they modify. For example, the Amharic equivalent for the English sentence *"He comes to library"* is *"እሱ ወደ ቤተ-መጽሃፍት መጣ" ("`Isu wede bEte-meShaft meTa")* here, the subject is "`I*su*" and the object is "*bEte-meShaft*" and the verb is "*meTa*". But, usually pronouns are omitted when used as a subject. For the above English sentence the usually way to say it in Amharic is, "*wede bEte-meShaft meTa* "The pronoun "`*Isu*" (He) is implicit in the sentence and come part of the verb. In this case the verb indicates the pronoun that is left out in the sentence.

Question formation is the same as a declarative sentence except the usage of question mark at the end. That is to ask the question *"did he go to the school?"* in Amharic, the sentence *"he went to school"* is ended with question mark instead of the Amharic full stop (.). The Amharic equivalent is *"እሱ ወደ ትምህርት ቤት ሄደ?" ("`Isu wede tmhrt bEt hEde?"*).

Sometimes, words that indicate the sentence is a question are added at the end of the sentence. In such cases the above question becomes *"`Isu wede tmhrt bEt hEde `IndE?". Here, the word "`IndE"* is added to indicate that it is a question.

3.5 AMBIGUITIES IN AMHARIC

Getahun [65]identified six types of ambiguity in Amharic: Phonological, Lexical, Structural, Referential, Semantic and Orthographic ambiguities. We now summarize each type of ambiguity and the examples are adopted from [65] are discussed next.

3.5.1 Phonological Ambiguity

Phonological ambiguity is a result due to the sound used for the word from the placement of pause with in a structure which occurs in speech .It can be illustrated through the following example:

ደግ ሰው ነበር

[deg + sew] neber

Kind person was

In the above sentence '+' sign shows where the pause is. When the sentence is pronounced with pause it means "He was a kind man" but the meaning differs if it is pronounced without pause .It will mean "They had preparation for a banquet".

3.5.2 Lexical Ambiguity

Lexical ambiguity refers to a case in which either a lexical unit belongs to different part-of-speech categories with different senses, or to a lexical unit for which there is more than one sense, while these different senses fall into the same part-of-speech category [36].There are three different factors that can cause lexical ambiguity which are: Categorical Ambiguity, Homonymy and Homophonous Affixes

Categorical Ambiguity

Categorical ambiguity is a result from lexical elements which have the same phonological form but belongs to different word class. This will be more described using the following ambiguous word:

አክርማ ሰጠችኝ

ekrma seT-ec-N ? gave-she-me

In the above example the underlined word *"ekirma"* is ambiguous since it has both nominal and a verbal meaning. It has two interpretations:

i.She gave me Akirma(a kind of grass).[With nominal meaning]

ii.She gave me something after delaying it for sometime.[With verbal meaning]

Homonymy

Homonyms are those lexical items with the same phonological form but with different meanings which will cause ambiguity. It can be illustrated with the following example:

በወሬ አልፈታም

bewerE elfetam

In the example the underlined *"bewerE"* is an ambiguous word having the following two different structures and readings shown below:

i. be-wer-E el-feta-m

with-month-my Neg-released-Neg

In this sense it means that "I will not be released in a month"

ii. be-wer-E el-fata-m

Neg-release-Neg

It means that “I will not get frustrated by any rumor”

Homophonous Affixes

This ambiguity result when affixes serve as different word classes. The following example show how homophonous affixes cause ambiguity.

ቤቱ ፈረሰ

bEt-u ferese

The above sentence is ambiguous because the suffix /-u/ serves as a definite article or as a third person masculine marker. It has two different meanings:

i. The house is destroyed. And

ii. His house is destroyed

3.5.3 Structural Ambiguity

Structural ambiguity resulted when a constituent of a structure has more than one possible position. By a structure we mean the way syntactic constituents are organized. The following is an example of such ambiguity:

የአረብ ታሪክ አስተማሪ

Ye-areb tarik estemari

of-Areb history teacher

The above sentence can have two different interpretations:

i. a person who teaches Arab history

ii. an Arab who teaches history

It can be further illustrated using structural organization of the sub-constituent */tarik/ ‘history’*. It is shown in the following labeled representation:

i. [[[Ye-areb tarik] [estemari]]]]

N N N

“Ye-areb history teacher”

ii. [Ye-areb [[tarik] [estemari]]]

N N N

3.5.4 Referential Ambiguity

This ambiguity arises when a pronoun has more than one possible antecedent, thus having as many reading as there are antecedents .The following sentence is an example of such ambiguity.

ካሳ ስለተመረቀ ተደሰተ

Kasa sletemereke tedesete

The above sentence has two different readings:

i.Kasa was pleased because he graduated.

ii.Somebody was pleased because Kasa graduated

3.5.5 Semantic Ambiguity

Semantic ambiguity is caused by polysemic, idiomatic and idiomatic and metaphorical constituents. The following sentence is an example Polysemic constituent which has multiple meanings.

መበራቱ ጠፋ

Mebrau tefa

The above sentences have two interpretations:

i. The light went off.

ii. Mebratu(name of a person) disappeared

Idioms refer to an expression that means something other than the literal meanings of its individual words. Idioms ambiguity can be illustrated using the following example:

berE welede

The literal meaning of the above example is *"An ox gave birth to a calf"* but the idiomatic expression refers to "*impossible* "to happen.

Metaphors have literal or non-literal (metaphoric) senses. The following is an example of metaphoric ambuigity:

አራስ ነብር

eras nebr

It has two different interpretations:

i.'inascible, hot tempered'

ii.'leopard with new-born cubs'

3.5.6 Orthographic Ambiguity

Orthographic Ambiguity is resulted from geminate and non-geminate sounds. The ambiguity can be resolved using context. Though in some cases it might not be possible like the following example:

መኪናው ይሰራል

Mekinaw yseral

The word "*yseral* "is the cause of ambiguity. The sentence is ambiguous between the following meanings.

i.The car works*("yseral")*

ii.The car will be repaired*("ysearal")*

3.6 SUMMARY

In this chapter, the basic introduction of Amharic language has been presented. The Amharic writing system, syntactic structure, its typical characteristics and the Amharic punctuation marks have been discussed, by emphasizing the key WSD research problems that should addressed. Finally, the chapter summarizes different type of Amharic ambiguities with its examples are also discussed in detail. For this study lexical ambiguity which was believed to be resolved by word sense disambiguation among the type of ambiguities that were discussed above.

CHAPTER FOUR

CORPUS PREPARATION AND SYSTEM ARCHITECTURE

4.1 CORPUS PREPARATION AND ACQUISITION OF SENSE EXAMPLES

As discussed in the literature review part, one of the mechanisms to acquire sense examples is to use a monolingual corpus of second language and translate the sense examples to the original language. For this study we used the corpus that was previously used in research on supervised WSD for Amharic and the best reported accuracy in using supervised classifier was 70 – 83% [23] because; firstly, due to absence of a standard sense annotated data and other resources available for Amharic WSD research. Secondly, it's convenient because the translated sentences were available. Finally, comparing the results with previous works (supervised Amharic WSD)

Solomon [23] used a total of 1045 sense example sentences for five Amharic ambiguous words and contexts were randomly extracted from the British National Corpus (BNC). The senses of the Amharic ambiguous words were first translated to their equivalent English words using Amharic-English Dictionary[66]. Then using the translated English word sense example sentences containing the word were acquired from the English corpus. For example the Amharic ambiguous word "ሾህል" has three senses that are *"cough"*, *"vow"*, *and "sharp"* [24]. Using these three senses, sense example sentences are acquired. The English sentences were examined thoroughly to check that it correctly represents the right sense of the Amharic word. In this work, those same sense examples were taken for this study.

Agirre & Martinez [67] have reported that accuracy of machine learning algorithms degrade significantly when the training and testing samples have different distributions for the senses. In this study we tried to use a balanced distribution of senses for the ambiguous words to maximize performance when enough sense examples are available. On average, about 100 example sentences were acquired for each sense of ambiguous words with the exception of two senses on which enough example senses were not acquired from the corpus. The distributions of senses are summarized in table 4.1 below.

Ambiguous Word	Sense	Count	
eTena	strengthen	100	200
	study	100	
mesal	cough	100	245
	sharp	72	
	vow	73	
me`sa`sat	taking care	100	200
	thin	100	
metrat	call	100	200
	clean	100	
qereSe	record	100	200
	shape	100	

Table 4.1 Distribution senses of Ambiguous words

4.2 SYSTEM ARCHITECTURE

The architecture of the system is shown in figure 4.1. The system takes sentences that contain the ambiguous words as an input .The sentences were preprocessed[1] to make them suitable for training and evaluating a selected algorithm. Then the unsupervised algorithm (clustering algorithm) builds model from the training set and evaluate the built model and displays performance valuation of the model. The detailed explanations of the processes presented in the next subsections.

