JIMMA UNIVERSITY

COLLEGE OF NATURAL SCIENCE

DEPARTMENT OF INFORMATION SCIENCE

TOWARDS THE SENSE DISAMBIGUATION OF AFAAN OROMOO WORDS USING HYBRID APPROACH (UNSUPERVISED MACHINE LEARNING AND RULE BASED)

By: WORKINEH TESEMA GUDISA

A Thesis Submitted to Jimma University Institute of Technology(JiT) in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology

Advisor: Debela Tesfaye (Ass.Prof)

Co-Advisor: Teferi Kebebew (MSc)

October 3, 2015

JIMMA UNIVERSITY

COLLEGE OF NATURAL SCIENCE

DEPARTMENT OF INFORMATION SCIENCE

TOWARDS THE SENSE DISAMBIGUATION OF AFAAN OROMOO WORDS USING HYBRID APPROACH (UNSUPERVISED MACHINE LEARNING AND RULE BASED)

Jimma, Ethiopia

October 3, 2015

Advisor Name: Debela Tesfaye(Assistant Professor) and Teferi Kebebew(Lecturer)
Rank: Ass. Pro
School/Department: Computing
Institute / University: Jimma University
City: Jimma

Declaration and Certification

This is to certify that the thesis entitled **"Towards the Sense Disambiguation of Afaan Oromoo Words Using Hybrid Approach (Unsupervised Machine Learning and Rule Based)"** submitted by **Workineh Tesema**, the M.Sc (IT) student of School of Computing, Jimma Institute of Technology, Jimma University, for the award of Master of Science in Information Technology, is a record of original work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the University and in my opinion the thesis has reached the standard needed for submission. The results embodied in the thesis have not been submitted to any other University or Institute for the award of any degree or diploma. I hereby declare that the student has incorporated the comments given during the mock defense to improve the work substantially.

Debela Tesfaye (Assistance Professor)	_____	_____
Advisor	Signature	Date
Teferi Kebebew (MSc)	_____	_____
Co-Advisor	Signature	Date
_____	_____	_____
External Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date
_____	_____	_____
Institute coordinator for Research, P, GS&C	Signature	Date

Date: 12/01/2015
Place: Jimma University

_____

_____

Dedication

This thesis work is dedicated to my father Tesema Gudisa, my mother Askala Jifar, all my brothers and sisters who were able to reap the fruit of their own.

Workineh Tesema Gudisa

October, 2015

Declaration

This study is my original work and has not been submitted as a partial requirement for a degree in any University and that all sources of material used for the study have been duly acknowledged.

Workineh Tesema Gudisa

October, 2015

Acknowledgment

First of all I would like to thank Jesus Christ who helped me to succeed in all my life long learn. Next, I would like to thank my advisor Debela Tesfaye for his valuable assistance in providing his genuine, professional advice and encouragement goes even beyond the accomplishment of this study. Some noteworthy aspects of this study are entirely because of him; all faults are in spite of him. He initiated me to do by giving precious comments on necessary points. My thanks go to him again, since it is difficult to mention his contribution to my achievements in words, it is better to say my heart has recorded it forever. And also I would like to acknowledge my co-advisor Teferi Kebebew for all his guidance at every step of the thesis work, for patiently listening to me even during uncountable hours, for advising me to do my study, for imparting so much valuable knowledge and for all his encouragement and words of kindness. Next, Words cannot express my deepest thanks to my Father Mr. Tesema Gudisa and mother Askala Jifar and my brothers Hirpa Tesema, Muleta Tesema, Naol Tesema and Ibsa Tesema for their moral and financial support I received from them throughout my life of learning. My honest thanks equally go to my sisters Sorse Tesema, Lamme Tesema and Shonkore Tesema for their help in providing me concerning background information. I would like to thanks Jimma University, Department of Afan Oromo Linguistic professional Lecturers Tefera, Waktola, Chimdi, Lenin and all my colleagues who have helped me on my progress; Chala Diriba and Worku Jimma.

Lastly but not least, I would like to thanks to ORTO and Oromia Folklore Bureau for their permission to data collection and professional contribution and Jimma University for financial support.

Table Content

Table of Contents **Page**

List of Tables

List of Figures

List of Appendix

List of Acronyms

AI: Artificial Intelligence

CSV: Comma-Separated Values

EM: Expect Maximization

HAC: Hierarchical Agglomerative Clustering

IR: Information Retrieval

KBD: Knowledge-Based Disambiguation

ML: Machine Learning

MT: Machine Translation

MAX: Maximum

MIN: Minimum

NLP: Natural Language Processing

ORTO: Oromia Radio and Television Organization (ORTO)

POS: Part-of-Speech

QA: Question Answering

VSM: Vector Space Model

WSD: Word Sense Disambiguation

Abstract

Word Sense Disambiguation is a technique in the field of Natural Language Processing where the main task is to find the appropriate sense in which ambiguous word occurs in a particular context. It is a fundamental problem for many natural language technology applications(Machine Translation, Text Summarization, Question and Answering, Information extraction and text mining and Information Retrieval). A word may have multiple senses and the problem is to find out which particular sense is appropriate in a given context. Ambiguity is a cause of poor performance in searching and retrieval system. The objective of this work is to develop hybrid word sense disambiguation which finds the sense of words based on surrounding contexts. Hence, this study presents a Word Sense Disambiguation strategy which combines an unsupervised approach that exploits sense in a corpus and manually crafted rule. The idea behind the approach is to overcome the problem a bottleneck for the machine learning approaches, while hybrid method can improve the accuracy and suitable when there is scarcity of training data. This makes our approach suitable for disambiguation when there is lack of resource and sense definitions. In this study, the context of a given word is captured using term co-occurrences within a defined window size of words. The optimal window sizes for extracting semantic contexts is window ±1 and ±2 words to the right and left of the ambiguous word. The similar contexts of a given senses of ambiguous word are clustered using hierarchical and partitional clustering. Each cluster representing a unique sense. Some ambiguous words have two senses to the five senses. The result argued that WSD yields an accuracy of 70% in Unsupervised Machine learning and 81.1% in Hybrid Approach. The machine learning were a useful information source for disambiguation but that it not as robust as a linguistic(rule based) [89]. Based on this, the integration of deep linguistic knowledge with machine learning improves disambiguation accuracy. Therefore, for Afan Oromo semantic has come to the conclusion that the sense of words are closely connected to the statistics of word usage. The achieved result was encouraging, despite it is less resource requirement. Yet; further experiments using different approaches that extend this work are needed for a better performance.

Key words: *Word Sense Disambiguation, Afan Oromo, Ambiguous Word, Disambiguation, Rule Based, Hybrid.*

Chapter One

Introduction

1.1. Background

In today's world, where World Wide Web technology is keeping on growing very fast, many users go to the web to search for information, for entertainment, to read documents and electronic books. Sometimes it is observed that the result of a search is not appropriate. The reason behind is, there is an ambiguous word in the query. According to [1] in information retrieval, the stored information items and the incoming search requests are normally represented by sets of content identifiers as keywords, index terms, or simply terms. Information Retrieval (IR) can potentially benefit from the correct meanings of words provided by Word Sense Disambiguation (WSD). Ambiguity is a cause of poor performance in IR systems. In the application to IR, WSD can bring different benefits. The queries may contain ambiguous words (terms), which have multiple meanings. The ambiguities of these query words negatively affect the number of retrieved documents. Identifying the correct meaning of the ambiguous words in both queries and documents can help improve retrieval of the documents. The other one is, query words may have tightly related meanings with other words not in the query. Making use of these relations between words can improve the number of retrieved relevant documents. IR is one of the Natural Language Processing (NLP) applications that paybacks from WSD which most of the words used to execute queries in IR systems have more than one meaning [2].

In natural language processing, WSD is the manner of determining which sense (meaning) of an ambiguous word is activated by the use of the ambiguous word in a particular context. It is a natural language problem, a given ambiguous word and its possible senses as defined by an occurrence of the ambiguous word in context into one or more of its sense [3]. Natural languages have ambiguous words which need to be disambiguated and thus the appropriate sense of an ambiguous word in a given context can be identified. An ambiguous word can take several senses depending on the context in which it appears. The same form and pronunciation can take different meanings in different contexts [4].

In this paper, ambiguous word and target word are used interchangeably

The ambiguity of human language is a greatly debated problem in many research areas. Human language is ambiguous, because the ambiguous words can be interpreted in multiple ways depending on the context in which it occurs [2].

It is considered as an open problem, that is a task whose solution is as hard as the most difficult problems in natural language [5]. Natural languages are ambiguous, hence an ambiguous word can take several senses depending on the context in which it appears. Consider the following simple sentences:

- *Afaan koo faaya kooti*.
- *Siifaan afaan bal'atti.*

In the first context, the meaning of the word *afaan* means 'the method of human communication or humans language'. In the second context, it means 'mouth or the opening and cavity in the lower part of the face, Surrounded by the lips'. In this case, the word form (*afaan*) is the same in both sentences. The word (*afaan*) sharing the same spelling and pronunciation, but have different senses. This multiple sense can be totally unrelated to each other. The word (*afaan*) has more than one senses, so it is too difficult to say the sense of (*afaan*) is language or mouth at a given moment. This is what makes word sense disambiguation difficult because of the given ambiguous word has more senses. However, the sense of any word was based on the contexts around the target word in sentences, paragraph and text. It is the context which decides the sense of the ambiguous words according to it appearing in the sentences [6].

The assignment of sense to word is accomplished by using major sources of information contained within the context in which the word appears [2]. All disambiguation work uses the context of the instance of the word to be disambiguated with either ruled based or information about the contexts of previously disambiguated instances of the ambiguous word derived from corpora (corpus based WSD). People decide the meaning of an ambiguous word based on the characteristic senses of a discussion or situation using their own evidences. But machines have no ability to decide such an ambiguous situation unless some procedures have been planted into the machines' memory.

The task of building computer programs that understand natural language is not straightforward [7] because, there are no decisive ways of identifying where one sense of an ambiguous word ends and the next begins. In computational linguistics, WSD is an open problem of natural language

processing which governs the process of identifying which sense of an ambiguous word (i.e. meaning) is used in a sentence, when the ambiguous word has multiple meanings [8]. This problem impacts on other computer-related processes, such as improving relevance of search and MT (Machine Translation). WSD as a heart of NLP because it is one of the most fundamental tasks in many applications in NLP directly or indirectly rely on WSD [65].

The purpose of this study is to combine both the rule based and machine learning approaches into a hybrid approach and to investigate word sense disambiguation for Afan Oromo words. In this case the approach is that the combination of both types of strategies, namely unsupervised machine learning and rule based can improve WSD effectiveness, because it overcomes the limitation of algorithms by rule based (linguistic knowledge). The motivation behind WSD is Afan Oromo to allow the users to make ample use of the available technologies because ambiguities present in any language provide great difficulty in the use of information technology as words in human language that occur in a particular context can be interpreted in more than one way depending on the context.

1.2. Statement Problem

Afan Oromo is an official language of Oromia Regional State (which is the largest regional State among the current Federal States in Ethiopia) [9]. It is one of the major languages that are widely used as a working language of the Regional State of Oromia, that is used in Ethiopia and Africa [10]. Afan Oromo has a large number of ethnic groups when compared with the rest of Ethiopian languages [11]. In Afan Oromo, like other natural languages, there exists same form of words, that has more than one different meanings. These words are homonyms which has multiple distinct meanings. For words having multiple senses, the challenge is to find out which particular sense is appropriate in a given context. In natural language, WSD is the problem of determining which sense (meaning) of a word is active in a particular context [12]. However, the task proved to be difficult for computer and believed that it is an open problem [14] and the problem is worse for under resourced languages like Afan Oromo.

Identifying the correct senses of the ambiguous words is easy for human being basically, sometimes it is difficult. However it is too tough for the machine to identify the correct sense of these words. For instance, applications like machine translation (English to Afan Oromo, Amharic to Afan Oromo) the role of the word sense disambiguation is great because it helps to find the meaning of

words. In such cases, it is required to have words that have translation for different senses and that are potentially ambiguous within a given domain [13].

In order to have a clear understanding of ambiguous words in the language, WSD for Afan Oromo language is also needed to be developed. Because, now a days as the development of technology is increasing rapidly, like any other language Afan Oromo has also started to use the technology for different purposes. Many different Afan Oromo documents are available in different sites which is an opportunity, yet creating a problem for information retrieval.

As already mentioned, human readers have the extraordinary capability to infer the meaning of a word from the context, even if they have never heard of the word before, and there are no visual cues present to support its interpretation and they can still make reasonable assumptions about the meaning. The surrounding contexts allow guessing the meaning of ambiguous words. However, this situation becomes more complicated if we want to use a computer to infer the meaning of words. Unlike humans, computer does not have any a priori knowledge of ambiguous words in natural languages like Afan Oromo. The words sharing the same spelling and pronunciation, but have different senses, which has multiple senses can be totally unrelated to each other. This is a fundamental problem for many established human language technology applications.

The intention of this study is to device an algorithm by adopting and integrating the already existing once for word sense disambiguation for Afan Oromo words. Currently, few researches in word sense disambiguation have been commenced for Ethiopian languages, particularly for Amharic in different domains by adopting different techniques. Our work presents a contribution towards developing natural language processing applications for Ethiopian languages exhibiting similar patterns with Afan Oromo. Specifically, it increases the scope of the word sense disambiguation research by investigating its applicability for Afan Oromo language. The other techniques used in this study are vector space model for similarity measure and clustering. The contexts are captured via extracting co-occurrences in a defined window of words from a corpus.

1.3. Objective of the Study

1.3.1. General Objective:

The general objective of this study is to develop the hybrid word sense disambiguation prototype to Afan Oromo ambiguous words.

1.3.2. Specific Objective:

In line with the general objective, the research is aimed at addressing the following specific objectives:

- To review related works so as to have a conceptual understanding of the state-of-the-art in Afan Oromo word sense disambiguation;
- To acquire the required corpus from different sources to use it for training and testing and to contribute as the language resource;
- To build a model using Unsupervised Machine Learning and Hybrid approach;
- To develop a prototype that disambiguate multiple sense words in Afan Oromo;
- To construct the rule that support the machine learning disambiguation model;
- To forward the direction of the WSD research;

1.4. Scope and Limitation of the Study

The scope of the study is limited to investigating word sense disambiguation for Afan Oromo words particularly at word level, which study what is the actual meaning of the ambiguous word in a given context. The technique used was a hybrid approach which combine machine learning (which is unsupervised) with detail background knowledge of the language (rule based). Due to the absence of standard train and test corpus, we had to prepare corpus for the experimentation which is unannotated machine readable free text. However, the amount of corpus prepared for this study is relatively small and requires further development. In this case, we have studied ambiguous words at word level in sentences based on surrounding contexts and modifiers. The main idea of this work is to explore word sense disambiguation using corpus and information from rule in Afan Oromo. The study investigated the meaning of words by looking in an unstructured corpus prepared from

different sources and official websites. The lack of annotated corpus and senses definition are among the constraints of this study.

1.5. Significance of the Study

WSD is considered as an input for many Natural Language Processing (NLP) applications. The researchers in the area of NLP of Afan Oromo Language could use the output of this thesis. Particularly, researchers interested in area of MT (Machine Translation), Question Answering (QA), Text Summarization are among the top beneficiaries. Ambiguity resolution has been pursued as a way to improve retrieval systems, and generally get better information access. Moreover, for the students who are always practicing to learn word meaning, the system is very helpful. The output of this study fixes the problem of word meaning and complexity in the society. Similarly, this work has an input for translation of English and Amharic to Afan Oromo. Interested users who want to browse web page by Afan Oromo could also use the system. The finding of this study is expected to support Afan Oromo researchers who deal with NLP applications. It also contributes to future researches and development in the area of NLP specifically in machine learning, Information retrieval as those areas require word sense disambiguation as complements.

1.6. Organization of the Thesis Report

The thesis is organized into five chapters comprising Introduction, Literature review, overview of Afan Oromo, Methodology, Experimentation and Evaluation, and Conclusion & Recommendations. The first chapter gives the general introduction of the study. The second chapter presents review made on different literatures regarding Word Sense Disambiguation together with its approaches and different machine learning techniques. The third chapter discusses the overview of the grammatical categories in Afan Oromo. The fourth chapter discusses the methodology and presents the process of data preparation for the system to be developed for Afan Oromo. It present the algorithms and rule preparation based on the analysis of rule categories as reviewed. The fifth chapter discusses the experimentation and discusses of the findings. Chapter sixth presents the conclusion and the recommendations as well as some directions to future works.

Chapter Two

2. Literature Review

2.1. Introduction

In this chapter, related literature review in the Word Sense Disambiguation (WSD) is presented. This part of study enlightens briefly on some of the work done by those researchers.

