Addis Ababa University

College of Natural Sciences

Afaan Oromo Word Sense Disambiguation Using WordNet

BIRHANE TESFAYE TILAHUN

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

Nov 2017

Addis Ababa University

College of Natural Sciences

BIRHANE TESFAYE TILAHUN

Advisors: *Yaregal Assabie(PhD)*
Dida Midekso (PhD)

This is to certify that the thesis prepared by *Birhane Tesfaye Tilahun*, titled: *Afaan Oromo Word Sense Disambiguation Using WordNet* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:	_____	_____	_____
Advisor:	_____	_____	_____
Examiner:	_____	_____	_____
Examiner:	_____	_____	_____

Abstract

All human languages have words that can mean different things in different contexts. In the natural language processing community, Word Sense Disambiguation (WSD) has been described as the task which selects the appropriate meaning (sense) to a given word in a text or discourse where this meaning is distinguishable from other senses potentially attributable to that word.

One of the several approaches proposed in the past is Michael Lesk's 1986 algorithm.

This algorithm is based on two assumptions. First, when two words are used in close proximity in a sentence, they must be talking of a related topic and second, if one sense each of the two words can be used to talk of the same topic, then their dictionary definitions must use some common words. For example, when the words "pine cone" occur together, they are talking of "evergreen trees", and indeed one meaning each of these two words has the words "evergreen" and "tree" in their definitions. Thus we can disambiguate neighboring words in a sentence by comparing their definitions and picking those senses whose definitions have the most number of common words. The main drawback of this algorithm is that dictionary definitions are often very short and just do not have enough words for this algorithm to work well. To overcome this problem Satanjeev Banerjee 2002 deal with this problem by adapting Lesk algorithm to the semantically organized lexical database called WordNet. Besides storing words and their meaning like a normal dictionary, WordNet also "connects" related words together.

To this end, we have developed a WSD system that identifies a sense of an Afaan Oromo ambiguous word by using information from Afaan Oromo WordNet. The system identifies the sense by checking different types of sense relationships between words that will help to identify the sense of a word, The conventional WordNet organizes nouns, verbs, adjectives and adverbs together into sets of synonyms called synsets each expressing a different concept. In contrast to the structure of conventional WordNet, we used a clue word based model of WordNet. The related words for each sense of a polysemy word are referred to as the clue words. These clue words are used to disambiguate the correct meaning of the polysemy word in the given context using knowledge based Word Sense Disambiguation (WSD) algorithms. The clue word can be a

noun, verb, adjective or adverb which can solve limitation of English WordNet which has limited number of cross pos relation(relation not between single part of speech).

The performance of the system is tested using 50 polysemy Afaan Oromo ambiguous words which are selected randomly. The performance of the WSD based on clue word based WordNet achieved 92%.

Keywords: word sense disambiguation, WordNet, clue word, sense relationships.

Dedication

This work is dedicated to my Grandmother, Tejitu chifera.

Acknowledgments

First of all I would like to praise and thank **GOD** for all his blessings and mercy . It is the Lord who makes everything possible for me.

My special gratitude go to my advisors Dr. Yaregal Assabie and Dr. Dida Midekso for all their guidance at every step of the way, for patiently listening to me for uncountable hours, for teaching me how to do research, for imparting so much valuable knowledge and for all their encouragement and words of kindness.

I would like to thank Dr. Amanuel Alemayehu who is Afaan Oromo Language expert for his guidance, invaluable ideas and comments on the development of Afaan Oromo WordNet.

I would like to thank all my friends, for their friendship, kindness and support during my stay. and I am very thankful for Eskedar Yirga.

I would like to thank the Gender Office of Addis Ababa University for giving me this scholarship chance to study my MSc. I very much appreciate the support provided by the Office of the Director, Department of Information technology and staffs from Oromia Revenue Authority.

Finally, I would like to thank my family for their precious love, their prayer, advise, encouragement and comprehensive support including taking care of my baby Hallelujah Henok. They were there with me throughout my thesis work. Thank you all.

Contents

List of Algorithms

List of Tables

List of Figures

List of Acronyms

BNC	British National Corpus
CIIL	Central Institute of Indian Languages
DSO	Defense Science Organization
EM	Expectation maximization
IE	Information extraction
IR	Information retrieval
kNN	k-nearest neighbor algorithm
LDOCE	Longman Dictionary of Contemporary English
MFS	Most Frequent Sense
MST	Maximum Spanning Tree
MT	Machine Translation
NLP	Natural Language Processing
NLTK	Natural Language Tool Kit
OALD	Oxford Advanced Learner's Dictionary of Current English
POS	Part-of-speech
SVM	Support Vector Machine
WSD	Word Sense Disambiguation
WSJ	Wall Street Journal

Chapter One :Introduction

1.1 Background

Natural language processing (NLP) is a field of Computer Science, Artificial Intelligence, and Computational Linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction. Many challenges in NLP involve natural language understanding, that is, enabling computers to derive meaning from human or natural language input, and others involve natural language generation [1].

A natural language refers to human languages (Amharic, Afaan Oromo, Tigrigna, English, Arabic, Chinese, etc.), as opposed to programming languages such as C++, Java, Pascal, etc. A natural language is represented using texts in spoken or written forms. The goal of NLP is to accomplish human-like language processing for various tasks and applications such as machine translation, information retrieval, question-answering, etc.

NLP is hard because of ambiguity and variability. The term ambiguity refers to a word, term, phrase or sentence that could mean several possible things. The term variability refers to lots of ways to express the same thing.

There are many commonly researched tasks in NLP. Disambiguation (ambiguity resolution) is one of those tasks which refers to the resolution of ambiguities that occur at different levels of linguistic analysis [1].

Word sense disambiguation (WSD) is an open problem of natural language processing . WSD is identifying which sense of a word (i.e. meaning) is used in a sentence, when the word has multiple meanings. This problem occurs at semantic level of linguistic analysis. The solution to this problem impacts other computer related writing, such as discourse, improving relevance of search engine, anaphora resolution, coherence, inference *etc.* [2].Many words have more than one meaning; we have to select the meaning which makes the most sense in the context. For this problem, we are typically given a list of words and associated word senses, e.g. from a dictionary or from an online resource such as Word Net. Most words in the natural languages are polysemous, that is, they have numerous meanings or sentences. Afaan Oromo language has many words that have multiple meanings.

For instance, Afaan Oromo word dhugaa has two different meaning ‘drink’ vs ‘truth’.

A human can easily understand which sense of “dhugaa” is intended in a sentence. It would also be useful if software could also detect which sense of “dhugaa” was intended.

Since the 1950s, many approaches have been proposed for assigning senses to words in a context. The three main approaches applied in the area of WSD field are knowledge-based approach, corpus based approach and hybrid approach which is a combination of the two. Knowledge based approach uses information provided by machine readable dictionaries (MRD), corpus based approach uses information gathered from training corpus and hybrid approach combines aspects of the two methodologies [3]. In this study, knowledge-based approach will be used to develop Afaan Oromo word sense disambiguation.

1.2 Motivation

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

1.3 Statement of the Problem

We need Afaan Oromo WSD for various applications such as machine translation, information extraction, question answering, information retrieval, text classification, text summarization, speech processing, text processing, grammatical analysis, content and thematic analysis and so on. The absence of efficient automatic WSD would make the development of those listed applications too difficult. Few attempts have been made towards the development of Afaan Oromo WSD. Tesfa Kebede [5] employed supervised machine-learning approach for Afaan Oromo WSD. The author used manually annotated training data containing instances of a target word to learn the context in which target words are used, Workineh Tesema [6] employed unsupervised machine learning approach for Afaan Oromo language and unannotated training data containing instances of the target word was used. However, the previous works [5],[6] have

limitations. The work of Tesfa Kebede[5] was limited to five ambiguous words only and available sense-annotated corpora are large and requires manually labeled sense examples, which is time taking and cumbersome when the number of corpus size increased and cluster the contexts of an ambiguous word into a number of groups. Workineh Tesema [6] used unannotated training data containing training instances whose concepts are not known. There was no need of any sense inventory and sense annotated corpora in this study. In previous studies, training instances of data are used in the form of manually annotated or unannotated training data that means researchers employed machine learning techniques.

However, there is still a gap to fill in this research area. The researchers have experimented only on a single word class which is using single part of speech for few words and training instance example of single target word gathered then the system performs disambiguation based on that training example. This disambiguation task is for single targeted word in input text. So to make WSD applicable there is a need to explore all words of Afaan Oromo. There is no work for Afaan Oromo language in developing word sense disambiguation that can disambiguate all words in a text. If the disambiguation can disambiguate each open class word (noun ,verb ,adverb and adjective) without restricting to single word class using sense from WordNet, it can generate appropriate sense of ambiguous word after identifying ambiguous word in an input text sentence.

In our study, knowledge based approach will be used. This method uses information extracted from structured data called a knowledge source which has information about a concept such as its definition or synonym and other relations to word like hypernym, meronym, clue noun, clue verb etc. In this selected approach, we will develop Afaan Oromo WordNet and use it as a source of information for disambiguation. This knowledge-based Afaan Oromo WSD method allows the system to disambiguate all words in a text rather than only sample words included in a training set.

1.4 Objectives

General Objective

The general objective of this study is to design and develop knowledge based Afaan Oromo Word Sense Disambiguation system .

Specific Objectives

The specific objectives of this study are:

- Conducting literature review and related works in Afaan Oromo and other languages to understand the approaches of WSD.
- Studying the ambiguous words of Afaan Oromo language and their contextual meaning for identifying sense of ambiguous words.
- Collecting data to develop WordNet for Afaan Oromo language.
- Designing architecture and algorithm for Afaan Oromo language to disambiguate words using WordNet.
- Developing a prototype of the system.
- Testing and evaluating the performance of the developed system.

1.5 Methods

Literature Review

Literature review will be done on different areas relevant to this work in order to understand ambiguities and Afaan Oromo language structures as well as approaches to word sense disambiguation. Specifically focus on reviewing literature for the techniques of word sense disambiguation approaches to develop Afaan Oromo WordNet Structure.

Data Collection

Afaan Oromo documents and information on Afaan Oromo ambiguous words will be collected from different websites, journals, magazines educational books and so on for understanding the characteristics of ambiguity. To develop Afaan Oromo WordNet, data will be collected from Afaan Oromo dictionary and homonyms words previously selected by linguistic experts.

Prototype Development

In order to evaluate the performance of the method a prototype system will be developed for the Afaan Oromo word sense disambiguation that can identify ambiguous word with its disambiguated sense. The performance of the prototype will be evaluated.

1.6 Scope and Limitations

The study aims to identifying ambiguous word and its senses from a manually developed Afaan Oromo WordNet.

The limitation of this study is that the system will not perform grammar and spelling correction. Furthermore, it does not work for speech data.

1.7 Application of Results

The main contribution of this research is on finding an efficient method of automatic Afaan Oromo WSD. Therefore, it will have a significant contribution for the development of a full-fledged automatic Afaan Oromo WSD. The output of this research work will also be a vital component for natural language applications that include Afaan Oromo language processing. This includes information retrieval, machine translation, question answering and information extraction.

1.8 Thesis Organization

The rest of this thesis is organized as follows. The second chapter is the literature review part. In this chapter, the conceptual review of WSD such as the overview of different methodological approaches and knowledge sources are presented. It also includes Afaan Oromo language which incorporates Afaan Oromo writing system, Afaan Oromo word classes and ambiguity in Afaan Oromo language. An overview of WordNet is also included in this chapter. The third chapter introduces review of related works in the area of WSD for Ethiopian languages as well as Non Ethiopian languages. Chapter four presents the design of the WSD system which embraces the architecture of the WSD, description of preparation of WordNet. Chapter five discusses the preparation of test dataset, experimental procedure, findings and challenges of the study in detail. The last chapter, chapter six, presents conclusion and future works.

Chapter Two: Literature Review

In this chapter literature in the field of word sense disambiguation (WSD) is reviewed. The chapter covers application of WSD, WSD tasks, knowledge sources, Word Net ,algorithms and discussion on major approaches that have been employed for WSD researches. Finally, Afaan Oromo language and Afaan Oromo ambiguity, especially semantic ambiguity is reviewed and discussed.

2.1 Word Sense Disambiguation

WSD is a potential intermediate task for many other NLP systems. The main field of application of WSD is machine translation, but it is used in almost about all kinds of linguistic researches. Among others, are discussed below [3],[7]:

• **Machine translation (MT):** This is the field in which the first attempts to perform WSD were carried out. WSD is required for machine translations as few words in every language have different translations based on the contexts of their use. There is no doubt that some kind of WSD is essential for the proper translation of polysemous words.

• **Information retrieval (IR):** WSD can be used in IR in order to discard occurrences of words in documents appearing with inappropriate senses. An accurate disambiguation of the document base, together with a possible disambiguation of the query words, would allow it to eliminate documents containing the same words used with different meanings (thus increasing precision) and to retrieve documents expressing the same meaning with different wordings (thus increasing recall).

• **Information extraction (IE) and text mining:** WSD plays an important role for information extraction in different research works, where it is interesting to distinguish between specific instances of concepts, as bioinformatics research, named entity recognition system, co-reference resolution etc.

• **Semantic parsing:** WSD can be applied in restricting the space of competing parses, especially, for the dependencies, such as prepositional phrases.

• **Speech synthesis and recognition**: WSD could be useful for the correct phonetisation of words in speech synthesis, and for word segmentation and homophone discrimination in speech recognition.

2.2 Word Sense Disambiguation Tasks

There are two variants of the generic WSD tasks[3]: the *lexical sample* task and the *all word* task. In the **lexical sample** task, a small pre-selected set of target words is chosen, along with an inventory of senses for each word from some lexicon. Since the set of words and the set of senses is small, supervised machine learning approaches are often used to handle lexical sample tasks. For each word, a number of corpus instances (context sentences) can be selected and hand-labeled with the correct sense of the target word in each. Classifier systems can then be trained using these labeled examples. Unlabeled target words in the context can then be labeled using such a trained classifier. Early work in word sense disambiguation focused solely on lexical sample tasks of this sort, building word-specific algorithms for disambiguating single words.

In contrast, in the **all-words** task systems are given entire texts and a lexicon with an inventory of senses for each entry, and are required to disambiguate every content word in the text. All words WSD systems are expected to disambiguate all open-class words in a text (i.e., nouns, verbs, adjectives, and adverbs). The all-words task is very similar to part-of-speech tagging, except with a much larger set of tags, since each lemma has its own set. A consequence of this larger set of tags is a serious data sparseness problem; there is unlikely to be adequate training data for every word in the test set. On the other hand, other approaches, such as knowledge- lean systems, rely on full-coverage knowledge resources, whose availability must be assured [7].

2.3 Knowledge sources

Knowledge is a fundamental component of WSD. Knowledge sources provide data which are essential to associate senses with words. They can vary from corpora of texts, either unlabeled or annotated with word senses, to machine-readable dictionaries, thesauri, glossaries, ontologies, etc.[7] .We can see the knowledge sources by classifying them into structured resources and unstructured resources.

2.3.1 Structured resources

According to Robert [7]and Nancy[8], structured resources include the following:

- ***Thesauri :*** provide information about relationships between words, like synonymy (e.g., car is a synonym of motorcar), antonym (representing opposite meanings, e.g., ugly is an antonym of beautiful). The most widely used thesaurus in the field of WSD is Roget's International Thesaurus[9]. The latest edition of the thesaurus contains 250,000 word entries organized in six classes and about 1000 categories.
- ***Machine-readable dictionaries (MRDs) :*** have become a popular source of knowledge for natural language processing since the 1980s, when the first dictionaries were made available in electronic format: among these Collins English Dictionary, the Oxford Advanced Learner's Dictionary of Current English, the Oxford Dictionary of English, and the Longman Dictionary of Contemporary English (LDOCE) and WordNet. WordNet is often considered one step beyond common MRDs, as it encodes a rich semantic network of concepts.
- ***Ontologies***: are specifications of conceptualizations of specific domains of interest[10], usually including taxonomy and a set of semantic relations. In this respect, WordNet and its extensions can be considered as ontologies, as well as the Omega Ontology, an effort to reorganize and conceptualize WordNet, the SUMO upper ontology, etc.

2.3.2 Unstructured resources

The unstructured knowledge resource is a corpus which is collections of texts used for learning language models. Corpora can be sense-annotated or raw (i.e., unlabeled). Both kinds of resources are used in WSD, and are most useful in supervised and unsupervised approaches, respectively[7].

2.4 WordNet

These days, the WordNet is becoming popular as a resource to be used in knowledge-based approach to disambiguate the meanings of polysemy words. WordNet is a lexical database developed at Princeton University for the English language[11]. The WordNet organizes nouns, verbs, adjectives and adverbs into the groups of synonyms and describes the relationships between these. WordNet [12] is a computational lexicon of English based on psycholinguistic

principles which encodes concepts in terms of sets of synonyms called *synsets*. Its latest version, WordNet 3.0, contains 147,478 words organized in over 117,792 synsets. For example, the concept of *automobile* is expressed with the following synset (recall superscript and subscript denote the word's sense identifier and part-of-speech tag, respectively): {*car* ^{1}n, *auto* ^{1}n, *automobile* ^{1}n, *machine* ^{4}n, *motorcar* ^{1}n}. Note that each word sense univocally identifies a single synset. For each synset**,** WordNet provides the following information:

- **A *gloss***, that is, a textual definition of the synset possibly with a set of usage examples (e.g., the gloss of *car* ^{1}n is "a 4-wheeled motor vehicle; usually propelled by an internal combustion engine; 'he needs a car to get to work' ").
- ***Lexical and semantic relations,*** which connect pairs of word senses and synsets, respectively.

2.4.1 Structure

The main relation among words in WordNet is synonymy, as between the words shut and close or car and automobile. Synonyms--words that denote the same concept and are interchangeable in many context are grouped into unordered sets (synsets). In English WordNet database, there are 18 tables [11]. The description of the tables is given below.