[1] Preprocessing include tokenization, stop word removal, stemming and context extraction

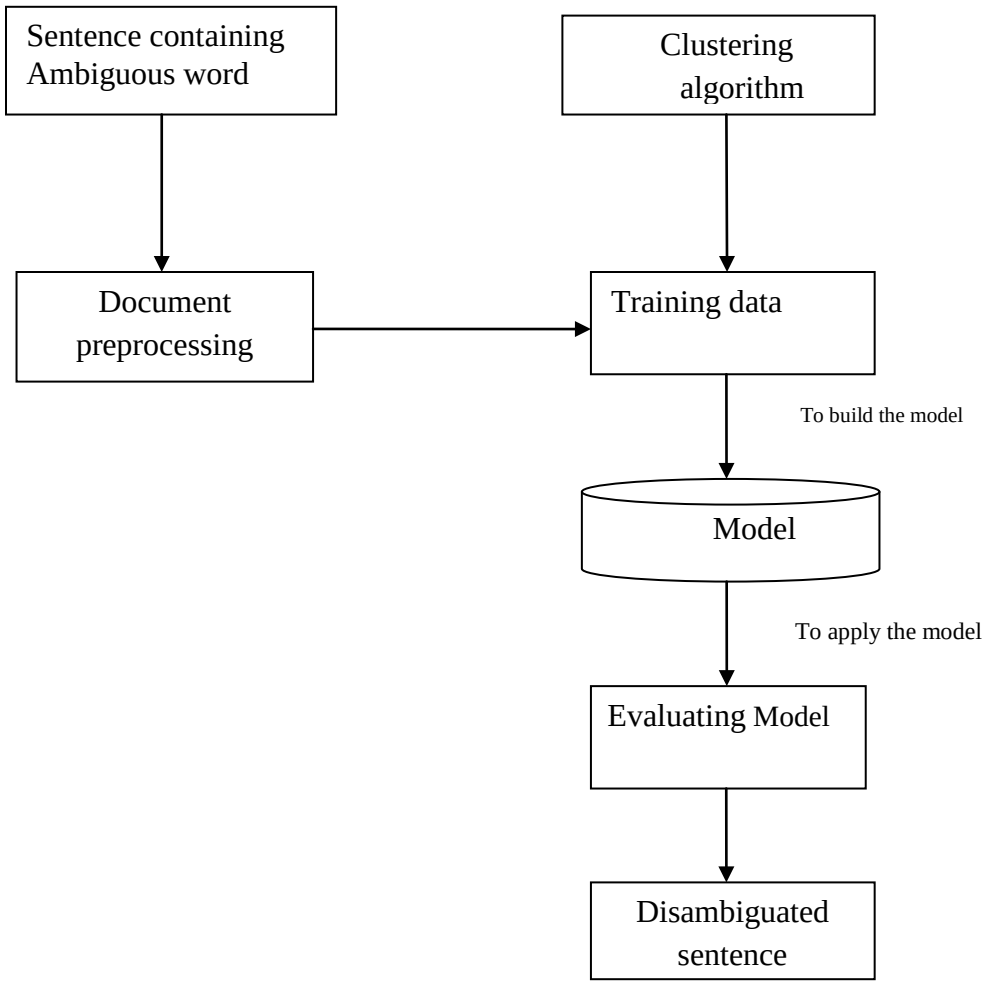

Figure4.1 Unsupervised Amharic Word Sense Disambiguation System Architecture

4.2.1 Document Preprocessing

4.2.1.1 Tokenization

Tokenization refers to the process of splitting the text into a set of tokens (usually words). This process detects the boundaries of a written text. Tokenizing of a given text depends on the characteristics of language of the text in which it is written. The Amharic language uses punctuation marks which demarcate words in a stream of characters which include *'hulet neTb'* (:), '*arat neTb* '(: :), '*derib sereze'* (፧), 'netela serez'(፡), exclamation mark '!' and question mark'?'. These punctuation marks don't have any relevance in identifying the meaning of ambiguous words using WSD.

4.2.1.2 Stop Word Removal

Like other languages, Amharic has its own stop words. Usually words such as article (e.g. *'ያ'ው', 'ይህ'*), conjunctions (*"'ና', 'ነገርግን', 'ወይም'*) and prepositions (e.g. *ውስጥ, ላይ*).Since stop words do not have significant discriminating powers in the meaning of ambiguous words , we filtered the sense examples with a stop-word list, to ensure only content bearing words are included. In addition to stop words, names of people and places are also filtered from the sense examples as they are not related to the meaning of words. Algorithm 4.1 presents the procedure for removing stop words from a give corpus file.

```
1. Open corpus and stop word list
2. While not end of corpus file is reached do
    Read terms
        For each term in the file
            If term in stop word list then
                Remove term
            End if
        End for
3. End while

4. Close files
```

Algorithm 4.1 Stop word removal algorithm

4.2.1.3 Stemming

The Amharic language makes use of prefixing, suffixing and infixing to create inflectional and derivational word forms. A stemmer is a system that tries to reduce various forms of a word to a single stem. In morphologically complex languages like Amharic, a stemmer will lead to significant improvements in WSD systems[31].

For this study, automatic removal of suffix and prefixes was done using adopted algorithm [23] and infixes are removed manually. After the semi-automatic removal, a manual inspection of the corpus was carried out in a way to correct a few errors in

exceptional cases in order to correct them. Algorithm 4.2 presents the procedure for removing prefixes and suffixes from a given corpus file.

```
1. open corpus, exception list and stop word list
2. While not end of corpus file is reached do
    Read terms
        For each term in the file
                If term starts with prefix
                        If term not in exception file list then
                                Remove prefix
                        End If
                End If
                If term ends with suffix
                        If term not in exception file list then
                                Remove suffix
                        End If
                End if
        End for
 3. End while

 4. Close files
```

Algorithm 4.2 Stemmer algorithm[2]

4.2.1.4 Transliteration

In addition to the above preprocessing, the Amharic documents need to be transliterated. For computational efficiency and simplicity of processing, transliteration of Amharic documents was used. Transliteration is the representation of the characters of one language by corresponding characters of another language (in this research Latin alphabets are used for transliteration). It enables easy, unambiguous and consistent communication of documents.

[2] *Stop word removal and stemmer algorithms are developed by our research group member (Solomon, Lakachew and Degfew)*

The transliteration of the Amharic corpus was conducted by using SERA[59]. In SERA, different variants of a single sound are represented using one English letter. For example, the above word *'ስርዓት'* contains two words *'ስ'* and *'ዓ'* that can be expressed using other Amharic alphabets *'ሥ'* and *'ዐ'* or *'አ'* respectively. By transliterating the entire text, it was possible to have normalized the representation of words in different forms to one common form. Therefore, the above six words to represent one meaning transliterated to the word ***'sr`at'***.

SERA, is case sensitive, i.e., upper and lower cases of the English alphabet representing different symbols in the Amharic alphabet. Therefore, other Amharic alphabets that have the same meaning and sound with different form are transliterated to the same form without affecting the meanings of words using SERA.

4.2.1.5 Context Extraction

Context in WSD refers to the words surrounding the ambiguous words which are used to decide the meaning of the ambiguous word.

For instance, the sentence: "የበሽታው የተለመደ ምልክት ያለማቋረጥ መሳል ከደረት ህመም ጋር ነው::"

After stop word removal and stemming, it will be "ተለመደ ምልክት ማቋረጥ መሳል ደረት ህመም" .

The contexts are words surrounding the ambiguous word **"መሳል"** which are *(ተለመደ,ምልክት,ማቋረጥ,ደረት,ህመም)*. In this study, the contexts of the ambiguous words are extracted using the algorithm in Algorithm 4.3 which is adopted from[69] and customized to fit to this study.

1. *initialize array buffers of strings*
2. *j=1*
3. *open corpus file*
4. *while not end of file is reached do*
   read sentence j
   i=1
   while end of sentence marker is not reached do
   read word i from sentence j
   if word i is the target word
   assign word i in to an array buffer
   assign the meaning(label) of word i to an array buffer
   if the n previous and following words from word i are within the sentence
   read the n previous words from the target word (word i) and assign them to array buffer
   read the n following words from the target word (word i) and assign to array buffer
   if not
   assign empty value to array buffer
   end inner while
   if not
   increment i by one
   end inner while
   increment j by one
5. *end outer while*
6. *write the content of the array to file*
7. close file

Algorithm 4.3 Context Extraction algorithm

4.3 TRAINING AND TESTING DATASETS

Once all the necessary preprocessing tasks were done on the corpus, training and evaluating the selected algorithms followed. For this study there is no need to split the data into training and test sets for evaluation because of unsupervised nature of clustering algorithms [70].Table 4.2 shows the description of attributes in the data set is presented. In the table, *Rcontext (i)* and *Lcontext (i)* refer to (ten words to the left and right) the words that surrounds the ambiguous word to the right and left respectively, where i ϵ (1,

2, …, 10), the target word holds the ambiguous word and Word class takes the senses of the ambiguous word, but the word classes are not practically used for experimentation (clustering senses) rather, they were used for evaluation of clustering assignments. If the i^{th} left or right word from the target word doesn't exist, an empty value will be assigned to mean that there is no context. We have found that, the longest sentences in the corpus constitute a maximum of ten words to the left and the right of the ambiguous word. So we used 10 words to the left and the right of the ambiguous word as possible contexts. We were further explained using the following example which was extracted from the corpus.

Lc ***Rc***

welaj lj guroro bexta titanes poliyo tktk mesal kufN rubEla mekelakeya ktbat meker

In the above example, the target word "*mesal*" which is the ambiguous word and its word class is "cough" (see appendix B) that is its sense in this context. ***Lc*** refers to the left context where as ***Rc*** refers to right context. There are seven left contexts and five right contexts surrounding the target word which are labeled as is shown in the example. But there are no five right contexts (6, 7, 8, 9, and 10) and there are no three left contexts (8, 9 and 10) which will be assigned as empty. Note the word classes are used for evaluation of clusters assignment.