2.2. Basic Concepts of Word Sense Disambiguation

In natural languages, [15] each word may have more than one meaning that is why a single word sometimes can have many senses. Disambiguation means to remove all ambiguity. WSD is a technique used in finding the meaning of a word in a sentence. A word can have multiple meanings and the exact meaning of the word is decided based upon context by humans. Computers also can follow similar technique i.e. decide meaning of an ambiguous word in a sentence using context information. Likewise, it is the process of identification to decide the appropriate meaning of an ambiguous word in a particular context. People decide the meaning of a word based on the characteristic senses of a discussion or situation using their own merits. Machines have no ability to decide such an ambiguous situation unless some practices have been planted into the machines' memory [16]. The word sense is extracted by human by connecting it to that specific context. But for corpus this function is performed by natural language running by the qualities of the term. It disambiguates ambiguity of an individual word that can be used (in different contexts) to express multiple meanings. The context is the only real way to recognize the significance of the homonym term. Generally, context can be used in the bag of words and relational information. Here the bag of words is the context that considered as words in some window surrounding the ambiguous word in terms of distance. The surrounding word of the ambiguous word decides correct choice of word. Assume, the ambiguity word *afaan* has surrounded with the following contexts. So, these contexts give us the clue of information what the meaning of this ambiguity word is. It seems that this ambiguity word identify its senses with the help of its contexts as shown the in Figure below:

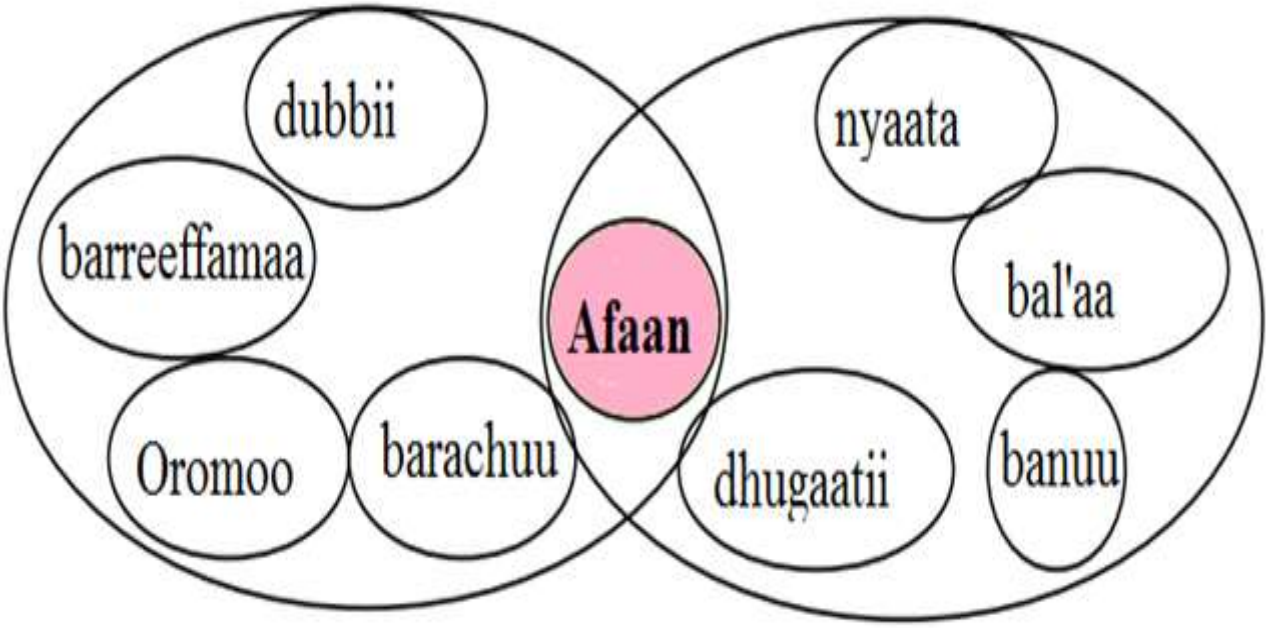


Figure 2.1. Surrounding Contexts of the term *Afaan*

2.3. Approaches to Word Sense Disambiguation

Regardless of the approach, WSD systems extract contextual information of an ambiguous word (in the corpus) and compare against the sense differentiation information stored in that word [13]. Many approaches have been used through the evolution of WSD research.

Based on the level of supervision the machine learning approaches are categorized as Supervised machine learning, Unsupervised machine learning and Bootstrapping machine learning approaches.

2.3.1. 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) [3]. 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 of 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 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, only classification algorithms used in this method.

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, time consuming, limiting the number of sense tagged words to be used and very few sense-tagged data are available.

2.3.2. 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 [3]. Unsupervised methods have the potential to overcome the knowledge acquisition bottleneck [17], 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 has 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. 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 [18]. 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 [2].

Nevertheless, there is another issue connected with the problem of the definition of a meaning, i.e., an issue of the creation of other resources used for automatic system performing WSD. This is

especially evident in the creation of corpora that is manually annotated (tagged) with the 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 inter annotator agreement (IA) and high cost of the 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 the estimation of an upper bound on performance on automatic WSD but there is also another measure [19].

2.3.3. Bootstrapping Machine Learning 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 [2] 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.

In addition to the above approaches, another criterion for categorization of approaches are use of data, knowledge (dictionary) [5].

2.3.4. Corpus - Based Approaches

A major challenging face WSD research is the ability to acquire a large number of words with their different contexts. Corpus-based approaches came up with an alternate solution to the challenge by obtaining information necessary for WSD directly from textual data which is 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.

2.3.5. Knowledge - Based Approaches

Knowledge-based WSD is based on lexical resources like dictionaries, thesauri, and corpora where Machine-readable dictionaries (MRDs) are the primary source of acquisition of data. There are various knowledge based approaches such as WSD uses Selectional Preferences, Lesk's algorithm, Walker's algorithm, WSD uses conceptual density and WSD uses Random Walk Algorithm. Rely on external knowledge resources (example, WordNet, Thesaurus), may use grammar rules for disambiguation and may use hand coded rules for disambiguation. Under this approach disambiguation is carried out using information from an explicit lexicon or knowledge base. Since corpus based approaches require a considerable amount of work to create a classifier for each word in a language. Knowledge-based approaches use an explicit lexon like, MRD, thesauri, computational lexicons such as WordNet or (hand-crafted) knowledge bases as information source to resolve lexical ambiguities for many words.

The created knowledge bases [20] which associate each sense in a dictionary with a signature composed of the list of words appearing in the definition of that sense. Disambiguation will be 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. 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 [18]. 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. 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 large electronic database containing useful lexical relations in linguistic psycholinguistic and computational [3]. Lexicon like WordNet is used for sense evaluation and for similarity measure in WSD. For example [21] created a knowledge base from WordNet's hierarchy

and apply a semantic similarity function to accomplish disambiguation, also for the purposes of information retrieval.

Finally, we can categorize WSD approaches as token based and type-based. Token based approaches associate a specific meaning with each occurrence of a word depending on the context in which it appears. In contrast, type-based disambiguation is based on the assumption that a word is consensually referred with the same sense within a single text. Consequently, these methods tend to infer a sense (called the predominant sense) for a word from the analysis of the entire text and possibly assign it to each occurrence within the text [5].

2.3.6. Hybrid Machine Learning Approach

The combinations of machine learning and rule based approaches have been used to make hybrid methods (in this study). A major weakness with unsupervised method is the accuracy and lack of ability to place a label on each discriminated word group. Combining an unsupervised method with a rule based method could overcome this weakness.

Since they obtain disambiguation information from both corpora and explicit linguistic 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. [22] used Bootstrapping approaches where initial data come 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.

Although above three approaches are very popular for WSD, they have weakness respectively. Semantic relations only cannot disambiguate any homonym word in any context. However, WordNet does not define the noun-verb relations. The supervised machine learning approach needs a great deal of training data that are expensive and sometimes hard to obtain. Therefore, a hybrid approach that uses both the machine and rule based methods to solve WSD problems. It is an approach that provides better results and make the results satisfy users' preferences.

2.4. Context Window Size

Context is the only means to identify the sense of an ambiguous word in sense disambiguation. All algorithms make use of the contexts to provide information for sense disambiguation. The context window size is defines the size of the window of context. A window size of *N* means that there will be a total of *N* words in the context window. If *N* is a number (positive), then there will be *N* word on the left and right side of the target word, where N is for example: 1, 2, 3, 4,5, etc. For example, if the window size is 2, then there will be 2 words on the left and right side of the target word. In order to disambiguate a given word, a small and wider context should be considered in the performance of the system to rise overall. However, a wider context implies more data and thus further features. The different methods and algorithms can benefit from the choice of the context size in a distinct way, which means that the optimal size of the context can depend on the used method and that the selection of the size of the context leads to a variation of the WSD output. [22] also claimed that the ambiguity word occurring in the data can be captured by a different size of the context.

2.5. Vector Space Model

Vector space model is the most important model to obtain a vector that represents each word in the sentences. This vector is usually chosen to capture the contexts in which words can appear in. The representation for a word is a point in a dimensional space. The dimensions stand for context items (for example, co-occurring words), and the coordinates depend on the co-occurrence counts. Each dimension corresponds to a separate word. If ambiguous word occurs with contexts word within the collected contexts, it has similarity on the vector. Similarly, all words in the contexts have its weight (the occurrence of word in the corpus); zero means the word has dissimilarity with the contexts. Several different ways of computing these values as weights (term) have been used. One of the best known schemes is weighting (frequency occurrence of the word with the contexts) [23].

The dimensionality of the vector is the number of occurrence of word in the contexts. As all vectors under consideration, a cosine value of zero means that the ambiguous word and contexts are dissimilar and have no match (i.e. the ambiguous word does not exist with the contexts). The context is formally a text that surrounds a language unit (e.g. a word) and helps to determine its interpretation. It represent the occurrences of target words as word vectors. From these vectors, vectors are formed and found that is a function of cosine similarity between the contexts [23].

The computation of meaning similarity as operationalized by vector-based models has found widespread use in many tasks within NLP. The popularity of vector-based models lies in unsupervised nature and ease of computation. The model represent the meaning of each word as a point in a dimensional space, where each component corresponds to some co-occurring contextual [24]. The advantage of taking such a geometric approach is that the similarity of word meanings can be easily quantified frequency by measuring the cosine similarity of the angle between them [25].

2.6. Clustering Algorithms

Clustering algorithms are generally categorized as hierarchical and partitional. The next section describes some common clustering algorithms. Here are general properties that characterize clustering algorithms [26].

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.

2.6.1. Hierarchical Agglomerative Algorithms

Hierarchical algorithms produce a nested partitioning of the data elements by merging clusters. Agglomerative algorithms iteratively merge clusters until all-encompassing cluster is formed [27], 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.

The different versions of agglomerative clustering differ in how they compute cluster similarity. The most common versions of the agglomerative clustering algorithm are single-link, complete-link and average-link clustering [22].

A. **Single Link Clustering**: The single link algorithm is a MIN version of the hierarchical agglomerative clustering method which is a bottom-up strategy, compare each point with each point.

Each context is placed in a separate cluster, and at each step merge the closest pair of clusters, until certain termination conditions are satisfied. For the single link, the distance of two clusters is defined as the minimum of the distance between any two points in the clusters. In single-link clustering the similarity between two clusters is the similarity between their most similar members (e.g. using the Euclidean distance) [28]. It is capable of discovering clusters of varying shapes like the clusters of Figure 2.2 a. However, single-link is not practical because it suffers from the chaining effect [27]. For example, in Figure 2.2 a, single-link clustering generates an elongated cluster because of a bridge of elements connecting two clusters.

B. **Complete Link Clustering**: The complete linkage algorithm is the MAX version of the hierarchical agglomerative clustering method which is a bottom-up strategy: compare each point with each point. Each context is placed in a separate cluster, and at each step merge the farthest pair of clusters, until certain termination conditions are satisfied.

In complete-link clustering, the similarity between two clusters is the similarity between their maximum similar members (e.g. using the Euclidean distance) [29]. Although complete-link clustering is not capable of discovering clusters like the two in Figure 2.2 b, it does not suffer from the chaining effect. Rather than producing straggly elongated clusters like single-link, complete-link generates compact clusters. Complete-link generates better clustering's than single-link in many applications[30]. Figure 2.2 a illustrates the cluster similarity of complete-link.

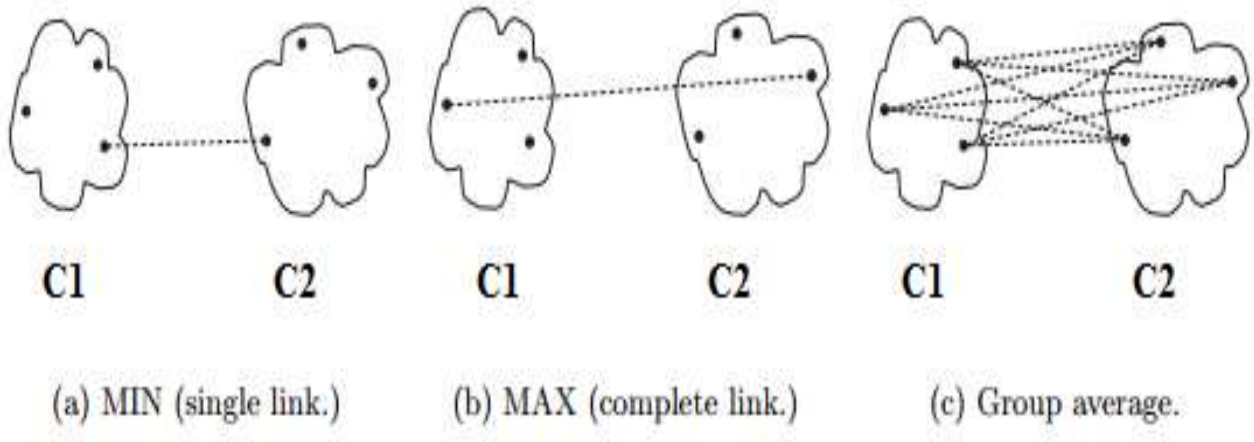


Figure 2.2. a) Single Link, b) Complete Link and c) Average Link

C. **Average Link Clustering:** Average-link clustering produces similar clusters to complete link clustering except that it is less susceptible to outliers [31]. It computes the similarity between two clusters, as the average similarity between all pairs of contexts across clusters (e.g. using the Euclidean distance). Figure 2.2 c shows merging decisions average linkage algorithms.

Generally, to use a hierarchical clustering procedure, need to express these distances in Weka tool. As an example, easily compute the Euclidean distance between cluster B and cluster C (generally referred to as d (B, C)) with regard to the two variables x and y by using the following formula, Weka by default has Euclidean distance.

$$d_{Euclidean}(B,C) = \sqrt{(x_B - x_C)^2 + (y_B - y_C)^2}$$

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 combined 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 [31].

A. K-means

This algorithm has the objective of classifying a set of *n* contexts into *k* clusters, based on the closeness to the cluster centers. The closeness to cluster centers is measured by the use of a Euclidean distance algorithm. K-means is an iterative clustering algorithm in which items are moved among sets of clusters until the desired set is reached. A high degree of similarity among senses in clusters is obtained, while a high degree of dissimilarity among senses in different clusters achieved simultaneously [31].

Although the k-means algorithm often produces good results, it is not time-efficient and does not scale well. By saving distance information from one iteration to the next, the actual number of distance calculations that must be made can be reduced. Some k-means variations examine ways to improve the chances of finding the global optimum. This often involves careful selection of initial clusters and means. Another variation is to allow clusters to be split and merged. The variance within a cluster is examined and if it is too large, a cluster is split. Similarly, if the distance between two cluster centroids is less than a predefined threshold, they will be combined.

k-means clustering [32] is a method of cluster analysis which aims to partition n observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. K-means [33] is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume *k* clusters) fixed a priori. The main idea is to define *k* centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid.

When no point is pending, the first step is completed and an early group age is done. At this point we need to recalculate k new centroids as bar centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

The most commonly used family of partitional algorithms is based on the K-means algorithm [31]. K-means clustering is often used on large data sets since its complexity is linear in *k*, 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 using the concept of a centroid where a centroid represents the center of a cluster. A centroid is usually not an element of 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 centroid. It is often difficult to define a centroid for categorical features.

K-means iteratively assigns each element to one of *k* clusters according to the centroid closest to it and recomputed the centroid of each cluster as the average of the cluster's elements. The following steps outline the algorithm for generating a set of *k* clusters:

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.*

B. Expectation Maximization

Expectation Maximization (EM) algorithm [34] is also an important algorithm of data mining. We used this algorithm when we are satisfied the result of k-means methods. An expectation maximization (EM) algorithm is an iterative method for finding maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM [35] iteration alternates between performing an expectation which computes the expectation of the log-likelihood evaluated using the current estimate for the parameters, and maximization which

computes parameters maximizing the expected log-likelihood found. These parameter-estimates are then used to determine the distribution of the latent variables.
Expectation maximization 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 [30].

Expectation maximization (EM) is a well-known algorithm used for clustering in the context of mixture models. 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 [30].

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 to 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 mixture model assumes all attributes to be independent random variables [30].
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.

2.7. Divisive Clustering

Although it is not as common as agglomerative clustering. 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 ($n^2 \log n$).

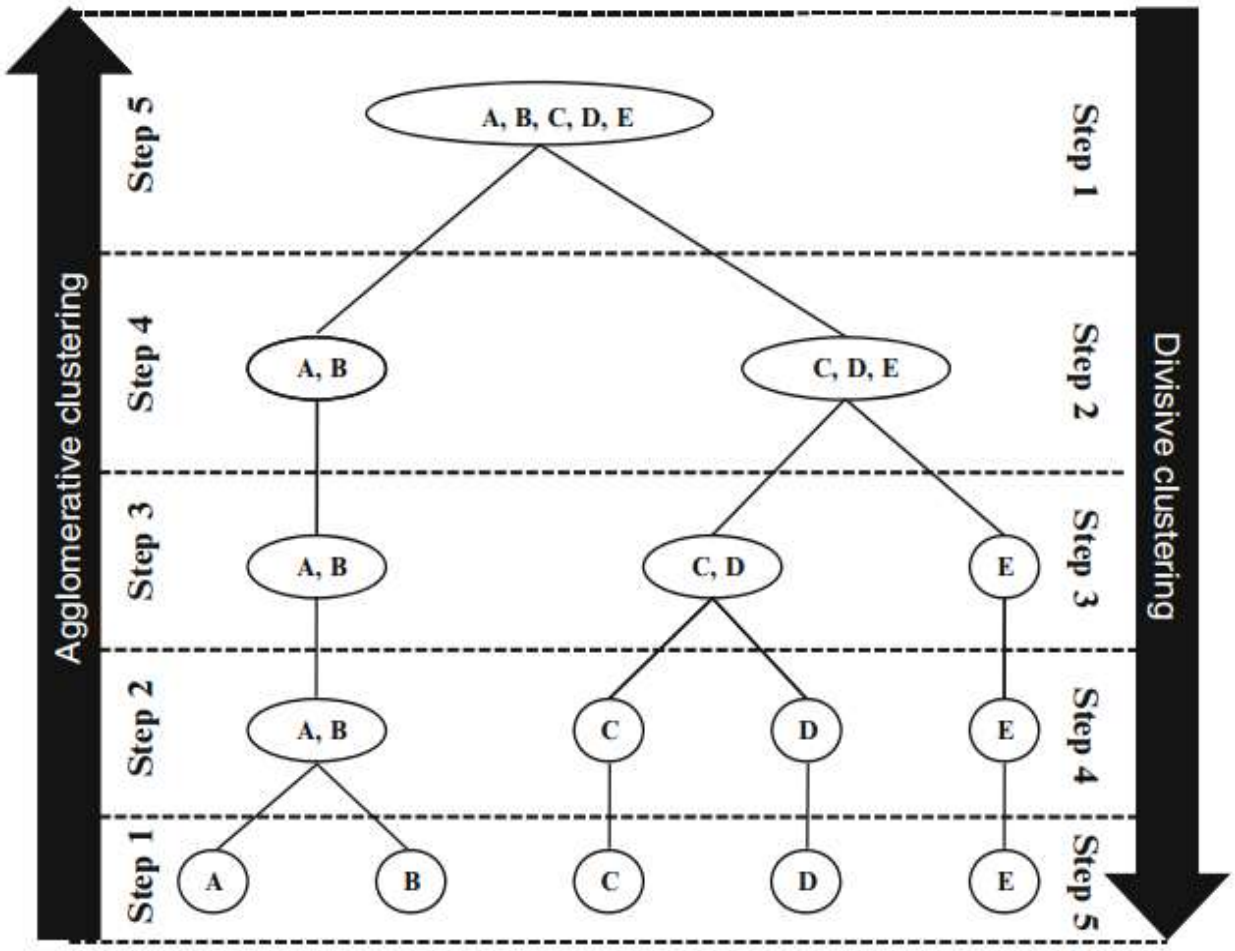


Figure 2.3. Hierarchical Clustering and Divisive Clustering [28]

2.8. Rule Based Approach

Rule-based disambiguation is the popular approach that uses hand-written rule for disambiguating. Hand-written rule are used to identify the correct meaning when a word has more than one possible meaning. Disambiguation is done by analyzing the linguistic features of the word, based on such factors as taking its preceding word, its following word and other aspects. This information is coded in the form of rules. The fact that a rule-based disambiguate that automatically learns its rule can perform so well should offer encouragement for researcher to further explore rule-based disambiguating, searching for a better and more expressive set of rule.