The Synset Table -The synsets table is one of the most important tables in the database. It is responsible for housing all the definitions within WordNet. Each row in the synset table has a synset id, a definition, a pos (parts of speech field) Each of WordNet's 117 000 synsets is linked to other synsets by means of a small number of "conceptual relations." Additionally, a synset contains a brief definition ("gloss") and, in most cases, one or more short sentences illustrating the use of the synset members. Word forms with several distinct meanings are represented in as many distinct synsets. Thus, each form-meaning pair in WordNet is unique.

The Words Table - WordNet also has a "words" table, that only has two fields: a word id, and a "lemma". The words table is responsible for housing all the lemmas (base words) within the WordNet database. There are 147,478 words in in the English WordNet database.

The Sense Table - The sense table is responsible for linking together words (in the words table), with definitions (in the synset table). The entries in the sense table are referred as "word-sense

pairs" - because each pairing of a wordid with a synset is one complete meaning of a word - a "sense of the word". There are a total of 207,235 word senses in the English WordNet database.

linktypes Table - defines all relation (link) types used in wordnet, about two dozen of them. Both lexlinks and semlinks tables use this table to define the type of each link. Some link types are marked as recursive, meaning that if "furniture" is, for example, a hypernim to a "chair", then a "chair" is a hyponym to "furniture".

Lexlinks Table - lexical links, i.e., relations between words. Example: sad - saddness (derivation)

semlinks Table - semantic links, i.e. relations between synsets. Example: chair - furniture (hypernym)

postypes Table - defines "parts of speech". Contains only the following values: n – noun, v – verb, a – adjective, r – adverb, s – adjective satellite.

Adjpositions –this table indicates the position of adjective class of word in the synset. Contains the following fields Synset id ,word id and position.

Adjpositiontypes - An adjective may be annotated with a syntactic marker indicating a limitation on the syntactic position .The adjective may have in relation to noun that it modifies. The syntactic markers are:

(p) predicate position

(a) prenominal (attributive) position

(ip) immediately postnominal position

Casedwords – table has three fields case word id, word id and cased. WordNet contains 408,50 cased words.

Lexdomains – this table has 45 lexical domains :- adjective has 3 lexical domains which are (all, pert, ppl); adverb has 1 lexical domain which is (all) ; noun has 26 lexical domains which are (tops, act, animal artifact, attribute, body, cognition, communication, event, feeling, food, group,. location, motive, object, person, phenomenon, plant, possession, process, quantity, linkdef, shape, state, substance, time); verb has 15 lexical domains which are (body, change,

cognition, communication, competition, consumption, contact, creation, emotion motion, perception, possession, social, stative, weather).

Morphmaps - only base forms of words are usually stored in WordNet, searches may be done on inflected forms. A set of morphology functions , morphy , is applied to the search string to generate a form that is present in WordNet. This table maps many forms of morphs to their root or base form.

Morphs –this table stores all morphemes types that indicate single base word. Example, gas gasses, gassed and gassing are morphs for single word gas.

Samples – this table has sample or example sentence for synsets.

Vframemaps –maps words and synsets to its verb frame.

Vframes - Each verb synset contains a list of generic sentence frames illustrating the types of simple sentences in which the verbs in the synset can be used.

Vframesentencemaps - maps words and synsets to its sample sentence.

Vframesentences – has sample verb frame sentences.

2.4.2 Relations

The most frequently encoded relation among synsets is the super-subordinate relation (also called hyperonymy, hyponymy or ISA relation) [12]. It links more general synsets like {furniture, piece_of_furniture} to increasingly specific ones like {bed} and {bunkbed}. Thus, WordNet states that the category furniture includes bed, which in turn includes bunkbed; conversely, concepts like bed and bunkbed make up the category furniture. All noun hierarchies ultimately go up the root node {entity}. Hyponymy relation is transitive: if an armchair is a kind of chair, and if a chair is a kind of furniture, then an armchair is a kind of furniture. WordNet distinguishes among types (common nouns) and instances (specific persons, countries and geographic entities). Thus, armchair is a type of chair, Barack Obama is an instance of a president. Instances are always leaf (terminal) nodes in their hierarchies.

Meronymy, the part-whole relation holds between synsets like {chair} and {back, backrest}, {seat} and {leg}. Parts are inherited from their superordinates: if a chair has legs, then an armchair has legs as well. Parts are not inherited “upward” as they may be characteristic only of

specific kinds of things rather than the class as a whole: chairs and kinds of chairs have legs, but not all kinds of furniture have legs.

Verb synsets are arranged into hierarchies as well. Verbs towards the bottom of the trees (troponyms) express increasingly specific manners characterizing an event, as in {communicate}-{talk}-{whisper}. The specific manner expressed depends on the semantic field; volume is just one dimension along which verbs can be elaborated. Others are speed (move-jog-run) or intensity of emotion (like-love-idolize). Verbs describing events that necessarily and unidirectional entail one another are linked: {buy}-{pay}, {succeed}-{try}, {show}-{see}, etc.

Adjectives are organized in terms of antonymy. Pairs of "direct" antonyms like wet-dry and young-old reflect the strong semantic contract of their members. Each of these polar adjectives in turn is linked to a number of "semantically similar" ones: dry is linked to parched, arid, dessicated and bone-dry and wet to soggy, waterlogged, etc. Semantically similar adjectives are "indirect antonyms" of the contral member of the opposite pole. Relational adjectives ("pertainyms") point to the nouns they are derived from (criminal-crime). There are only few adverbs in WordNet (hardly, mostly, really, etc.) as the majority of English adverbs are straight forwardly derived from adjectives via morphological affixation (surprisingly, strangely, etc.)

2.5 Cross-POS relations

The majority of the WordNet's relations connect words from the same part of speech (POS). Thus, WordNet really consists of four sub-nets, one each for nouns, verbs, adjectives and adverbs, with few cross-POS pointers. Cross-POS relations in English WordNet includes "morphosemantic" links and semantic role of the noun with respect to the verb.

Morphosemantic links: link between semantically similar words sharing a stem with the same meaning: observe (verb), observant (adjective) observation, observatory (nouns).

Semantic Role links: noun-verb pairs which indicate semantic role of the noun with respect to the verb: {sleeper, sleeping car} is the LOCATION for {sleep} and {painter}is the AGENT of {paint}, while {painting, picture} is RESULT of {paint}.

2.6 WSD Approaches

Word sense disambiguation approaches are classified into three categories which are knowledge based approach , corpus-based approach and hybrid approach. Corpus-based approach is further classified as supervised corpus-based approach and unsupervised corpus-based approach[3],[4].

2.6.1 Knowledge-based approach

Knowledge-based approaches are based on different knowledge sources as machine readable dictionaries or sense inventories, thesauri etc. WordNet is the mostly used machine readable dictionaries in research field[13]. Knowledge based methods for WSD are usually applicable to *all words* in unrestricted text, as opposed to corpus-based techniques, which are applicable only to those words for which annotated corpora are available. There are various algorithms to knowledge based WSD [13]. Generally, four main types of knowledge-based algorithms are : Lesk algorithm, Semantic Similarity, Selectional Preferences and Heuristic.

i. Lesk algorithm

The Lesk algorithm [14] is one of the first algorithms developed for the semantic disambiguation of all words in unrestricted text. The only resource required by the algorithm is a set of dictionary entries, one for each possible word sense, and knowledge about the immediate context where the sense disambiguation is performed. This is the first machine readable dictionary based algorithm built for word sense disambiguation. This algorithm depends on the overlap of the dictionary definitions of the words in a sentence. In this approach [14],[15] , first of all a short phrase (containing an ambiguous word) is selected from the sentence. Then, dictionary definitions (glosses) for the different senses of the ambiguous word and the other meaningful words present in the phrase are collected from an online dictionary. Next, all the glosses of the key word are compared with the glosses of other words. The sense for which the maximum number of overlaps occurs, represents the desired sense of the ambiguous word.

The main idea behind the original definition of the algorithm is to disambiguate words by finding the overlap among their sense definitions. Namely, given two words, $W1$ and $W2$, each with $NW1$ and $NW2$ senses defined in a dictionary, for each possible sense pair $W1$ i and $W2$ j, $i = 1..NW1$, $j = 1..NW2$, we first determine the overlap of the corresponding definitions by counting

the number of words they have in common. Next, the sense pair with the highest overlap is selected, and therefore a sense is assigned to each word in the initial word pair. Algorithm 2.1 illustrates the main steps of the algorithm[14].

```
for each sense i of W1
 for each sense j of W2
     compute Overlap(i,j) //the number of words in common
     between the definitions of sense i and sense j //
find i and j for which Overlap(i,j) is maximized
assign sense i to W1 and sense j to W2
```

Algorithm 2.1: Dictionary-based Lesk algorithm [14]

As an example, consider the task of disambiguating the words *pine* and *cone* in the word pair *pine cone.* The *Oxford Advanced Learner's Dictionary* gives four senses for *pine* and three senses for *cone*:

Pine

1 seven kinds of evergreen tree with needle-shaped leaves

2 pine

3 waste away through sorrow or illness

4 pine for something, pine to do something

cone

1 solid body which narrows to a point

2 something of this shape, whether solid or hollow

3* fruit of certain evergreen trees (fir, pine)

The first definition for *pine* and the third definition for *cone* have the maximum overlap among all possible sense combinations, with three words in common: *evergreen*, *tree*, and *pine*, and therefore these are the meanings selected by the Lesk algorithm for the given pair *pine cone.* The

Lesk algorithm was evaluated on a sample of ambiguous word pairs manually annotated with respect to the *Oxford Advanced Learner's Dictionary*; a precision of 50–70% was observed.

Since the original definition of the Lesk algorithm in 1986, a major problem with the original algorithm is the exponential growth of the search space when trying to disambiguate a pair for with more than two words. The following sentence: "I saw the man who is 98 years old and still can walk and tell jokes" contains nine words that have more than two meanings (information extracted from WordNet): see (26), man (11), year (4), old (8), can (5), still (4), walk (10), tell (8), joke (3). In total there are 43,929,600 possible combinations, thus finding the optimal combination is impractical and almost impossible as a result several variations of the algorithm have been proposed, Variations of the Lesk Algorithm are Simplified Lesk Algorithm, Simulated Annealing , corpus based Lesk algorithm and Adapted Lesk Algorithm .

The simplest version of the algorithm, often called the **Simplified Lesk** algorithm by Kilgarriff and Rosenzweig presented in Algorithm 2.2[16]:

The primary problem with either the original or simplified approaches, however, is that the dictionary entries for the target words are short, and may not provide enough chance of overlap with the context.

```
For each sense s of that word,
 Set weight(s) to zero.
Identify set of unique words W in surrounding sentence.
For each word w in W,
  For each sense s,
   If w occurs in the definition or example sentences of s,
        Add weight(w) to weight(s).
Choose sense with greatest weight(s)
```

Algorithm 2.2: Simplified Lesk algorithm [16]

• **Simulated Annealing**

The major problem with the original Lesk algorithm is that it leads to a combinatorial explosion when applied to the disambiguation of more than two words. As cited in [13] ,Cowie et al.(1992) proposed simulated annealing as a solution to this problem. Cowie defined a function *E* that reflects the combination of word senses in a given text, and whose minimum should correspond to the correct choice of word senses. For a given combination of senses, all corresponding definitions from a dictionary are collected, and each word appearing at least once in these definitions receives a score equal to its number of occurrences. Adding all these scores together gives the redundancy of the text. The *E* function is then defined as the inverse of redundancy, and the goal is to find a combination of senses that minimizes this function. To this end, an initial combination of senses is determined (e.g., pick the most frequent sense for each word), and then several iterations are performed, where the sense of a random word in the text is replaced with a different sense, and the new selection is considered as correct only if it reduces the value of the *E* function. The iterations stop when there is no change in the configuration of senses[13]. The algorithm shown in Algorithm 2.3.

```
Define a function E = combination of word senses in a given text.
Find the combination of senses that leads to highest definition
overlap (redundancy)
       Start with E = the most frequent sense for each word
          At each iteration, replace the sense of a random word
          in the set with a different sense, and measure E
       Stop iterating when there is no change in the
     configuration of  senses
```

Algorithm 2.3: simulated annealing algorithm[13]

• **Corpus based Lesk algorithm**

A corpus based Lesk algorithm [13] is other variation of the Lesk algorithm which is frequently used to solve the semantic ambiguity of a target word, using manually annotated corpora. This corpus-based variation has the capability to augment the sense-centered context of a word with additional tagged examples. Subsequently, the most likely sense for a new occurrence of the ambiguous target word is identified as the one with the highest dictionary overlap between the sense-centered contexts and the new context. The steps of corpus based algorithm are presented in Algorithm 2.4. The weight of a word is defined using a measure borrowed from the information retrieval community: Weight (w) is the inverse document frequency (IDF) of the word w over the examples and dictionary definitions. The IDF of a word is

$$IDF = -\log(p(w)) \quad (1)$$

where p(w) is estimated as the fraction of examples and definitions including the word w[13].

```
for each sense i of W
   set Weight(i) to 0
for each [unique] word w in surrounding context of W
     if w appears in the training examples or Dictionary
     definition of sense i
       add Weight(w) to Weight(i)
 choose sense i with highest Weight(i)
```

Algorithm 2.4: Corpus based Lesk algorithm [13]

- **Augmented Semantic Spaces**

Another variation of the Lesk algorithm , also called the adapted Lesk algorithm, was introduced by Banerjee and Pedersen[15]. In this algorithm, the definitions of related words are used in addition to the definitions of the word itself to determine the most likely sense for a word in a given context. Banerjee and Pedersen employ a function similar to the one defined by Cowie to determine a score for each possible combination of senses in a text, and attempt to identify the

sense configuration that leads to the highest score. While the original Lesk algorithm considers strictly the definition of a word meaning as a source of contextual information for a given sense, this algorithm is based on the WordNet hierarchy. The algorithm takes into account hypernyms, hyponyms, holonyms, meronyms, troponyms, attribute relations, and their associated definitions to build an enlarged context for a given word meaning. Algorithm 2.5 describes steps followed by augmented semantic space[15].

```
let best_score_till_now = 0
loop until all candidate combinations are done
 let w[i...N] = get_next_candidate_combination(w)
 let combination_score = 0
 for i ranging over 1 <= i < N
   for j ranging over i < j <= N
     for r1 ranging over (self hypernym hyponym holonym meronym
                          troponym attribute)
        for r2 ranging over (self hypernym hyponym holonym meronym
                             troponym attribute)
           let combination_score = combination_score +
                       get_score(gloss(r1(w[i])), gloss(r2(w[j])));
         end for
      end for
     end for
 end for
 if combination_score > best_score_till_now
    let best_score_till_now = combination_score
    let best_candidate_till_now[1...N] = w[1...n]
 end if
 end loop
 output best_candidate_till_now
```

Algorithm 2.5 Augmented Semantic Spaces algorithm[15]

ii. Semantic Similarity

Words in a discourse must be related in meaning for the discourse to be coherent. This is a natural property of human language and at the same time one of the most powerful constraints used in automatic word sense disambiguation. Words that share a common context are usually closely related in meaning, and therefore the appropriate senses can be selected by choosing those meanings found within the smallest semantic distance[17].Measures of semantic similarity computed over semantic networks(distance between concepts).Depending on the size of the context they span, these measures are in turn divided into two main categories:

1. Methods applicable to a local context, where semantic measures are used to disambiguate words connected by a) syntactic relations; to words connected by syntactic dependencies with the target word. b) their locality. this kind of semantic constraint is often able to provide unity to an entire discourse, Depending on the size of the context they span its scope has been usually limited to a small number of words found in the immediate vicinity of a target word. These methods target the local context of a given word, and do not take into account additional contextual information found outside a certain window size.
2. Methods applicable to a global context, where lexical chains are derived based on measures of semantic similarity (a lexical chain is a thread of meaning drawn throughout an entire text). This rely on a global context and attempt to build threads of meaning throughout an entire text, with their scope extended beyond a small window centered on target words. Lexical chains are an example of such semantic relations drawn across several words in a text. Similar to the Lesk algorithm, these similarity methods become extremely computationally intensive when more than two words are involved. However, solutions designed to increase the efficiency of the Lesk algorithm are equally applicable here, as for instance the algorithm proposed in [18] in which each ambiguous word in disambiguated individually, using a method similar in spirit with to the simplified Lesk algorithm.

iii. Selectional Preferences

Selectional preferences [19] find information of the likely relations of word types, and denote common sense using the knowledge source. For example, modeling-dress, walk-shoes are the words with semantic relationship. In this approach, improper word senses are omitted and only those senses are selected which have harmony with common sense rules. The basic idea behind this approach is to count how many times this kind of word pair occurs with syntactic relation. From this count, senses of words will be identified. This method captures information about the possible relations between word categories, and represents commonsense knowledge about classes of concepts. EAT-FOOD, DRINK-LIQUID, are examples of such semantic constraints, which can be used to rule out incorrect word meanings and select only those senses that are in harmony with commonsense rules. For instance, given the sentence *Mary drank burgundy*, the 'color' sense of *burgundy* does not fit in context since the verb *to drink* requires a liquid as a direct object[13].

Frequency counts of word-to-word relations are useful measures to account for the *semantic fit* between words. Given two words *W*1 and *W*2, and the syntactic relation *R* that connects them, the semantic fit between these words can be quantified by counting in a large corpus the number of times that the two words occur in the relation *R*, which can be formalized here as *Count* (*W*1,*W*2 ,*R*). An alternative method is to use conditional probabilities to estimate the semantic fit of a given relation. Under the same assumption that selectional preferences are learned for two words *W*1 and *W*2 connected by a relation *R*, the conditional probability is determined as in

$$P(w1 \mid w2, R) = \frac{\text{Count}(w1,w2,R)}{\text{Count}(w2,R)} \quad (2)$$

where the word *W*2 imposes the selectional preferences on *W*1. The constraint can be expressed in the other direction as well, with conditional probabilities where the roles of the two words are reversed. While selectional preferences are intuitive, and occur to us in a natural way, it is difficult to put them into practice to solve the problem of WSD.

iv. Heuristic Method

Heuristic methods, consisting of simple rules that can reliably assign a sense to certain word categories, including: Most frequent sense , One sense per collocation, One sense per discourse [13]. The most frequent sense works by finding all likely senses that a word can have and it is basically right that one sense occurs often than the others. One sense per discourse says that a word will preserve its meaning among all its occurrences in a given text. Finally, one sense per collocation is same as one sense per discourse except it is assumed that words that are nearer, provide strong and consistent signals to the sense of a word.