No	Attribute	Description	Value
1	Lcontext(i)	Used to hold the i^{th} left word from the ambiguous word	Any word in the corpus
2	Rcontext(i)	Used to hold the i^{th} Right word from the ambiguous word	Any word in the corpus
3	Target Word	Holds ambiguous word	Ambiguous word
4	Word Class	Holds the label of the target word class	Different sense of ambiguous word

Table 4.2 Description of attributes used for this study

4.4 EVALUATION TECHNIQUE

Evaluation of clustering result can be done in many ways[71]. Some of them are based on external criteria, i.e., the comparison of the resulting clustering solution with some pre-existing categories that were created manually. On the other hand, one can use internal criteria without resorting to gold standard clustering. The most important drawback of evaluation using internal criteria is that good score does not always correspond to good results of clustering in a given application[72]. For the purpose of this study, annotated corpus was used for evaluation. The problem with WSD is its small size. Therefore there is a risk of not capturing all of the peculiarities and biases of some large corpora in WSD.

We evaluate our method using sources of sense-tagged corpus. In supervised learning sense-tagged corpus is used to induce a classifier that is then applied to classify test data. Our approach, however, purely unsupervised and the sense tagged corpus was used to carry out an evaluation of the discovered sense groups. The way Weka evaluates the clustering's depends on the cluster mode you select[26]. For this study, class to cluster evaluation mode was selected in current implementation of Weka 3.6.4package in order to satisfy our evaluation method. In this mode Weka first ignores the class attribute and generates the clustering. Then during the test phase it assigns classes to the clusters, based on the majority value of the class attribute within each cluster. Then it computes the clustering error, based on this assignment and also shows the corresponding confusion matrix [26]. Based on the above technique its prediction accuracy was used to measure how well it has been able to generalize the clustering result.

4.5 SELECTED ALGORITHMS FOR TESTING

For this work, we have selected five clustering algorithms for experimentation with the existing implementation in Weka 3.6.4 package but we tried to choose algorithms representing a few different approaches or techniques (that is, partitional, hierarchical and probabilistic approach) to the problem of clustering.

As a task of WSD is a contextual one, we cluster contexts (text snippets) containing ambiguous word. From the context some real-valued features are extracted. So the

context is a vector of features *V* in high dimensional space. The feature vector comprises of attribute-value pairs, where the attributes are those contextual clues important for clustering.

First, we started with simple K-means algorithms, which represent simple, hard and flat clustering methods (see section 2.6.2). This algorithm has its drawbacks in terms of computational complexity, i.e., O (k (n−k) ^{2}), where n is number of contexts to cluster and k is number of centroid. This approach was applied in our experiments, as we have relatively small datasets.

Second, we choose agglomerative single, average and complete link clustering algorithms as representative family of hierarchical clustering algorithms (see section 2.6.1). Last but not least, we test also the Expectation Maximization algorithms also known as the EM which is probabilistic clustering algorithms (see section 2.6.3). It solves the maximization problem containing hidden (incomplete) information by an iterative approach[73]. In the setting of WSD, incomplete data means that the contextual features are not directly associated with word senses. The WSD is equivalent to choosing a sense that maximizes the conditional probability P (X|Y, Θ). And also its performance is still highly competitive. The precision and recall of the concept model in[74] reach 67.2% and 65.1% respectively.

4.6 SUMMARY

In this chapter, the design of the WSD system for Amharic was presented and discussed. Using the design, the process of corpus preparation, training and testing data sets, experimental evaluation technique and selected clustering algorithms for experimentation were illustrated in detail. The next chapter deals with the experimentation and discussion on the results of the experiment.

CHAPTER FIVE

EXPERIMENTATION AND DISCUSSION

5.1 INTRODUCTION

As discussed in the previous chapter, unsupervised word sense disambiguation was selected for this study. In this machine learning paradigm, clustering procedure tend to use a set of unlabeled data and automatically find sense distinctions. Usually those methods involve some form of clustering. An unsupervised WSD system deals with grouping of contexts for given word that express the same meaning without providing explicit sense labels for each group (e.g., without using a dictionary).

For WSD, learning the unsupervised machine learning procedures not required to providing explicit sense labels, where each data set example is described by a feature vector within each target word and in their sense label. The feature vector comprises of attribute-value pairs, where the attributes are those contextual clues important for clustering.

For these study five ambiguous words namely *mesal, qeretse, atena, metrat* and *mesasat* are trained for each ambiguous word with their corresponding data sets that are defined in Chapter four. The experimental settings are the same as in experiments presented in [23], where approaches based on supervised was tested. We use the same data set because we want to have a clear point of comparison with the previous work (supervised approach). We use the same features vector as in [23], i.e., bag of words, to simplify discussion. Contrary to[23], there is no need to split the data into training and test sets for evaluation because of unsupervised nature of clustering algorithms [70], [75]. These features are extracted from text in the following process. First a text window surrounding ambiguous word of ±10 segments (word) is constructed. Then the occurrence of a target word is noted in a feature vector for every dimension corresponds to different word. Then using Euclidian distance function, which is default in Weka package, can be used for measuring similarities between contexts. In this chapter the experimental procedures with the analysis of the experiment results was presented.

5.2 EXPERIMENTATION PROCEDURE

In this study a total of four experiments were conducted using simple K means, EM and agglomerative single, complete and average link clustering algorithms that are implemented in Weka 3.6.4 Package.

The first experiment was conducted to check to what extent stemming and stop word removal of Amharic words in the corpus will affect the accuracy of unsupervised Amharic WSD.

The second experiment sought to investigate the effect of different context sizes on disambiguation accuracy for Amharic ambiguous word, and to find out, if the standard two-word window applicable for other languages and especially English [75] holds for Amharic. In this regard, different training data sets was prepared for each ambiguous words, where the contextual information was obtained from 1-left and 1-right to 10-left and 10-right consequent surrounding words are prepared for each ambiguous word.

The third experimentation was conducted to see the effect of sense distribution on the performance of the algorithms that are listed above. The finally experiment was carried out to compare the accuracy of selected algorithms that are investigated in this study. These are (simple K means, EM, and Agglomerative single, complete and average link algorithms).

5.3 DISCUSSION OF RESULTS

This section presents and discusses the experimentation outputs for four of the experiments that are mentioned earlier.

Experiment I: The effect of stemming on the accuracy of the result (selected ambiguous word).

As discussed earlier, stemming has been found to give a significant improvement on performance of WSD for morphologically complex languages. This experiment is performed to test whether this applies to unsupervised WSD for Amharic. “Class to cluster” evaluation mode was selected to test the experiments. In this mode Weka first

ignored the class attribute and generated the clustering. Then during the test phase it assigned classes to the clusters, based on the majority value of the class attribute within each cluster. Then it computed the clustering error, based on this assignment and also showed the corresponding confusion matrix. From these its prediction accuracy was used to measure how well it has been able to generalize from the training data to evaluate the model.

The result of this experiment presented as follows:

Ambiguous word	Accuracy									
	K-Means		E M		Single Link		Complete Link		Average Link	
	Before	After	Before	After	Before	After	Before	After	Before	After
eTena	59.1	**63.3**	69.1	**75**	51.3	**51.5**	58.3	**58.7**	50.4	**51.4**
mesal	50.2	**67.8**	51.2	**67.1**	**44.3**	40.7	50.1	**52.4**	40.1	**40.6**
me`sa`sat	57.3	**61.9**	73.9	**74**	**50.6**	**50.6**	67.9	**72.2**	**50.8**	50.6
metrat	62.6	**71.6**	69.2	**73.1**	**50.6**	50.2	59.6	**63.3**	50.4	**52.7**
qereSe	53.7	**62.8**	68	**68.1**	53.6	**54.6**	54.6	**56.2**	52.5	**55.8**

Table 5.1 The effect of stemming on accuracy of the classifier using class to cluster evaluation test option.

As indicated in table 5.1 for all words, stemming improved the accuracy of all ambiguous words in simple K-means, Expectation Maximization and agglomerative complete clustering algorithms. But, for those ambiguous word '*mesal'* and '*metrat'* in single link and '*me`sa`sat'* in average link agglomerative clustering algorithms, stemming doesn't improve the accuracy of the algorithm. On average in all of each clustering algorithms, stemming improved the accuracy of the algorithms. The reason behind the enhanced accuracy might be that, stemming brings variants of a word into their common stem. This minimizes the consideration of the variants of a word as different word by WSD model.

As stated earlier, WSD models determine the meaning of a word by learning the pattern of surrounding words. If stemming is done, the variant of a word is taken as the same

pattern, which will improve the accuracy of the algorithms. For example, before stemming, surrounding words "ሰዎቹ" and "ሰዎች" would be assumed as different but, basically they are the variants of the same word " ሰው". After stemming, these words are taken as the same pattern. Therefore, in subsequent experiments the stemmed dataset was used as it enhanced the performance of the models.

Experiment II: Determining optimal context window

In English a standard two-word window on either side of the ambiguous word is found to be enough for disambiguation[75]. For Amharic supervised WSD, three window size on both sides for the ambiguous word is found to be enough[23]. But, this has not been established for Amharic unsupervised WSD. For this study experiments were carried out ten times for each classifier to determine an average optimal window size from one-one window to ten-ten window on both side of the ambiguous word.