The advantages of this approach are that it is easy to incorporate domain knowledge into the linguistic knowledge which provides highly accurate results. Furthermore, the linguistic knowledge acquired for one natural language processing system may be reused to build knowledge required for a similar task in another system [36].

However, rule based is the most method used in NLP of disambiguation. When there is lack of available resources and their limitations, the rule based approach was used which rely on hand-constructed linguistic rule in systems and resources. The use of rule based transformations are based on a core of solid linguistic knowledge [36].

Obviously, the manual creation of rule is an expensive and time consuming effort, which must be repeated every time the disambiguation scenario changes. Knowledge of the linguistic used for WSD are either lexical knowledge released to the public or world knowledge learned from a training corpus [37]. The detail description of WSD approaches are in the following table 2.1.

Algorithm Category	**Strengths**	**Weaknesses**
Rule Based	Accuracy	It takes time, needs expert and also it is very expensive
Hybrid Approach	Accuracy and it mix from others	-
Unsupervised Approach	No pre-training necessary Works on multiple languages with no modification to the algorithm	Merely discriminates between word senses; not disambiguate word senses
Supervised Approach	It needs pre-training (labeled data)	Dependent on pre-annotated corpora for training data
Knowledge Based	Accuracy	Rely on precompiled lexical knowledge resources

Table 2.1.Summary of Word Sense Disambiguation Approaches

2.9. History of Word Sense Disambiguation (WSD)

WSD is one of the open and oldest problems in computational linguistics. It was in the 1940s that WSD was first formulated as a separate task [8]. 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. [8] 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. Very soon it became clear that WSD was a very difficult problem, also given the limited means available for computation. Indeed, its acknowledged hardness [38] 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 [39], 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[20].

The research by [29] 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. [40] 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 [39] and several others which was a foresight to the so-called statistical revolution" in the 1990s. [41] 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, thereby introducing scientific rigor and uniformity. SensEval eventually became the primary forum for all WSD evaluations. In SensEval-3, a conclusion was reached that WSD in itself has reached a performance level, and no significant rise in the results obtained already is possible. It has been since then, that people started thinking about new directions in which WSD research can go. In particular, in recent years there has been considerable growth in the areas of parallel bilingual corpora, and unsupervised corpus-based WSD. This study employed hybrid WSD and attempts to draw upon the idea that hybrid WSD is the way to go in future.

2.10. Related Works

The work from various books, papers, articles, journals and web pages has been referred to this of the study. Some of them are reviewed at the following:

According to [42] unsupervised machine learning approach for Amharic using five selected algorithms were used; these are Simple k-means, EM (Expect Maximization) and agglomerative single, average and complete link clustering algorithms. The tested unsupervised machine learning method that deals with clustering of contexts for a given word that express the same meaning. The above mentioned work concluded that simple k means and EM clustering algorithms achieved higher accuracy on the task of WSD for selected ambiguous word, provided with balanced sense distribution in the corpus.

According to [43] the meaning of words are extracted based on live contexts using supervised and unsupervised approaches. Unsupervised approaches use online dictionary for learning, and supervised approaches use manual learning sets. Hand tagged data are populated which might not be

effective and sufficient for learning procedure. This limitation of information is the main flaw of the supervised approach. They developed approach focuses to overcome the limitation using a learning set which is enriched in a dynamic way of maintaining new data. The trivial filtering method is utilized to achieve appropriate training data. They introduce a mixed methodology having "Modified Lesk" approach and "Bag-of-Words" having enriched bags using learning methods. The approach establishes the superiority over individual "Modified Lesk" and "Bag-of-Words" approaches based on experimentation.

The research by [44] developed a prototype for Amharic language which is semi-supervised method of WSD. Their finding showed that Semi-supervised learning using a bootstrapping algorithm performs better. It is more adaptive on WSD for Amharic language. Specifically, Adtree, Adaboost and bagging are potential algorithms to be applied to Amharic WSD systems using semi-supervised learning methods. The authors concluded that a window size of 3-3 can be a standard window size of Amharic WSD systems development.

According to [45] developed a supervised learning approach for Amharic WSD. However, supervised machine learning approach of WSD performs better by human intervention, but it has knowledge acquisition bottleneck i.e. it requires manually labelled sense examples which take much time, very laborious and therefore very expensive to create when the corpus size increases. Development of these resources requires huge amount of human efforts and typically takes years for a building. As these resources are not available for Afan Oromo, supervised techniques cannot be immediately applied. Few attempts have been made on unsupervised WSD like [22], which seeks minimum human involvement, in the form of providing a few seed words that occur with each sense of the ambiguous word for bootstrapping the algorithm. The algorithm then classifies each occurrence of the ambiguous word in a corpus into several clusters so that all the occurrences are in the same sense within a cluster. Additional co-occurrences are collected in this process, which are then used for disambiguating unseen texts from the held-out corpus.

One research by [46] explored Word Sense Disambiguation using supervised, unsupervised, and knowledge-based approaches for WSD. The finding showed that supervised methods, undoubtedly perform better than the other approaches. However, relying on the availability of large training corpora for different domains, languages, and tasks is not a realistic assumption. Their finding also

furnishes an idea of a few of the WSD algorithms and their performances, which compares and assesses the need of the word sense disambiguation.

According to [47] unsupervised WSD does not require much time in creating high quality resources and perform great in terms of accuracy. The other reason not to go for supervised approach is that, even if all possible resources are available to build a great supervised approach, it cannot be implemented in other languages easily. The resources have to be replicated for all possible languages. Another disadvantage of using the supervised approaches is, by using fixed sense repositories, the fixed number of senses present in that repository. It cannot discover new senses of words, which are not present in the sense corpus. Hence, only considering the accuracy of the approach is not a good idea, but considering its versatility and portability to other languages and domains is also equally important. These are the reasons that many unsupervised approaches were being tried by many researchers in WSD.

The research by [2] conducted the overall survey and early work on word sense extraction in context. The author started with a word sense discussion which was done by [48], which describes the need for word sense extraction in machine translation. The very first efforts at automatic sense disambiguation were created in the framework of machine [48] traces the foundation of a strategy to word sense removal in a word that is the following. The useful issue is: "What minimum value of N (Neighboring word) will, at least in a tolerable fraction of cases, lead to the correct choice of meaning for the central word?" Then the authors concentrated on early test with preceding query done by [49] which extracts sense of the main word by getting a couple of phrases on both sides of a central word. The impressive truth about it is that early function on word sense removal is the amount to which the strategies and essential difficulties towards the difficulty were foreseen and created in those days. It gives overall earlier work on word sense extraction. It provides idea about selecting values of neighboring word for extracting word sense in a particular context.

According to [3] surveyed the field of word sense disambiguation. The author asserted that human language is ambiguous, so that many words can be interpreted in multiple ways depending on the context in which they occur. Most of the time humans do not even think about the ambiguities of language, machines need to process unstructured textual information and transform them into data structures which must be analyzed in order to determine the underlying meaning. The computational identification of the meaning of words in context is WSD. The author, distinguished two main

approaches to WSD namely: i) Supervised WSD: These strategies use machine-learning methods to understand a classifier from tagged training sets, that's models of examples secured in conditions of a quantity of attributes as well as their proper sense tag (or course). ii) Unsupervised WSD: These procedures are centered on unlabeled corpora and do not utilize any manually sense-tagged corpus to supply a sense alternative for a word in context. It gives detailed survey of word sense extraction with approaches. It is useful to select an approach for sensing word in context.

The research by [50] stated that lexical meaning of words using vector spaces and linear algebra is paramount. He argued that the meanings of words will be represented using vectors, as part of a high-dimensional "semantic space". The fine-grained structure of this space is provided by considering the contexts in which words occur in large corpora of text. Words can easily be compared for similarity in the vector space, using any of the standard similarity or distance measures available from linear algebra, for example the cosine of the angle between two vectors. Other authors said that the hypothesis underlying distributional models of word meanings is the so-called distributional hypothesis: the idea that "Words that occur in similar contexts tend to have similar meanings" [51]. According to [50] now the basis vectors correspond to whole sequences of grammatical relations, relating the ambiguous word and contexts. Which paths to choose is a parameter of the approach, with the idea that some paths will be more informative than others.

The research stated by [52] addressed that the task of computing vector space representations for the meaning of word occurrences, which can vary widely according to context. This task is a crucial step towards a more robust, vector-based compositional account of sentence meaning. The authors argued that existing models for this task do not take syntactic structure sufficiently into account. In the paper, it was concluded that structured vector space models of word meaning relying on this procedure of vector composition are limited both in their scope and scalability. The best-known work in this category is that of [25]. He first computes "first-order" vector representations for word meaning by collecting co-occurrence counts from the entire corpus. Then, he determines "second-order " vectors for individual word instances in their context, which is taken to be a simple surface window, by summing up all first-order vectors of the words in this context. The resulting vectors form sense clusters. In addition to a vector representing a word's lexical meaning, it contains vectors representing the word's selectional preferences. These selectional preferences play a central role in the computation of meaning in context.

The other research by [53] stated the unsupervised learning algorithm for sense disambiguation that, when trained on unannotated English text, rivals the performance of supervised techniques that require time-consuming hand annotations. The algorithm is based on two powerful constraints that words tend to have one sense per discourse and one sense per collocation exploited in an iterative bootstrapping procedure.

According to [54] has a pioneered work in the hierarchical clustering of word senses. In his disambiguation experiments, Schutze used post-hoc alignment of clusters to word senses. Because the top-level cluster partitions based purely on distributional information do not necessarily align with standard sense distinctions, he generated up to 10 sense clusters and manually assigned each to a fixed sense label (based on the hand-inspection of 10-20 sentences per cluster).

The research by [55] published a work on early application of bootstrapping to augment training sets for a supervised sense tagger. She trained her fully supervised algorithm in hand-labelled sentences, applied the result to new data and added the most confidently tagged examples to the training set. Regrettably, this algorithm was only described in two sentences and was not developed further. Other unsupervised methods have shown great promise. [56] developed a method using co-occurrence statistics in independent monolingual corpora of two languages to guide lexical choice in machine translation. Translation of a Hebrew verb-object pair such as lahtom (sign or seal) and hoze (contract or treaty) is determined using the most probable combination of words in an English monolingual corpus. This work showed that leveraging bilingual lexicons and monolingual language models can overcome the need for aligning bilingual corpora.

One research by [57] presented a model that represents word meaning in context by vectors which are modified according to the words in the target's syntactic context. Contextualization of a vector is realized by reweighting its components, based on distributional information about the contexts. Vector-space models of meaning lend themselves as a basis for determining a soft and gradual concept of semantic similarity (for example, through the cosine measure), which does not rely on a fixed set of dictionary senses with their well-known problems [58].

Inspired by earlier work of [59], who developed a network algorithm to extract context specific vector representations for words in context, [60] investigated the systematic combination of distributional representations of word meaning along syntactic structure. They used to represent the meaning of a complex expression that consists of two syntactically related words w and w' by a

vector obtained by combining the word vectors of w and w', and find that component-wise multiplication performs best for the task under consideration. They consider their proposal primarily under the aspect of compositionality, but it can also be taken to be a method to contextualize a target word through its dependents.

A different approach has been taken by [61] and Reisinger and [62]. Instead of "refining" vector representations ranging over all words in a corpus by means of vector composition, they start out from "token" vectors for individual instances of words in context and then group these token vectors into different sense specific clusters. Once features are selected, SenseClusters creates a vector for each test instance to be discriminated where each selected feature is represented by an entry/index. Each vector shows if the feature represented by the corresponding index occurs or not in the context of the instance (binary vectors), or how often the feature occurs in the context (frequency vectors). This is referred to as a first order context vector, since this representation directly indicates which features make up the contexts. Here we are following [63], who likewise took this approach to feature representation. [25] utilized second order context vectors that represent the context of a target word to be discriminated by taking the average of the first order vectors associated with the unigrams that occur in that context. In SenseClusters we have extended this idea such that these first order vectors can also be based on co–occurrence or bigram features from the training corpus.

Rule based systems exploit the hand crafter rule for WSD task. The rule based systems require extensive work of expert linguists and thus can result in near human accuracy. The research done by [64] Afan Oromo rule based Afan Oromo Grammar Checker, shows a promising result. The finding of the study shows that rule based is an approach used in the morphologically rich language like Afan Oromo. This rule based approach for languages, such as Afan Oromo, advanced tools has been lacking and are still in the early stages. The rule based method is advisable for under resourced languages.

One research by [65] the result of the study shows that the limited availability of resources in the form of digital corpora and annotated, the rule based method is applied. All senses are discovered using a set of rules and knowledge base for later use in the disambiguation process. The finding shows an improvement in assigning correctly the corresponding disambiguation over the baseline method. All the generated information is stored in the Sense Knowledge base. Since this information can affect the accuracy of the disambiguation process.

According to [66], there are three important characteristics of an ambiguous word: grammatical information about the ambiguous word to be disambiguated, words that are syntactically related, and words that are topically related to the ambiguous word. The semantic and syntactic information to disambiguate an ambiguous word, consideration is taken in the first two types. For each entry of the Sense Knowledge base, it consists of the following: ambiguous and related words, sense of the ambiguous and related words, and part of speech of the related word. Moreover, this information can be converted into an understandable rule format that best describes the relationship between, the ambiguous word and the related word. The rules can be easily constructed to describe the relationship between, the ambiguous and related words.

According to [67] the automatic generation and the evaluation of sets of rules for word sense disambiguation (WSD). The ultimate aim is to identify high-quality rules that can be used as knowledge sources in a relational WSD model. The knowledge-based approaches which depend on the manual encoding of accurate linguistic knowledge and disambiguation rules.

Chapter Three

3. Overview of Afan Oromo

3.1. Introduction

Afan Oromo, also called Oromiffaa or Afaan Oromoo, is a member of the Cushitic branch of the Afro-Asiatic language family [68]. It is the third most widely spoken language in Africa, after Hausa and Arabic. Its original homeland is an area that includes much of what is today Ethiopia, Somalia, Sudan and northern Kenya and some parts of other East African countries [69]. Currently, it is an official language of Oromia Regional State (which is the biggest region among the current Federal States in Ethiopia). It is used by Oromo people, who are the largest ethnic group in Ethiopia, which amounts to 50 % of the total population in 2007[70]. With regard to the writing system, Qubee (a Latin-based alphabet) has been adopted and become the official script of Afan Oromo from 1991 [10].

Among the major languages that are widely spoken and used in Ethiopia, Afan Oromo has the largest speakers [69]. It is considered to be one of the five most widely spoken languages among the roughly one thousand languages of Africa [71]. Afan Oromo, although relatively widely distributed within Ethiopia and some neighboring countries like Kenya, Tanzania and Somalia, is one of the most resource scarce languages [72]. It is part of the Lowland East Cushitic group within the Cushitic family of the Afro-Asiatic phylum [69], unlike Amharic (an official language of Ethiopia) which belongs to Semitic language family. Although it is difficult to identify the actual number of Afan Oromo speaking societies (as a mother tongue), due to lack of appropriate and current information sources, according to the census taken in 2007 it was estimated that 50 % of Ethiopians are ethnic Oromo [73].

It is widely used as both written and spoken language in Ethiopia and neighboring countries [73]. Currently it is used as an instructional media for primary, junior secondary schools and Ethiopian Universities in the region and out of the region like Mekele, Dilla and Arbamich Universities. It is also given as a subject starting from grade one to high school throughout the schools in Oromia region. It is also broadcasted via televisions and radio stations locally and internationally. Furthermore, few literature works, newspapers (Bariisaa, Kallacha Oromiyaa and Oromiyaa),

magazines, educational resources, official documents and religious writings are written and published in this language [10].

3.2. Language Structure

Afan Oromo has a very rich morphology like other African and Ethiopian languages [72]. With regard to the writing system, Qubee (Latin-based alphabet) has been adopted and became the official script of Afan Oromo since 1842 [72]. The writing system of the language is straightforward, which is designed based on the Latin script. Thus letters in the English language are also in Afan Oromo except the way it is spelled. A detailed description of Afan Oromo Writing System can be found in any text related to the language, but [74] discussed writing system of the language.

3.3. Alphabet (Qubee Afaan Oromoo)

Afan Oromo uses Qubee (Latin based alphabet) that consists of thirty three basic letters, of which five are vowels, twenty-four are consonants, out of which seven are paired letters and fall together (a combination of two consonant characters such as 'ch'). The Afan Oromo alphabet characterized by capital and small letters as in the case of the English alphabet. In Afan Oromo language, as in English language, the vowels are sound makers and are sound by themselves. Vowels in Afan Oromo are characterized as short and long vowels. The complete list of the Afan Oromo alphabets is found on the manuscript by [68] and [75]. The basic alphabet in Afan Oromo does not contain 'p', 'v' and 'z', because there are no native words in Afan Oromo that formed from these characters. However, in writing Afan Oromo language, they are used to refer to foreign words such as "*polisii*" ("police").

3.4. Sentence Structure

Afan Oromo and English are different in sentence structuring. Afan Oromo uses subject-object-verb (SOV) language. SOV is the type of language in which the subject, object and verb appear in that order. Subject-verb-object (SVO) is a sentence structure where the subject comes first, the verb second and the third object. For instance, in the Afan Oromo sentence "*Mooneeraan bilisa bahe*". "*Mooneeraa* "is a subject, "bilisa" is an object and "bahe" is a verb. Therefore, it has SOV structure. The translation of the sentence in English is "*Mooneeraa* has got freedom" which has SVO structure. There is also a difference in the formation of adjectives in Afan Oromo and English. In Afan Oromo

adjectives follow a noun or pronoun; their normal position is close to the noun they modify while in English adjectives usually precede the noun. For instance, *namicha gaarii* (good man), *gaarii* (adj.) follows *namicha* (noun).

3.5. Adjectives

An adjective is a word which describes or modifies a noun or pronoun. A modifier is a word that limits, changes, or alters the meaning of another word. Unlike English, adjectives are usually placed after the noun in Afan Oromo. For instance, in *Qotiyyoo diimaa gate* " He lost red Ox" the adjective *diimaa* comes after the noun *Qotiyyoo* [10].