2.6.2 Corpus-based approaches

Corpus-based approaches are those that build a classification model from examples. These methods involve two phases: learning and classification[21]. The learning phase consists of learning a sense classification model from the training examples. The classification process consists of the application of this model to new examples in order to assign the output senses. Most of the algorithms and techniques to build models from examples come from the machine learning area of AI, such as supervised and unsupervised approach[21].

i. Supervised Corpus-based WSD

The supervised approaches applied to WSD systems use machine-learning technique from manually created sense-annotated data. Training set will be used for classifier to learn and this training set consists of examples related to target word. These tags are manually created from a dictionary. Basically, this WSD algorithm gives good result than other approaches. The most commonly used methods in supervised WSD are discussed below:

- **Decision List**

A decision list [22] is a set of "if-then-else" rules. Training sets are used in decision list to induce the set of features for a given word. Using those rules, few parameters like feature-value, sense, score are created. Based on the decreasing scores, final order of rules is generated, which creates the decision list. When any word is considered, first its occurrence is calculated and its representation in terms of feature vector is used to create the decision list, from where the score is calculated. The maximum score for a vector represents the sense.

- **Decision Tree**

A decision tree is a tree structure which uses classification rules in a tree structure that recursively divides the training data set. Parent node of a decision tree denotes a test which is going to be applied on a feature value [23]. Each branch denotes an output of the test. The exact sense of the word is represented in the leaf node.

- **Naïve Bayes**

Naive Bayes classifier [24] is a probabilistic classifier which is based on Bayes Theorem. This approach classifies text documents using two parameters: the conditional probability of each sense (Si) of a word (w) and the features (fj) in the context. The maximum value evaluated from the Bayes formula represents the most appropriate sense in the context.

- **Neural Networks**

In the neural network based computational model [26],artificial neurons are used for data processing using connectionist approach. Input of this learning program is the pairs of input features, and the goal is to partition the training context into non-overlapping sets. Next, to produce a larger activation these newly formed pairs and link weights are gradually adjusted. Neural networks can be used to represent words as nodes and these words will activate the ideas to which they are semantically related. The inputs are propagated from the input layer to the output layer through the all intermediate layers. The input can easily be propagated through the network and manipulated to arrive at an output. It is difficult to compute a clear output from a network where the connections are spread in all directions and form loops. Feed forward networks are usually a better choice for problems that are not time dependent and predict a diverse range of applications.

- **Exemplar-Based or Instance-Based Learning**

This supervised algorithm builds classification model from examples [27]. This model will store examples as point in feature space and new examples will be considered for classification.

These examples are gradually added to the model. The k-nearest neighbor algorithm is based on this methodology. In this procedure, first of all a certain number of examples is collected; after that the Hamming distance of an example is calculated by using k–NN algorithm. This distance

calculates the closeness of the input with respect to the stored examples. The k >1 represents the majority sense of the output sense among the k-nearest neighbors.

- **Support Vector Machine**

Support vector machine based algorithm [28] use the theory of structural risk minimization. The goal of this approach is to separate positive examples from negative examples with maximum margin and margin is the distance of hyperplane to the nearest of the positive and negative examples. The positive and negative examples which are closest to the hyperplane are called support vector. The SVM based algorithms are used to classify few examples into two distinct classes. This algorithm finds a hyperplane between these two classes, so that, the separation margin between these two classes becomes maximum. The classification of the test example depends on the side of the hyperplane, where the test example lies in. The input features can be mapped into a high dimensional space also, but in that case, to reduce the computational cost of the training and the testing procedure in high dimensional space, some kernel functions are used. A regularization parameter is used in case of non-separable training examples. The default value of this parameter is considered as 1. This regularization procedure controls the trade-off between the large margin and the low training error.

ii. Unsupervised Corpus-based WSD

Unsupervised WSD methods do not depend on external knowledge sources or sense inventories, machine readable dictionaries or sense-annotated data set [3]. These algorithms generally do not assign meaning to the words instead they discriminate the word meanings based on information, found in un-annotated corpora. This approach has two types of distributional approaches; the first one is monolingual corpora and the other one is translation equivalence based on parallel corpora. These techniques are further categorized into two types; type-based and token-based approach. The type-based approach disambiguates by clustering instances of a target word and token-based approach disambiguates by clustering context of a target word. Main approaches of unsupervised are as follow:

A. Context Clustering

A context clustering method is based on clustering techniques in which first context vectors are created and then they will be grouped into clusters to identify the meaning of the word [29]. This method uses vector space as word space and its dimensions are words only. Also in this method, a word which is in a corpus will be denoted as vector and how many times it occurs will be counted within its context. After that, co-occurrence matrix is created and similarity measures are applied. Then discrimination is performed using any clustering technique.

A distributed K-means clustering method is used in the offline procedure. In this approach, the Google n-gram (n=5) corpus Version-II is considered as a compressed summary of the web. This corpus consists of 207 billion tokens selected from the LDC-released Version-I, which consisted of 1.2 billion tokens. These 5-grams are extracted from about 9.7 billion sentences. All these 5-grams are tagged with part-of-speech (POS) according to their original sentences. Then the resulting clusters are utilized for WSD in a Naïve Bayesian classifier.

B. Word Clustering

Word clustering is similar to context clustering in terms of finding sense. But here, the method clusters those words which are semantically identical. The approach uses Lin's method and identifies the identical words which are similar to the target word. Similarity among those words is calculated using the features they are sharing. Then clustering algorithm is applied to discrimination among senses. If a collection of words is taken, first the similarity among them is identified by using measures. Then ,words are arranged in an order according to the similarity and create similarity tree. At the starting stage, only one node is there and for each word available in the list, iteration is applied. Finally, pruning is applied to the tree. As a result, it generates sub-trees. The sub-tree where the root is the initial word that we have taken to find ambiguity, gives the senses of that word[3].

C. Co-occurrence Graph

This method creates co-occurrence graph with edge E and vertex V. where E is added if the words co-occur in the relation according to syntax in the same text or paragraph and V represents the words in the text. For a given input target word, first, the graph is created and then adjacency matrix for the graph is determined. After that, the Markov clustering method is applied to the

graph to find the meaning of the word. Each edge of the graph is assigned a weight which represents the co-occurring frequency of those words. Weight for edge {m,n} is given by the formula:

$$w_{mn} = 1 - \max\{P(w_m \mid w_n), P(w_n \mid w_m)\} \quad (3)$$

Where $P(w_m|w_n)$ is the $freq_{mn}/freq_n$ where $freq_{mn}$ is the co-occurrence frequency of words w_m and w_n, $freq_n$ is the occurrence frequency of w_n. Word with high frequency is assigned the weight 0, and the words which are rarely co-occurring, assigned the weight 1. Edges, whose weights exceed certain threshold, are omitted. Then an iterative algorithm is applied to graph and the node having highest relative degree, is selected as hub. Algorithm comes to an end, when frequency of a word to its hub reaches to below threshold. At last, whole hub is denoted as sense of the given target word. The hubs of the target word which have zero weight are linked and the minimum spanning tree is created from the graph. This spanning tree is used to disambiguate the actual sense of the target word.[7].

D. Spanning tree based approach

Word sense induction is the task of identifying the set of senses of an ambiguous word in an automated way. These methods find the word senses from a text with an idea that a given word carries a specific sense in a particular context when it co-occurs with the same neighboring words[3]. In this approach, first a co-occurrence graph (Gq) is constructed. Then all the nodes whose degree is 1 are eliminated from Gq. The maximum spanning tree (MST) TGq of the graph is determined. Then, the minimum weight edge e_TGq is removed from the graph one by one, until the N connected components that are the word clusters are formed or until there remains no more edges to eliminate.

2.7 Afaan Oromo Language

Afaan Oromo is among the major languages that are widely spoken and used in Ethiopia and neighbor countries like Kenya and Somalia. Currently, it is an official language of Oromia regional state (which is the largest regional state in Ethiopia). It is used basically by Oromo people, who are the largest ethnic group in Ethiopia, which amounts to 34.5% of the total population (Census, 2017). With regard to the writing system, Qubee (a Latin-based alphabet) has been adopted and become the official script of Afaan Oromo since 1991 [30] .

Qubee has 33 characters representing distinct sounds. The 26 basic letters of qubee are ordered in similar way with other languages that use Latin based alphabet. In qubee the additional seven compound letters are often ordered following the 26 “basic” characters. Qubee is given in Appendix A.

2.7.1 Word class

List of word classes of Afaan Oromo and their equivalent of the English word classes is given in Table 2.1 [31].

Table 2.1 Afaan Oromo word class

Gochima	Verb(v)
Maqaa	Noun(n)
Addeessa	Adjective(adj)
Dabalgochima	Adverb(adv)
Raajeffanno	Interjection(int)
Walingaa	Conjunction(conj)
Durgala	Preposition(prep)
Maqadhaal	Pronoun(pron)

2.7.2 Afaan Oromo Sentence Structure

Afaan Oromo uses subject-object-verb (SOV) structure unlike English which has (SVO) structure. For instance, in the Afaan Oromo sentence “Birhaneen *baratuu* dha”, “Birhaneen “is the subject, “*baratuu*” is the object and “dha” is the verb. The translation of the sentence in English is “Birhane is a student”. There is also a difference in the formation of adjectives in Afaan Oromo and English. In Afaan 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, in “ ilma gaarii” (good boy), gaarii (adj.) follows ilma (noun).

2.7.3 Articles

Afaan Oromo does not require articles that appear before nouns unlike that of English. As a result of this translation of noun phrases is difficult. In English, there are three main semantic choices for article insertion: definite article (the), indefinite article (a, an, some, any) and no article. In Afaan 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 "namtiicha" to indicate certainty.

2.7.4 Punctuation Marks

Punctuation marks used in both Afaan Oromo and English languages are the same and are used for the same purpose with the exception of apostrophe. Apostrophe mark (') in English shows possession but in Afaan Oromo it is used in writing to represent a glitch (called hudhaa) sound. It plays an important role in the Afaan Oromo reading and writing system. For example, it is used to write the word in which most of the time two vowels are appeared together like "du'a" to mean ("die") with the exception of some words like "har'a" to mean "today" which is identified from the sound created.

2.7.5 Morphology

Morphology is a branch of linguistic that studies and describes the internal structure of words and how words are formed in a language. There are two branches of morphology: inflectional and derivational. Inflectional morphology deals with combination of a word stem with a grammatical morpheme in the same word class. In inflectional morphology, inflectional morphemes, morphemes that serve a purely grammatical function, which never create a new word but only a different form of the same word, are added in words. However, derivational morphology deals with combination of a word stem with a grammatical morpheme that yields different word class. Thus, in derivational morphology, there are methods of forming new lexemes from already existing ones by affixing derivational morphemes, morphemes that change the meaning or lexical category of the words to which they are attached.

Like in a number of other African and Ethiopian languages, Afaan Oromo has a very complex and rich morphology [32]. It has the basic features of agglutinative languages involving very extensive inflectional and derivational morphological processes. In agglutinative languages like Afaan Oromo, most of the grammatical information is conveyed through affixes, (that is,

prefixes and suffixes) attached to the root or stem of words. Although Afaan Oromo words have some prefixes and infixes, suffixes are the predominant morphological features in the language. Almost all Afaan Oromo nouns in a given text have person, number, gender and possession markers which are concatenated and affixed to a stem or singular noun form. In addition, Afaan Oromo noun plural markers or forms can have several alternatives. For instance, in comparison to the English noun plural marker, s (-es), there are more than ten major and very common plural markers in Afaan Oromo including: -oota, -oolii, -wwan, -lee, -an, een, -eeyyii, -oo, etc.). As an example, the Afaan Oromo singular noun mana (house) can take the following different plural forms: manoota (mana + oota), manneen (mana + een), manawwan (mana + wwan). The construction and usages of such alternative affixes and attachments are governed by the morphological and syntactic rules of the language. Table 2.2 indicates Afaan Oromo plural markers.

Case

May be an uncommon grammatical class of noun, pronoun, adjective, participle or numeral whose esteem reflects the grammatical work performed by that expressions over a phrase, clause, or sentence. Afaan Oromo thing need a reference structure alternately base structure that is utilized when the thing is the article of a verb, the item of a preposition or postposition, or an ostensible predicative.

- ***mana*** 'house', ***mana binne*** 'we bought a house'
- ***hamma*** 'until', ***dhuma*** 'end', ***hamma dhuma*** 'until (the) end'
- ***mana keessa***, 'inside (a/the) house'
- ***inni*** 'he', ***barsiisaa*** 'teacher'
- ***inni barsiisaa (dha)*** 'he is a teacher'

A thing might additionally show up Previously, a standout amongst six other grammatical cases (Nominative, Genitive, Dative, Instrumental, Locative, Ablative).

Table 2.2 Afaan oromo plural markers

Suffixes that delete the last vowel{ -oota,- oolii, oolee}			
Noun	**Gloss**	**Plural**	**Gloss**
Barataa	Student	Barattoota	students
Warra	Parent	Warroolii	parents
Suffixes that don't delete the last vowel{ -wwan, -lee}			
Noun	**Gloss**	**Plural**	**Gloss**
Gaaffii	Question	Gaaffiiwwan	questions
Jabbii	Calf	Jabbiilee	Calves
Suffixes that double last consonant{ -een, -(a)n}			
Noun	**Gloss**	**Plural**	**Gloss**
Beera	old woman	Beerran	old women
Eessuma	uncle (maternal)	Eessumman	Uncles
Wasiila	uncle (paternal)	Wasiillan	Uncles
Muka	Tree	Mukkeen	Trees
Mana	House	Manneen	Houses
Suffixes that drop –eessa/eensa{ -eeyyii}			
Noun	**Gloss**	**Plural**	**Gloss**
sooressa	Rich	Sooreyyii	rich people
waraabessa	Hyena	Waraabeyyii	Hyenas

Nominative

Nominative. Those nominative will be utilized for nouns that would those subjects for clauses. Table 2.3 indicates Afaan oromo nominative case markers.

Example: Ibsaa (a name), Ibsaan 'Ibsaa (nom.)', konkolaataa '(a) car', qaba 'he has':.

Ibsaan konkolaataa qaba 'Ibsaa need An car'.

Table 2.3 Afaan oromo nominative case markers example

Description	**Case Form Example**	**Case**
Practically nouns finishing in short vowels for a first absolute consonant drop the last vowel and include -ni to structure the nominative. Taking after sure consonants, digestion progressions whichever those n alternately that consonant (the subtle elements rely on upon those dialect)	• *nama* 'man', *namni* 'man (nom.)' • *namoota* 'men'; *namootni*, *namoonni* 'men (nom.)' (*t* + *n* may assimilate to *nn*)	Nominative
Though a last short vowel may be preceded by two consonants or An geminated consonant, -i is suffixed	• *ibsa* 'statement', *ibsi* 'statement (nom.)' • *namicha* 'the man', *namichi* 'the man (nom.)' (the *ch* in the definite suffix -*icha* is actually geminated, though not normally written as such)	Nominative
If the noun ends in a long vowel, -n is suffixed to this. This pattern applies to infinitives, which end in -uu	• *maqaa* 'name', *maqaan* 'name (nom.)' • *nyachuu* 'to eat, eating', *nyachuun* 'to eat, eating (nom.)'	Nominative
If the noun ends in n, the nominative is identical to the base form	• *afaan* 'mouth, language (base form or nom.)'	Nominative
Some feminine nouns ending in a short vowel add -ti. Again assimilation occurs in some cases	• *haadha* 'mother', *haati* (*dh* + *t* assimilates to *t*) • *lafa* 'earth', *lafti*	Nominative

Genitive

Those genitive may be utilized for ownership or "belonging"; it corresponds approximately on English from claiming or -'s. Those genitive may be generally structured by protracting a last short vowel, by including -ii on a last consonant, Also Toward taking off An last long vowel unaltered. The possessor thing takes after the possessed thing in a genitive phrase. A large number such expressions with particular specialized foul implications bring been included of the Oromo vocabulary Previously, late a considerable length of time.

- *obboleetti* 'sister', *namicha* 'the man', *obboleetti namichaa* 'the man's sister'
- *hojii* 'job', *Caaltuu*, woman's name, *hojii Caaltuu*, 'Caaltuu's job'
- *barumsa* 'field of study', *afaan* 'mouth, language', *barumsa afaanii* 'linguistics'

In place of the genitive it is also possible to use the relative marker kan (m.) / tan (f.) preceding the possessor.

- *obboleetti kan namicha* 'the man's sister'

Dative

Those dative may be utilized for nouns that representable the beneficiary (to) or the advocate (for) about occasion. Those dative manifestation of a verb infinitive (which demonstrations like a thing done Oromo) demonstrates motivation. Those dative takes a standout amongst those Emulating forms:.