Window size	eTena	mesal	me`sa`sat	metrat	qereSe
1-1	55.2	68.9	**74.4**	**73.1**	55.1
2-2	67.6	69.8	67.2	71.5	72
3-3	**79.4**	**76.8**	65.1	71.5	**73.1**
4-4	73.4	68.9	57.5	71.5	73.1
5-5	73.8	65.6	52.9	71.5	70.4
6-6	57.3	65.6	52.9	71.5	60.9
7-7	53.7	65.6	52.9	71.5	57.2
8-8	56.7	65.6	51.3	71.5	56.3
9-9	57.3	65.6	51.3	71.5	55.1
10-10	57.3	65.6	51.3	71.5	55.1

Table 5.2 Summery of Window Size experiment for Simple K-Means clustering algorithms

Window size	eTena	mesal	me`sa`sat	metrat	qereSe
1-1	73.6	73.6	**75.9**	70.4	69.9
2-2	70.7	70.7	73.4	72	**74.1**
3-3	**76.9**	**76.9**	73.9	**73.1**	67.9
4-4	73.9	73.9	73.4	70.9	65.8
5-5	69.6	69.6	73.9	68.8	63
6-6	77.4	77.4	73.9	67.8	69.3
7-7	74.3	74.3	73.9	69.3	72.5
8-8	74.8	74.8	73.9	67.2	70.9
9-9	74.8	74.8	73.9	66.7	67.2
10-10	75.8	75.8	74.6	66.7	60.9

Table 5.3 Summery of Window Size experiment for Simple E.M clustering algorithms

Window size	eTena	mesal	Me`sa`sat	metrat	qereSe
1-1	52.9	56.9	74.9	**72**	55.1
2-2	**70.7**	54.4	**77.1**	69.4	57.2
3-3	55.3	**57.4**	51.8	68.3	50.3
4-4	62.9	54.4	50.8	66.2	56.7
5-5	67.6	51.1	52.4	63.5	**59.8**
6-6	51.1	50.1	76.4	61.4	55.1
7-7	50.6	49.2	69.8	52.4	56.7
8-8	63.5	50.5	75.4	61.4	53
9-9	51.6	50.3	75.4	59.3	52.4
10-10	61.6	49.7	75.4	59.8	56.1

Table 5.4 Summery of Window Size experimentation for CL clustering algorithm

Window size	eTena	mesal	Me`sa`sat	metrat	qereSe
1-1	**52.1**	40.5	50.5	53	54
2-2	51.6	40.5	**50.8**	51.9	55.1
3-3	51.6	40.7	50.8	51.9	**55.4**
4-4	50.6	**41.5**	50.8	53.9	55.4
5-5	50.6	40.8	50.8	**54**	55.4
6-6	51.6	40.5	50.8	52.4	55.1
7-7	51.6	40.5	50.8	52.4	55.1
8-8	51.6	40.8	50.3	52.4	55.1
9-9	51.6	40.4	50.3	52.4	55.1
10-10	51.6	40.5	50.3	52.4	55.1

Table 5.5 Summery of Window size experimentation for Single Link clustering algorithm

Window size	eTena	mesal	Me`sa`sat	metrat	qereSe
1-1	**52.6**	40.5	50.3	51.9	55.1
2-2	52.1	**41.7**	50.3	51.9	**58.3**
3-3	51.6	41.1	50.3	51.9	46.7
4-4	50.6	41.1	50.8	**52.6**	55.1
5-5	50.6	41.3	**51.3**	52.6	55.6
6-6	51.6	40.3	50.8	52.6	55.6
7-7	51.6	40.3	50.8	52.6	55.1
8-8	51.6	40.3	50.3	52.6	55.1
9-9	51.6	40.3	50.3	52.6	55.1
10-10	51.6	40.3	50.3	52.6	55.1

Table 5.6 Summery of Window size experimentation for Average Link clustering algorithm

As shown in table 5.2 and 5.3, for three of the ambiguous word for simple k means (*eTena, mesal* and *qereSe*) and EM (*eTena, mesal a*nd *metrat*) the maximum accuracy was achieved on three-three word window size. Whereas, for simple k means (*me`sa`sat* and *metrat)* the highest accuracy was attained on one-one word window and for EM (*me`sa`sat* and *qereSe*) the highest accuracy was attained on one-one and two-two word window respectively.

As shown table 5.4, 5.5 and 5.6, summary of experiments for different word windows for Agglomerative Single (SL), complete link (CL) and Average Link (AL) clustering algorithms. In Agglomerative complete link and single link clustering algorithms, two of the ambiguous word for CL (*eTena* and *me`sa`sat*) and for SL (*mesal* and *qereSe*) the maximum accuracy was achieved in two-two word window. Whereas for CL (*mesal, metrat and qereSe*) the highest accuracy was achieved on three-three, one-one and five-five word windows and SL (*eTena,* me`sa`sat and *metrat*) the highest accuracy was achieved on three-three, five-five and four-four word window respectively. On the other hand, in agglomerative Average link clustering algorithms, as shown in table 5.6, five of ambiguous word, (*eTena*, mesal, *me`sa`sat*, *metrat* and *qereSe*) the maximum accuracy was dispersed over one-one to five-five word window for each ambiguous word. As a result, average link clustering algorithm was not included in experimentation and discussion for determining optimal context window.

As shown in the table 5.2 to 5.6, all the five ambiguous words and for each clustering algorithms, the result agreed with the findings in other language that the nearest words surrounding the ambiguous word give more disambiguation information than words far from the ambiguous word[75].

For this study, since in all ambiguous words, the average accuracy of windows after 3-3 window for Simple k means and EM clustering algorithms was less than that of a 3-3 window. Window size of 3-3 was considered to be effective for Simple K means and EM clustering algorithms. On the other hand for agglomerative SL and CL clustering algorithms the average accuracy of word windows two-two considered to be effective.

Using a 3-3 (k means and EM) and 2-2 (SL and CL) window size for the final accuracy of the Ambiguous words were summarized in table 5.7:

Ambiguous word	Window Size			
	Three-Three		Two -Two	
	K means	EM	Single Link	Complete Link
eTena	79.4	76.9	52.1	70.4
mesal	76.8	76.4	41.7	54.4
me`sa`sat	65.1	73.9	50.3	71.1
metrat	71.5	73.1	51.9	69.4
qereSe	73.1	67.9	58.3	57.2

Table 5.7 Summary of Accuracy of classifiers using 3-3(K means and EM) and 2-2(SL and CL) window size

As indicated in table 5.7, the algorithms achieved accuracy within the range of 65.1 – 79.4 in simple k means, 67.9 - 76.9 in EM, 54.4 – 71.1 in Complete Link and 51.9 – 58.3 for Single link clustering Algorithms. Ide and Veronis [27] have shown that unsupervised method yields current state-of-the-art accuracy in the range 60–70 %, WSD has been one of the most important open problems in NLP.

Experiment III: Effect of Distribution of training data on accuracy

WSD performance can be affected by the distribution of training data for each sense. In this study, a balanced distribution of training data was employed to maximize performance of the result. But, naturally the sense distribution might vary for each sense. To test the effect of distribution of senses on accuracy, experiments using three-three (for EM and k means) and two-two (for CL) word window sizes were conducted. For these experiments the following algorithms were selected according to the results recorded in experiment II (optimal context window). These are Simple k means, EM and Complete Link clustering algorithms. According to experiment II, EM, simple k mean and

Complete Link algorithm results an accuracy of 1, 2 and 3 respectively. We intentionally vary the distribution of senses and compared the result with balanced senses distribution (experiment II).

Ambiguous word	Sense Distribution	Balance		Accuracy			Unbalance		Accuracy		
				K-M	EM	CL			K-M	EM	CL
eTena	Strengthen	100	200	**79.4**	76.9	**70.4**	100	140	77.4	**78.4**	52.6
	Study	100					40				
mesal	Cough	100	245	**68.9**	71.8	**54.4**	65	187	67.5	**77.4**	50.1
	Sharp	72					72				
	Vow	73					50				
Me`sa`sat	Taking care	100	200	**62.1**	**73.9**	**71.1**	100	160	60.3	71.4	70.3
	Thin	100					60				
metrat	Call	100	200	**71.5**	**72**	**69.4**	60	160	66.3	66.3	64.4
	Clean	100					100				
qereSe	Record	100	200	**73.1**	**67.9**	57.2	100	170	60.6	56.7	**61.2**
	Shape	100					70				

Table 5.8 Summery of experimentation on effect of sense distribution on accuracy

As presented in table 5.5, the accuracy of unbalanced sense distribution resulted less accuracy for all five ambiguous words in simple k means, three of the ambiguous word (except, *eTen* and mesal) in EM and four of the ambiguous word (except, *qereSe*) in agglomerative Complete Link. The finding supports the findings in other studies that the accuracy of the algorithms degrades significantly when the training samples have different distributions for the senses as there will be bias to the highest number of sense distribution[31].

Experiment IV. Comparison of clustering algorithms that are tested in this study and compare the results supervised algorithms.

According to the result of the above experiments (III), Expectation maximization and simple k means clustering algorithms achieved almost the same result in optimal context window size and the effect of stemming on the accuracy of the result. On the other hand, effect of sense distribution in accuracy in experiment III, simple k means algorithm performs a better result of all clustering algorithms. Next to simple k means, EM performed a good result.