3.6. Articles

Afan Oromo does not require articles that appeared before nouns, unlike that of English. In English there are three main semantic choices for article insertion: definite article (the), indefinite article (a, an, some, any) and no article. In Afan Oromo, however, the last vowel of the noun is dropped and suffixes (*-icha,-ittii,-attii*) are added to show definiteness instead of using definite article. For example, "the man" is "*namticha*" to indicate certainty [68].

3.7. Punctuation Marks

Punctuation marks used in both Afan Oromo and English languages are the same and used for the same purpose with the exception of the apostrophe. Apostrophe mark (') in English shows possession, but in Afan Oromo it is used in writing to represent a glitch (called *hudhaa*) sound. It plays an important role in Afan Oromo reading and writing system. For example, it is used to write the word in which most of the time two vowels appeared together like "*ba'e*" to mean ("get out") with the exception of some words like "*ja'a*" to mean "six" which is identified from the sound created. Sometimes apostrophe mark (') in Afan Oromo interchangeable with the spelling "h". For instance, "*ba'e*", "*ja'a*" can be interchanged by the spelling "h" like "*bahe*", "*jaha*" respectively still the senses of the words is not changed.

3.8. Conjunctions

Conjunctions are used to connect words, phrases or clauses. In Afan Oromo there are different words that are used as a conjunction. Conjunctions in Afan Oromo are "*fi*", "*haa ta'u malee*", "*garuu*", "*akkasumas*". Example: *Amboo, Finfinnee fi Jimmaan magaalota Oromiyaati*.(Ambo, Finfine and Jimma are Oromian cities.)

3.9. Word and Sentence Boundaries

In Afan Oromo, like in other languages, the blank character (space) shows the end of one word. Moreover, parenthesis, brackets, quotes are being used to show a word boundary. Furthermore, sentence boundaries punctuations are almost similar to English language i.e. a sentence may end with a period (.), a question mark (?), or an exclamation point (!) [76].

3.10. Word Segmentation

The word, in Afan Oromo "*jecha*" is the smallest unit of a language. There are different methods for separating words from each other. This method might vary from one language to another. In some languages, the written or textual script does not have whitespace characters between the words. However, in most Latin languages a word is separated from other words, by white space characters [77]. Afan Oromo is one of Cushitic family that uses Latin script for textual purpose and it uses white space character to separate words from each other's. For example, "*Bilisummaan Finfinnee deeme*". In this sentence the word "*Bilisummaan*"," *Finfinnee*" and" *deeme*" are separated from each other by white space character. Therefore, the task of taking an input sentence and inserting legitimate word boundaries, called word segmentation, is performed using the white space characters.

3.11. Homonymy

A homonym is a group of words sharing same spelling and pronunciation, but have different senses. These multiple senses can be totally unrelated to each other. Homonymy is defined as a relation that holds between words that have the same form with unrelated meanings. The items taking part in such a relation are homonyms [16]. 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:

Afaan koo faaya kooti My language is My beautifulness, My mouth is my beautifulness

In the example the underlined "*Afaan*" is an ambiguous word having the same pronunciation and spelling but different meaning. This notation indicates that these are two separate lexemes, with distinct and unrelated meanings, that happen to share an orthographic form.

3.12. Synonym

Polysemy is the special case of homonymy where multiple senses of the word are related to each other [79]. The phenomenon of synonymy is sufficiently widespread to account for the popularity. The notion of synonymy, has a deceptively simple definition different lexeme with the same meaning. The synonyms can substitute for one another in a sentence without changing either the meaning or the acceptability of the sentence [16].

As an example: *Seenaan gargaarsa naaf godhe*. "Sena helped me".

The underlined word *gargaarsa* can be substituted by the word "*deeggarsa*" which is not used in this sentence. These two words can each other without changing the meaning of the sentence. By this, two lexemes share a central core meaning.

3.13. Polysemy

The phenomenon of a single lexeme with multiple related meanings is polysemy. The notion of polysemy allows to state that sense is related to, and possibly derived from, without asserting that it is a distinct lexeme [16]. For example: *Qorachuun mala dhayiiti*. The study is the way of putting solution.

Here the word underlined *Qorachuun* which mean *Iyyaafachuu* “call out” and *gaafachuu* “ask” are two related meanings according the contexts of the sentence. As one suspect, the task of distinguishing homonymy from polysemy is not quite a straightforward. There are two criteria that are typically invoked to determine whether or not the meanings of two lexemes are related or not: the history, or etymology, of the lexemes and how the words are conceived by native speakers [16]. In the absence of detailed etymological evidence, a useful intuition to use in distinguishing homonymy from polysemy is the notion of coincidence. On the other hand, it is far more difficult to accept cases of polysemy as coincidences.

Chapter Four

4. Methodology

4.1. Introduction

This chapter describes the methods employed in this research. In order to develop disambiguation model for Afan Oromo we followed three steps process which involve: (a) text preprocessing which take input and corpus, tokenize to remove stop words and perform normalization (b) extract context terms providing clue about the senses of the ambiguous term using two techniques (window size and rule based), (c) clustering to group similar context terms of the given ambiguous terms, the number of clusters representing the number of senses encoded by the ambiguous term. In order to cluster similar context terms we computed the degree of similarity using the vectors constructed from co-occurrence information. The architecture (pipe line) of the system with the underlying steps is presented in figure 4.1 below:

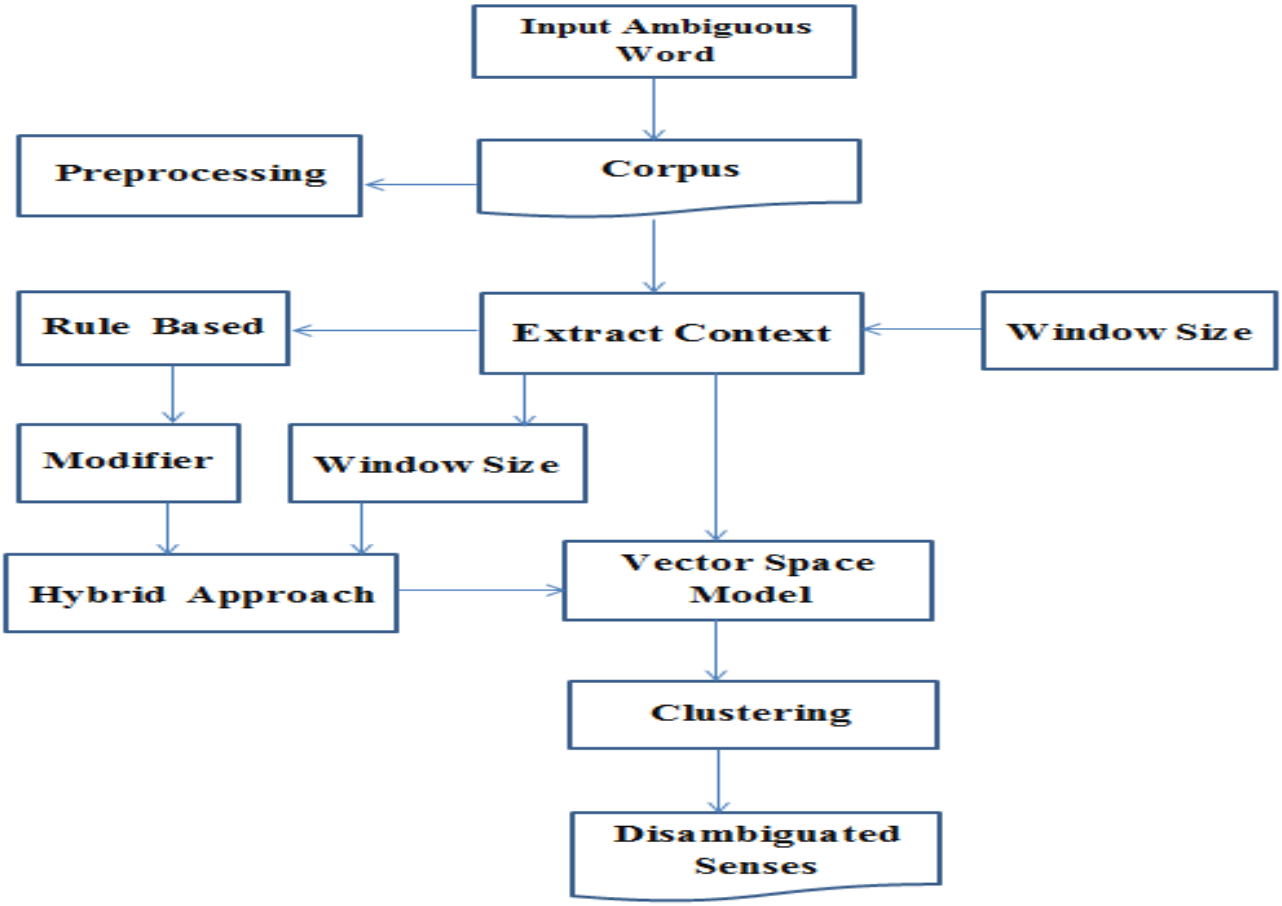


Figure 4.1. The Architecture (pipe line) of the System

The description of the corpus preparation procedures followed by the various algorithms employed are described in Section 4.3 and Section 4.6 respectively.

4.2. Method

To achieve the objectives stated in Section 1.3, this study relies on the patterns learned from the corpus (unsupervised approach) in combination with the manually crafted rules for clustering similar contexts of ambiguous word and extracting the contexts respectively. The motivation behind the use of hybrid is mainly aroused form the fundamental problem of corpus-based approach in relation to the sparseness of the training contexts. The idea of this research is therefore to combine both the rule based and unsupervised machine learning approaches into a hybrid approach. Such method of word sense disambiguation as also employed in this work, combine the advantages from machine learning and rule based, potentially yielding better results. Hence, the reason why the hybrid approach was used is taking the availability and reliability of linguistic knowledge on the top of the semantic techniques and training methods learned from corpus for learning the role of the words in its context. We have described the techniques employed under each task in the following sections.

4.3. Corpus

This section deals with the collection of data and preprocessing the collected data. For this study, we have prepared new corpus for the WSD of Afan Oromo. The size of the corpus is an important factor in the quality of the disambiguated senses, with the general message that more data is better data [78]. However, in such a work, it is very difficult to obtain standardized dataset for under resourced language. The procedure for collecting and preprocessing corpus is described here below.

4.3.1. Corpus Acquisitions and Preparations

The corpus acquisition and preparation are set of techniques required for gathering and compiling data for training and testing algorithms. The lack of resource has led researcher to explore the use of unannotated raw corpus to perform hybrid disambiguation. It should be noted that unsupervised disambiguation cannot actually label specific terms as a referring to a specific concept that would require more information than is available.

The mechanism to acquire resource is to use the data from various sources and hence it is not previously used in any research on Afan Oromo. The collected data is machine readable free text. It is collected from newspapers (Bariisaa, Kallacha Oromiyaa and Oromiyaa. Bariisaa is a weekly newspaper, whereas the rest two come out once in two weeks), bulletins, news (ORTO), government and official websites. Moreover, ORTO found in Adama releases daily news through radio and television broadcast and on its official website [10]. To reduce the data sparsity we have used data from these sources since they are believed to represent texts addressing various issues of the language. Actually, the collected data was not directly used for the purpose. We have applied preprocessing (tokenize, stop word and normalize) techniques before it used. Hence, the acquired corpus is a free sentence of the corpus, we have prepared and filtered to make as it ready for the study in the following Sections.

4.3.2. Corpus Preprocessing

The preprocessing step includes tokenizing, stop-word removal and normalization (change characters to the same form). For stop word removal, we have used the Afan Oromo stop-word compiled from different text in addition to the stop-word list prepared by [79]. In case of the normalization, we have simply convert the corpus into smaller case (the same case), hence it has no effect on the meaning of the words.

A. Tokenization

The corpus which is a set of sentences first tokenized into words. Since, Afan Oromo uses Latin alphabet the sentences can split using similar word boundary detection techniques like the use white space in English. In this work tokenization is needed for some purposes that are to select contexts, stop word removal and for further steps. For instance if we have the sentence *akkanatti hin haftu ni dabarti* in our corpus. We have tokenized into set of words on the white space, like *akkanatti, hin, haftu, ni* and *dabarti.* After tokenization we will apply our algorithm to extract contexts among the tokens as discussed in the coming sections (Section 4.6.1.1).

B. Stop Word Removal

After tokenization take place, we have removed Afan Oromo stop words, hence it has no effect on meaning of the words. In this work, Stop word removal is used to remove stop words from the selected contexts because the absence or presence of these words has no contribution to identify appropriate sense. Not all tokenized words are necessary for this work hence, one word carry the meaning than other words and other words that have no own meaning. For instance, words such as (‘*as’*, ‘*achi’*), conjunctions(“*’fi’*, ‘*akkasumas’*, ‘*kana malees’*). Since stop words do not have significant discriminating powers in the meaning of ambiguous words; we filtered stop-words list to ensure only content bearing words are included. Few list of stop words in Afan Oromo is shown in the table 4.2 below:

Stop Words	**Meaning (English)**
Kana	This
Achi	There
Yookan	Or
Haa ta’u malee	However
Irra	On
Jala	Under
As	Here
Kanaaf	So
Jedhama	Called
Ammo	However, but
Garuu	But
Bira	Beside, at, near of
Ala	Outside, out
Akka	Such as, like, according to

Table 4.2. Stop Words (source: unpublished [80])

C. Normalization

In this work, some characters of the same words are sometimes represented in uppercase or lowercase in the corpus as well as in the user input and hence we have normalized them into lowercase. The purpose of normalization in this case is to make similar the words in different cases in our corpus. This is the observation that many different strings of characters often convey essentially identical meanings. Given that to get at the meaning that underlies the words, it seems reasonable to normalize superficial variations by converting them to the same form. The most common types of normalization are case folding (converting all words to lower case). Case folding is easy in Afan Oromo for example *Qabsoo* similar to *qabsoo*.

4.4. Implementation Tools

To perform this experiment, the algorithms were implemented in Java, NetBeans 8.0.2 which run on the prepared corpus and the clustering were performed in Weka 3.6.5 tool. Java package is a general-purpose and open source programming language. Moreover, it is optimized for software quality, developer productivity, program portability, and component integration. Nowadays Java is commonly used around the world for Internet scripting, systems programming, user interfaces, product customization, numeric programming, and more. It is generally considered to be among the top four or five most widely-used programming languages in the world.

The Weka 3.6.5 is the other tool used to implement WSD specifically context clustering, using the hierarchical clustering (which are single link, complete link and average link) and partitional clustering (EM, K-means) algorithms built into the tool. The tool has by default Euclidian distance which is used to measure the distance between clusters.

4.5. Testing

The WSD system was trained on unannotated corpus, which constitutes ambiguous words. The work was tested by using most frequent 15 ambiguous words collected from the public via questions (in an Appendix A) from random native speakers of the language. To this end, we collected the words from 20 individuals and selected the top 15 frequent ambiguous terms. The senses encoded by the words in the test set are determined from the information obtained from the language experts. The selected ambiguous words are listed below:

No	Ambiguous Word	Possible Senses	Defined Number of Senses
1	Afaan	Language / Mouth	2
2	Bahe	Freedom / Highland / Cloth / Witness / Dead(pass)	5
3	Boqote	Break (rest) / Died	2
4	Darbe	Cross/ Pass from class to class / Died / broadcast	4
5	Diige	Fence / Absence on Meeting / cancel to start new	3
6	Dubbatate	Struggle / Wedding	2
7	Tume	Make / Hit / Contraceptive	3
8	Haare	Sad / Burn	2
9	Handhuura	Center / Gift / Navel	3
10	Ija	Eye/ Tree Fruit / Vengeance/ Wide-eyed/ a little	5
11	Ji'a	Stars / Month	2
12	Lookoo	Pretty / Rope	2
13	Dhahe	Follow / Hit /Fail	3
14	Mirga	Direction / Human right / Brave	3
15	Waraabuu	Hyena / Fetch / Record	3

Table 4.1. List of Ambiguous Words for testing

4.6. Algorithms

In our approach, two important features need to be extracted: the first one is determining all possible contexts (the candidate sense words) of the ambiguous words and the other one is to group these various contexts (senses) of the word, each group representing a specific sense of the ambiguous word. To this end, we developed two kinds of approaches towards the word sense disambiguation. The first approach is completely machine learning in its nature. Unsupervised machine learning approach extracts the two important features (the various contexts of the ambiguous words and their clustering) without supplying linguistic rule (Section 4.6.1.2). The second approach combined unsupervised machine learning algorithm and rule based (background linguistic). Such algorithm is called hybrid approach, which is a mixture of rule based (background knowledge) and corpus-based approaches applied. In this latter approach we combined the rule learned from a linguistic feature of Afan Oromo with the semantic feature learned from corpus using unsupervised machine learning approaches. First, the manually crafted linguistic rule extract the various contexts assumed by the ambiguous word used. We then relied on the unsupervised machine learning algorithm to cluster these contexts as described in Section 4.6.1.4. Hence, this latter approach extracts all possible contexts assumed by the ambiguous word employing rule-based approach, which make use of linguistic knowledge, whereas the corpus-based approaches use information acquired from the corpus; and the hybrid approach merges characteristics from both approaches.

The machine learning approach followed towards the WSD is basically unsupervised and mainly used for clustering the possible contexts for giving words that express the same meaning. Hence we didn't provide explicit sense labels for each group as the machine learning approach is unsupervised. Yet, small list of ambiguous words are required to test the algorithm.

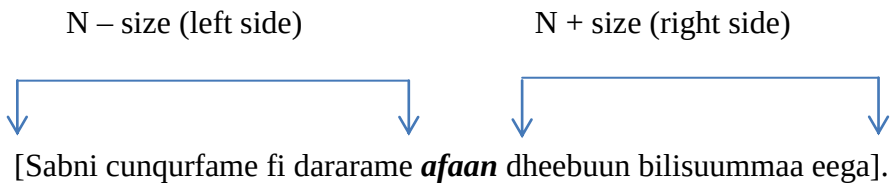
4.6.1. Machine Learning Approach

As discussed in the above (Section 4.6), two important features need to be extracted. We have described the procedures followed to extract the features in the subsequent Sections.

4.6.1.1. Extracting the Context Words

In the machine learning, the surrounding contexts were extracted by sliding window of *n* words. The reason why the contexts extracted is that, the words that occur in similar contexts tend to have similar meanings. This idea, known as the 'distributional hypothesis', has been proposed by various scholars. In order to extract the contexts from a set of sentences the role of window is great. Because a word's meaning cannot be fully grasped unless one takes the context into account. Meaning and context can be captured in terms of (more or less direct) neighborhood, i.e. words co-occurring.

Most of WSD algorithms make use of the contexts to provide information for sense disambiguation. As already mentioned in this approach, we followed automatic procedure to extract all possible contexts assumed by the ambiguous word. This is the idea of the study, which is context based meaning of words. We have determined a set of contexts based on the frequency co-occurrence of context words in the corpus with target words, by determining a window size contexts to the left and to the right.