- Lengthening of a final short vowel (ambiguously also signifying the genitive)
  - ✓ *namicha* 'the man', *namichaa* 'to the man, of the man'
- *-f* following a long vowel or a lengthened short vowel; *-iif* following a consonant
  - ✓ *intala* 'girl, daughter', *intalaaf* 'to a girl, daughter'
  - ✓ *saree* 'dog', *sareef* 'to a dog'
  - ✓ *baruu* 'to learn', *baruuf* 'in order to learn'
  - ✓ *bishaan* 'water', *bishaaniif* 'for water'
  - ✓ *-dhaa* or *-dhaaf* following a long vowel
  - ✓ *saree* 'dog'; *sareedhaa*, *sareedhaaf* 'to a dog'

- *-tti* (with no change to a preceding vowel), especially with verbs of speaking
  - ✓ *Caaltuu* woman's name, *himi* 'tell, say (imperative)', *Caaltuutti himi* 'tell Caaltuu'

Instrumental

The instrumental molding is utilized for nouns that representable those instrument flying ("with"), the intends ("by"), those agenize ("by"), the reason, or those the long haul of an off chance. The creation of the instrumental molding parallels that of the dative on a few extent. Table 2.4 indicates Afaan oromo instrumental case markers.

Table 2.4 Afaan oromo instrumental case markers example

Description	**Case Form Example**
-n following a long vowel or a lengthened short vowel; *-iin* following a consonant	• *harka* 'hand', *harkaan* 'by hand, with a hand' • *halkan* 'night', *halkaniin* 'at night'
-tiin following a long vowel or a lengthened short vowel	• *Afaan Oromo* 'Oromo (language)', *Afaan Oromotiin* 'in Oromo'
-dhaan following a long vowel	• *yeroo* 'time', *yeroodhaan* 'on time' • *bawuu* 'to come out, coming out', *bawuudhaan* 'by coming out'

Locative

The locative is utilized to nouns that speak to all areas of occasions or states, approximately at. For additional particular locations, Oromo employments prepositions or postpositions. Postpositions might additionally take the locative postfix documentation. The locative likewise appears should cover to some degree with the instrumental, now and then Hosting An fleeting capacity. Those locative will be shaped for those addition documentation -tti. *Arsiitti* 'in Arsii'

- *harka* 'hand', *harkatti* 'in hand'
- *guyyaa* 'day', *guyyaatti* 'per day'
- *jala, jalatti* 'under'

Ablative

The ablative may be used to speak to the sourball for an event; it corresponds nearly will English from. The ablative, connected on postpositions Also locative adverbs and additionally nouns proper, will be shaped in the Emulating ways. Table 2.5 indicates Afaan oromo ablative case markers.

Table 2.5 Afaan oromo ablative case markers example

Description	**Case Form Example**
• When the word ends in a short vowel, this vowel is lengthened (as for the genitive)	✓ *biyya* 'country', *biyyaa* 'from country' ✓ *keessa* 'inside, in', *keessaa* 'from inside'
• When the word ends in a long vowel, -*dhaa* is added (as for one alternative for the dative)	✓ *Finfinneedhaa* 'from Finfinnee (Addis Ababa)' ✓ *gabaa* 'market', *gabaadhaa* 'from market'
• When the word ends in a consonant, -*ii* is added (as for the genitive)	✓ *Hararii* 'from Harar'
• Following a noun in the genitive, -*tii* is added	✓ *mana* 'house', *buna* 'coffee', *mana bunaa* 'cafe', *mana bunaatii* 'from cafe'
• An alternative to the ablative is the postposition *irraa* 'from' whose initial vowel may be dropped in the process	✓ *gabaa* 'market', *gabaa irraa, gabaarraa* 'from market'

Frequent gender markers in Afaan Oromo include -eessa/-eettii, -a/-ttii or –aa/tuu. The language uses -eessa for masculine and -eettii for feminine. Natural female gender corresponds to grammatical feminine as in the case of sun, moon etc. names of towns, countries, rivers are also feminine. There are also suffixes like -a, -e that indicate present and past form of masculine markers respectively. -ti and -tii for present feminine marker and -te past tense marker -du for making adjective form. Table 2.6 indicates Afaan oromo gender markers.

Table 2.6 Afaan oromo gender markers

Afaan Oromo	**Construction**	**Gender**	**English**
Obboleessa	***obbol + eessa***	Male	Brother
Obboleettii	***obbol + eettii***	Female	Sister
beekaa	***beek+aa***	Male	Knowledgeable
beektuu	***beek + tuu***	Female	Knowledgeable

Demonstrative pronouns like kun (this), sun (that) are used to express definiteness. In some Afan Oromo dialects the suffix -icha for male and -ittii(n) for female and for undermining usually has a singularize function is used where other languages would use a definite article. For example: Afaanichi, afaanicha, Jaartittiin, jaartittii, Jaarsichi, jaarsicha, re`ettiin, re`e.

In Afaan Oromo, derivational suffixes enable a new word, often with a different grammatical category to be built from stem/root of other words. But the distribution of suffixes is unpredictable since some nouns are formed with different suffixes. Nouns can be derived from another noun or verb.

Abstract nouns are derived from other nouns by adding the suffix -ummaa,-eenya or -ooma to

the noun stems. Table 2.7 shows Sample Afaan oromo noun that are constructed by adding suffix to other noun.

Table 2.7 Sample Afaan oromo noun by adding suffix to other noun

Noun	**Gloss**	**derived noun**	**Gloss**
Gooftaa	boss/lord	Gooftummaa	lordship
Nama	Man	Namooma	Humanity
Nagaa	Peace	Nageenya	Peaceful
Jabaa	Strong	Jabeenya	Strength

Nominal can be derived from the verb stem by suffixing the morphemes like -aa,-eenya, -tuu, -ina, -noo, -ii, -ee, -iinsa, -iisa, -umsa, -maata, -aatii. Table 2.8 shows Sample Afaan oromo noun that are constructed by adding suffix to verb.

Table 2.8 Sample Afaan oromo noun that are constructed by adding suffix to verb.

Verb	**Gloss**	**derived noun**	**Gloss**
qab-	to have	qabeenya	Property
rak-	to suffer	rakkina	Problem
hubat-	to understand	hubannomoo	Understanding
falm-	to argue	falmii	Argument
tiks-	to shepherd	tiksee	shepherd/guardian
dalag-	to work	dalaga	work/job
barsiis-	to teach	barsiisaa	Teacher
bulch-	to govern	bulchiinsa	Government
qot-	to farm	qotiisa	Farming
bar-	to learn	barumsa	Education
fur-	to solve	furmaata	solution
lol-	to fight	loltuu	soldier

Afaan Oromo adjectives have case, person, number, gender, and possession markers similar to Afaan Oromo nouns. Afaan Oromo verbs are also highly inflected for gender, person, number, tenses, voice, and transitivity. Furthermore, prepositions, postpositions and article markers are often indicated through affixes in Afaan Oromo. The extensive inflectional and derivational features of Afaan Oromo are presenting various challenges for a number of NLP tasks in the language.

Morphological variations of the context words might not have any serious consequences for the performance of the WSD algorithms for morphologically poor language like English, however, this approach may not work well for morphologically rich languages like Afaan Oromo. In such languages, an ambiguous word might occur in several morphological forms and hence, without morphological analysis it would be impossible, even to identify these forms as ambiguous word forms, for assigning the correct sense [34]. A morphological-analyzer reduces the different forms of an ambiguous word into their root forms and plays an important role in this regard.

2.8 Ambiguities

Ambiguity can be referred as the ability of having more than one meaning or being understood in more than one way. Like all natural languages, Afaan Oromo language also has ambiguities. Ambiguity can occur at various levels of NLP. There is no previously studied and identified types of ambiguity in Afaan Oromo language. Tesfa Kebede [5] adapts types of ambiguity studied for Amharic language . Lexical ambiguity, phonological ambiguity, referential ambiguity and semantic ambiguity are the different types of ambiguity. We will summarize each type of ambiguity as follows and the examples are adopted from[5].

A. Lexical Ambiguity

Lexical Ambiguity is the ambiguity of a single word. A word can be ambiguous with respect to its syntactic class.

Example:

Lataan kubbaa miilaa xabata. ‘Lata plays football’.

Lataan miilaa dheera qaba. ‘Lata has a long leg’.

In the first sentence, miilaa ‘leg’ takes the position of adjective to describe the noun kubbaa ‘ball’. But in the second sentence, miilaa is a noun described by dheera ‘long’.

Lexical ambiguity can be resolved by lexical category disambiguation i.e, parts-of-speech tagging. There are different factors that can cause lexical ambiguity. Here we consider categorical ambiguity and homonymy.

- **Categorical Ambiguity**

Categorical ambiguity results from lexical elements which have the same phonological form but belongs to different word class. This will be more described using the following ambiguous word: Barsiisan kutaa ***seena*** jira. In the example, the bold italic word "***seena***" is ambiguous since it has both nominal and a verbal meaning:

The teacher is getting into the class room. [With nominal meaning]

The teacher is in the history room. [With verbal meaning]

- **Homonymy**

Homonyms are those lexical items with the same phonological form but with different meanings which will cause ambiguity. It can be illustrated with the following example: Tolaan ***ulfina*** gudda qaba. In the example the word "***ulfina***" is an ambiguous word having the following two different senses:

Tolaa has a huge weight

Tolaa is a respected person

B. Phonological Ambiguity

Phonological ambiguity results due to the sound used for the word from the placement of pause within a structure which occurs in speech. It can be illustrated through the following example: **Karaa + itti du'e / karaatti du'e** In the sentence, "+"sign shows the place where the pause has occurred. When the sentence is pronounced with pause, it means "*the way he was killed*" but the meaning differs if it is pronounced without pause. It will mean "*He died on the road*".

C. Referential ambiguity

Referential ambiguity arises when a word or phrase in the context of a particular sentence refers to two or more properties or things. Usually, the context tells us which meaning is intended, but when it doesn't we may choose the wrong meaning. If we are not sure which reference is intended by the speaker, we will misunderstand the speaker's meaning, if we assign the wrong meaning to the word . For example, ***Tolaan nama gudda dha (tolaa is a big man)*** you have to guess whether *gudda* (big) refers to his height (dheera dha), his weight (furdaa dha), social status

(kabajamaa dha) or something else. As another example: ***Gaadisaan gatii ebifaamef gamade***. The sentence has three different meanings:

Gadisa was pleased because he graduated.

Somebody was pleased because Gaadisa graduated.

D. Semantic Ambiguity

Semantic ambiguity is the phenomenon when a word has multiple meanings. It is caused by polysemic and idiomatic constituents. The following sentence is an example of polysemic constituent which has multiple meanings. **Abaabon Lalisee gudate jira.** The sentence has two interpretations:

The flower has grown.

Lalise's flower has grown.

Idioms refer to an expression that means something other than the literal meanings of its individual words. Idioms ambiguity can be illustrated using the following example: **Inni dhiiga kooti.** The literal meaning of the example is "*that is my blood*" but the idiomatic expression refers to "that is my relative".

2.9 Summary

This chapter discusses about application of WSD, WSD tasks, and types of Knowledge sources for WSD as well as Afaan Oromo language and Afaan Oromo ambiguity. WordNet and survey of the major approaches to WSD have been discussed. The next chapter discusses related works in the area of WSD that have been done for different languages.

Chapter Three: Related Work

Word sense disambiguation is one of the most popular research areas in the field of natural language processing. Some of the research works have been done in different languages. In this chapter, we review the research works conducted on the development of Word Sense Disambiguation.

3.1 WSD for Ethiopian Language

3.1.1 WSD for Amharic Language

Teshome Kassie [35] has aimed to demonstrates how linguistic disambiguation based on semantic vector analysis can improve the effectiveness of an Amharic document query retrieval algorithm. Query criteria based document retrieval is important for Accurate document retrieval in every knowledge domain. If words can have different meanings in different contexts The ability to retrieve appropriate documents is made more difficult as a result when search engines could disambiguate those words, more accurate retrieval of documents should be able to be achieved.

The author used an Amharic disambiguation algorithm based on the principles of semantic vectors and implemented in Java. The disambiguation algorithm was then used to develop a document search engine. A set of 865 Ethiopian Amharic language legal statute documents were selected as the document population that would be searched. Ten queries containing Amharic keywords with ambiguous meaning were selected. Author compare semantic vector query algorithm with Lucene algorithm. Each query was run using both algorithms. The 20 most relevant documents were identified for each query from each algorithm. For each query, the list of documents retrieved by each algorithm was compared to the list of documents identified by the expert. The number of correct (consistent with the expert's choices)documents retrieved by each algorithm was measured. Results are that the semantic vector algorithm was superior for 6 of the 10 queries (Lucene was superior on 2 queries, and on two they were tied). The semantic vector algorithm averaged 82% correct identification of documents whereas the Lucene algorithm was only 49% accurate. The Author concluded that for Amharic legal statute documents, for queries that include ambiguous keywords, the semantic vector algorithm is superior over Lucene algorithm.

Solomon Mekonnen [36] used a corpus based approach to disambiguation, where machine learning techniques are applied to a corpus of Amharic sentences so as to acquire disambiguation information automatically. A total of 1045 English sense examples for the five ambiguous words **(** አጠና(eTena), መሳል(mesal), መሣሣት(me`sa`sat), መጥራት(metrat), and ቀረጸ(qereSe) **)** are collected from British National Corpus (BNC) and the sense examples are translated to Amharic using dictionary. The sense examples are manually annotated and preprocessed to make it ready for experiment. Naive-Bayes classifier is employed from Weka 3.62 package in both the training and testing phases to perform the supervised learning on the preprocessed dataset using 10-fold cross-validation. The author concluded that, Naïve Bayes methods achieve higher accuracy on the task of WSD for selected ambiguous words using five clustering algorithms those are Simple k means, EM and agglomerative single, average and complete link . In this research total of four experiments has been conducted. The first experiment was to check the effect of stemming and stop word removal, the second one was to investigate the effect of different context sizes, the third was to see the effect of sense distribution and the last experiment was to compare the accuracy of selected algorithms and achieved accuracy within the range of 70% to 83% which is very encouraging.

However Corpus based approach suffers from the so-called knowledge acquisition bottleneck. It needs large quantities of sense examples to learn disambiguation rules .This is very challenging for linguistic resource-deficient languages like Amharic and further experiments for other ambiguous words and using different approaches needs to be conducted.

Solomon Assemu [37] presents a corpus based approach to word sense disambiguation that only requires information that can be automatically extracted from untagged text. author used unsupervised techniques to address the problem of automatically deciding the correct sense of an ambiguous word based on its surrounding context. This study, report experiments on five selected Amharic ambiguous words, these are አጠና (eTena), መሳል (mesal), መሣሣት (me`sa`sat), መጥራት (metrat), and ቀረጸ (qereSe).For the purposes of this research, unsupervised machine learning technique was applied to a corpus of Amharic sentences so as to acquire disambiguation information automatically. A total of 1045 English sense examples for the five ambiguous words were collected from British National Corpus (BNC). The sense examples were translated to

Amharic using the Amharic-English dictionary and preprocessed to make it ready for experimentation. The author tested five clustering algorithms (simple k means, hierarchical agglomerative: Single, Average and complete link and Expectation Maximization algorithms) in the existing implementation of Weka 3.6.4 package. “Class to cluster” evaluation mode was selected to learn the selected algorithms in the preprocessed dataset. The author achieved accuracy within the range of 65.1 to 79.4 % for simple k means, 67.9 to 76.9 for EM and 54.4 to 71.1 for complete link clustering algorithms for five ambiguous words.

Getahun Wassie [38] have conducted a research with the main objective of design a WSD (word sense disambiguation) prototype model for Amharic words using semi-supervised learning method to extract training sets which minimizes the amount of the required human intervention and it can produce considerable improvement in learning accuracy. The semi-supervised method narrows the gap of supervised and unsupervised methods by making use of labelled and unlabeled training data. In other tokens, it minimizes knowledge-acquisition bottleneck of supervised learning, and it improves poor performance of unsupervised learning. One of its main differences from the previously tested Amharic WSD models is the existence of training data from some labelled and many unlabeled datasets. They used a python program which they made it in line with their preprocessing tasks. The experiment of semi-supervised methods using bootstrapping algorithms was conducted on the five Amharic WSD datasets following semi-supervised clustering assumption. These words were atena (አጠና), derese (ደረሰ), tenesa (ተነሳ), bela (በላ) and ale (አለ). Separate data sets using the five ambiguous words were prepared for the development of this Amharic WSD prototype. The experiment showed that the average performance results of Adaboost, Bagging and ADtree algorithms are 84.90%, 81.25% and 88.45%. And the researchers concluded that Semi-supervised learning using bootstrapping algorithm performs better in their 35 study and it is more adaptive on WSD for Amharic. They also found that, a window size of 3-3 can be a standard window size for Amharic WSD systems development.

Segid Hassen [39] has implemented a knowledge-based word sense disambiguation method that employs Amharic WordNet that was developed in his thesis work. Knowledge-based Amharic WSD extracts knowledge from word definitions and relations among words and senses. The system consists of preprocessing, morphological analysis and disambiguation components besides Amharic WordNet database. The system assigns the appropriate sense of ambiguous words in a sentence using Amharic WordNet by using sense overlap and related words. The researcher conducted two major experiments. The first one is evaluating the effect of Amharic WordNet with and without morphological analyzer and the second one is determining an optimal windows size for Amharic WSD. For Amharic WordNet with morphological analyzer and Amharic WordNet without morphological analyzer obtained accuracy of 57.5% and 80%, respectively and found two-word window on each side of the ambiguous word is enough for Amharic WSD.

3.1.2 WSD for Afaan Oromo Language

Tesfa Kebede [5] presents a research work on Word Sense Disambiguation for Afaan Oromo Language. A corpus based approach to disambiguation is employed where supervised machine learning techniques are applied to a corpus of Afaan Oromo language, to acquire disambiguation information automatically. It also applied Naïve Baye‟s theorem to find the prior probability and likelihood ratio of the sense in the given context. The author collected total of 1240 Afaan Oromo sense examples for selected five ambiguous words namely ***sanyii, karaa, horii, sirna and qoqhii.*** The sense examples were also manually tagged with their correct senses and preprocessed to make it ready for experimentation. Hence, these sense examples were used as a corpus for disambiguation. As stated by author a standard approach to WSD is to consider the context of the ambiguous word and use the information from its neighboring or collocation words. As result the contextual features used in this thesis were co-occurrence feature which indicate word occurrence within some number of words to the left or right of the ambiguous word. For the purpose of evaluating the system, a statistical technique called k-fold cross-validation was applied using standard performance evaluation metrics. The researcher achieved an accuracy of 79% and found four-word window on each side of the ambiguous word is enough for Afaan Oromo WSD. However Corpus based approach suffers from the so-called knowledge acquisition bottleneck it needs large quantities of sense examples to learn disambiguation rules this is very challenging for linguistic resource-deficient languages like Afaan Oromo.