In Agglomerative Single, Complete and Average Link clustering algorithms, the worst result was recorded in Average Link algorithms next to Single Link. But, the results of Agglomerative Complete Link clustering algorithm was interesting with compared to Single and Average Link but, it's still a problem when the number of senses (classes) increased. For example the Ambiguous word "*mesal*", there are three senses *'sharp'*, *'vow' and 'cough'*. In this ambiguous word Complete Link performs a least result as off Simple k means and EM algorithms (that is, 76.8, 76.4, **54.4** and 67.8, 67.1, **52.4** in experimentation II and Experimentation I respectively).

The results of this study (the accuracy of unsupervised Amharic WSD algorithms) were compared to the accuracy of supervised Amharic WSD algorithms, because the evaluation corresponds to our evaluation, both studies used average accuracy to measure the result of the ambiguous words. Also both approaches were tested on the same dataset and using the same feature set the supervised Amharic WSD algorithm achieved the accuracy of 70.1 to 83.2% as reported by[23]. On the other hand, the best clustering algorithm was an accuracy of 65.1 to 79.4 % in Simple k means, 67.9 to 76.9 in EM and 54.4 to 71.1 in Complete Link clustering algorithms respectively. The results of unsupervised Amharic WSD Algorithms were encouraging that compared to Supervised Amharic WSD algorithm reported by [23] because, usually unsupervised approaches have trouble with sense annotated datasets[31].

5.4 SUMMARY

In this chapter the experimental procedures together with presentation and discussion of four experiments were covered. The first experiment was showed that stemming significantly improved the accuracy of the result. In successive experiments using classes to cluster evaluation mode, an experiment also carried out to determine optimal window size for the all ambiguous word; as a result, three-three word window was found most favorable window size for Expectation Maximization and Simple k means and two-two word window for agglomerative Single and Complete Link clustering algorithms. Using three-three and two-two window, the final accuracy of unsupervised Amharic WSD algorithms were achieved within the range of 65.1 to 79.4 % for Simple k means, 67.9 to 76.9 in EM, 54.4 – 71.1 in Complete Link and 51.9 – 58.3 for Single link clustering Algorithms which was encouraging that compared to Supervised Amharic WSD reported by [23]. The finally, experiment was carried out to test the effect of sense distribution on the accuracy of the algorithm and it was found that balanced sense distribution scored a better result in increased accuracy than unbalanced sense distribution.

CHAPTER SIX

CONCLUSIONS AND RECOMMENDATIONS

6.1 CONCLUSIONS

The overall focus of this research is Word Sense Disambiguation which addresses the problem of automatically deciding the correct sense of an ambiguous word based on its surrounding context's. WSD is considered tool for NLP like MT and IR and other applications.WSD is considered to be one of the most challenging of all NLP research areas due to its reliance on a varied range of linguistic and statistical knowledge.

The problem of WSD, addressed for Amharic which is one of less studied language. Though Amharic has many ambiguous words only five ambiguous words were selected and model for each ambiguous word were built. The words are *eTena, mesal, me`sa`sat, metrat,* and *qereSe*.

The most popular approaches to WSD rely on supervised machine learning methods, where a machine learning classifier is required to be trained on manually labeled training instances, to generate a classifier model that can be used to classify future instances. But manually labeling (annotation) training instances is too costly and time consuming. Mihalcea [6] identified that the cost of annotation preparing corpuses for supervised classification algorithm is higher, because large effort is required during manual annotation.

In this study, unsupervised machine learning approach using five selected algorithms were used; these are Simple k means, EM and agglomerative single, average and complete link clustering algorithms. This method avoids the problem of knowledge acquisition bottleneck, that is, lack of large-scale resources manually annotated with word senses. This approach to WSD has been based on the idea that the same sense of a word will have similar neighboring words. They are able to induce word senses from input text by clustering word occurrences, and then classifying new occurrences into the induced clusters. They do not rely on labeled training text and, do not make use of any

machine–readable resources like dictionaries, thesauri, ontology, etc. However, the main disadvantage of fully unsupervised systems is that, as they do not exploit any dictionary, they cannot rely on a shared reference inventory of senses and the result accuracy is less than that of supervised WSD method [37]. For this study, a monolingual corpus of English language was used to acquire sense examples and the sense examples were translated back to Amharic which is one approach of tackling knowledge acquisition bottleneck.

Based on selected algorithms, experiments on Weka 3.6.4 package, we conclude that simple k means and EM clustering algorithms were achieved higher accuracy on the task of WSD for selected ambiguous word, provided with balanced sense distribution in corpus. We have achieved accuracy within the range of 65.1 to 79.4 % for Simple k means, 67.9 to 76.9 for EM and 54.4 to 71.1 for Complete Link clustering algorithms for the five ambiguous words. But the worst results were recorded in single and Average link clustering algorithm.

We also found that, stemming of Amharic words in the corpus enhanced the accuracy of the algorithms. The accuracy was increased after stemming was applied to words in the corpus.

For Amharic unsupervised WSD, there is no standard optimal context window size before, which refers to the number of surrounding words that are sufficient for extracting useful disambiguation. Based on this study, we have found that three-word window on each side of the ambiguous word was enough for disambiguation for Simple k means and EM, and two-word window for Complete Link algorithms for all ambiguous words.

We also found that balance sense distribution of ambiguous word was a crucial factor in improved the accuracy using Simple k means, EM and complete link clustering algorithms. The best accuracy in our experiment was seen for all ambiguous words that have a balanced sense distribution.

Finally, the results of this study; accuracy of unsupervised machine learning approach for WSD to Amharic words was compared to the accuracy of supervised Amharic WSD

algorithm reported by [23]. The best unsupervised Amharic WSD algorithms have an accuracy of 65.1 to 79.4 % for Simple k means, 67.9 to 76.9 for EM and 54.4 to 71.1 for Complete Link respectively. These results (unsupervised Amharic WSD) were encouraging as compared supervised Amharic WSD , as usually unsupervised approaches have troubles with sense annotated datasets as of 70.1 to 83.2% reported supervised Amharic WSD [23]

In total, the chosen methodology, unsupervised machine learning approach for Amharic word sense disambiguation has been justified in terms of its theoretical foundations as well as the results obtained in our experiments for selected Amharic Ambiguous words.

6.2 RECOMMENDATIONS

We have the following recommendations which include the developments of resources and future research directions for WSD for Amharic text:

1. Researches in WSD for other languages use linguistic resources like Thesaurus, Lexicon like WordNet, machine readable dictionaries and machine translation software. In this study, we faced a significant challenge as Amharic lacks those resources. Taking into account their contribution to WSD and other researches concerned institutions should develop these resources.
2. For other language a standard sense annotated data are available for WSD research and also for testing a WSD systems. We don't have such data for Amharic language which makes the study to be limited for five ambiguous words. So, there need to be an initiative to prepare the data for WSD research.
3. Future research directions for WSD in Amharic include:
   - Extending this experimentation using Supervised and unsupervised WSD for other ambiguous words in addition to those covered in the research.
   - Due to time limitation, in this study only five clustering algorithms were experimented that are implemented in Weka 3.6.4 package. But other algorithms like Clustering by Committee (CBC), Growing Hierarchical Self-Organizing Map (GHSOM) and Graph-based algorithms has been tested as they are used and found to yield impressive result for other language[16], [25].

- In addition to corpus based approach, there are also knowledge based and hybrid approach (combination of knowledge base and corpus based approach) which are used for WSD for other language and found a good result [6], [34]. These approaches need to be investigated for Amharic as well.
- As unsupervised WSD are based on unlabeled corpora, and do not exploit any manually sense-tagged corpus to provide a sense choice for a word in context, these results less accuracy than other approaches. A research should be conducted using bootstrapping approach with is required little training data and yields a very high performance. For example, an evaluation of Yarowsky's bootstrapping algorithm leads to very high performance over 90% accuracy on a small-scale data set [1]. This approach overcome the main problems of supervision and the data scarcity problem specially lack of annotated data like Amharic.

REFERENCES

1. Yarowsky, D. *unsupervised word sense disambiguation rivaling supervised methods*. in *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. 1995. Cambridge, M.A.

2. Clara, Allan M., Lotofus., and Elizabeth, F., *A spreading-activation theory of semantic processing.* Psychological Review, 1975. **Vol 86**(6).

3. Anderson, J. R., *Language, Memory, and Thought.*1976: Hillsdale, NJ.

4. Anderson, J. R., *A Spreading Activation Theory of Memory.* Journal of Verbal Learning and Verbal Behavior, 1983. **Vol 22**.

5. Masterman, M. and M. Masterman, *Semantic message detection for machine translation using interlingua*, in *Semantic message detection for machine translation using interlingua,*1969: Her Majesty's Stationery Office, London.

6. Mihalcea, R. *Knowledge-based methods for WSD*. in *Proceedings of the 3rd InternationalWorkshop on the Evaluation of Systems for the Semantic Analysis of Text*. 2006. (Senseval-3, Barcelona, Spain).

7. Robert, A., *The structure of the Merriam-Webster Pocket Dictionary. Austin, TX: University of Texas at Austin*, 1980.

8. Hearst and Marti A., *Noun homograph disambiguation using local context in large corpora*, in *presented at Proceedings of the 7th Annual Conference of the University of Waterloo Centre for the New OED and Text Research,*19 99: Oxford, United Kingdom.

9. Gale, W., Church, K. , and Yarowsky, D. . *Estimating upper and lower bounds on the performance of word sense disambiguation programs*. in *in Proceedings of the 30th Conference of the Association for Computational Linguistics*. 1992. Newark, Delaware.