Once the context words are extracted, the next step cluster similar contexts based on their inherent semantics, the number of the cluster representing the number of senses assumed by the ambiguous word. The procedure employed for clustering the context is described below.

4.6.1.2. Clustering the Contexts

For each context extracted in Section 4.6.1.1 above, we constructed a vector space matrix from co-occurrences. After the co-occurrence matrix, the cosine similarity was computed based on the angle between vectors of the contexts.

These cosine similarity values were used to cluster similar contexts by using Weka 3.6.5 tool. The cosine similarity result which is in .CSV (Comma Separated Value) file format was entered into the Weka tool. The Weka tool cluster the contexts based on cosine similarity result. Each cluster represents a unique sense. To this end, we used hierarchical (single, complete, and average link) clustering, and (K-means, EM) algorithms from partitional clustering to merge the contexts based on their cosine values. The vector space model and the clustering algorithms are described in Sections 4.6.1.3 and 4.6.1.4 respectively.

4.6.1.3. Vector Space Model

The vector space model is used to represent the meaning of a word, hence it used to measure semantic similarity. The rationale behind the vector space model is that, words that occur in similar contexts tend to have similar meanings (as it described earlier). We construct vectors for each extracted context terms of the ambiguous word for measuring their similarity. For the given pair of context terms the algorithm extract words co-occurring in a predefined window. After extracting the term co-occurrences we computed weight based on their frequency. We did the same for the other pair of the context term. In order to identify the similarity between the context words the cosine similarity of the vectors were computed. The cosine similarity of the vectors is then considered to determine the semantic similarity between the terms. It is preferable because it takes into account variability of data and features' relative frequencies as well as useful to intend with cluster as it captures similarity overall.

To illustrate this with a simple example: consider a co-occurrence matrix populated by simple frequency counting: if the term *i* co-occurs 5 times with term *j* in the corpus, we have used 5 in the f_{ij}(frequency of term *i* and *j*) to compute the cosine similarity. The co-occurrences are normally counted within a context window spanning some number of words. In our case, the vectors are extracted by taking words regardless of their passion, hence, it doesn't consider the ordering of the words in the corpus. It uses the angle between vectors instead of distance in order to measure the similarity between the context terms.

$$sim_{COS}(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{\|\vec{x}\|\|\vec{y}\|} = \frac{\sum_{k=1}^{n} x_i y_i}{\sqrt{\sum_{k=1}^{n} x_i^2}\sqrt{\sum_{k=1}^{n} y_i^2}} \quad \text{.....1}$$

Where SimCos – is cosine similarity

x – is the dimension of the vector one

y – is the dimension of the second vector

4.6.1.4. Clustering Algorithms

As already mentioned, we cluster context terms of the ambiguous words using their similarity values produced in Section 4.6.1.3. The clustering algorithms used in this study are hierarchical agglomerative clustering, which include single link, complete link, average link and EM and K-means clustering from partitional clustering (see details in Section 2.6). These clustering algorithms have their own unique nature. The hierarchical clustering begin by assuming that each context of an ambiguous word forms its own cluster (and therefore represents a unique sense). Then, it merges the contexts that have the minimum dissimilarity between them (and are therefore most alike). The partitional clustering algorithms started by partitioning into predefined *k* sizes [82]. It found the one which is the nearest to initial centroid. A centroid is usually not an element of the cluster. Rather, it 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 centroid. The minimum specified cutoff which determines the number of clusters is taken. In this case, the minimum specified cutoff of the number of clusters is two hence one ambiguity word has at least two senses.

4.7. Hybrid Approach

According to [22] WSD heavily relies on knowledge of machine learning and linguistics. The hybrid approach in this work, constitute the unsupervised machine learning approach to cluster the contexts followed by hand crafted rule to extract the modifiers of the ambiguous word as described in the following Sections.

4.7.1. Constructing Rule for Extracting Contexts

The linguistic knowledge of the language plays an important role to create the rule. The linguistic knowledge required for the natural language can be obtained in different ways. In this study, the rules were created based on the inherent structure of Afan Oromo in forming meaning of words. However, an effort has been to develop the rule of the language (more details in Section 4.7.1.2).

The most common way of representing language is via rules [68]. The rule underlies many linguistic theories of the language, which turn into a set of rules [82]. The modifiers and contextual information were the basis of the linguistic properties of Afan Oromo word meaning. In this study, the modifier of the ambiguous word is used in order to get the semantic clues of the particular sense in which the ambiguous word is used. To achieve this we analyzed the structure of Afan Oromo sentence formation with respect to modifier patterns to develop the rule. The constructed rule used for extracting all modifiers modifying the ambiguous term. The modifiers are word or phrase which provide information about a word and give more description about the words it modify. The modifiers (can be single word or phrase) establish understanding of the ambiguous word.

In Afan Oromo, the meaning of words fundamentally based on the words preceded by the word (modifies) [83] [84]. Hence, the words are described (modified) by the Noun or Verb preceding them. However, since there is no annotated corpus of this language we didn't identify whether this assertion is holds true or not. The ambiguous word may appear in the middle or the end of a sentence, but modifiers always becomes before the word it modifies. As an example, "*Seenaan kaleessa daara bahe*." [Seenaa got cloth yesterday]. From the construction of this sentence, the word "*bahe*" is modified by the Noun "*daara*". According to Afan Oromo structure, the Noun and Verb always appear before the word they modify [66].

4.7.1.1. Modifiers

In Afan Oromo like other languages, the word meaning has own its rule. However, the correct sense cannot be only found by choosing the one that is related to another. Promising techniques relied on linguistic knowledge also for extracting semantic features, in our case to mine context of the ambiguous term via of modifiers specializing its meaning [72].

The modifiers have a great role to decide on the word meaning according to its role in the sentence. The modifiers can appear before the target word (the word it modify or describe). Like English, the sentences would be pretty boring without modifiers to provide excitement and intrigue. A modifier adds detail or limits or changes the meaning of the other word or phrase.

In Afan Oromo, the words preceding a specific word are more likely to influence the meaning of a word.

For example, [*Finfinneen **magaala** Oromiyaati eenyumatti hin kenninu*]**.**

Finfine is Oromian city nobody will take it from Oromia.

In this example, the core of the sentence is "[*Finfinneen magaala Oromiyaati eenyumatti hin kenninu*] Finfine is Oromian city nobody will take it from Oromia". The word "[*Finfine*]" is a modifier; it gives an extra information that is part of the sentence. In this case, it is a Noun modifier, because it is modifying the ambiguity word "*magaala*". A modifier should be placed next to the word it describes.

4.7.1.2. Modifiers Rule

Disambiguation is done by analyzing the linguistic features of the word and its preceding word. The rule-based section of our approach disambiguate word automatically using rule in order to complement the features learned from training data. This information is coded in the form of rules. As it discussed in the Section 4.7.1.1, the modifiers always precede the target word in Afan Oromo. Based on this notion, the rule developed was as follows:

- **IF** ambiguous word preceded by Modifiers **THEN** collect the modifiers to disambiguate.
- **IF** ambiguous word is **NOUN**, the modifiers immediately following ambiguous word
- **IF** ambiguous word was **VERB**, the modifiers immediately preceding ambiguous word

4.8. Evaluation Method

To our knowledge, there were no previous standard Afan Oromo word sense disambiguation dataset for evaluation as presented in Section 4.5. For this reason we did not evaluate against the other systems. In this work, the evaluation were undertaken on the basis of precision and recall. Precision is defined as the percentage of correctly disambiguated words out the total of disambiguated words. Recall is defined as the percentage of correctly disambiguated words out of the total number of ambiguous words [85].

$$\textbf{Precision}(\%) = \frac{\text{\# correctly disambiguated words}}{\text{\# disambiguated words}}$$

$$\textbf{Recall}(\%) = \frac{\text{\# correctly disambiguated words}}{\text{\# total number of ambiguous words}}$$

On the other hand, the clustering algorithms were evaluated comparing the result produced by the clustering algorithm with the manually grouped similar contexts of the ambiguous words in the test set by experts. The evaluation constitutes the following two points:

1. To evaluate how much the produced clusters are comply with the clusters prepared by human experts as a benchmark..

   In order to achieve this we used the following criteria:

   - How many of the clustered contexts are correct, i.e. to evaluate if all the similar contexts of the ambiguous words are placed in the same group.

2. Given the number of senses assumed by the ambiguous words in the test, judge the system on the basis of the number of senses identified by the system.

Similarly, in order to achieve this the following steps performed:

a. Start with a small list of ambiguous words in the test with known number of senses N.
b. Run the algorithm on the test to identify the possible senses based on it's the number of clusters of the context as extracted from the big corpus
c. Count the number of clusters
d. Compare it against the already prepared sense clusters by experts

Chapter Five

5. Experimentation Result, Discussion and Evaluation

5.1. Introduction

In this chapter we have described the training data (Section 5.1.1), the experimental procedure and the result of the unsupervised machine learning and the hybrid WSD approach in Section 5.2, 5.3 respectively.

The following figure shows the user interface of the sense disambiguation used during the experimentation.

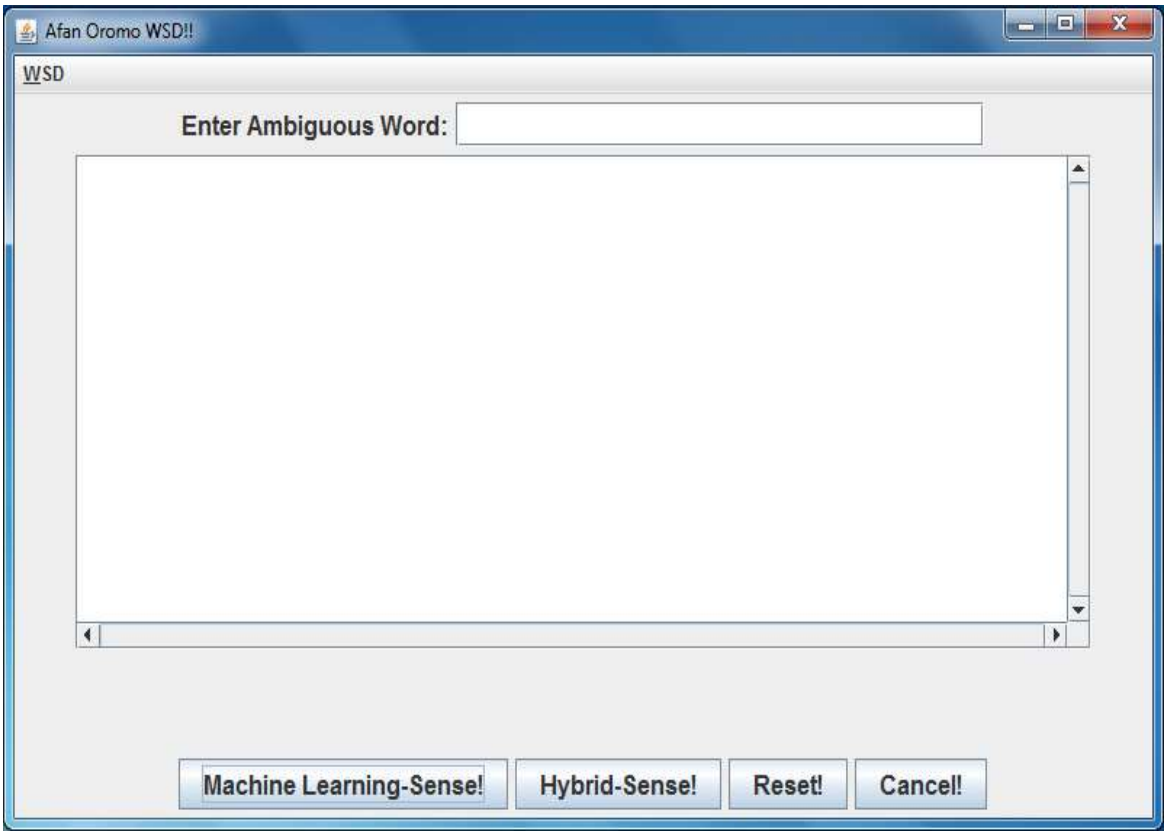


Figure 5.1. User Interface of the WSD

To disambiguate the given ambiguous word, it takes one ambiguous word at a time in the interface provided and produce the cosine similarity to cluster.

5.1.1. Data

In this work, two types of data were used: (a) the small list of ambiguous words to test the algorithms. Fifteen (15) highly frequent ambiguous words were selected from the language speakers using the procedure described in Section 4.5, (b) the big corpus containing thousands of sentences to extract contexts and its vectors to represent the group of contexts. Hence, the compiled corpus for the experiment is a free text. It is composed of a set of 57,062 lines of sentences and 412,366 words. This corpus is prepared for the purpose of this study as there is no standard corpus for Afan Oromo language. However, it is unlabeled data and never used in any researcher and it needs to be developed further (as discussed in Section 4.3.1).

5.2. Experiment on the Machine Learning Approach

As mentioned in chapter four, the machine learning approach uses a window of *N* words surrounding the ambiguous word to extract contexts. The extracted context words are then clustered together based on their similarity values using clustering algorithms and vector space model respectively.

In the following section, we have presented the experimental procedure and result on various windows size and suggested the optimal window size for Afan Oromo.

5.2.1. Experiment on the Context Window Sizes

Defining an optimal window size plays a crucial role to identify the information surrounding the ambiguous word. In English a standard window of two-words on either side of the ambiguous word is found to be enough for disambiguation [66]. Similarly, for Afan Oromo WSD, two window size on both sides for the ambiguous word is found to be enough. Therefore, this has been established for Afan Oromo WSD. The investigated experiment shows that the window sizes around the ambiguous word influences WSD performance with various window sizes.

The system provides context terms to the left and right side of ambiguous word (excluding stop words) based on the provided window size. In this experiment, window size of up to ±10 words on both sides of ambiguous word have been tried for Afan Oromo. Starting with the first term in the test set, extract words appearing *N*-words left and right of the ambiguous words. In order to measure the performance of the window size for the disambiguation, we run the disambiguator using all the

window sizes (starting from 1 to 10) and observe the differences in the performance of the disambiguator.

As evidenced by the experiment, differences in the window sizes yield different results. The result has proved that window of two-words to the left and right of the ambiguous word achieved the best performance than other windows. This result converges with the result obtained by [66]. Table 5.1 below shows the performance of the disambiguate using different window sizes.

Window Size (N)	**Accuracy (%)**
1	73.34
2	66.67
3	60.0
4	55.5
5	51.6
6	46.3
7	41.1
8	36.0
9	31.8
10	28.6

Table 5.1. Determining Optimal Window Size

As can be seen from table 5.1, a narrow window of context, one and two words to either side (73.34% and 66.67% respectively), was found to perform better than wider windows (28.6%). The accuracy is conducted by measuring the performance of WSD with varying the window size (the tested ambiguous word found in sentences). It is very likely that smaller window sizes have yield significantly higher accuracy than other windows and different windows gave different results. From this experiment, we conclude that smaller window sizes usually lead to accuracy while bigger window sizes relatively low accuracy.

5.2.2. Experiment on the Clustering

The underlying idea of using a clustering algorithm is to cluster contexts which are providing a useful way to discover semantically related senses and the same sense of a word has similar neighboring words. In this case, the clustering algorithms directly use the vector representations (cosine similarity measure) of the contexts of the ambiguous word (extracted using rules or the window of words) as input. The basic point of sense clustering is that related word senses must be associated and grouped in the same cluster. Hence, it identifies groups of senses, which are assumed to represent different meanings. The Weka clustering tool is used to group similar contexts together assuming, the induced clusters represent the different senses of the ambiguous word.

The contexts of the ambiguous words are clustered using three hierarchical clustering algorithms and two partitional clustering algorithms. Each and every of clustering algorithm has its own measures of similarity so as the cluster formed by the different clustering algorithms were produce different result. One algorithm may make 5 clusters the other may define only 2 cluster groups. The number of clusters formed by the clustering algorithms depends upon the uniqueness and the dissimilarity present in the cosine similarity.

An important point here is how to decide which constitutes good clustering, since it is commonly acknowledged that there is no absolute best criterion which would be independent of the final aim of the clustering. Consequently, it is the researcher who supply the criterion that best suits their particular needs and the result of the clustering algorithm can be interpreted in different ways (see detail in Section 4.6.1.4). One approach is to group data in an exclusive way, so that if a certain item of data belongs to a definite cluster, then it could not be included in another cluster. Another approach, so-called overlapping clustering, uses unclear sets of cluster data in such a way that each item of data may belong to two or more clusters with different degrees of membership.

Given a set of context words of the ambiguous word, with the cosine similarity values corresponding to their similarity, the hierarchical clustering begins by setting each item of cosine as a cluster and proceeds by uniting the two nearest clusters. After a few iterations it reaches the final clusters wanted. The partitional clustering is partitioning the contexts into *k* clusters based on the closeness to the cluster centers and finding the maximum probabilistic.

5.2.3. Experimental Results

As already mentioned, the evaluation of the clustering algorithms is from two dimensions (refer Section 4.8 for more detail):

a. Evaluate the accuracy of the clustering on the basis of the number of clusters produced
b. On the bases of the correctness of the produced clusters

We tested the algorithm on fifteen natural ambiguous words as shown on Table 5.2 below.

Ambiguous Word	Accuracy (%)				
	Single Link	Complete Link	Average Link	K-Means	EM
Afaan	50%	50%	50%	50%	50%
Bahe	60%	60%	60%	60%	80%
Boqote	50%	50%	50%	50%	50%
Darbe	50%	50%	50%	50%	50%
Diige	50%	50%	50%	66.67%	66.67%
Dubbatate	50%	50%	50%	50%	50%
Dhahe	66.67%	66.67%	66.67%	66.67%	66.67%
Haare	50%	50%	50%	50%	50%
Handhuura	66.67%	66.67%	66.67%	50%	66.67%
Ji'a	50%	50%	50%	50%	50%
Tume	66.67%	66.67%	66.67%	66.67	66.67%
Lookoo	50%	50%	50%	50%	50%
Ija	40%	40%	40%	60%	80%
Mirga	50%	50%	50%	66.67%	66.67%
Waraabuu	66.67%	66.67%	66.67%	66.67%	66.67%

Table 5.2. Disambiguation using evidence from Machine Learning

The average accuracy for test terms were 56.2% for the machine learning approach. Thus, in total there are 15 ambiguous words to be discriminated, 6 words with 2 senses, 6 words with 3 senses, 2 words with 5 senses, and 1 word with 4 senses. Five different configurations of clustering are run for each word, leading to a total of 75 experiments. The results are shown on Table 5.2. As indicated on table 5.2, when the contexts of the word senses that are not similar are clustered together resulting in low performance and vice versa.