Workineh Tesema [6] followed unsupervised corpus-based methods of word sense discrimination and do not rely on external knowledge sources such as machine readable dictionaries, concept hierarchies or sense-tagged text. The algorithm first gathers all the contexts of a given word and then cluster them by using Cluster BY Committee (CBC) algorithm. The clusters are labeled using frequency information and the number of clusters produced is considered as the number of the senses of the target word. CBC takes a word type as input, and finds clusters of words that represent each sense of the word . The clusters will be made up of synonyms or words that are related to the discovered senses. The premise behind the approach is that words having multiple senses or ambiguous words have different distinct groups of contexts. Usually such contexts are expressed in terms of words. Therefore one can identify senses of a given word by retrieving all the words occurring in all the contexts of the words and group the words depending on the contexts. This approach relied on the co-occurrence information to retrieve the contexts. Accordingly all the words appearing with the target word in a fixed sized window of words is considered as context. The extracted words are grouped and labeled. Each group is considered as a sense of the target word.

For evaluation author used a portion of Wikipedia text to train and evaluate the algorithm. The system is therefore made to identify the sense of certain sets of word from Wikipedia. Author selected 10 words 6 ambiguous and the remaining 4 unambiguous words and provide to the system after that the result produced by the system manually evaluated. The task is basically clustering of the 10 words in to the correct classes: ambiguous or unambiguous. The strategy to evaluate the performance is therefore counting the number of words assigned to the correct class ambiguous or unambiguous class. Accordingly out of the 6 ambiguous words 3 of them are correctly classified as ambiguous giving the performance level of 51%. Similarly out of the 4 unambiguous words 3 of them are correctly classified as unambiguous giving the performance level of 49%. The average performance level is therefore 50%. However the performance of word sense disambiguation algorithm can be calculated by counting the number of senses produced for a given word and the correctness of the label produced by the system which is not done by this research work.

3.2 WSD for Non Ethiopian Language

3.2.1 WSD for English Language

In 1986, Lesk Michael [14] developed an algorithm called Lesk algorithm to identify senses of polysemy words and used the overlap of word definition from the Oxford Advanced Learner's Dictionary of Current English (OALD) to disambiguate the word senses. This algorithm is based on two assumptions. First, when two words are used in close proximity in a sentence, they must be talking of a related topic and second, if one sense each of the two words can be used to talk of the same topic, then their dictionary definitions must use some common words. For example, when the words ”pine cone” occur together, they are talking of ”evergreen trees”, and indeed one meaning each of these two words has the words ”evergreen” and ”tree” in their definitions. Thus

The algorithm can disambiguate neighboring words in a sentence by comparing their definitions and picking those senses whose definitions have the most number of common words. The biggest drawback of this algorithm is that dictionary definitions are often very short and just do not have enough words for this algorithm to work well.

Banerjee and Pedersen [15] try to deal with the problem of Lesk algorithm [14] by adapting the original Lesk algorithm to use the lexical database WordNet. Besides storing words and their meaning like a normal dictionary, WordNet also ”connects” related words together. The authors overcome the problem of short definitions by looking for common words not only between the definitions of the words being disambiguated, but also between the definitions of words that are closely related to them in WordNet. The authors used Senseval-2 word sense disambiguation exercise to evaluate their system and the overall accuracy was found to be 32%. To compare two glosses, they used the longest sequence of one or more consecutive words that occur in both glosses. For each overlap, a square of the number of words in the overlap is calculated and the final score is the sum of all overlaps.

Faris and Cheng [40] executed word sense disambiguation for the English language utilizing a Knowledge-based approach. The authors recommend a hearty knowledge-based answer for the saying feeling disambiguation issue for the English language. Ambiguous expressions are determined utilizing not main those word's part-of-speech, as well as the relevant data found in the sentence. The authors depicted a two-phase word-sense disambiguation result. The initial

stage may be answerable for identifying and locating all the possible knowledge objects corresponding to each term in the given sentence. It utilizes morphological principle standards taught to the system to convert words into their base forms, and happens at the first step of parsing a sentence, namely, the syntax step. The second stage is answerable for resolving the ambiguity among all possibilities to correctly identify the intended meaning. Because of those nature of the ambiguities, the authors arrange ambiguities under two classifications. One category may be resolved based on the grammar's requirement that a certain pos be at a specific place of the given sentence, and hence can be resolved during the parsing stage of a sentence. The other category is resolved during the understanding of the thought, which uses the context information available from the rest of the sentence. There solution depends on the capability of the program to infer the categories of various objects. Resolving an ambiguous word based on the word's pos is possible when the parse tree is unambiguous. However, problems may arise when multiple parse trees can be formed due to the absence of an optional term and the presence of a term with an ambiguous pos.

3.2.2 WSD for Hindi Language

As cited on [3] Haroon, R.P. (2010) has given the first attempt for an automatic WSD in Malayalam. The author used the knowledge based approach. One approach is based on a hand devised knowledge source and the other is based on the concept of conceptual density by using Malayalam WordNet as the lexical resource. The author has used the Lesk and Walker algorithm. In this algorithm, the collection of the contextual words is prepared for a target word. Next, different bags, containing few words of specific sense are generated from the knowledge source. After that, the overlap between the contextual words and the bags are measured. A score of 1 is added to that sense, if any overlap is there. Highest score for a sense is selected as the winner. Other approach is the conceptual density based algorithms find the semantic relatedness between the words The semantic relatedness is measured in many ways. One way is considering the path, depth and information content of words in the WordNet. For each sentence, first the sentence is tokenized, next, in a sequence of steps, the stop words are removed and stemming is performed. Then, the ambiguous word is detected. If an ambiguous word is found, that word is shifted into one document and sense lookup is performed. After that, the nouns are extracted from the sentence and saved as a document. For each sense in the sense lookup, the depth with

each noun is calculated. If there are multiple nouns, depth of each is added and taken as the depth. The sense, which results in lower depth (highest conceptual density) is selected as the correct sense.

Rakesh and Ravinder [41] have proposed a WSD algorithm for removing ambiguity from the text document. The authors used the Modified Lesk Algorithm for WSD. Two hypotheses have been considered in this approach. First, the co-occurring words in a sentence are be disambiguated by assigning the most closely related senses to them. The second hypothesis is considered as , the definitions of related senses have maximum overlap.

Sinha M. et. al. [42] developed automatic WSD for Hindi language using Hindi WordNet at IIT

Bombay. The authors used statistical method for determining the senses. The system could disambiguate the nouns only. They compared the context of the polysemy word in a sentence with the contexts constructed from the WordNet. They evaluated the system using the Hindi corpora provided by the Central Institute of Indian Languages (CIIL). The accuracy of their algorithm was found in the range from 40% to 70%. They used simple overlap method to determine the winner sense.

3.2.3 WSD for Nepali Language

Shrestha N. et. al [43] used the Lesk algorithm to disambiguate the Nepali ambiguous words. They modified the Lesk algorithm in such a way that only the words in the sentence without their synset, gloss, examples and hypernym are taken as context words. Each word in the context words is compared with each word in the collection of words formed by the synset, gloss, examples and hypernym of each sense of the target word. They did not include the synset, gloss,

example and hypernym of the context words in the collection of context words. Moreover, the number of example for each sense of the target word was only one.

Dhungana and Shakya [44] used the adapted Lesk algorithm to disambiguate the polysemy word in Nepali language. The experiments performed on 348 words (including the different senses of 59 polysemy words and context words) with the test data containing 201 Nepali sentences shows the accuracy of their system to be 88.05%. This accuracy is found to be increased by 16.41%, when compared to the accuracy of the system developed by[43].

Udaya R Dhungana et. al [45] presented a new model of WordNet that is used to disambiguate the correct sense of polysemy word based on the clue words for Nepali language . They refer related words for each sense of a polysemy word as well as single sense word as the clue words. The conventional WordNet organizes nouns, verbs, adjectives and adverbs together into sets of synonyms called synsets each expressing a different concept. In contrast to the structure of WordNet, researchers developed a new model of WordNet that organizes the different senses of polysemy words as well as the single sense words based on the clue words. These clue words for each sense of a polysemy word as well as for single sense word are used to disambiguate the correct meaning of the polysemy word in the given context using knowledge based Word Sense Disambiguation (WSD) algorithms. The clue word can be a noun, verb, adjective or adverb. Unlike in WordNet, researchers grouped each sense of a polysemy word based on the verb, noun, adverb and adjective with which the sense of the polysemy word can be used in a sentence.

The researches on Afaan Oromo WSD Tesfa Kebede [5], Workineh Tesema [6] conducted a study on selected words from similar word classes using machine learning techniques. Both the researchers come up with a promising result. In this thesis we will have investigate a more generic Afaan oromo WSD for all-word classes and all word task.

3.3 Summary

This part reviewed separate word sense disambiguation meets expectations that would identified with this postulation worth of effort Possibly over WSD approach, systems in how to resolve ambiguity, for language conduct What's more done assessment. In the reviewed research works, knowledge-based What's more corpus-based methodologies need aid utilized to saying feeling disambiguation should distinguish those accurate sense for ambiguous saying In view of setting. However, corpus-based WSD meets expectations relies once corpus evidence, which is used to train a model utilizing labeled alternately untagged corpus and hence, they need aid altogether unreasonably. Research works on [[15],[39],[42],[43] ,[44] ,[45]] used WordNet and it was noticed that the WordNet is very useful resource that is used for word sense disambiguation. However, it is not exactly suitable for knowledge-based, overlap selection WSD approaches. The reason stated by [45] is the WordNet is built for general purpose in NLP tasks but is not focused for WSD. WordNet contains huge amount of information for words that are arranged with semantic relation. Only very few words taken from the WordNet used to disambiguate the different senses of a multi-sense word so in all WSD methods, that used WordNet to take a large number of words to disambiguate the meaning of multi-sense word waste all the efforts such as processing time for CPU and memory to store large number of unused words. Furthermore, it is noticed that the words taken from the WordNet to disambiguate the multiple meaning of the multi-sense word itself creates the ambiguity resulting in decrease in accuracy. Another important point stated by [45] is that when deeper levels of the hypernym from the WordNet are used as like by [44], if the amount of information increased to make context large, due to the entry have more common information for all senses, correctly disambiguated words are also incorrectly disambiguated. The reason is that the deeper (probably from second and/or third) levels of Hypernym of all the different sense of a polysemy word are same, there is no any basis to disambiguate the different sense of the polysemy word and there is no solid answer for which level of hypernym disambiguation is the best. To this end we use knowledge based approach in this thesis to pick the sense whose definition is most similar to the context of the ambiguous word by means of textual overlap.

Chapter Four: Design Of Afaan Oromo WSD

4.1 Introduction

Among the different word sense disambiguation approaches proposed by different researchers we have implemented knowledge based approach. This knowledge-based overlap of sense selection WSD algorithm uses sense of ambiguous word from WordNet. For this research we used Afaan Oromo WordNet as knowledge source. There is no well-constructed Afaan Oromo WordNet; so we have constructed it by adapting English WordNet structure and modify it in order to make it suitable for our experiment.

The system accepts input and disambiguates a given ambiguous word in the text by checking overlaps between clue words belongs to different senses of the ambiguous word in Afaan Oromo WordNet and words around ambiguous word in the input text. In the next sub sections, the architecture of the WSD system with detail description of components and their algorithms as well Afaan oromo WordNet preparation are discussed.

4.2 Architecture Of Afaan Oromo WSD System

As shown in the figure 4.1 below, the system first accepts text as an input and then preprocesses the text. The texts are preprocessed to make suitable for further processing. The preprocessing task involves tokenization, normalization, and stop word removal. After that the system use morhplogical examination to decrease different manifestations of a expressions on a solitary root word. Morphologic examination may be vital for morphologically rich language such as Afaan Oromo on it may be difficult on store every expected expressions for WordNet. There is no morphological analyzer for Afaan Oromo language so we use manually developed morphological analyzer using morph and morph map tables of our AO wordnet.Finally word sense disambiguation task performed this task includes ambigious word identification and sense retrival.The system uses information from Afaan Oromo WordNet.

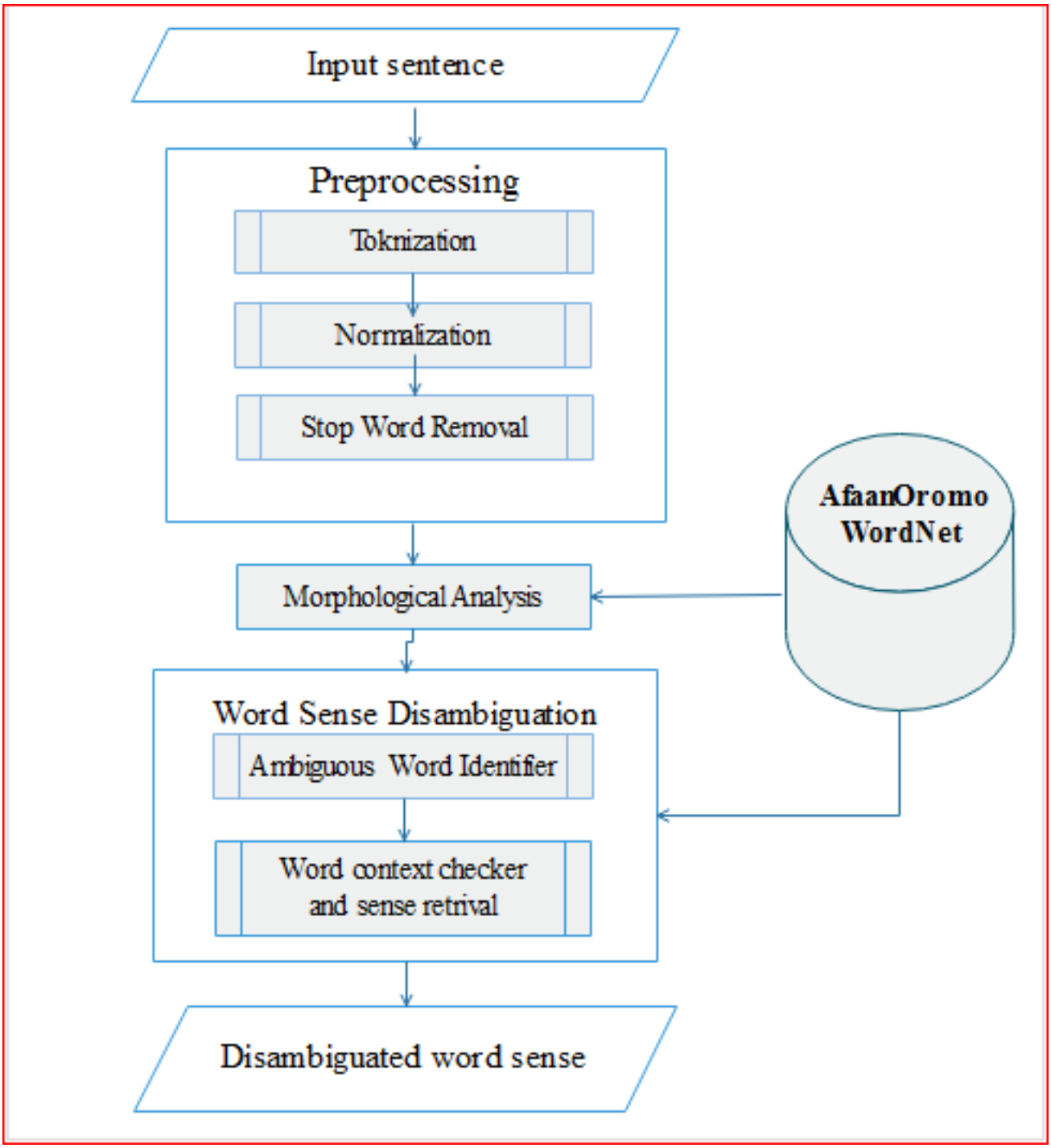

Figure 4.1: Architecture of knowledge based Afaan Oromo Word Sense Disambiguation system

4.2.1 Preprocessing

The preprocessing phase is required to prepare the data for further processing, which includes tokenization, normalization and stop word removal.

- **Tokenization (Sentence Segmentation)**

Sentence segmentation is also known as sentence boundary identification or tokenization. This module identifies sentence boundaries between clauses, phrases or sentences, by first splitting the all text into words. Word demarcation in Afaan Oromo is handled following white space. Thus, Afaan Oromo tokenization parses text into its constituent words usually by considering the white space and punctuation marks. Punctuation mark usage in Afaan Oromo is similar to that of English which includes semicolon (;), comma (,), full stop (.), question mark (?) and exclamation mark (!). These punctuation marks are removed from the text because they don't have any relevance in identifying the meaning of ambiguous words in WSD.

- **Normalization**

It is the process of transforming tokenized texts into a single canonical form that it might not have had before. Normalizing texts before storing or processing it allows for separation of distresses since input is assured to be consistent before operations are performed on it. Text can also be normalized for storing and searching in a database. For instance, if a search for "Caccabaa" is to match the word "caccabaa", then, the text would be converted to a single case "caccabaa. Even though, there are various types of normalization, for our work, it is intended for converting all tokenized texts to lower cases and prepare convenient environment as well.

- **Stop Word removal**

Stop word removal is a module used to remove stop words from the input text. Every language has its own list of stop words: words that have no significant discriminating powers in the meaning of ambiguous words. Stop words mainly consist of prepositions, conjunctions, articles, and particles. Stop words, are the high frequency words in a language which do not contribute much to the topic of the sentence. In English, such words include, 'a' , 'an' , 'the', 'of' , 'to' etc. In Afaan Oromo (**ammo** to mean however, but ; **garuu** to mean but; **bira** to mean beside, at, near of ; **ala** to mean outside, out ; **akka** to mean such as, like, according to etc.).