10. Lesk, M. . *Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone*. in *Proceedings of the SIGDOC Conference*. 1986. Toronto, Ontario.

11. FDRE Statistic Agency. *Census summery of Ethiopia* 2007 [cited 2011 Accessed on March 02]; Available from: http://www.CSA.gov.et/ Census-Summery-Final-Report.pdf.

12. Roberto, N. , *Word Sense Disambiguation: A Survey.* ACM Computing Surveys, 2009. **Vol 41**(2).

13. Yehenew, S., *Design and Development of Human-aided Rule-based English Amharic Machine translation prototype, Master's Thesis*, 2004, Addis Abeba university.

14. Yoseph, S., *Application of multilingual Thesauri for cross language information retrieval(CLIR)[Amharic –English CLIR for the legal Environment] , Master's Thesis*, 2004, Addis Ababa University.

15. Atelach A., Askar, L., Richard, C., and Jussi, K. *Dictionary based Amharic-English Information Retrieval*. in *Proceedings of the third Workshop of the Cross-Language Evaluation (CLEF)*. 2004. Bath, England.

16. Lin, D. and Pantel, P. . *Discovering word senses from text*. in *In Proceedings of the 8th ACM SIGKDDInternational Conference on Knowledge Discovery and Data Mining*. 2002. Edmonton, Alta., Canada.

17. Atelach, A., *Automatic Sentence Parsing for Amharic Text, An Experiment Using Probabilistic Context Free Grammars, Masters Thesis*, 2000, Addis Abeba University.

18. Daniel, A, *An integrated approach to automatic complex sentence parsing for Amharic text, Masters Thesis*, 2002, Addis Ababa University.

19. Wube, A., *Rule Based Syntactic Disambiguation Parser for Amharic Sentence*, *Masters Thesis*, 2004, Addis Ababa University.

20. Teshome, K., *Word Sense disambiguation for amharic text retrieval: a case study for legal documents. Master Thesis*, 1999, Addis Ababa University.

21. Xinglong, W. and John, C. . *Word Sense Disambiguation Using Automatically Translated Sense Examples*. in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*. 2005. Association for Computational Linguistics.

22. Dagan, I. and Itai, A. , *Word sense disambiguation using a second language monolingual corpus.* Computational Linguistics, 1994. **Vol 20**(4).

23. Solomon, M., *Word Sense Disambiguation for Amharic words , A Machine Learning Approach*, *Master's Thesis*, 2010, Addis Abeba University.

24. Girma, G. *በአማርኛ ሥርአተ-ጽሕፈት ውስጥ የድምፀ-ምክሸ ሆሄያት አጠቃቀም ማስታወሻ.* 2007 [cited 2010 Retrived on April 10]; Available from: http://www.nlp.amharic.org/resources/lexical/word-lists/homonyms/homonym-collected-by- girma-getahun/.

25. Bartosz, Broda and Wojciech, Mazur, *Evaluation of clustering algorithms for polish word sense disambiguation*, in *Proceedings of the IMCSIT*.2010, IEEE. p. 25–32.

26. Ian H.Witten and Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques.* Second: ed2005: Morgan Kaufmann publications.

27. Ide, N. and Veronis, J., *Word Sense Disambiguation: The State of the Art.* Computational Linguistics, 1998.

28. Voorhees, E. M., *Using WordNet for text retrieval*, in *WordNet: An Electronic Lexical Database.*1998, MIT Press.

29. Han, J. and Kamber, M. , *Data Mining – Concepts and Techniques.* 2001: Morgan Kaufmann.

30. Pasca, M. and Harabagiu, S. *The informative role of WordNet in Open-Domain Question Answering*. in *Proceedings of NAACL-01 Workshop on WordNet and Other Lexical Resources*. 2001. Pittsburgh, PA.

31. Karov, Y. and Shimon, E., *Similarity-based word sense disambiguation.* Computational Linguistics, 1998. **Vol 24**(1).

32. Zhang, T., Ramakrishnan, R., and Livny, M. *BIRCH: An efficient data clustering method for very large databases*. in *Proceedings of SIGMOD-96*. 1996. Montreal, Canada.

33. Yorick Wilks, *Formal semantics of natural language.*1975: Cambridge University Press, Cambridge, UK.

34. King, B., *Step-wise clustering procedures.* Journal of the American Statistical Association, 1967. **Vol 69**: p. 86–101.

35. Yarowsky, D. . *Word-sense disambiguation using statistical models of Roget's categories trained on large corpora*. in *Proceedings of the Fourteenth International Conference on Computational Linguistics*. 1992. Nantes, France.

36. Brown, Peter F., Pietra, Stephen A. Della, Pietra, Vincent J. Della, and L., Robert. *Word-sense disambiguation using statistical methods*. in *In Proceedings of the29th Annual Meeting of the ACL*. 1991.

37. Doina, T., *Word sense disambiguation by machine learning approach: a short survey.* Informatica, 2004. **Vol XLIX**(2).

38. Smeaton, A.F., *Linguistic Approaches to Text Management: An Appraisal of Progress.* Journal of Document & Text Management, 1995. **Vol 2**(2).

39. Resnik, Philip and David, Y. , *Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation.* Natural Language Engineering, 2000. **Vol 5**(2).

40. Artstein, R. and Poesio, M., *Inter-coder agreement for computational linguistics.* Computational Linguistics, 2008. **Vol 34**(4): p. 555–596.

41. Hutchins, J. and Sommers, H., *Introduction to Machine Translation.*1992: Academic Press.

42. Nello C., John S., and Huma L. *Latent semantic kernels*. in *Proceedings of 18th International Conference on Machine Learning*. 2001.

43. Jain, A. K., Murty, M. N., and Flynn, P. J. , *Data clustering: a review.* ACM Computing Surveys, 1999. **Vol 31**(3): p. 264–323.

44. Sneath, P. H. A. and Sokal, R. R., *Numerical Taxonomy: The Principles and Practice of Numerical Classification.*1973, London, UK: Freeman.

45. Nagy, G. . *State of the art in pattern recognition*. in *Proceedings of IEEE*. 1968.

46. Jain, A. K. and Dubes, R. C., *Algorithms for Clustering Data.*1988: Prentice-Hall.

47. Kaufmann, L. and Rousseeuw, P. J. , *Finding Groups in Data: an Introdution to Cluster Analysis.*1990: Wiley and Sons.

48. Manning, C. D. and Schütze, H. , *Foundations of Statistical Natural Language Processing.*1999: MIT Press.

49. Kaufmann, L. and Rousseeuw, P. J. , *Clustering by means of medoids*, in *In Dodge, Y. (Ed.) Statistical Data Analysis based on the L 1 Norm.*1987: Elsevier/North Holland, Amsterdam. p. 405–416.

50. Duda, R. and Hart., P., *Pattern Classification and Scene Analysis.* Wiley, New York, NY., 1973.

51. Cutting, D. R., Karger, D., Pedersen, J., and Tukey, J. W. . *Scatter/Gather: A cluster-based approach to browsing large document collections*. in *In Proceedings of SIGIR-92*. 1992. Copenhagen, Denmark.

52. Guha, S., Rastogi, R., and Kyuseok, S. . *ROCK: A robust clustering algorithm for categorical attributes*. in *In Proceedings of ICDE'99. pp. 512–521*. 1999. Sydney, Australia.

53. Guha, S., Rastogi, R., and Shim, K. . *Cure: An efficient clustering algorithm for large databases*. in *Proceedings of SIGMOD-98. pp. 73–84*. 1998. Seattle, WA.

54. Karypis, G., Han, E.-H., and Kumar, V. , *Chameleon: A hierarchical clustering algorithm using dynamic modeling.* IEEE Computer: Special Issue on Data Analysis and Mining, 1999. **Vol 32**(8): p. 68–75.

55. Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. *A density-based algorithm for discovering clusters in large spatial databases with noise*. in *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. 1996. Portland, OR.

56. Ankerst, M., Breunig, M.M., Kriegel, H.-P., and Sander, J. . *OPTICS: Ordering Points to Identify the Clustering Structure*. in *In Proceedings of ACM SIGMOD International Conference on Management of Data* 1999. Philadelphia, PA.

57. Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. . *Automatic subspace clustering of high dimensional data for data mining applications*. in *In Proceedings of ACM SIGMOD International Conference on Management of Data*. 1998. Seattle, Washington.

58. Marvin L. Bender, Head W. Sydeny, and Roger Cowley, *The Ethiopian Writing System,* ed. I.B.e.a.E.L.i. Ethiopia1976, London: Oxford University press.

59. Yacob, D. . *System for Ethiopic Representation in ASCII (SERA)*. 1996 [cited 2010 Accessed on 12 March]; Available from: from: http://www.abyssiniacybergateway.net/fidel/.

60. Getachew Haile, *The Problems of Amharic Writing System*, 1967, Unpublished.

61. Berhane, Girmaye, *Word formation in amharic.* Journal of Ethiopian Languages and Literature, 1992: p. 50–74.

62. Saba, A and Dafydd, G., *Finite state morphology of amharic*, in *International Conference on Recent Advances n Natural language processing*2005: Borovets. p. 47–51.