No	**Clustering Algorithms**	**Accuracy (%)**
1	Single Link	54.4%
2	Complete Link	54.4%
3	Average Link	54.4%
4	K-Means	56.9%
5	EM	60.7%

Table 5.3. The Accuracy of the Machine Learning

The test by unsupervised machine learning method, that deals with clustering of contexts for a given word that express the same meaning. The simple K-Means and EM clustering algorithms achieved much accuracy on the task of WSD for selected ambiguous word. The partitional clustering which include K-means and EM resulted 56.9 % and 60.7 % respectively achieved performance in clustering.

5.2.4. Summary of the Machine Learning Approach

Name of the Algorithms	No of Clusters	Cluster Instances	No of Iterations	Within clusters sum of squared errors	Time taken to build model	Log likelihood	Unclustered Instances
Single Link	24	0 - (50%) 1 - (50%)	-	-	0	-	-
Complete Link	24	0 - (70%) 1 - (30%)	-	-	0	-	-
Average Link	24	0 - (70%) 1 - (30%)	-	-	0	-	-
K-Means	26	0 - (40%) 1 - (60%)	2	7.66736	0	-	-
EM	28	0 - (70%) 1 - (30%)	-	-	0.4	26.37927	-

Table 5.4. Summary of Machine Learning Clustering Algorithms

As shown on table 5.4 the Weka experiment, the result of clustering algorithms varies with the distance they used. The single link algorithm uses the shortest is the most similar senses. Contrary, the complete link algorithm uses the maximum distance is the most similar senses and average clustering use the average of both algorithms.

A good clustering algorithm was automatically discovered an approximately same number of clusters as senses of that word. As can be seen from table 5.2, most of the ambiguous words have 2 to 5 senses. The experimental evaluation showed that K-Means and EM always lead to significantly higher accuracy than hierarchical algorithms which suggests that partitional algorithms are well-suited for clustering cosine similarity due to not only their relatively low computational requirements, but also comparable or even better clustering performance.

We verify that our cluster estimate the number of clusters in a dataset by observing the experimental results with varying number of *k* clusters for the parameter. As shown on table 5.3, K-means and EM relatively considerable than other clustering algorithms. In the case of K-means, the results are

entirely dependent on the value of *k* i.e. the number of clusters. There is no way of knowing how many clusters exist. The same algorithm can be applied to the same cosine value, which can may produce two or three clusters. There is no general theoretical solution to find the optimal number of clusters for any given data. For this reason, the result produced by algorithms cross checked with the result by experts. EM is the analytical methods available to achieve probability distribution parameters, in all probability the value of the variable is given.

As can be seen from table 5.3 above, out of the total clusters grouped by experts, the possible clusters are produced by different algorithms that correspond to different senses. An important issue in clustering is how to decide what is the best set of clusters for a given dataset, in terms of both the number of clusters and the membership of those clusters. For this reason we compare the number of clusters produced with the number of senses chosen by a group of human subjects (experts). However, counting the number of clusters only doesn't report about the accuracy of the system if one doesn't take the quality of the clusters into account. Hence, we have also evaluated how good the clusters are by comparing them against the clusters produced by experts. Put, differently, we measured the number of contexts assigned to correct and incorrect clusters (senses).

Based on the experiment, few of the clusters wrongly clustered with different contexts which has a different sense, for instance the pair of contexts *dhugaa* and *ragaa* are appearing in different clusters when clustering as shown in Figure 5.2 and Figure 5.3 below. The hierarchical clustering algorithms are not much satisfying result because of they are on a limited amount of information (single link), or they assume that all the cosine value in the cluster is very similar to each other at distance (complete link). The average link measuring the similarity of both these clustering algorithms, as the average of the pairwise similarity of the cosine value from each cluster.

5.3. Experiment on the Hybrid Approach

The second part of the experiment in this work is combining unsupervised machine learning and rule based approaches. Unlike unsupervised machine learning in Section 5.2, the hybrid approach relies on both machine learning and rule based, which is hand-constructed rules that are acquired from the structure of language rather than automatically trained from data. The rule based WSD benefit from various sources of linguistic knowledge [86]. This approach is used when there is a scarcity of data. The rule depend on domain knowledge to bridges the gap caused by data scarcity.

Contrary, unsupervised machine learning which uses window size, the hybrid approach used rules to extract modifiers of the ambiguous word and consider them as contexts. These modifiers are therefore identified according to the developed rule planted. In the hybrid approach, the context extraction process involves finding the distribution of all possible modifiers of the ambiguous words. Differently, the window which varies window sizes to find the contexts, in the case of rule based it is limited to the modifiers preceding ambiguous word. In this case, the modifiers aren't varying the sizes as window size.

The advantages of this approach is that it is easy to incorporate the linguistic knowledge [40]. In this research, we try to answer "Does the addition of linguistic knowledge, improve a word sense disambiguation performance for Afan Oromo ambiguous word?". As the conducted experiment shows the obtained result was more robust than unsupervised machine learning. In this case, with worldly knowledge and reasoning, it is necessary to assist disambiguation. [22] certain knowledge sources provide evidence as to the word sense used or disallowed by a particular context, but most of them use linguistic knowledge. [87]the hybrid define a contextual representation is a characterization of the linguistic contexts in which a word appears.

However, in this work the rule based approach does not independently used to disambiguate, rather it is combined with unsupervised machine learning, just to mine the contexts of the ambiguous word. The modifiers (Section 4.7.1.1) are identified accordingly the rule planted to it. Similar to unsupervised machine learning, we have used the same test set in the hybrid approach. The following sections show the experiment taken place by hybrid approach.

5.3.1. Effect of Hybrid Approach on Accuracy of Disambiguation

In this hybrid experiment, most of the tested ambiguous words have relatively higher performance when compared with machine learning. As indicated in table 5.5 below, the hybrid approach result the accuracy of nearly 73.9% on the nine of the test set and achieved poor performance on the remaining of the test set. The reason behind the enhanced accuracy might be that, due to the nature of unsupervised machine learning trained from free text corpus and hybrid method mix the advantages from both methods especially from the rules. From defined number of senses by experts for each term, the accuracy of disambiguation was very encouraging.

Ambiguous Word	Accuracy (%)				
	Single Link	**Complete Link**	**Average Link**	**K-Means**	**EM**
Afaan	50%	50%	50%	50%	50%
Bahe	80%	60%	80%	80%	80%
Boqote	50%	50%	50%	100%	100%
Darbe	75%	75%	75%	75%	75%
Diige	66.67%	66.67%	66.67%	66.67%	66.67%
Dubbatate	50%	50%	50%	100%	100%
Tume	66.67%	66.67%	66.67%	66.67	66.67%
Haare	50%	50%	50%	50%	50%
Handhuura	66.67%	66.67%	66.67%	50%	100%
Ija	60%	60%	60%	80%	80%
Ji’a	50%	50%	50%	100%	50%
Lookoo	50%	50%	50%	50%	100%
Dhahe	66.67%	66.67%	66.67%	66.67%	66.67%
Mirga	66.67%	66.67%	66.67%	66.67%	66.67%
Waraabuu	66.67%	66.67%	66.67%	66.67%	66.67%

Table 5.5. Disambiguation using evidence from Hybrid Approach

As can be shown on table 5.5, an average accuracy of test set was 65.5% which is encouraging than the unsupervised machine learning work. Table 5.5, shows the accuracies for each senses. We calculated the accuracy for all such tests, which yielded disambiguation accuracy. From this

experiment table 5.5, hybrid approach finding the right number of clusters is equivalent to finding the number of senses.

No	Clustering Algorithms	Accuracy (%)
1	Single Link	61%
2	Complete Link	59.7%
3	Average Link	61%
4	K-Means	71.2%
5	EM	74.6%

Table 5.6. The Accuracy of the Hybrid Approach

The accuracy of any clustering algorithm depends on the quality of the produced clusters. A good clustering algorithm produces high-quality clusters such that inter-cluster similarity is minimized and intra-cluster similarity is maximized as shown on table 5.6. A perfect clustering solution was the one that leads to clusters that contain right contexts.

5.3.2. Summary of the Hybrid Approach

Name of the Algorithm	No of Clusters	Cluster Instances	No of Iterations	Within clusters sum of squared errors	Time taken to build model	Log likelihood	Unclustered Instances
Single Link	28	0: 50% 1: 50%	-	-	0	-	0
Complete Link	27	0: 50% 1: 50%	-	-	0	-	0
Average Link	28	0: 50% 1: 50%	-	-	0	-	0
K-Means	31	0: 75% 1: 25%	2	2.0	0	-	0
EM	32	0: 85% 1: 15%	-	-	0.04	50.1999	0

Table 5.7. Summary of Hybrid Approach Clustering Algorithms

The way of using clustering algorithms in this hybrid approach also similar to the algorithms that are used in unsupervised machine learning. Likewise, the manner of these algorithms used, determined and evaluated also similar but their performance is different. To evaluate the accuracy, we consider the basis of number of clusters produced and correctness of the produced clusters. Additionally, the time (rate of convergence) and the number of iterations were also the other evaluations (as shown on table 5.7).

Based on the experiment on table 5.5 and table 5.6, from the total five clusters, the EM and K-Means clusters which yield significantly higher accuracy than other clusters when it compared with machine learning result (as shown on table 5.2 and table 5.3). EM is the best-suited algorithm for the given datasets, since it depends on the probability distributions where each distribution represents a cluster. As presented in the above Section, if the log likelihood probability was maximum shows that high accuracy. Similarly, K-Means clustering performance was also better performance. However; Single, complete and average link clusters result not much surprise when compared with EM and K-Means clusters. Therefore; for WSD to Afan Oromo, EM and K-means enhance the accuracy of sense clustering than hierarchical single, complete and average clustering algorithms.

5.4. Discussion

In this paper, we reported unsupervised machine learning and hybrid approach for Afan Oromo WSD. We described the experiment we conducted to compare the performance of the two approaches. These approaches were evaluated with a benchmark test set (by experts), to facilitate the comparison with the results of both approaches. The accuracy results of our method, when compare machine learning and hybrid results (table 5.2 and table 5.5) are promising and proving method respectively[22].

In the machine learning experiment, the system yields an accuracy of 56.2% for all senses as well as in the hybrid experiment, the system yields an accuracy of 65.5%. Looking at the smallest dataset available to both machine learning and hybrid, they found an average figure of different yielded results. Initially, we evaluated our WSD method with all the 15 words. This lead, to a total of 15 words tested in this evaluation, and these ambiguous words have two senses to five senses. Six terms have two senses (the terms with two senses are *afaan, boqote, dubbatate, haare, ji'a, lookoo*), and six terms have three senses (the terms with three senses are *diige, tume, handhuura, dhahe, mirga,*

waraabuu) and two terms have five senses (the terms with five senses are *bahe, ija*) and the left one has four senses (the term with four senses is *darbe*) out of 15 ambiguous terms [94]. The table 5.2 and table 5.5 show the number of senses in detail.

The hybrid approach is promising result as compared with the performance of independent machine learning approach or independent rule based approaches to WSD task (as shown in table 5.2 and table 5.5). The rules were helpful to extract contexts overcoming the shortage of training data. This rule is beneficial for WSD, especially the integration of deep linguistic knowledge with algorithms markedly improves disambiguation accuracy. We argue that using hybrid approach to find senses, significantly increase disambiguation performance. This result also seems to support the result obtained by [88]. In this study we claim that using either a machine learning or hybrid approach (as discussed in Section 4.6 and Section 4.7 respectively) possible to find the sense of any ambiguous word in a given contexts but the hybrid approach was recommended when there are scarcity of training data and the accuracy needed [36].

The result found that using a window size of ±2 words either side of the target word offered the accuracy of disambiguation than using the whole sentence. Therefore; smaller value of the window size, which leads to the proper choice of sense for the target word. Based on this result, we conclude that for Afan Oromo window 2 was recommended unlike Amharic language [45] which window size of 3 is recommended.

The conducted experiment shows that, the semantic has come to the conclusion that the meaning of words are closely connected to the statistics of word usage, which are working with window size and vectors value derived from event frequencies; that is, we are dealing with Vector Space Model (cosine similarity) and clustering (Euclidean distance) [89]. By using cosine similarity we include important semantic information in the purely statistical process of selecting the appropriate sense for a particular word. This benefits both unsupervised, hybrid approaches to WSD by increasing the chances of matching a particular contexts.

As shown in table 5.8, the result obtained by unsupervised machine learning and hybrid approach were different as the semantic information extracted by the algorithms is distinct from the rule. However, the inclusion of unsupervised machine learning only as features does not always improve performance. This is [89] that machine learning algorithms were a useful information source for disambiguation but that it not as robust as a linguistic (modifiers in this case). The most likely reason

for this is that our approach relies on automatically assigned immediately preceding words while machine learning are needs to left and right of unannotated data set. On the other hand, the machine learning is noisy while the rule is more reliable and prove to be a most useful linguistic knowledge for WSD.

As the conducted experiment showed, each clusters have a context group, where the sense of these context groups are hopefully different. The underlying assumption is that the senses found in similar contexts are similar meanings. Then, new occurrences of the context can be classified into the closest induced clusters (senses). All contexts of related senses are included in the clustering and thus performed over all the contexts in the sentences. The underlying hypothesis is that ambiguous word contexts clustering captures the reflected unity among the contexts and each cluster reveal possible relationships existing among these contexts (as seen in table 5.3 and table 5.6) [82].

As the evaluation of the system indicates that, unsupervised machine learning usually 70% on the test, while the hybrid approach accuracy 81%. The below table 5.8 contains the evaluation performance of the WSD:

No	**Unsupervised Machine Learning**		**Hybrid Approach**	
	Precision	Recall	Precision	Recall
Correctly Disambiguated Senses	70%	46%	81%	60%

Table 5.8. Evaluation of WSD

From Table 5.8, the hybrid approach WSD is an encouraging result than unsupervised machine learning approach. This is due to unsupervised approach is not as hybrid approach, especially hierarchical clustering result was noisy. As already discussed before, the obtained result in both approach were different. Therefore, the linguistic knowledge (hybrid approach) the best approach to solve WSD than machine learning algorithms in Afan Oromo [64] as shown in the experiment

(Table 5.8). However, the overall system performance gained thus far is not surprusing since this language was under resourced materials and tools.

From the finding of this experiment the addition of deep linguistic knowledge to a WSD system is a significant rise in disambiguation accuracy and coverage as compared with results discussed so far. It is especially interesting that using the preceding modifiers of the ambiguous word perform better result. We can conclude that modifiers contain a lot of valuable clues for disambiguation [72].

The WSD developed for Afan Oromo has its own strength and weakeness sides. As the result showed that, the experiment attempts to disambiguate any ambiguous words if it running in corpus rather than limiting itself to treating a restricted ambiguous words. This is one of the strongest sides of this WSD. It is argued that this approach is more likely to assist the creation of practical systems. This system has the first work, that integerated different algorithms to find the appropriate sense of ambiguous word in Afan Oromo.

However, there is some ambiguous word on which the performance of our approach is actually low (as shown in Table 5.2 and Table 5.5). The system reported that the vector space model was affected by the data sparsity. The frequency of co-occurrence of most context words are zero due to the limited corpus size of the language. This result affects our cosine similarity values (as shown in Table 5.10 below). The other weakness of the system is context clustering in hierarchical clustering which was a noisy and yielded low performance as compared to K-Means and EM clustering (as shown in Table 5.3 and Table 5.6).

Walk-through Using an Example

A. Machine Learning

In this section, a walk-through of the experiment is described using an example in the following sections. As described in Section 5.2.1, the contexts are identified using window sizes. For instance, assuming the ambiguity word is *bahe* the unsupervised machine learning WSD, extracts the following contexts words.

No	Contexts
1	Bilisa
2	Ragaa
3	Dhugaa
4	Qabsoo
5	Tabba
6	Gaara
7	Uccuu
8	Duute
9	Daara
10	Lubbuu

Table 5.9. Extracted Context Using Window Size

As already mentioned, we take a property of vector space model that the values of the contexts in a vector space derived from event frequencies. We constructed a vector for each context and calculate the angle between these contexts using cosine similarity measure. The entry in the table 5.10 below shows the cosine similarity of the contexts.

	Bilisa	Qabsoo	Gaara	Tabba	Daara	Uccuu	Duute	Lubbuu	Ragaa	Dhugaa
Bilisa	**1**	**0.98**	0.09	0.05	0.01	0.15	0.04	0.17	0.07	0.19
Qabsoo	**0.99**	**1**	0.09	0.015	0.063	0.14	0.02	0.246	0.111	0.25
Gaara	0.07	0.089	**1**	**0.98**	0.062	0.027	0.004	0.20	0.11	0.10
Tabba	0.09	0.07	**0.97**	**1**	0.05	0.06	0.04	0.09	0.06	0.12
Daara	0.40	0.12	0.22	0.46	**1**	**0.98**	0.0	0.0	0.014	0.016
Uccuu	0.52	0.14	0.34	0.63	**0.98**	**1**	0.0	0.0	0.031	0.013
Duute	0.41	0.08	0.22	0.41	0.0	0.0	**1**	**0.91**	0.0	0.038
Lubbuu	0.248	0.247	0.11	0.20	0.033	0.08	**0.94**	**1**	0.066	0.27
Ragaa	0.43	0.14	0.287	0.49	0.083	0.009	0.0	0.12	**1**	**0.98**
Dhugaa	0.41	0.20	0.22	0.36	0.07	0.08	0.05	0.28	**0.95**	**1**

Table 5.10. Cosine Similarity Measure Representations

For the clustering result, we have loaded this cosine values in CSV format into the Weka tool that is shown in the Table 5.10. If the cosine value is not in CSV format we need to be converting it to .CSV (Comma-Separated Values) file. The reason for the format change is the incompatibility of the tool and to use in the further steps. Weka tool present with visual dendrogram graphically and non-hierarchical clustering algorithms.

From the total contexts, the Weka provided five clusters for the ambiguity word *bahe*. Out of the total, it correctly clustered three senses. While the rest two pairs of cluster were incorrectly clustered with different senses. Based on the experiment shows that the ambiguity word *bahe* with given contexts has the following senses; the first cluster include *bilisa and qabsoo* at dissimilarity 1.15, the second cluster include *gaara* and *tabba* at dissimilarity 1.23 and the third cluster include *daara* and *uccuu* at dissimilarity 1.42, the two wrongly clustered contexts are *dhugaa* with *lubbuu*, and *ragaa* with *duute*. But it should cluster *dhugaa* with *ragaa*, *lubbuu* with *duute* to give the senses of witness and death/pass respectively as experts evaluated.