We remove these words and focus on our main subject/topic, to solve ambiguity. Removing these words will improve the efficiency of the system because this words are not checked from WordNet and need to be removed during preprocessing phase. There are various techniques used to remove stop words. Among this IDF (inverse document frequency) value and dictionary lookup are the common one. The IDF approach assumes words that appear in many documents as stop words. However, most of the existing stop words removal techniques are based on a dictionary lookup that contains a list of stop words. Dictionary lookup was employed for this study also. For the purpose of this research work, list of around 100 stop words that is compiled from Afaan Oromo books during implementation of a stemmer by Debela Tesfaye [32] is used. The algorithm is described in Algorithm 4.1. List of stop word indicated in Annex B.

```
Begin
 Input segmented text
  For each word in a segmented text
  If a word in segmented text is stop word
                      #compare with stop word list compiled
    Remove the word from the segmented text
  End for
 Output List of Nonstop word
Stop
```

Algorithm 4.1: Stop word removal algorithm

4.2.2 Morphological Analysis

Natural language applications, such as question-answering, speech recognition, information retrieval, and machine translation, rely on a lexicon of possible forms in the relevant language. Morphological analysis is important for morphologically complex languages like Afaan Oromo because it is practically impossible to store all possible words in a lexicon, and many words have close to 0 probability of occurrence in any given corpus. The correspondence between words in Afaan Oromo will often be many-to-one [46]. Morphological variations of the words might have serious consequences for the performance of the WSD algorithms for morphologically rich language like Afaan Oromo. In this thesis, we used manually constructed morphological analyzer using dictionary based method. The constructed table of morphological analysis is put in Afaan Oromo WordNet. Algorithm 4.2 shows how morphological analysis component of AOWSD system works.

```
Begin
Input :Nonstop word  List
      Load Afaan Oromo WordNet
     For each word in a Word List
       If exist word in morph list of  AOWN
          Extract root word belongs to that morph
#morph list holds different morph with a variety of different
possibilities and mapped to single lemma.
       Else
          End
      End For
  Return root word list
Output: list of root word
Stop
```

Algorithm 4.2: Morphological analysis algorithm

4.2.3 Word Sense Disambiguation

This component is composed of two sub-components: ambiguous word identifier, context checker and sense retrieval.

- **Ambiguous Word Identifier**

This component identifies ambiguous word from root words by comparing all root words with their senses from Afaan Oromo WordNet. It counts sense of a word and if the word has multiple senses it is set as an ambiguous word. The algorithm is shown in the Algorithm 4.3. As we use all words word sense disambiguation method which Attempt to disambiguate all open-class words in a text each word in the input text is disambiguated separately, starting with the first word and working left to right. At each stage, the word being disambiguated is called the target word, and the surrounding words form the context window. Ambiguous word Identifier is a component used to identify the ambiguous word from the input sentence based on information provided on Afaan Oromo WordNet and Afaan Oromo WordNet is used as a knowledge source for this study. Afaan Oromo WordNet is used to identify the ambiguous word in this study and contains a list of senses for given words from the input sentence. The Ambiguous Word Identifier is to be checked whether each root word exist in the Afaan Oromo WordNet or Not. Words that are found in Afaan Oromo WordNet have their own sense on AOWN. So if root word exist in AOWN this module checks whether root word belongs to one synset in AOWN or not if it belongs to one synset this word added to non-ambiguous word list (NAW) else if it belongs to more than one synset it will added to ambiguous word list(AW) .If words do not exist in AOWN the word is discarded.

For example, if the following sentence is the input sentence: "kasala aaraa isaa aarsii fixi." First, the input sentence is preprocessed. After morphological analysis, only four words will be left (i.e. kasala,aar,aars,fix) in the input sentence. Then each root word and belonging synset to them is counted in AOWN. In our case, the root word **"**aars**"** belongs to two synset. So that, "aars" is detected as ambiguous word in the input sentence and "aars" is the root word for the word "aarsii". Therefore, "aars" is ambiguous word and the sense is retrieved in AOWN based on the context of the sentence. The algorithm is shown in Algorithm 4.3.

```
Begin
Input : List of root words
   For each word W in root word list  do:
     If: W is belonging to only one synset in AOWN (Afaan Oromo WordNet)
      Then: Add  W to NAW (Non ambiguous word)list.
     Else if: W  is belonging to a several(more than one)synsets in AOWN
      Then: Add W to AW(Ambiguous word)list.
       # synset holds information about words and its senses in AOWN
    End
   Return Ambiguous Words list AW ; Non Ambiguous Words list NAW
Output: List of Ambiguous Words  AW and Non Ambiguous Words NAW.
Stop
```

Algorithm 4.3 : Ambiguous Word Identifier (AWI) Algorithm

- **Context checker and Sense Retrieval**

Context in WSD refers to the words surrounding the ambiguous words, which are used to decide the meaning of the ambiguous word. This component generally checks the relations among concepts by considering relation between ambiguous word and the context words. In our case clue words has relations with ambiguous word . Frequently co-occurring words are taken as clue words we gather clue words from different document for each ambiguous word and after collecting ambiguous word and there frequently co-occurring words as clue word the lists are approved by linguistic expert from Afaan oromo department. On our AO word net clue words has relation with ambiguous word the relation between them can be synonymy, hypernym ,antonymy, meronym and contextual relation. As we stated in literature review and related work section of this paper In the previous overlap based word sense disambiguation researches word gloss and limited number of relation like synonymy used often but those resources are not

enough for WSD task. As we know, words in WordNet are interlinked or related to each other. Relations among words in English WordNet are between single word class (pos) so there are limited number of cross pos relations in English WordNet .

Words in clue wordlists are belongs to different synonym set of ambiguous word .When we retrieving sense of ambiguous word we can get information about to which synonymy set ambiguous word belongs to and the gloss of belonging synonymy set as well. In this phase the system calculates overlap between context bag(which are non-ambiguous word list) from the input sentence and clue word list(frequently co-occurring words with ambiguous word) in AOWN then after the sense with highest overlap count selected as best sense of ambiguous word with in the given context. When we take the input sentence example "kasala aaraa isaa aarsii fixi."

After morphological analysis, only four words will be left (i.e. kasala,aar,aars,fix) in the input sentence. Then Ambiguous Word Identifier module counts the synset belongs to each root word from Afaan Oromo WordNet then the word belongs to more than one synset chosen as ambiguous word and added to Ambiguous Word List(AW) and the rest of words also added to non-Ambiguous Word List (NAW). So on this indicated example we have two list of words Ambiguous Word List { aars } and non-Ambiguous Word List { kasala,aar,fix }

Then the senses for each word in Ambiguous Word List is retrieved by Context checker and Sense Retrieval component of AOWSD.

The sense for ambiguous word list { aars } is

Sense1: Bobeessuu dhaan qilleensa gubataa kan gadi lakkisuu.

Sense2: Dallansiisuu.

In this module after sense for ambiguous word retrieved clue word list that are belongs to each sense of ambiguous word also retrieved.

For sense 1:

{arrab,boochisuu,dallan,deem,dheekkams,dhiit,dub,goch,habootuu,ifa,ija,ilaal,jech,jib,kaballaa, mudd,muf,qimmiduu,reebicha,ruk,waqar,waraan }

For sense 2:

{aar,abiddaa,aduu,belbel,boba'aa,bobba'a,bobbeessuu,chidii,cilee,danf,dikee,dungoo,Elektrikii, fool,galaba,gub,hulchiis,ibid,ibsaa,ixaana,kasala,kiribiit,kurraazii,muka,qayya,qoraa,qumbii,siga araa,tamboo,yaa'u }

Finally the comparison of non-ambiguous word list with clue word list of sense of ambiguous word is done .Non Ambiguous Word List { kasala,aar,fix } used as context bag and to which sense set of clue word it has overlap checked. This list has two overlap with sense 2 .So sense 2 extracted as sense of ambiguous word { aars } which is" Bobeessuu dhaan qilleensa gubataa kan gadi lakkisuu".The algorithm is shown in the Algorithm 4.4.

Table 4.1 sample polysemy word "**Aarsu**" with its sense and clue words

Polysemy word	**Sense**	**Clue word lists**
Aarsu	1.Dallansiisuu ykn Mufachiisuu	arrab,boochisuu,dallan,deem,dheekkams,dhiit,dub,goch,habootuu, ifa,ija,ilaal,jech,jib,kaballaa,mudd,muf,qimmiduu,reebicha,ruk,waqar,waraan
	2.Bobeessuu ykn Qilleensagubataa gad-lakkisuu	**aar**,abiddaa,aduu,belbel,boba'aa,bobba'a,bobbeessuu,chidii,cilee,danf,dikee, dungoo,elektrikii,fool,galaba,gub,hulchiis,ibid,ibsaa,ixaana,**kasala**,kiribiit,kur raazii,muka,qayya,qoraa,qumbii,sigaaraa,tamboo,yaa'u

Words in clue wordlists are belongs to different synonym set of ambiguous word .When we retrieving sense of ambiguous word we can get information about to which synonymy set ambiguous word belongs to and the gloss of belonging synonymy set as well. In this phase the system calculates overlap between context bag and clue word list then after the sense with highest overlap count selected as best sense of ambiguous word with in the given context.

Begin

Input: List of Ambiguous Words AW and Non Ambiguous Words NAW.

For all words in NAW(context words) **do:**

For all words in AW **do:**

Extract associated senses (s_1……s_n) from AOWN

End

For each senses (s_1……s_n) of AW **do:**

Extract associated **clue word list** (cl_1…… cl_n) from AOWN

// no of clue word list = no of senses of ambiguous word

End

For each clue word list(cl_1…..cl_n **)** of senses **do:**

Calculate overlap between word in NAW and word in clue word list

If overlap**(**NAW, cl_i**) >** overlap**(**NAW, cl_j**)**

Then best sense =sense attached to cl_i

//clue word list that has maximum overlap with NAW (context words)

Extract sense and the **sense gloss** of clue word list cli.

End

End

Return sense of ambiguous word

Output: sense of the ambiguous word

Stop

Algorithm 4.4 : Context checker and sense retrieval Algorithm

When we see overall system architecture using an example if user has this input text " kitaaba kee boqonnaa sadaffaa banii dubbisi."

The disambiguation system follow all word task disambiguation system because of this not only target words cheeked rather all words in user input text disambiguated. Let as discussed overall system architecture using this entered user input text. As we see from figure 4.1 the first component of knowledge base AOWSD system is Preprocessing in this phase input text tokenized then the tokenized text normalized after normalization stop words removed.

Preprocessing

- " kitaaba kee boqonnaa sadaffaa banii dubbisi."
    - kitaaba ,kee ,boqonnaa ,sadaffaa, banii, dubbisi.
    - kitaaba ,kee ,boqonnaa ,sadaffaa, banii, dubbisi.

after preprocessing the system expected to result **normalized nonstop word** list of tokens.

- kitaaba ,boqonnaa, sadaffaa, banii ,dubbisi

Morphology analysis

This phase accepts normalized nonstop word list from preprocessing phase then root words of each word in list extracted .To do this the component checks every words from AO WordNet then retrieve attached (mapped) root word to each words. Expected result from this component is word and its root word to which it mapped.

- Kitaaba➔kitaab
- boqonnaa➔boqonnaa
- sadaffaa➔sad
- banii➔ban
- dubbisi➔dubb

word sense disambiguation

On this phase the system expected to do ambiguous word identification and sense retrieval for each ambiguous word. This component accepts normalized nonstop word root list of words as input. From previous example { kitaab, boqonnaa, sad, ban ,dubb } is expected result from morphological analysis and input to this word sense disambiguation component.

Ambiguous word identifier : is sub component of word sense disambiguation component .As this system is all word task word sense disambiguation system this component will check the ambiguity of all words in the list . To do so each word in the list compared with AO WordNet. When we see the list in the example:{ kitaab, boqonnaa, sad, ban , dubb } the number of synset belongs to each word counted from beginning to the end of the stated word list then the word belongs to more than one synset in AO WordNet selected as ambiguous word and word belongs to more than one synset added to ambiguous word list and word belongs to one synset added to non-ambiguous word list. In our AO WordNet synset has information about word and its sense .

Word ***kitaab*** has one sense so expected to belong to one synset.

Word ***boqonnaa*** has two sense so expected to belong to two synset.

Word ***sad*** has one sense so expected to belong to one synset.

Word ***ban*** has one sense so expected to belong to one synset.

Word ***dub*** has one sense so expected to belong to one synset.

After the number of synset that belongs to a word checked word that belongs to more than one synset identified as ambiguous word. From the previous user input text the system expected to result word ***boqonnaa*** as ambiguous word. Then the word ***boqonnaa*** added to ambiguous word list and other words added to non-ambiguous word list. Final expected result from this component are Ambiguous word{ ***boqonnaa*** } and Non ambiguous word list { kitaab , sad, ban , dubb }.

Context checker and Sense Retrieval

This component is sub component of word sense disambiguation component. Context checker and Sense Retrieval accepts two types of list of words from **Ambiguous word identifier** sub component of word sense disambiguation component which are **ambiguous word list** and **non-ambiguous word list.** ambiguous words in ambiguous word list are used to be disambiguated in this phase and the non-ambiguous word list are used as information for disambiguation.

The overall Task in this phase are generalized as for each non-ambiguous word list in previous example { kitaab , sad, ban , dubb } we check there relation with ambiguous word list.

The step followed in this phase discussed below:

Step one :For all word in non-ambiguous word list { kitaab , sad, ban , dubb } from AWI

Step two: For ambiguous word list { boqonnaa } from AWI

Step three: Retrieve sense of each ambiguous word and expected result from this module will be sense of the word "boqonnaa" .The system will result N number of sense(S_1…..S_n). From the example the synset belongs to the word "boqonnaa" will retrieved.

In this phase we expected information about the belonging synsets which will be synset id and gloss definition.

synsetid	wordid	lemma	senseid	sens...	pos	definition
161	161	boqonnaa	164	1	ma	Afuura baafachuu ;Haaraga lfannaa
1113	161	boqonnaa	165	2	ma	Kutaa gurguddaa kitaabaa ,Qooddii kitaabaa

Fig 4.2: Expected synset information of ambiguous word "boqonnaa**"**

The word "boqonnaa" has single word id but belongs to different synset because it is a member of different synsets in other word it has multi sense which is polysemy.

In this step associated senses to ambiguous word retrieved.

S1 Afuura baafachuu ;Haaraga lfannaa

S2 Kutaa gurguddaa kitaabaa ,Qooddii kitaabaa

Step four: For each sense (S_1…..S_n) of ambiguous word in ambiguous word list retrieve associated clue word list(Cl_1…….. Cl_n)

In this case ambiguous word list { boqonnaa } for each sense of a word "boqonnaa" retrieve clue words lists. Each sense has its own clue word list. from previous step ambiguous word "boqonnaa " has two sense so it will have two list of clue words (no of clue word list = no of senses of ambiguous word) .

Cl1 of S1{ nama,da'umsaa,dhibee,heerumaa,jaart,afuura,tapha,dadhab,dhukkub,hoj,kubba miillaa,bal,ulf,deem,fuudh,ijooll,kubbaa,ciis,waaggaa,dheer,gabaab,jaars}

Cl2 of S2{ kut,walaloo,asii,diraamaa,kitaaba leenjii, qoosaa, barreef,karoor,kitaaba ,qulqulluu,qorannoo,cigoo,kitaab,kitaabaa,quraana,dabtara,kitaaba barataa}

Step five: in this step context bag of input text in our case non ambiguous word list are compared with retrieved clue word list then the overlap between clue word list of ambiguous word associated to each sense and the context bag of input text will be cheeked.

If overlap (NAW, cl1) > overlap (NAW, cl2) then best sense will be the sense attached to cl1 When we compare overlap between{ kitaab , sad, ban , dubb } and Cl1 as well overlap between { kitaab , sad, ban , dubb } and Cl2.

Cl_1 {nama,da'umsaa,dhibee,heerumaa,jaart,afuura,tapha,dadhab,dhukkub,hoj,kubbamiillaa,bal,ulf,deem,fuudh,ijooll,kubbaa,ciis,waaggaa,dheer,gabaab,jaars} *with* { kitaab , sad, ban , dubb } has no overlap :- context bag { kitaab , sad, ban , dubb } with Cl_1 has 0 overlap;

Cl_2 {kut,walaloo,asii,diraamaa,kitaabaleenjii,qoosaa,barreef,karoor,kitaaba,qulqulluu,qorannoo,cigoo,kitaab,kitaabaa,quraana,dabtara,kitaaba barataa} *with* { kitaab , sad, ban , dubb } has 1 overlap :- context bag { kitaab , sad, ban , dubb } with Cl_2 has 1 overlap so the sense of clue word list with maximum overlap selected as best sense in this case sense of Cl2 which is S2has maximum overlap .

Finally the gloss of the sense with maximum number of clue word overlap selected . The system expected to result gloss associated with S2: " Kutaa gurguddaa kitaabaa ; qooddii kitaabaa".

4.3 Afaan Oromo WordNet preparation

The origin of WordNet is to build a lexical-conceptual model, consisting of both lexical units and the relations between such units, structured into a relational semantic network. For Afaan oromo language, we don't have previously-constructed WordNet, so we constructed the WordNet manually by following the principle of English WordNet structure. The WordNet consists of words with their synsets and gloss definitions for each sense of a given word. We have used Afaan Oromo dictionary that was prepared by Addis Ababa University entitled "Galmee Jechota Afaan Oromo"[47], hinshene's dictionaries entitled Dungo Oromo-Amharic- English dictionary [31] and Bakkalcha English–Oromo dictionary[48]. Although there are different relationships between words, for this research we have basically used synonymy and clue word relationships to identify the meaning of an ambiguous word.