63. Baye Yemam, *የአማርኛ፣ሰዋስው□□*1987 ዓ.ም፡ ት.መ.ማ.ማ.ድ. .።።.

64. Dawkins, C.H. , *The Fundamentals of Amharic*1969: A.A Sudan interior mission.

65. Getahun A, *the Analysis of Ambiguity in Amharic.* JES, 2001. **Vol XXXIV**(2).

66. Amsalu Aklilu, *Amharic English Dictionary.* 1987, Addis Ababa: Kuraz Publishers.

67. Agirre, E. and Martinez, D. *Exploring automatic word sense disambiguation with decision lists and the web*. in *Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content*. 2000.

68. Broda, M. B. and Piasecki, M. . *Evaluating lexcsd — a weaklysupervised method on improved semantically annotated corpus in a large scale experiment*. in *proceedings of Intelligent Information Systems, Eds*. 2010.

69. Gebeyehu, K. , *The application of decision tree for part of speech (pos) tagging for amharic, Master Thesis.* , 2009, Addis Ababa University.

70. Zhao, Y. and Karypis, G., *Empirical and theoretical comparisons of selected criterion functions for document clustering.* Machine Learning, 2004. **Vol 55**(3): p. 311-331.

71. Forster, R., *Document clustering in large german corpora using natural language processing, Ph.D. dissertation*, 2006, University of Zurich.

72. Manning, C. D., Raghavan, P., and Schütze, H., *Introduction to Information Retrieval*. 2008: Cambridge University Press.

73. Dempster, A., laird, N., and Rubin, D. , *Maximum likelihood from incomplete data via the EM algorithm.* . Roy. Statist. Soc. , 1977. **Vol 39**: p. 1-38.

74. Bruce, R. and Wiebe, J. *Word-sense Disambiguation Using Decomposable Models*. in *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*. 1994. Las Cruces, New Mexico.

75. Kaplan, A. , *An experimental study of ambiguity and context.* Mechanical Translation. , 1955. **Vol 2**(2).

Appendix A. Selected ambiguous words and their Amharic meaning adopted from Girma (25).

ድምፀ-ሞክሼ ቃላት	ፍቺዎች
መሣሣት መሣሣት	መቅጠን፤ ዘርዛራ ወይም ሥሥ መኾን በሥሥት ወይም በጥንቃቄ መያዝ
መሳል መሳል መሳል	መለመን፤መጠየቅ ኡህ ኡህ ማለት መቀኑራጫ ጠረዝን ማትባት
መጥራት መጥራት	ጥሪ ማረግ መጽዳጽ፤ ከጉድፍ፣ ወዘተ. መላቀቅ
ቀረጸ ቀረጸ	ድምጽን ቀዳ ምስል አወጣ
አጠና አጠና	ጥናት አደረገ፤ ተረዳ፤ መረመረ አጠነከረ፤ አበረታ

Appendix B. Sample list of English sense examples used with their Amharic equivalent translation.

1. I made a vow to St.Gabriel to fast for two days

   ለቀዱስ ገብርኤል ሁለት ቀን ለመፆም ተሳልኩ ::

2. Over and above this, men might vow individuals or possessions to God as a thank-offering.

   ከዚህ በተጨማሪ ወንዶች ሰዎችን ወይም ያላቸውን ንብረት እግዚያብሔርን ለማመስገን ይሳሉ ነበር ::

3. This harmony might be expressed as an offering which accompanies a vow of some kind or as a thank-offering or free-will offering.

   ይህ ስምምነት የሚገለፀው የሆነ ነገር በፍቃደኝነት ለመስጠት ወይም ለማመስገን በመሳል ነው ::

4. Jacob vowed a vow, saying, “If God will be with me, and will keep me in this way that I go, and will give me bread to eat, and clothing to put on, so that I come again to my father’s house in peace, and Yahweh will be my God,

   ያእቆብ እግዚአብሔር በሄድኩበት ከጠበቅከኝ፣ የምበላውን እና የምለብሰውን ከሰጠኸኝ፣ ወደ አባቴ ቤት እመለሳለው እግዚአብሔርም አምላኬ ይሁናል ብሎ ተሳለ ::

5. I am the God of Bethel, where you vowed a vow to me. Now arise, get out from this land, and return to the land of your birth.’”

   እኔ ስለት የተሳልክልኝ የቤቴልሔም አምላክ ነኝ፤አሁን ከዚህ ምድር ተነሳና ወደ ተወለድክበት ምድር ተመለስ::

6. Now, after years of hard work, we are in sight of immunizing all the world's children against polio, tuberculosis, diphtheria, whooping cough, tetanus and measles.

   ከብዙ ጠነካራ ሥራ በኋላ ሁሉንም ያዓለም ሕፃናት ከ ፖሊዮ ፣ የሳንባ ነቀርሳ፣ የጉሮሮ በሽታ፣ የትክትክ ሳል ፣ ቲታነስ እና ኩፍኝ ክትባት መስጠት ችለናል ::

7. In the United Kingdom, parents are advised to have their children immunised against diphtheria, tetanus, polio, whooping cough, measles and rubella .

   በእንግሊዝ ወላጆች ልጆቻቸውን ከ ጉሮሮ በሽታ፣ ቲታነስ ፣ፖሊዮ፣ የትክትክ ሳል፣ ኩፍኝ፣ ሩቤላ የሚከላከል ክትባት እነዲያስከትቡ ተመከሩ ::

8. The days of being forced to get out of it on cough medicine are well behind.

   የሳል መድኃኒት አልቆ የምንቸገርበት ጊዜ አልፏል ::

9. Greenough and colleagues showed that babies who did not require respiratory support had a high prevalence of wheeze and cough in the first year of life.

   ግሪኖፍ እና ጓደኞቹ የመተንፈሻ አካላት እርዳታ የማያስፈልጋቸው ህፃናት ማቃተትና ሳል በመጀመሪያ ዓመታቸው እንደሚያጋጥማቸው አሳዩ::

10. The commonest symptom for the disease is coughing persistently, with frequent chest infection.

    የበሽታው የተለመዱ ምልክቶች በተደጋጋሚ መሳል እና የሚደጋገም የደረት ህመም ናቸው ::

11. Charlton applied a sharp knife, carving it into steaks in the kitchen.

    ቻርልተን ማብሰያ ቤት ውስጥ ጥብሱን ለመከተፍ የተሳለ ቢላ ተጠቀመ ::

12. A terrific place to have breakfast in, not a knife sharp enough to cut a lemon.

    ቁርስ ለመብላት የማይመች ቦታ ነው፤ ሎሚ ለመቁረጥ የሚሆን እንኳን የተሳለ ቢላ የለም ::

13. The film's sharp sword has many edges.

    ፊልሙ ላይ ያሉት የተሳሉ ጎራዴዎች ብዙ ጠርዞች አላቸው ::

14. ‘Even now, the memories are sharp as broken glass.

    አሁንም ትዝታዎቹ ልክ እንደ ተሰበረ ብርጭቆ የተሳሉ ናቸው ::

15. Again do not round over the sharp edges when sanding.

    አሁንም አሸዋ ስታፈስ በተሳሉ ጠርዞች ላይ አትዙር ።።

16. To call Graf and Kohde-Kilsch a team on this showing is a misnomer.

    ግራፍንና ኮህድ ከልሰችን በዚህ ትርይታቸው ቡድን ብሎ መጥራት ትክክል አይሆንም ።

17. Azeglio Vicini, can call on an almost full-strength squad for a game in which he hopes Italy will prove his assertion.

    አዚግልዮ ቪኒቺ የጣሊያንን ብቃት የሳያል ተብሎ የሚጠበቅ ጠንካራ ተብሎ ሊጠራ የሚችል ቡድን ይዟል ።

18. The house was fall of memories; but even to call them memories was to imply that Jack had put them behind him; and he had not.

    ቤቱ በትዝታ የተሞላ ነው ፤አንደውም ትዝታ ተብሎ የሚጠራው ጃክ ከጀርባው አርጓቸው ነው በሚል ነው፤ግንአይደለም።።

19. ‘All the words he uses are what you would call anti-feminist,’ said a police officer.

    ፖሊሱ እንዳለው እሱ የሚጠቀማቸው ቀላት በሙሉ ፀረሴት ተብለው ሊጠሩ የሚችሉ ናቸው ።

20. Sorry, that's what we call the Monday morning meeting where we discuss what's going on.

    ይቅርታ ሁሉንም ነገር የምንወያበት የሰኞ ጠዋቱ ስብሰባ ብለን የምንጠራው ስብሰባ ይህ ነበር።

21. **A** man is only ‘acceptable’ to females if he is ‘nice and clean’.

    ወንድ የሴቶች ተቀባይነት የሚያገኘው ጥሩ እና ጠራ ያለ ሲሆን ነው ።።

22. It was poorly and sparsely furnished; a brave effort had been made to keep it tidy and clean.

ጥሩ ባልሆነና በተራራቀ ሁኔታ ነው እቃዎቹ የተቀመጡት፤ የፀዱና የጠሩ ለማረግ ጥሩ ጥረት ተደርጓል ::

23. In our minds ‘eating everything that is placed in front of us’ is associated with ‘well done, that's a nice clean plate’.