One thing that is clear from the experiment is that the senses are clustered in 0 - 4 clusters (5 clusters) where cluster 0 is the pair of contexts which are *bilisa* and *qabsoo* clustered and make the sense of *got freedom,* cluster 1 is the pair of contexts *gaara* and *tabba* are grouped to make the sense of *highland* and cluster 2 is *daara* and *uccuu* are merged to make the sense of *cloth,* the other two clusters: cluster 3 and cluster 4 are incorrectly clustered and cannot make a sense.

The above experiment was summarized as the following:

a. Cluster 0: this set of senses comprises the contexts which have highly similar. The two contexts merge together under this cluster are the context words *bilisa* and *qabsoo,* this context has similar sense which shows freedom or political releases.
b. Cluster 1: this set of senses comprises the context words, namely *gaara* and *tabba* which are clustered together to make the sense of highland or topography
c. Cluster 2: this set of senses comprises the context words, namely *daara* and *uccuu* which are clustered to show the similar sense which is the cloth
d. Cluster 3: this set of senses comprises the context words, namely *dhugaa* and *lubbuu,* which are incorrectly clustered together to make inappropriate sense.
e. Cluster 4: this set of senses comprises the context words, namely *ragaa* and *duute,* which are incorrectly clustered together to make the sense.

The figure 5.2 below dendrogram shows the more description of these experiments:

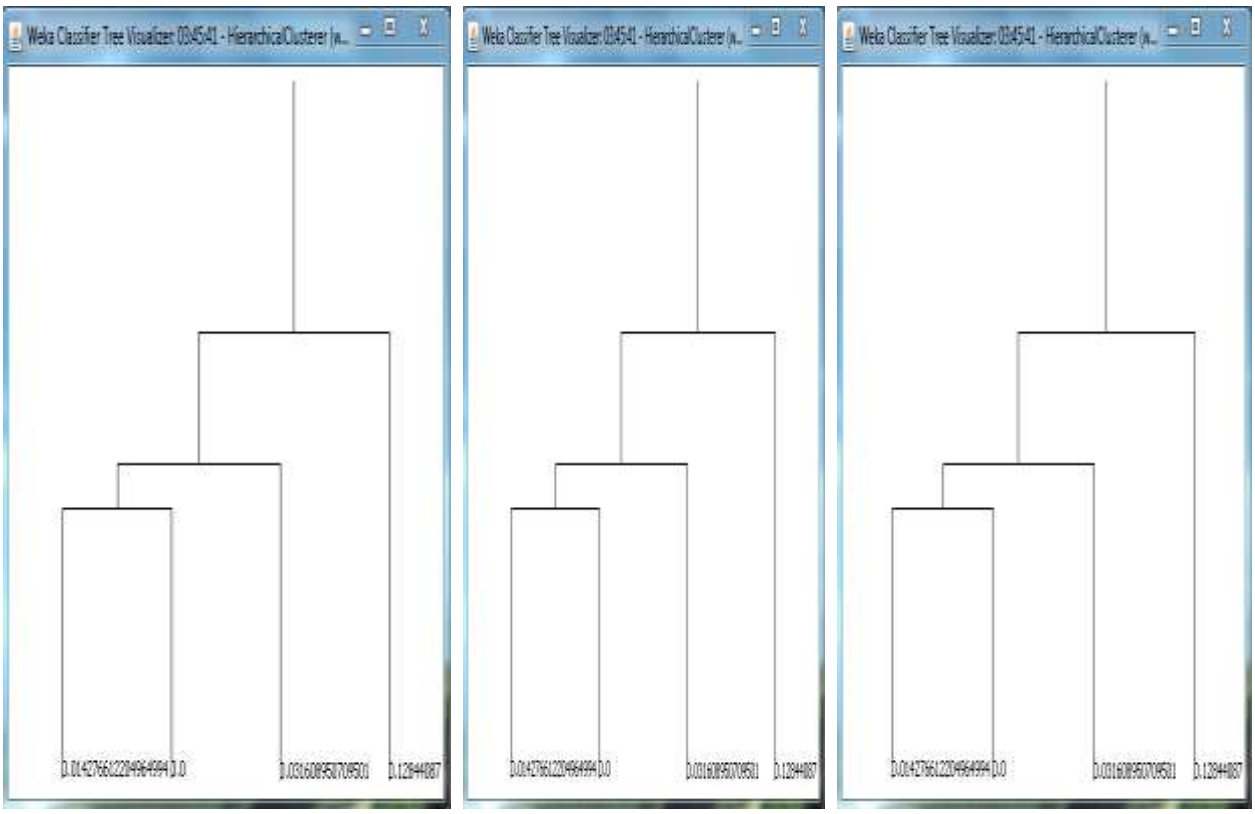


Figure 5.2. Single-Link(left), Complete-Link(Middle) and Average-Link(right) Dendrogram

The result obtained by the three hierarchical clustering are similar from the above dendrogram (as shown in Figure 5.2). The first step in the clustering process is to look for the pair of contexts that are the most similar, that are the closest in the sense of having the lowest dissimilarity this is the pair assume *bilisa* and *qabsoo*. These two samples are then joined at a level in the first step of the dendrogram, or clustering tree.

As figure 5.2 indicates that, in the hierarchical clustering experiment individual senses are leaf nodes in a binary tree of clusters, and are treated as singleton clusters. Any intermediate node is the centroid of its two children,which are more similar to each other than any other cluster as well as the root of the tree contains all clusters and therefore contains all senses.

It appears from this dendrogram that, the data can be represented by three clusters. However, as the number of cases increases, it may not be obvious. Indeed, one of the biggest problems with this cluster analysis is identifying the optimum number of clusters. As the fusion process continues increasingly dissimilar clusters must be fused. Deciding upon the optimum number of clusters is

largely subjective, although looking at a graph of the level of similarity at fusion versus a number of clusters may help.

As evident from the visualization Figure 5.3 below, the output has been classified into 5 clusters using K-Means and EM clustering. The algorithm took 2 iterations to complete and reach the result. The distortion (average within the cluster sum of squared errors) amongst the clusters themselves is 7.66 units. EM is not nested (it means not hierarchical result). The result of this cluster result is written to a class indices. The values indicate the class indices, where a value '0' refers to the first cluster; a value of '1' refers to the second cluster. As the experiment shows that EM and K-Means produce better result, as compared with hierarchical clusters. In case of the quality of the clusters, EM and K-Means are relatively good quality as compared with others. Our result demonstrate that our criteria match the predictability presented by experts, indicating there is no single perfect cluster criterion. Instead, it is necessary to select the promising criterion to match a human subject's generalization needs.

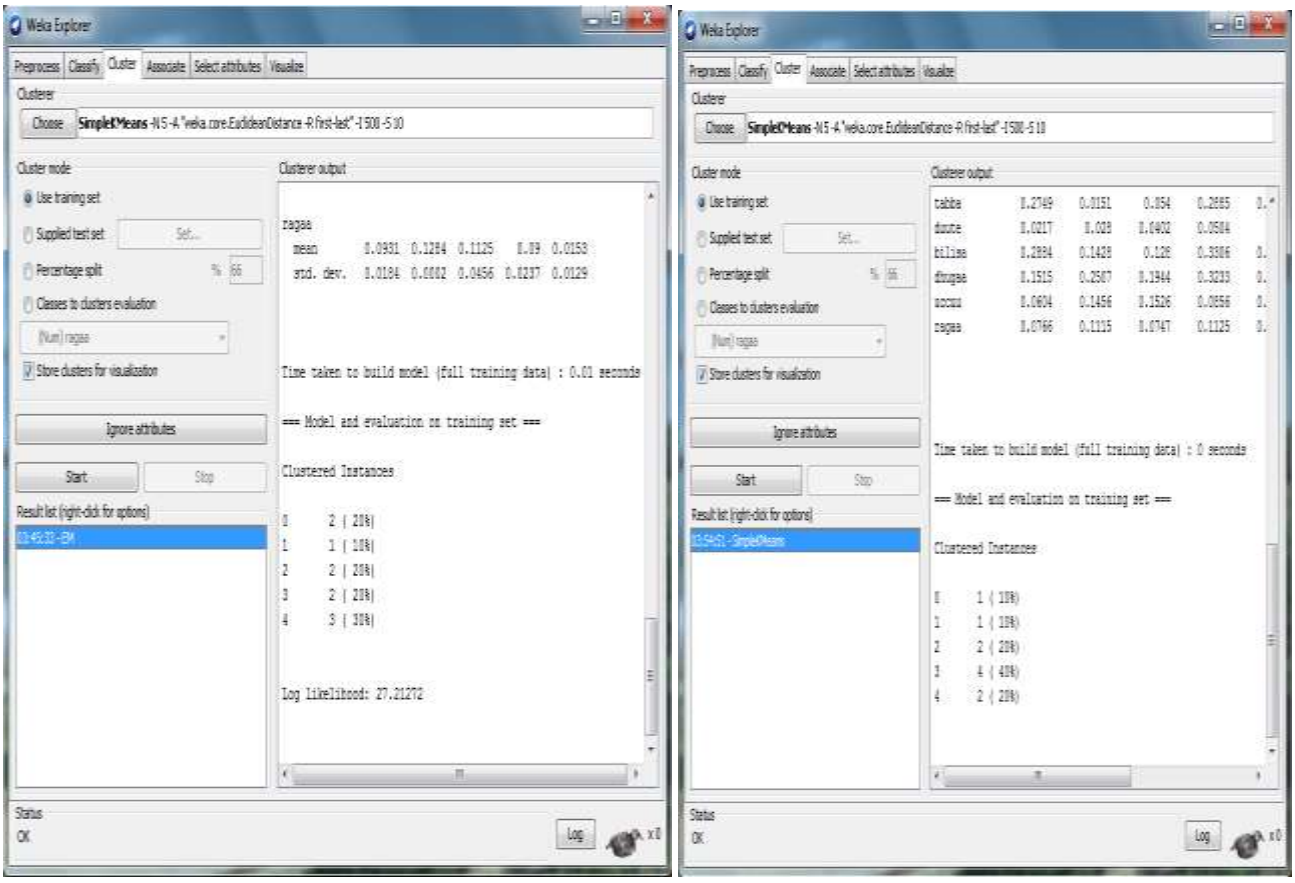


Figure 5.3. Expectation Maximization and K-Means Clustering

B. Hybrid Approach

No	The Modifiers
1	Bilisa
2	Qabsoo
3	Tabba
4	Gaara
5	Dhugaa
6	Ragaa

Table 5.11. Extracted Modifiers for ambiguity word *bahe*

As can be seen from table 5.11, the modifiers identified accordingly the rule developed. Similar to machine learning, the identified modifiers with its cosine similarity in .CSV file loaded to Weka tool. The process of using clustering algorithms was similar to in unsupervised machine learning experiment. As the experiment shows that senses are similar between each other and are dissimilar to the senses belonging to different clusters. The clustering is which two or more senses are considered belong to the same cluster if it defines a concept common to all these senses.

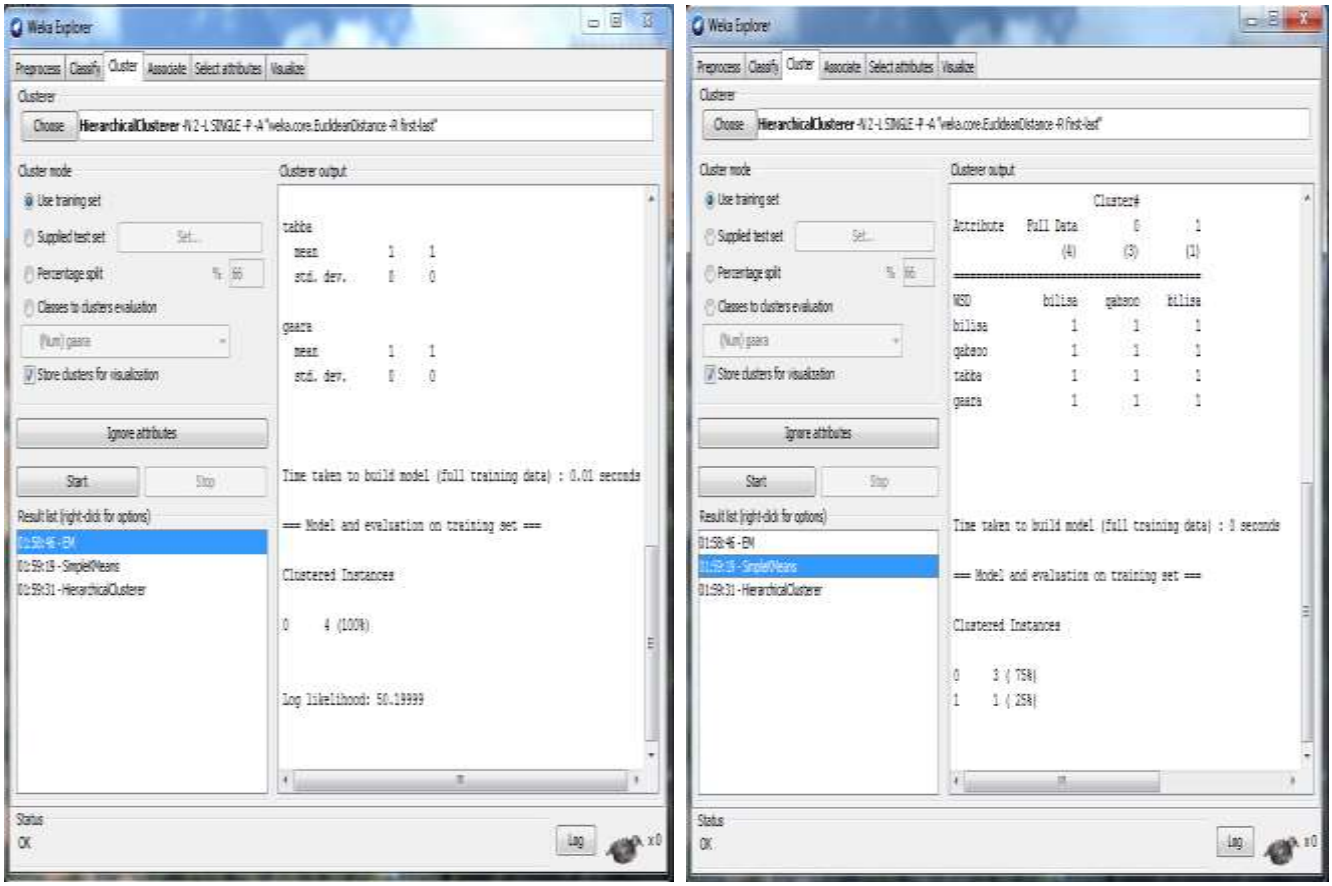


Figure 5.4. EM (left side) and K-Means (right side)

The other interesting way to examine the data in these cluster is inspect it visually. One of the options from this pop-up menu is visualizer cluster assignments. A window pops up that the results and see them visually. This is shown in a chart how the clusters are grouped in terms of instances. Also, turn up the “Jitter” which is artificially scatter the plot points to allow to see more easily.

As a general, the WSD performance was promising result as it is hybrid approach which compiled unsupervised machine learning and rule based approaches. The hybrid approach was very important for WSD system especially for under resourced language like Afan Oromo. In this work, it has essential role in this study in terms of accuracy and data scarcity. Similarly, it has a strict sense of well-formedness in mind and imposes linguistic constraints to satisfy well formedness. It relies on hand-constructed rules that are to be acquired from language specialists rather than automatically trained from data [64]. Based on the experiment, the possible senses of the test ambiguous word according to the given contexts has the following senses on table 5.12.

No	Ambiguous Words	No of Senses	List of Senses	Sense
1	Afaan	2	Sense 1	Language
			Sense 2	Mouth
2	Bahe	5	Sense 1	Freedom
			Sense 2	Highland
			Sense 3	Cloth
			Sense 4	Witness
			Sense 5	Dead / Pass
3	Boqote	2	Sense 1	Break / Rest
			Sense 2	Died
4	Darbe	4	Sense 1	Cross
			Sense 2	Class to class pass
			Sense 3	Died
			Sense 4	Broadcast
5	Diige	3	Sense 1	Absence on Meeting
			Sense 2	Fence
			Sense 3	cancel to start new
6	Dubbatate	2	Sense 1	Struggle
			Sense 2	Wedding
7	Tume	3	Sense 1	Hit
			Sense 2	Make
			Sense 3	Contraceptive
8	Haare	2	Sense 1	Sad
			Sense 2	Burn
9	Handhuura	3	Sense 1	Center
			Sense 2	Gift
			Sense 3	Navel
10	Ija	5	Sense 1	Eye
			Sense 2	Fruit of Tree
11	Ji’a	2	Sense 1	Stars
			Sense 2	Month
12	Lookoo	2	Sense 1	Pretty
			Sense 2	Rope
13	Dhahe	3	Sense 1	Hit
			Sense 2	Follow
			Sense 3	Fail
14	Mirga	3	Sense 1	Direction
			Sense 2	Human right
			Sense 3	Brave
15	Waraabuu	3	Sense 1	Fetch
			Sense 2	Hyena
			Sense 3	Record

Table 5.12. Summary of the Test

Chapter Six

6. Conclusion and Recommendation

6.1. Conclusion

The study deals with Afan Oromo word sense disambiguation. As discussed in the previous chapters, in this research, hybrid word sense disambiguation was recommended for Afan Oromo and for under resourced languages in general. To this end, we have forwarded the conclusion and recommendation as presented in the following sections.

The goal of Information Retrieval is striving to answer users' requests correctly as intended by the user. The contribution of NLP in achieving such goal of Information Retrieval Systems has been clearly pointed out. Furthermore, it has been pointed out how NLP plays a significant role in enhancing the computer's capability to process texts. To that end, word sense disambiguation is one component of NLP contributing a lot to the effort of solving the problem of Information Retrieval Systems in answering users' requests by introducing semantics of a query term and index terms.

The overall focus of this research is to investigate Word Sense Disambiguation which addresses the problem of deciding the correct sense of an ambiguous word based on its surrounding context's and the modifiers. In this study, a hybrid machine learning approach which combines unsupervised machine learning and rule based methods. To this end, we relied on several techniques which include: N-gram models (N- windows of words) to extract context words, vector space model for measuring similarity between the context words and clustering algorithms (Single, Complete and Average clustering, EM and K-means clustering algorithms) to group related context words. This hybrid method, avoid the problem of the knowledge acquisition bottleneck, that is, lack of large-scale resources and annotated corpora. Our approach to WSD has been based on the idea that the semantics of the context words belonging in the same sense of a word will have similar neighboring words. The context is hence a source of information and is the only means to identify the meaning of an ambiguous word. In order to achieve this, the approach does not rely on labeled training text and doesn't make use of any expensive resources like dictionaries, thesauri, and WordNet [22].

For under resourced Ethiopian language like Afan Oromo the hybrid approach is recommended. Since here is no annotated corpus (even difficult to obtain electronic materials for such language), hybrid approach plays a great role to disambiguate. The hybrid approach relies on hand-constructed rules that are acquired from language specialists rather than automatically trained from data.