Unlike English WordNet, our Afaan oromo WordNet has 9 tables. We eliminated some tables from the English WordNet and added some feature to it because English WordNet database was constructed for general NLP purpose not focusing on word sense disambiguation task only. There are few cross-POS relations in English WordNet but we used many cross-POS relations in this work like clue noun, clue verb relation. The main relation among words in WordNet is synonymy[12]. Synonyms words that denote the same concept and are interchangeable in many contexts are grouped into unordered sets (synsets). The organization of a single sense is also done in the same way. Figure 4.3 depicts Afaan Oromo WordNet tables. The Afaan Oromo WordNet has 50 polysemy word and contain 1175 synsets and 1,105 word. Sample polysemy words in Afaan Oromo WordNet is given in Annex C. During preparation of Afaan Oromo WordNet we collect 500 polysemy words and randomly select 50 polysemy words from them. This 50 polysemy words are checked by linguistic expert they are ambiguous or not. Additionally clue words belongs to each sense of ambiguous word manually gathered from different resources and there belongingness to each sense also checked so approved list of ambiguous word, their sense with clue words are used in this work.

The description of Afaan Oromo WordNet tables is given below.

The Synset Table -The synsets table is one of the most important tables in the AO WordNet database. It is responsible for housing all the definitions within WordNet. Each row in the synset table has a synsetid, a definition, a pos (parts of speech field) There are over 1,175 synsets in our AO WordNet database.

The Words Table - WordNet also has a "words" table, that only has two fields: a "word id", and a "lemma". The words table is responsible for housing all the lemmas (base words) within the WordNet database. There are 1,105 words in in the AO WordNet database.

The Sense Table - The sense table is responsible for linking together words (in the words table), with definitions (in the synset table). The entries in the sense table are referred as "word-sense pairs" - because each pairing of a word id with a synset is one complete meaning of a word - a "sense of the word". There are a total of 1,158 word senses in the AO WordNet database.

linktypes Table - defines all relation (link) types used in WordNet, In this work we have a lot of link types noun to noun, noun to verb, noun to adjective, noun to adverb. We use many cross pos relation unlike English WordNet. We call it this relation type as clue word.

Lexlinks Table - lexical links, i.e., relations between words. Example: horii - horiif (derivation)

semlinks Table - semantic links, i.e. relations between synsets. Example: horii - qarshii (synonymy)

postypes Table - defines "parts of speech". Contains only the following values: Ma – noun, Go –verb, Add– adjective, Dab – adverb.

Morphmaps - only base forms of words are usually stored in WordNet, searches may be done on inflected forms. A set of morphology functions , morphy , is applied to the search string to generate a form that is present in WordNet. This table maps many forms of morphs to their root or base form.

Morphs –this table stores all morphemes types that indicate single base word. Example morphs horii, horiin, horiif mapped to lemma horii.

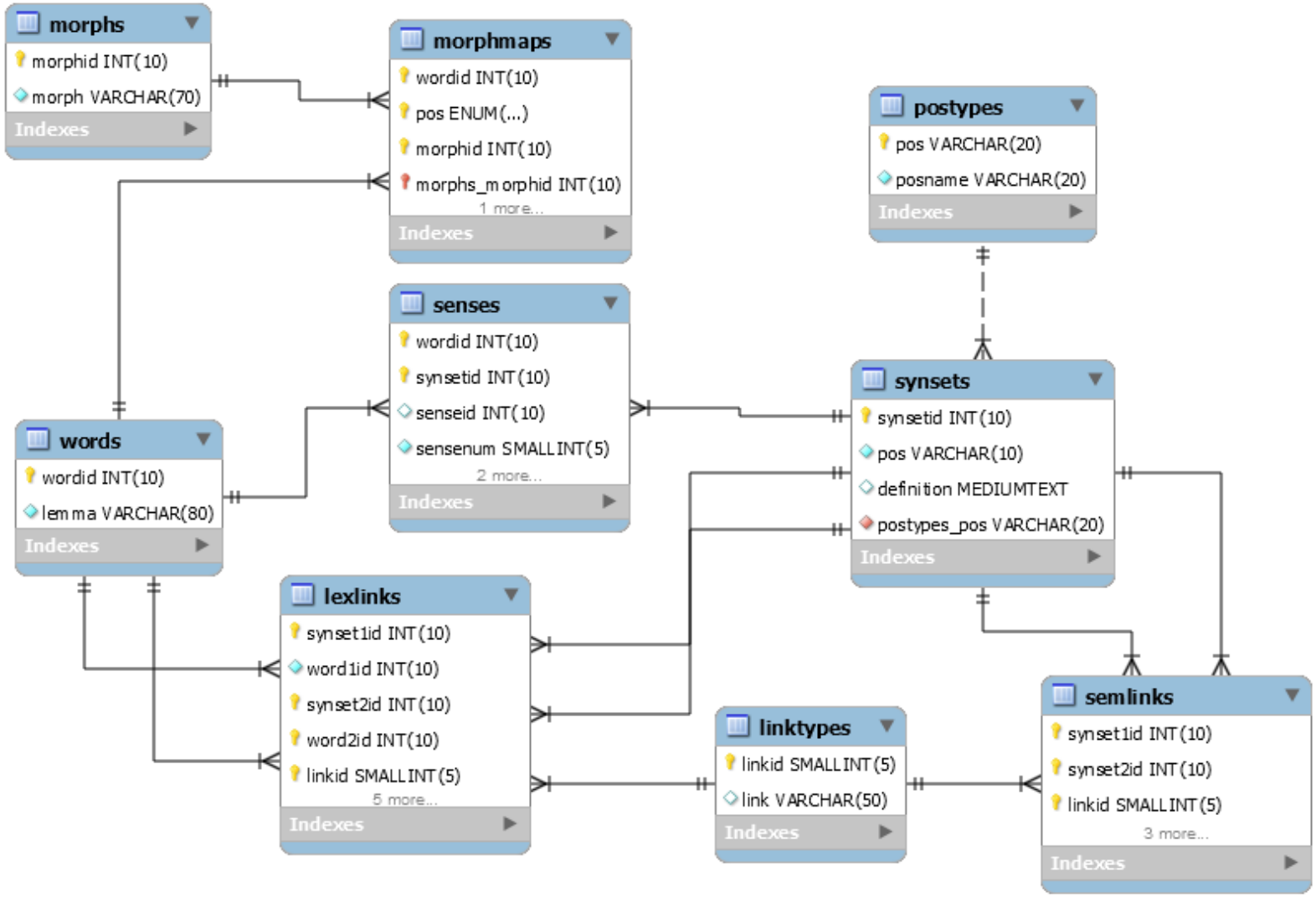

Fig 4.3: AO WordNet Schema tables

Chapter five: Implementation and Evaluation

5.1 Introduction

The main goal of any WSD research is to resolve ambiguity of different senses listed in a dictionary, thesaurus or another source of a given word. Similarly, this research attempts to find a way to automatically determine the sense of ambiguous Afaan Oromo words in a given context. For this purpose, we have developed an Afaan Oromo WSD system prototype. As described in the literature review section there are different approaches for WSD, however ,we have selected the knowledge based approach which checks overlap and relation of words. In this thesis the WSD system prototype we have developed has Three main parts: preprocessing ,Morphological Analysis and word sense disambiguation additionally AO WordNet which is manually constructed. As discussed in the previous chapter, the sense of an ambiguous word in a given sentence can be identified by checking overlap and relation between the words in input sentence and list of related words to ambiguous word in the WordNet. To check relation of words, we used different relation types like synonymy ,hypernym , meronym in addition to this basic relation that was used for English WordNet data base we used cross pos relations like clue noun, clue verb etc. Many word that has different types of relation to ambiguous word gathered in this work and defined as clue words .

5.2 Prototype

5.2.1 Development Tools

Java programming language has been used to develop the prototype because Java is dynamic in nature and it can be run in any platform and MySQL server is used to develop Afaan Oromo WordNet.

5.2.2 Snapshots

Figure 5.1 shows the interface of AOWSD system. our AOWSD system has Three main components which are Preprocessing, Morphological analysis and Word Sense Disambiguation.

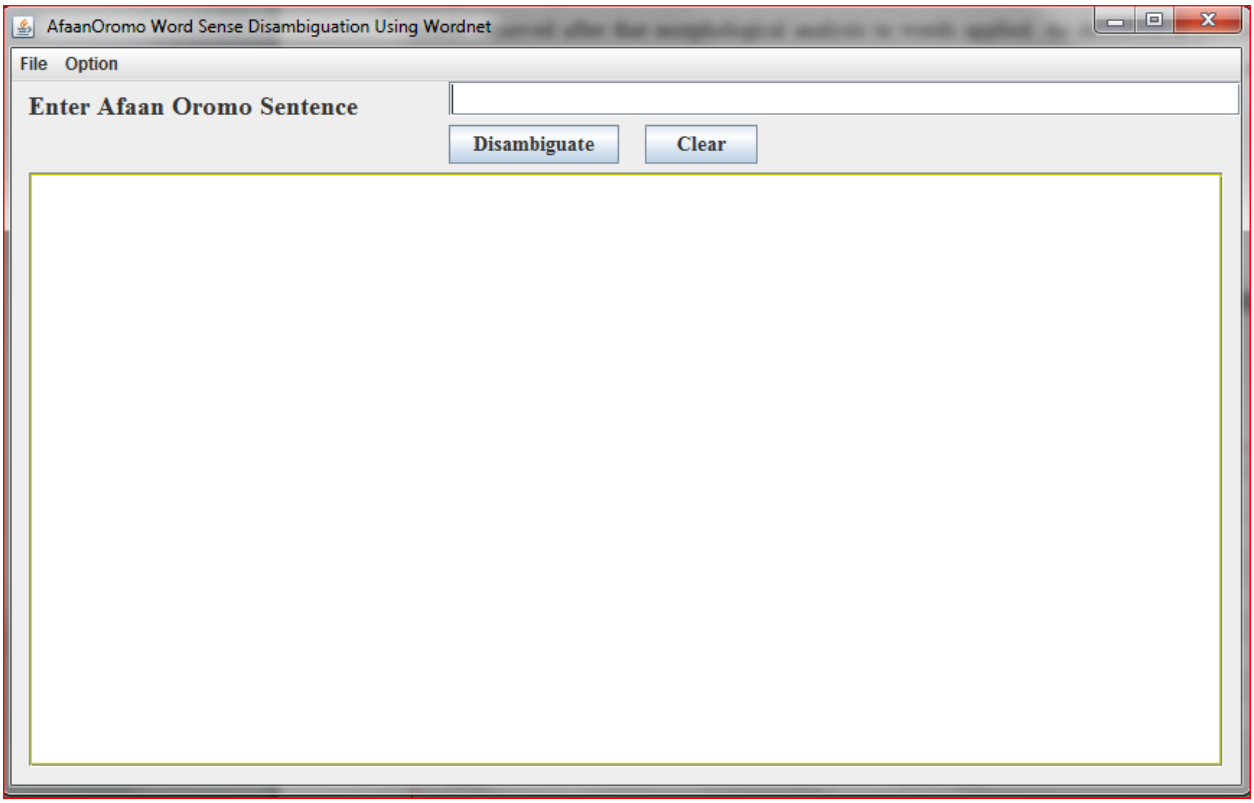

Figure 5.1: Interface of AOWSD system

Preprocessing: In this component of AOWSD input sentence from user tokenized then normalized after that stop words removed.

Morphological analysis: After preprocessing morphological analysis to words applied. As described in previous chapter ,we uses manually constructed morphological analysis using morph table in our AO WordNet. Figure 5.2 shows sample snapshot of the morphological analysis component.

Resultset 1 | Resultset 2

lemma	pos	morph
dand	n	danda'an
dand	n	danda'u
dand	n	danda'a
dand	n	danda'u
dand	n	danda'u'
dand	n	dandeessa
dand	n	dandeessaa
dand	n	dandeesse
dand	n	dandeessee
dand	n	dandeessi
dand	n	dandeessisa

Figure 5.2:Sample result set Screenshot for morphology analysis sub Component of AOWSD

Word Sense Disambiguation : This phase has two sub components which are

1. Ambiguous Word Identifier
2. Context checker and Sense Retrieval

Ambiguous Word Identifier: Which identifies Ambiguous word by counting sense of a given word. If a given word belongs to more than one synset this component set it as an ambiguous word. Sample screen shoot image for this component is depicted in Figure5.3.

During preparation of Afaan Oromo WordNet, we collected around 500 ambiguous word. As working on all the words is difficult , We randomly selected 50 polysemy words.

Resultset 1 | Resultset 2

lemma	pos	sensenum	synsetid	SUBSTRING(definition FROM 1 FOR...
horii	ma	1	551	Maallaqa
horii	ma	2	1143	Beellada manaa

Figure5.3:Sample result set for Ambiguous Word Identifier subcomponent of AOWSD

As we see from figure 5.3 single word **horii** belongs to two synset so our Ambiguous Word Identifier module select the word horii as ambiguous.

Context checker and Sense Retrieval: checks relation of clue words list with input sentence context words. First, we find collection of clue words or indicators for each sense of a polysemy word. If a word has two senses ,it will have two sets of collection of clue words that are belongs to two different sense of that word .After this, we find the collection of words from the context window as non-ambiguous word list. Finally, it checks to which sets of collection of clue words the collection of words from the context window overlaps. Figure 5.4 depicts this context checker and sense retrieval subcomponent of word sense disambiguation.

During preparation of clue words we collected context word from the corpus and the words that are most frequently co-occurring and also has a relation with ambiguous word. In our AOWSD system, we try to link co-occurring words to the appropriate sense of ambiguous word.

Resultset 1 | Resultset 2

ssensenum	lemma	link	linkedlemma	SUBSTRING(sdefinition FROM 1 ...
1	horii	clue words	birrii	Maallaqa
1	horii	clue words	gurgur	Maallaqa
1	horii	clue words	qus	Maallaqa
1	horii	clue words	bitanii gurguruun	Maallaqa
1	horii	clue words	liq	Maallaqa
2	horii	clue words	fonii	Beellada manaa
2	horii	clue words	kazoo	Beellada manaa
2	horii	clue words	bineensa	Beellada manaa
2	horii	clue words	nama	Beellada manaa
2	horii	clue words	ciis	Beellada manaa

Figure 5.4:Sample result set Screenshot for Context checker and Sense Retrieval subcomponent

- Generally Figure 5.5 and Figure 5.6 shows how the system analyzes the sense of ambiguous word “aars”

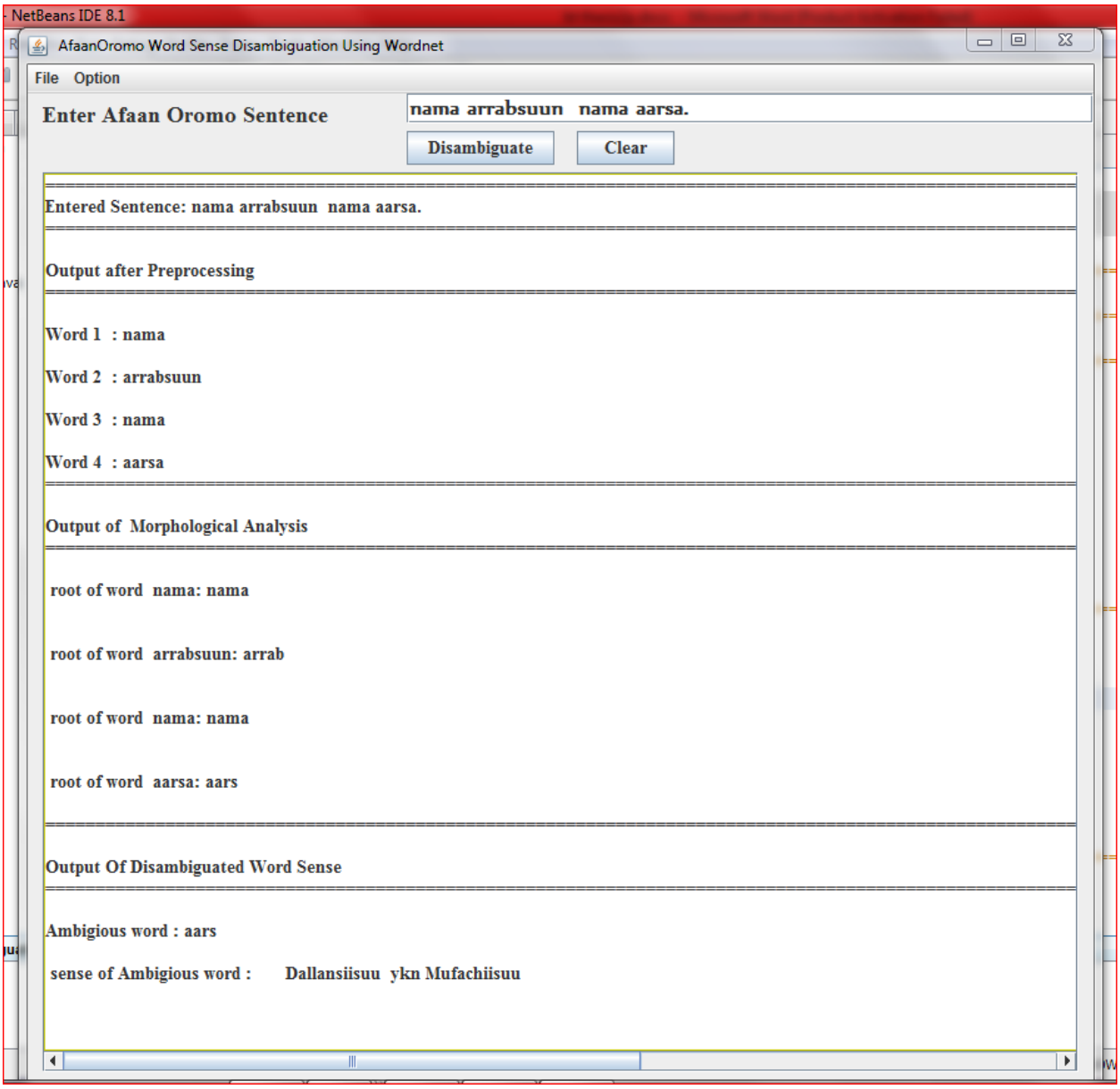

Figure 5.5:Analysis of the sentence “nama arrabsuun nama aarsa.”