በዐዕምሯችን ፍለፊታችን የቀረበልንን ምግብ መብላት በጥሩ ከተሰራና ጠራ ያለ ብርድልብስ ጋር ይያያዛል ::

24. Her hair flew out behind her, and the clean air struck her face.

ፀጉሯ ከኋላዋ ይውለበለባል፤ እነዲሁም የጠራ አየር ፊቷን ይመታዋል ::

25. It's quite a job, keeping the windows clean.

ሥራው የመስኮቶችን ጥራት መጠበቅ ነው ::

26. Curled up on his armchair, thin as a wood shaving, he looks far too slight to carry this immense spectacle.

በእጅ ወንበር ላይ ተጠቅልሎ፤ ልክ እንደ እንጨት መፈግፈጊያ ሣሥቶ፤ ግርማሞገሱን ለማሳየት ብዙ የሚቀረው ይመስላል::

27. Above all the accounts, technical and economic, of the lift's operation, are quite incredibly thin.

ከሁሉም በላይ ሁሉም አካውንቶቹ ቴክኒካዊ እና ኢኮኖሚያዊ የቀኝ ሥራዎች በሚገርም ሁኔታ የሳሱ ናቸው ::

28. Armed with a thin red and white linen cloth.

በሣሣ ቀይ እና ነጭ ላይነን ልብስ ታጥቀዋል ::

29. Higher than predicted gravity values occur over the oceans because they are underlain by thin and relatively dense crust.

ከተገመተው በላይ በባህሮች ላይ የመሬት ስበት የሚጨምረው በሣሣ እና በንፅፅር ጥቅጥቅ ባለ ቅርፊት ስለሚሸፈን ነው፡፡

30. These colours are so strong that you have to thin quite a bit to gain softer tones.

እነዚህ ቀለሞች እጅግ ጠንካራ ናቸው፤ የለሰለሰ ቶን ለማግኘት ቲኒሽ መሣሣት አለበት ፡፡

31. ‘You're very good at taking care of people,’

ለሰዎች በመሣሣት በጣም ጥሩ ነህ ፡፡

32. Wilson taking care of me and treating me like a lady — because there was a little something between us.

ዊልሰን ልክ እንደ ሴት እየሣሣልኝ እና እየተንከባከበኝ ነው ምክንያቱ በመሀካላችን ቲኒሽ ነገር ነበር ፡፡

33. It may be decorating a flat for a person, to taking care of the cat of an elderly hospitalised lady.

ሆስፒታል ያለችን የሴት ድመት መሣሣት፤ ለሰው ጎማን እንደሚያስጌጥ ሊሆን ይችላል ፡፡

34. Her husband Barry's taking care of the other two kids — he's a real capable boy.’

ባለቤቷ ባሪ ሌሎቹን ሁለት ልጆች እየሣሣላቸው ነው በእውነቱ አቅም ያለው ልጅ ነው ።

35. She was given the special responsibility of taking care of me, and I owe her my life.

ለእኔ የመሣሣት ልዩ ሀላፊነት ትሰጥቷል፤ እና ሕይወቴን እሰጣታለው ።

36. Until we study the life cycles of animals in fine detail, we cannot know precisely which creatures depend upon what.

የእንስሳትን የሕይወት ዑደት በተብራራ ሁኔታ ሳናጠና፤ አንዱ ፍጥረታት በምን ጥገኛ እንደሁኑ በትክክል ማወቅ አንችልም ፡፡

37. Their Social Class and Educational Opportunity gave a new impetus both to the study of these themes and to action upon them.

ማህበረሰቡ ያላቸው ቦታ እና ያገኙት የትምህርት ዕድል ይህን ዘርፍ እና የሚሰሩትን ድርጊት እንዲያጠኑ ኃይል ሰጥቷቸዋል ።።

38. If they can make us more aware of the Earth and our relationship with it then their study will have been worthwhile.

ስለምድር እና ከምድር ጋር ስላለን ግኙኝነት ብዙ እንድናውቅ ካረጉን ጥናታቸው ጠቃሚ ይሆናል ።።

39. The study of Scripture, he suggested, did nothing to hinder an inquisitive man's delight in the study of nature.

የመፅሀፍ ቅዱስ ጥናት የሰው ልጅ ስለ ተፈጥሮ መመራመር ፍላጎትን እንደማያደናቅፍ አሳሰበ።።

40. It is also particularly easy to study, because sound can be recorded and reproduced by a tape-recorder.

ላማጥናት በጣም ቀላል ነው፤ ምክንያቱም ድምፆች በቴፕ መቅጃ ተቀድተው ሊባዙ ይችላሉ ።።

41. Setting out Labour's case for international co-operation to strengthen world security and combat environmental degradation

የሰራተኞችን ጉዳይ መወሰን ዓለምአቀፍ ትብብር የዓለምን ሰላምን እና ለማጥናት እና የአከባቢ ብክለትን ለመከላከል።።

42. Saving and investment, and to strengthen private and public institutions; and steps to protect the poor during the transition.

ቁጠባ እና ኢንቨስትምንት የግል እና የህዝብ ተቋማትን ለማጥናት እና በሽግግሩ ጊዜ ድሀን ለመከላከል።።

43. Ministry of Higher Education aims to strengthen teacher competence at local level by encouraging teachers.

የከፍተኛ ትምህርት ሚኒስቴር አስተማሪዎቹን በማበረታት የአስተማሪዎቹን አቅም በአከባበው ደረጃ ለማጥናት አቅዷል፡፡

44. All listed companies have an active audit committee will strengthen the auditor's position vis-a-vis client management.

የተዘረዘሩት ድርጅቶች የኡዲተሩን ቦታ ሊያጠና የሚችል ትጉ የኡዲት ኮሚቴ አላቸው ::

45. ‘He will strengthen our squad and give us a lot more options this season.

የኛን ስኳድ ያጠናልናል እንዲሁም በዚህ ዓመት ብዙ አማራጮች ይሰጠናል ::

46. Chances are that it won't make an ideal radio record any more than ‘Candle in the Wind.

ንፋስ ላይ ካለ ሻማ ላይ የተሻለ ጥሩ የሬዲዮ ቀረጻ መስራት አይቻልም ::

47. Millions of journalists have begun a record Christmas break.

በሚሊዮን የሚቆጠሩ ጋዜጠኞች የገናን እረፍት መቅረጽ ጀመሩ ::

48. Since leaving the ranks of a solo career, John Cale has gone on to record a catalogue of solo work that is both voluminous and impressive.

የሶሎ ሙያውን ከተወ በኋላ ጀን ኬል በጠም ብዙ የሚመስጡ የሶሎ ሥራዎችን ሊቀረጽ ነው ::

49. During the hearing Mike Morley had tricked his way into the prison to record the interview with Nilsen.

ክሱ በሚሰማበት ጊዜ ሞርሊ የኒልሰንን ቃለመጠይቅ ለመቅረጽ መሄዱን ካደ ::

50. Ironically, David Guest's victory came on the day that television cameras were allowed for the first time to record proceedings in a Scottish court.

የሚደንቀው የዴቪድ ገስት ድል የመጣው ቴሌቪዥኖች የስኮትላነድን ፍርድ ቤት ሂደት እንዲቀርጹ በተፈቀደላቸው በመጀመሪያው ቀን ነበር ::

51. In spite of the many things it has achieved over the last hundred years — and we have all been shaped by that — it has got itself boxed in by one issue.

ባለፊት መቶ ዓመታት ብዙ ነገር ቢያሳካም፤ እንዲሁም በዛ ብንቀረጽም በአንድ ጉዳይ ብቻ ነው ራሱን የወሰነው ::

52. The rapid growth of film and media studies in colleges and schools has been dominated and shaped by the cultural theories of the left.

በኮሌጆችና በትምህርትቤቶች ፈጣኑ የፊልምና የሚዲያ ጥናት በባህላዊው የቀኝ ጽንሰሐሳብ የተቀረጸና ተፅእኖ ያረፈበት ነው ::

53. Drug laws are shaped by vested economic interest, and the real reason that some drugs remain illegal is to allow the law to intervene in the lives of those the state perceives as threatening.

የመድኃኒት ህጎች የተቀረጹት የኢኮኖሚውን ፍለጎት ባገናዘበ መልኩ ነው፤ እንዲሁም እውነተኛው አንዳነድ መድኃኒቶች ህገወጥ የሆኑበት እውነተኛ ምክንያት መንግስት አስጊ በሚላቸው ሰዎች ኑሮ ውስጥ ጣልቃ ለመግባት እንዲመቸው ነው::

54. The stories were generally shaped by his values and his literary genius allowed every social detail to be authentic and appropriate.

ታሪኮቹ በአጠቃላይ የተቀረጹት በእሴቶቹና በያነዳንዱ የማህበረሰቡ ጥቃቅን ነገሮች ያለው የሥነፅሁፍ ችሎታ አስተማማኝና ትክክለኛ መሆኑ ነው ::

55. These pupils are shaped by many other factors than their schooling

ተማሪዎቹ ከሚማሩት ትምህርት በተጨማሪ በሌሎች ብዙ ተፅዕኖዎች ተቀርጸዋል ::

Appendix C. Lists of Affixes removed from the token (Atelach, 2002).

Prefixes
ለ
ስለ
በ
በየ
እንደ
እንደየ
እየ
ከ
ወደ
ወደየ
የ

Suffixes	
ም	ኣቻችን
ምና	ኣቻችንም
ና	ው
ንም	ዎቻቸው
ንና	ዎቻቸውን
እና	ቻቻቸውንም
ኡ	ዎች
ኦች	ዎቻችን
ኦችም	ዎችን
ኦችን	

Declaration

I declare that the thesis is my original work and has not been presented for a degree in any other university.

_____

Date

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

_____

Advisor