Prior to this research, for Afan Oromo, there is no work recommending standard optimal context window size, which refers to the number of surrounding words that are sufficient for extracting useful contexts. Based on this study, the window size of ± 2 on both sides of the ambiguous word was found to be enough for disambiguation. Similar to English, window of ± 2 is the standard window applicable for disambiguation [92] in Afan Oromo. The nearest words surrounding the ambiguous word give more information than words far from the ambiguous word and consequent surrounding words to the left and to the right provide any information for the purpose of disambiguation.

There are several types of clustering algorithms. In this paper we relied on hierarchical and probabilistic algorithms. A hierarchical clustering is based on the union between the two nearest clusters. The beginning condition is realized by setting every datum as a cluster. After a few iterations, it reaches the final clusters wanted. The probabilistic algorithms use around model matching using probabilities as opposed to distances to decide clusters. EM and K-Means is an example of this type of clustering algorithm [95].

We did experiments on five different clustering algorithms namely K-Means, EM, single, complete and average link. Based on the result of the experiment out of the five algorithms simple K-Means and EM algorithms are the best of all to identify the meaning of ambiguous word in a context. We believe that the observed poor performance of hierarchical agglomerative algorithms is because of the errors they make during early agglomeration. The superiority of partitional algorithm also suggests that partitional clustering algorithms are well-suited for obtaining clustering solutions due to not only their relatively low computational requirements, but also comparable or better performance.

We also presented an evaluation methodology for measuring the precision and recall of discovered senses. In our experiments, we proved that the hybrid approach outperformed the machine learning approach.

In this work the hybrid approach and unsupervised machine learning achieved an accuracy of 81% and 70% respectively. We found that better results were achieved using a combination of rule and unsupervised machine learning features commonly used for WSD. The results obtained were encouraging as there is lack of resource of the language because of shortage of dictionary, labelled corpus, thesaurus.

6.2. Recommendation

The following recommendations are forwarded based on our findings with regard to the developments of resources and future research directions to WSD for Afan Oromo:

- Researches in WSD for other languages use knowledge resources like WordNet, lexicon, machine readable dictionaries and machine translation software. In this study, we faced a significant challenge as Afan Oromo lacks those resources. Taking into account their contribution to WSD and other research concerned institutions should develop these resources.
- For other language a standard sense annotated data are available for WSD research and for testing a standardized WSD system. The researcher doesn't have such data for Afan Oromo language. So, there need to be an initiative to prepare the data for WSD research.
- For the under resourced Ethiopian language like Afan Oromo, the linguistic knowledge was made better result, it needs linguistic background knowledge of the language.
- All the result showed that the techniques are fairly successful and effective in the disambiguation task. Thus, more research work should be exerted to carry out further improvements on the performance of these techniques.
- The hybrid method has been used. Thus, more research work should be employed to carry out further improvements on the performance using Supervised, Semi-Supervised approaches.
- The size of the corpus used for this research was limited and doesn't has good quality. These limitations affected the accuracy of word sense disambiguation because if resources used are small, we may not be able to find all possible senses. Therefore, some work should be done with large and high quality corpus to minimize these problems.

Future research directions for WSD in Afan Oromo include:

- In addition to corpus based approach, there are also knowledge source based and hybrid approach (combination of knowledge source based and corpus based approach) which are used for WSD in another language and found a good result [93] like WordNet, annotate data. These approaches need to be investigated for Afan Oromo 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. This approach overcomes the main problems of supervision and the data scarcity problem, especially the lack of annotated data like Afan Oromo. For example, an evaluation of Yarowsky's bootstrapping algorithm leads to very high performance over 90% accuracy on a small-scale data set [22]. This approach overcomes the main problems of supervision and the data scarcity problem, especially the lack of annotated data like Afan Oromo.

References

1. *Salton G(1981) The Measurement of Term Importance in Automatic Indexing: In Journal of the American Society for Information Science, vol.32.*
2. *Ide, Nancy and Jean Véronis (1998)Word sense disambiguation: The state of the art, Computational Linguistics, vol.24.*
3. *Roberto Navigli (2009) Word Sense Disambiguation: A Survey, ACM Computing Surveys, USA, Washington Dc.*
4. *David Yarowsky and Radu Florian (2002)Evaluating sense disambiguation across diverse parameter spaces. Natural Language Engineering.*
5. *Mohd Shahid Husain and Mohd Rizwan Beg (2013) Word Sense Ambiguity: A Survey, Lucknow, India, Volume 02.*
6. *Harris(1998) Distributional structure: Word, Cambridge University Press, Cambridge, UK.*
7. *Getachew M.(2011) Parts of Speech Tagging for Afaan Oromo, M.Sc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia.*
8. *Karov, Y. and Shimon, E(1998), Similarity-based word sense disambiguation. computational Linguistics, Vol.24.*
9. *Abebe Keno(2002) Case Systems in Oromo, MA Thesis. Ababa University, Ethiopia.*
10. *Girma Debele (2014)Afan Oromo News Text Summarizer, Master's thesis, Pohang University of Science and Technology, Pohang, Korea.*
11. *Daniel Bekele(2011) Afaan Oromo Information Retrieval (CLIR): A Corpus Based Approach, M.Sc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia.*
12. *Agirre, E. and Martinez, D.(2000) Exploring automatic word sense disambiguation with decision lists and the web, Toronto, Ontario.*
13. *Nancy Ide and Jean Véronis, (2007)Word Sense Disambiguation Algorithms and Applications, Ney work, Spring, vol.33.*
14. *Bar-Hillel, Yehoshua (1960)The present status of automatic translation of languages. Advances in Computers, ed. by Franz Alt, New York: Academic Press.*
15. *Ravi Mante, Mahesh Kshirsagar and Dr. Prashant Chatur(2014) A Review Of Literature On Word Sense Disambiguation, PhD, Government college of engineering Amravati, Maharashtra.*

16. *Jurafsky, D., Martin, J.(2009) Speech and Language Processing (2nd Edition) Pearson Education.*
17. *Gale, W., Church, K., & Yarowsky, D. (1993)A method for disambiguating word senses in a large corpus, Computers and the Humanities, vol.26.*
18. *Doina Tatar, Gabriela Serba(2001) A New Algorithm for Word Sense Disambiguation, Studia Universitatis "Babes-Bolyai", Seria Informatica, Vol.16.*
19. *Artstein, R. and Poesio, M.(2008) Inter-coder agreement for computational linguistics. Computational Linguistics, Vol 34.*
20. *Lesk, M. (1986) Automatic sense disambiguation using machine readable dictionaries, Toronto, Ontario.*
21. *Smeaton, A.F(1995) Linguistic Approaches to Text Management: An Appraisal of Progress, Journal of Document & Text Management, Vol.2.*
22. *David Yarowsky(1994)Decision lists for lexical ambiguity resolution: application to accent restoration in Spanish and French, USA, Stroudsburg.*
23. *Marco Baroni and Gemma Boleda(2009) Distributional Semantics: Natural Language Processing.*
24. *Scott McDonald(2000)Environmental Determinants of Lexical Processing Effort. Ph.D. thesis, University of Edinburgh.*
25. *H. Schutze(1998)Automatic word sense discrimination, Computational Linguistics, vol.24(1).*
26. *Jain, A. K., Murty, M. N., and Flynn, P. J. (1999), Data clustering: a review, ACM Computing Surveys, Vol. 31.*
27. *Nagy, G(1968) State of the art in pattern recognition, Cambridge University Press, Cambridge, UK.*
28. *Sneath, P. H. A. and Sokal, R. R(1973) Numerical Taxonomy: The Principles and Practice of Numerical Classification, London, UK: Freeman.*
29. *King, B(1967) Step-wise clustering procedures, Journal of the American Statistical Association, Vol.69.*
30. *Jain, A. K. and Dubes (1988), R. C. Algorithms for Clustering Data, Prentice-Hall.*
31. *Han, J. and Kamber, M.(2001) Data Mining–Concepts and Techniques: Morgan Kaufmann.*
32. *Juanying Xie, Shuai Jiang, Weixin Xie, Xinbo Gao (2009) An Efficient Global K-means Clustering Algorithm, Vol.6.*

33. *M. N. Murty and A. K. Jain,(1999) "Data clustering: a review," ACM Computing Surveys, vol. 31.*
34. *A. Dempster, N. Laird, and D. Rubin(1977) Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Vol. 39.*
35. *Celeux, G. and Govaert, G. (1992) A classification EM algorithm for clustering and two stochastic versions. Computational statistics and data analysis, vol.14.*
36. *Abdel Monem, K. Shaalan, A. Rafea, H. Baraka (2008) Generating Arabic Text in Multilingual Speech-to-Speech Machine Translation Framework, Machine Translation,Springer,vol.20(4).*
37. *Amaal Saleh Hasan (2010) Word Sense Disambiguation and Semantics techniques, MSc thesis, Sultan Qaboos University.*
38. *Gang Li, Guangzeng Kou, Ercui Zhou, and Ling Zhang(2009) Symmetric Trends: Optimal Local Context Window in Chinese Word Sense Disambiguation, USA, Washington DC.*
39. *Yorick Wilks(1975) Formal semantics of natural language: Cambridge University Press, Cambridge, UK.*
40. *Gale, W., Church, K., & Yarowsky, D. (1993)A method for disambiguating word senses in a large corpus. Computers and the Humanities, vol.26(5).*
41. *Brown, P., S. Della Pietra, V. Della Pietra, and R. Mercer(1991) A statistical approach to sense disambiguation in machine translation, Pacific Grove CA.*
42. *Solomon A.(2011) Unsupervised Machine Learning Approach For Word Sense Disambiguation To Amharic Words, M.Sc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia.*
43. *Alok Ranjan Pal, Anirban Kundu, Abhay Singh, Raj Shekhar, Kunal Sinha(2013) A Hybrid Approach To Word Sense Disambiguation Combining Supervised And Unsupervised Learning, West Bengal, India.*
44. *Getahun Wassie (2014) A Word Sense Disambiguation Model for Amharic Words using Semi-Supervised Learning Paradigm, School of graduate studies, Addis Ababa University, Ethiopia.*
45. *Solomon A.(2011) Unsupervised Machine Learning Approach For Word Sense Disambiguation To Amharic Words, M.Sc. Thesis, Addis Ababa University, Addis Ababa, Ethiopia.*

46. *J. Sreedhar, S. Viswanadha Raju, A. Vinaya Babu, Amjan Shaik, P. Pavan Kumar(2012) Word Sense Disambiguation: An Empirical Survey, Hyderabad, India.*
47. *Shaikh Samiulla Zakirhussain(2013)Unsupervised Word Sense Disambiguation, Indian Institute of Technology, Bombay.*
48. *Warren Weaver(1955) Machine Translation of Languages, MIT Press, Cambridge.*
49. *Kaplan A(1955)An experimental study of ambiguity and context, Mechanical Translation, Vol.2.*
50. *Stephen Clark (2014)Vector Space Models of Lexical Meaning, Handbook of Contemporary Semantics, 2nd ed, edited by Shalom Lappin and Chris Fox.*
51. *Turney, Peter D. & Patrick Pantel (2010) From frequency to meaning: Vector space models of semantics, Journal of Artificial Intelligence Research, vol.37*
52. *Katrin Erk and Sebastian Padó(2008) A structured vector space model for word meaning in context. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, HI, USA.*
53. *Yarowsky, D.(2007) Unsupervised word sense disambiguation rivaling supervised methods, In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, Cambridge, M.A.*
54. *Schutze, H. (1992)Dimensions of meaning: In Supercomputing '92: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing. IEEE Computer Society Press, Los Alamitos, CA.*
55. *Hearst, Marti(1991) "Noun Homograph Disambiguation Using Local Context in Large Text Corpora," in Using Corpora, University of Waterloo, Waterloo, Ontario.*
56. *Dagan, Ido and Alon Itai(1994) Word Sense Disambiguation Using a Second Language Monolingual Corpus, Computational Linguistics, vol.20.*
57. *Stefan Thater, Hagen Fürstenau, and Manfred Pinkal(2011)Contextualizing semantic representations using syntactically enriched vector models: In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden.*
58. *Adam Kilgarriff(1997) I don't believe in word senses. Computers and Humanities, vol.31(2)*
59. *Honolulu(2008) A Structured Vector Space Model for Word Meaning in Context Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing.*
60. *Jeff Mitchell and Mirella Lapata(2008) Vector-based models of semantic composition: In Proceedings of ACL-08: HLT, Columbus, USA.*

61. *Katrin Erk and Sebastian Padó(2008) A structured vector space model for word meaning in context. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, HI, USA.*
62. *Joseph Reisinger and Raymond J Mooney(2010)Multi-prototype vector-space models of word meaning, In Proceedings of HLT-NAACL.*
63. *Pedersen B. S.(1997) Lexical ambiguity in machine translation: expressing regularities in the polysemy of Danish Motion Verbs. PhD Thesis,Center for Sprogteknologi,Copenhagen.*
64. *Debela Tesfaye(2011)A Rule-based Afaan Oromo Grammar Checker, Addis Ababa University, Addis Ababa, Ethiopia.*
65. *Francisco Oliveira, Fai Wong, Yiping Li, Jie Zheng(2005) Unsupervised Word Sense Disambiguation and Rules Extraction using non-aligned bilingual corpus, MSc thesis, Tsinghua University, Beijing.*
66. *Ide, N., & Veronis, J. (1998) Introduction to the special issue on word sense disambiguation: the state of the art. Comput. Linguist., vol.24(1).*
67. *Specia L.,Oliveira-Neto S. F., Nunes M.G.V.,Stevenson, M. (2005) An automatic approach to create a sense tagged corpus for word sense disambiguation in machine translation:To be published in Proceedings of the 2nd Workshop of the Meaning Project, Trento.*
68. *Tullu Guya(2003) CaasLuga Afaan Oromoo: Jildii-1,Gumii Qormaata Afaan Oromootiin Komishinii "Aadaa fi Turizimii Oromiyaa", Finfinnee.*
69. *Abera N. (1988) Long vowels in Afan Oromo: A generic approach, Master's thesis, School of graduate studies, Addis Ababa University, Ethiopia.*
70. *Debela Tesfaye (2010) Designing a Stemmer for Afan Oromo Text: A hybrid approach, Master's thesis, School of graduate studies, Addis Ababa University, Ethiopia.*
https://om.wikipedia.org/wiki/Oromoo Accessed at 1/1/2015
71. *Gragg and Gene B.(1996)Oromo of Wollega: non-semetic languages of Ethiopia , East lansing, Michigan state university press.*
72. *Tilahun Gamta(1995) Seera Afaan Oromoo, Finfinnee, Boolee Press.*
73. *Kula Kekeba Tune, Vasudeva Varma and Prasad Pingali, (2007) "Evaluation of Oromo English Cross-Language Information Retrieval", IJCAI 2007 Workshop on CLIA, Hyderabad, India.*
74. *Wakshum Mekonnen (2000) Development of Stemming Algorithm for Oromo Texts. MA Thesis.*

75. *Tilahun Gamta(1989)Oromo-English Dictionary: Addis Ababa. University Press.*
76. *Getachew Rabirra(2014) Furtuu: Seerluga Afaan Oromoo, Finfinnee Oromiyaa press.*
77. *Atelach Alemu Argaw and Lars Asker(2010) An Amharic Stemmer: Reducing Words to their Citation Forms, M.S Thesis, Department of Computer and Systems Sciences, Stockholm University, Sweden.*
78. *Rada R., Mili H., Bicknell E. and Blettner M.(2009) Development an Application of a Metric on Semantic Nets, in IEEE Transactions on Systems,Man and Cybernetics, vol. 19.*
79. *Geoffrey Leech(1991)The state of the art in corpus linguistics, in English Corpus*
80. *Diriba Megersa(2002)An Automatic Sentence Parser For Oromo Language Using Supervised Learning Technique, M.Sc. Thesis, Addis Ababa University, Ethiopia.*
81. *Curran, James R. (2004) From Distributional to Semantic Similarity, Ph.D. thesis, University of Edinburgh.*
82. *Fei Shao, Yanjiao Cao(2005) A New Real-time Clustering Algorithm, Department of Computer Science and Technology, Chongqing University of Technology Chongqing 400050, China.Linguistics:Linguistic Studies in Honour of Jan Svartvik, London, Longman.*
83. *Malmkjaar, Kirsten(1995)The linguistics Encyclopedia, New York: Routledge.*
84. *McCarthy D. and Carroll J.(2003) Disambiguating nouns, verbs and adjectives using automatically acquired selectional preferences, Computational Linguistics, Vol. 29.*
85. *Flickinger, D.(2015)Natural Language Engineering-Efficient Processing with HPSG: Methods, Systems, Evaluation. At: http://www.coli.uni-sb.de/nlesi/ Accessed at 3/15/2015.*
86. *Agirre, E. and Martinez, D.(2001)Exploring automatic word sense disambiguation with decision lists and the web: In Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content, London, Longman.*
87. *Jung-Wei Fan, MS1 and Carol Friedman(2008)Word Sense Disambiguation via Semantic Type Classification,http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2655949/ Accessed at 1/8/2015*
88. *Rada Mihalcea, Ted Pedersen(2005) Advances in Word Sense Disambiguation, Tutorial at ACL Conference.*
89. *Kaplan,A.(1955) An experimental study of ambiguity and context, Mechanical Translation, Vol 2.*
90. *Forster, R.(2006) Document clustering in large German corpora using natural language processing, Ph.D. dissertation, University of Zurich.*
91. *Slava Kisilevich, Florian Mansmann, Daniel Keim(2001) P-DBSCAN: A density based*

clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos, University of Konstanz.

92. *Tesmegen Negassa (1995) word formation: the structure of independent and dependent clauses in oromo, Addis Abeba, Bole.*
93. *George Miller.(1990) Wordnet: An on-line lexical database. International Journal of Lexicography, vol.3(4).*
94. *Ted Pedersen and Rebecca Bruce(1997)Distinguishing word senses in untagged text: In Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, New York, Routledge.*
95. *Rui Xu and Donald Wunsch (2005) Survey of Clustering Algorithms and Transactions On Neural Networks, Cambridge University Press, Cambridge, UK. Vol. 16.*

Appendix A: Question For Collection of Ambiguous Words

The questions are prepared and converted to Afaan Oromoo language to make suitable to collect the ambiguous words test from public.

Saala: _____ Umurii:_____

Sadarka Barumsa: kutaa 1-5:_____

kutaa 6-12:_____

Barataa Kolleejjii:_____

Barataa Yuuniversitii:_____

Barsiisaa:_____

Q/Bulaa:_____

Kan biraa:_____

1. Jechoota hiika lamaa ol qaban (ambiguous words) kan naannawaa keessanitti fayyadamtu naaf tarreessi?

Lakk	Jechoota	Baay'ina Hiikaa	Hiika
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			