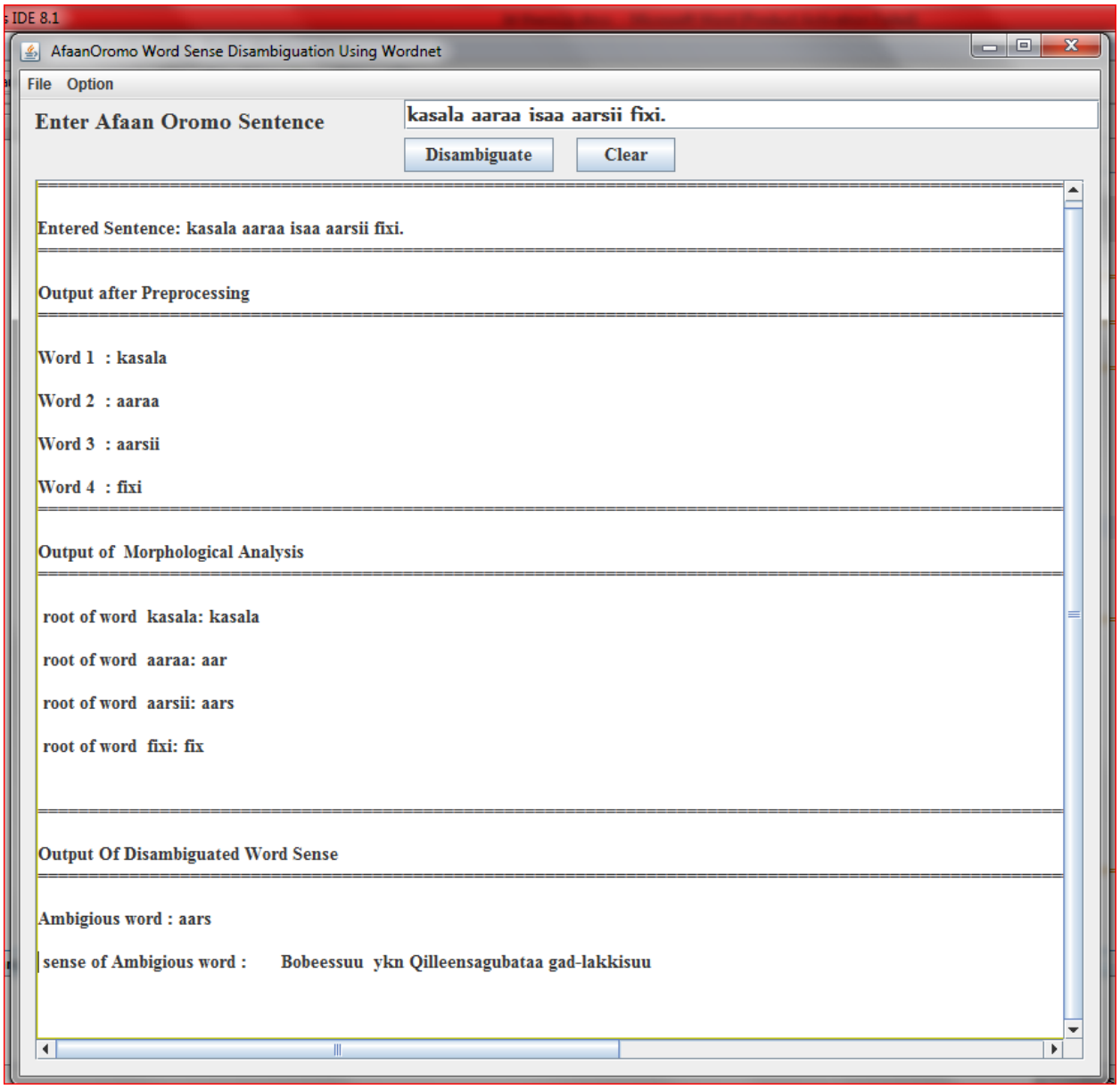

Figure 5.6: Analysis of the sentence “kasala aaraa isaa aarsii fixi.”

5.3 Evaluation

In this section, experiments are conducted to evaluate the performance of the proposed approach. Evaluating the performance of the WSD system is an important part of the research, which discusses the actual work of the research. However, it is a very difficult task since there is no standard rule for WSD evaluation for all languages. We performed an evaluation of the proposed knowledge-based WSD algorithm accuracy. We have conducted experiments on 50 ambiguous words. The system is evaluated based on evaluation measures. Evaluation measures are the assessment of word sense disambiguation systems usually performed in terms of evaluation measures borrowed from the field of information retrieval [7]. Based on this we used the metrics such as precision P, recall R, F1-measure and accuracy.

1. Precision determines how good are the answers given by the system being assessed. The precision P of a system is computed as the percentage of correct answers given by the automatic system, that is:

$$\text{Precision (P)} = \frac{TP}{TP+FP}$$

2. Recall R is defined as the number of correct answers given by the automatic system over the total number of answers to be given:

$$\text{Recall (R)} = \frac{TP}{TP+FN}$$

3. Measure which determines the weighted harmonic mean of precision and recall, called the F1-measure or balanced F-score, is defined as

$$\text{F1- Measure} = \frac{2*P*R}{P+R}$$

4. $\text{Accuracy} = \frac{TP+TN}{Pt+Nt}$

Where TP , TN, FP and FN refer to true positives, true negatives, false positives and false negatives respectively and Pt and Nt refer to the total number of positive and negative examples in the test set respectively. In a binary class based classification context, the terms positive and negative are associated with membership to sense.

5.3.1 Preparation of Test Dataset

For testing task we have prepared test dataset. The prepared test data set has 138 test sentences which have a number of four sentence in average for each polysemy word. The test data set collected by the help of linguistic experts from Afaan Oromo linguistic department. Annex D shows sample test data set.

5.3.2 Experimental Results

We run the system using 138 test sentences. In the experiment, we found that out of the polysemy words in 138 test sentences 128 are correctly disambiguated. This shows the accuracy of the system with sample Afaan oromo WordNet found to be 92%. About 10 % polysemy words are not correctly Disambiguated basically due to the reason there is no standard automatic morphological analyzer for Afaan oromo language.

Table 5.1 assessment result of AO word sense disambiguation systems

Evaluation measures	**Value**
Precision	90%
Recall	95%
F1-measure	92.4%
Accuracy	92%

Chapter six: Conclusion and Future Works

6.1 Conclusion

The main aspiration of this research work is to investigate a more generic approach for all-words based Afaan oromo WSD. Previous thesis works in the area of WSD conducted a study on lexical sample task and their experiments were on restricted part of speech. In this research, we explored a new method towards all words task by using knowledge based approach which uses Afaan Oromo WordNet relations of words. To this end, we developed a WSD system based on implementation of knowledge based approach using Afaan oromo WordNet. This research work is the first attempt to develop a word sense disambiguation system for Afaan Oromo language using Afaan Oromo WordNet. Since there is no linguistic resources prepared i.e. WordNet, thesaurus, machine readable dictionaries and others for Afaan Oromo Language, which is important for WSD purpose, we prepared Afaan Oromo WordNet manually for this study.

Our solution depends on the capability of the program to infer the sense of an ambiguous word based on the ambiguous word's relation to clue words . This makes our solution applicable to a huge amount of knowledge and robust in its ability to disambiguate unforeseen cases as more knowledge is acquired. Resolving an ambiguous word based on the ambiguous word's relation to clue words is possible when any words that indicate the sense of ambiguous word collected as clue word without restricting the relation among word on predefined single POS relations only. This work used many single POS as well as cross POS relations between words in AO WordNet. Using those relations best suits for knowledge-based WSD algorithms resulting in higher accuracy. Moreover, this reduces the computational effort for the system and saves the system memory while processing. Thus, for an accurate disambiguation, selecting the appropriate context is crucial.

During the preparation of Afaan Oromo WordNet , we have selected 1105 words including 50 ambiguous words. Based on these ambiguous words, we extracted many clue words this clue words are belongs to different senses of ambiguous word. The proposed system has three component and AO WordNet as knowledge resource. The three components are preprocessing, morphological analysis and word sense disambiguation.

The Java language was used to implement the system, and MYSQL DBMS used to construct AO WordNet. To evaluate the proposed system, a test corpus is constructed by collecting texts from various resources. 138 selected Afaan Oromo sentences collected as test sentence afterwards checked by language experts. The accuracy of the system tested using dividing the number of sentences correctly disambiguated, over the total number of sentences in the test examples results 92 %. Previous systems focused only on target words that means system disambiguate only target words to do so training set instance(labeled or unlabeled form) gathered for only single target words and the target word is mostly from single word class (single part of speech) as opposed to this our afaan oromo word sense disambiguation system disambiguate all words from user input text and unrestricted .

The result of this study will produce experimental evidences that demonstrate the use of WordNet by connecting words with clue words that can sufficiently disambiguate the meaning of the polysemy word rather than connecting words by semantic relations for the development of Afaan oromo WSD system. The study contribute for future researches and development in the different areas of NLP as WSD is an intermediate task for many other NLP applications. In addition, this study show the possibility of applying a WSD system that would work for all Afaan Oromo word classes and for all words which will have a way for future researchers to investigate and study more on all-words WSD for Afaan Oromo. Morphological analysis and Afaan Oromo WordNet are also the significance of this system.

6.2 Future Works

The method utilized within this thesis is that the WordNet is very useful resource for language learning and language processing tasks. Organizing the different senses of polysemy words and context words based on clue words best suits for knowledge-based WSD algorithms resulting in higher accuracy. Moreover, this reduces the computational effort for the system and saves the system memory while processing. Word sense disambiguation researches require variety of linguistic resources like thesaurus, WordNet and machine readable dictionaries in which we faced a significant challenge as Afaan Oromo language lacks those resources.

The main task behind word sense disambiguation is identifying the appropriate sense of a given ambiguous word in input text. However there is no standard collected list of ambiguous word and there sense in Afaan Oromo Language .We gathered ambiguous words and sense indicating

words from different dictionary in this thesis work. For identifying the sense of ambiguous words we used words that frequently co-occur with ambiguous word as indicator for appropriate sense. Previously these resources are not available identifying ambiguous word and the relations between words were challenging. Therefore, we forward the following future works for WSD for Afaan Oromo texts:

- Researches in WSD for other language use linguistic resources like thesaurus and machine readable dictionaries. For Afaan Oromo language those resources are not yet been developed. We recommend those resources to be included in the future work.
- The Afaan Oromo WSD developed identifies word senses using information from WordNet. We have tried to construct a simple WordNet that contains small number of linked or related words which affects greatly the performance of the Afaan Oromo WSD. No full-fledged Afaan Oromo WordNet is available and constructing it manually is tedious. Constructing such lexical knowledge base for WSD is important and time efficient. Hence we recommend to work more on constructing an Afaan Oromo WordNet.
- Since WSD is a central problem in the field of NLP, different attempts are being done in this area for other local languages using different approaches. For the future ,we suggest for researchers who are interested to work on WSD for other language to follow this approach and to explore more.
- We study WSD for words that have semantic type of ambiguity which is polysemic. For the future, we recommend that researchers work on disambiguation of other types of ambiguities in Afaan Oromo language.
- There are many polysemy words in Afaan Oromo language. This study includes 50 polysemy words. For the future, we suggest that researchers work on other polysemy words.
- Morphological analyzer is vital component for morphological rich language like afaan oromo. This study uses manually developed morphological analysis. We recommend the development of this resource to enhance word sense disambiguation.

References

[1] Charniak, Eugene:" Introduction to artificial intelligence", page 2. Addison-Wesley, 1984 .

[2] B. S.P. Mishra, Satchidananda Dehuri, Euiwhan Kim, Gi-Name Wang."Techniques and Environments for Big Data Analysis: Parallel, Cloud, and Grid" pp 60 .

[3] Alok Ranjan Pall and Diganta Saha2, "Word Sense Disambiguation: A Survey," IJCTCM, vol. 5, no. 3, pp. 1-16, July 2015.

[4] Andres M., Armando S., German R., Manuel P., "Combining Knowledge and Corpus-based Word Sense Disambiguation Methods," *Journal of Artificial Intelligence Research*, vol. 23, no. 1, pp. 299-330, January 2005.

[5] Tesfa Kebede, Word Sense Disambiguation for Afaan Oromo Language, u*npublished Master's Thesis*, Department of Computer Science, Addis Ababa University, Addis Ababa, 2013.

[6] Workineh Tesema, Word Sense Disambiguation for Afaan Oromo Language, u*npublished Master's Thesis*, Department Of information technology, Jimma University, Jimma, 2014.

[7] Navigli Robert, "Word Sense Disambiguation: A Survey," *ACM Computing Surveys*, vol. 41, no. 2, pp. 1-69, February 2009.

[8] Nancy Ide and Jean Veronis , "Word Sense disambiguation: The state of the Art," *Computational Linguistics*, vol. 24, no. 1, pp. 2-40, March 1998.

[9] ROGET, P. M. *Roget's International Thesaurus*, 1st ed. Cromwell, New York, NY, 1911.

[10] GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing. In *Proceedings of the International Workshop on Formal Ontology* (Padova, Italy),1993.

[11] Miller, G.A., "WordNet: A Lexical Database", Comm. ACM, Vol. 38, No. 11, Pp. 39-41, 1993

[12] https://wordnet.princeton.edu/ Retrieved:9/19/2017.

[13] *E. Agirre and P. Edmonds (eds.), Word Sense Disambiguation: Algorithms and Applications,* 107–131. © 2007 *Springer.* (book)

[14] Lesk, M. . *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone.* In SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation, pages 24-26, New York, NY, USA. ACM.,1986.

[15] Banerjee, S., Pedersen, T., "An adapted Lesk algorithm for word sense disambiguation using WordNet", In Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, February 2002.

[16] Kilgarriff and J. Rosenzweig. 2000. English SENSEVAL:Report and Results. In Proceedings of the 2nd International Conference on Language Resourcesand Evaluation, LREC, Athens, Greece.

[17] Mittal, K. and Jain, A.,"word sense disambiguation method using semantic similarity measures and owa operator", ictact journal on soft computing: special issue on soft –computing theory, application and implications in engineering and technology, january, 2015, volume: 05, issue: 02.

[18] Rada, Roy, Hafedh Mili, Ellen Bicknell & Maria Blettner. 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man,and Cybernetics*, 19(1): 17–30.

[19] Diana, M.C., Carroll, J., "Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Selectional Preferences", Computational Linguistics, Volume 29, Number 4, pp. 639-654.

[20] Patrick, Y. and Timothy, B.,(2006) "Verb Sense Disambiguation Using Selectional Preferences Extracted with a State-of-the-art Semantic Role Labeler", Proceedings of the 2006 Australasian Language Technology Workshop (ALTW2006), pages 139–148.

[21] Gerard Escudero Bakx, *Machine Learning Techniques For Word Sense Disambiguation*. Barcelona, Spain, 2006.

[22] Parameswarappa, S. and Narayana V.N,(2013) "Kannada Word Sense Disambiguation Using Decision List", Volume 2, Issue 3, May – June 2013, pp. 272-278.

[23] Singh, R. L., Ghosh, K. , Nongmeikapam, K. and Bandyopadhyay, S.,(2014) "A decision tree based word sense disambiguation system in manipuri language", Advanced Computing: An International Journal (ACIJ), Vol.5, No.4, July 2014, pp 17-22.

[24] Le, C. and Shimazu, A.,(2004)"High WSD accuracy using Naive Bayesian classifier with rich features", PACLIC 18, December 8th-10th, 2004, Waseda University, Tokyo, pp. 105-114.

[25] Aung, N. T. T., Soe, K. M., Thein, N. L.,(2011)"A Word Sense Disambiguation System Using Naïve Bayesian Algorithm for Myanmar Language", International Journal of Scientific & Engineering Research Volume 2, Issue 9, September-2011, pp. 1-7.

[26] McCulloch and Pitts , MCCULLOCH, W. AND PITTS, W. A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 5, 115–133, 1943.

[27] Brody, S., Navigli, R., Lapata, M.,(2006) "Ensemble Methods for Unsupervised WSD", Proceedings the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 97–104, Sydney, July 2006.

[28] Buscaldi, D., Rosso, P., Pla, F., Segarra, E. and Arnal, E. S.,(2006)"Verb Sense Disambiguation Using Support Vector Machines: Impact of WordNet Extracted Features", A. Gelbukh (Ed.): CICLing 2006, LNCS 3878, pp. 192–195.

[29] Niu, C., Li, W., Srihari, R. K., Li, H., Crist, L.,(2004) "Context Clustering for Word Sense Disambiguation Based on Modeling Pairwise Context Similarities", SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, July 2004.

[30] Tilahun Gamta . "Qube Afaan Oromo: Reasons for Choosing the Latin Script for Developing an Oromo Alphabet", The Journal of Oromo Studies 1(1),1993.

[31] Hinsene Mekuria.. Dungoo, " Oromo-Amharic-English Dictionary", Addis Ababa, Elleni p.p.Plc,2012

[32] Debela Tesfaye. "Designing a Stemmer for Afan Oromo Text: A hybrid approach", Master's thesis, School of graduate studies, Addis Ababa University, Ethiopia, 2010.

[33] Abebe Abeshu, "Automatic morphological synthesizer for Afaan Oromo", Master's thesis, Faculty Natural science, Addis Ababa University, 2010

[34] Baskaran Sankaran, k. Vijay-Shanker, Influence of morphology in word sense disambiguation for Tamil, Anna University and University of Delaware Proceedings of International Conference on Natural Language Processing, 2003.

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

[36] Solomon Mekonnen, Word Sense Disambiguation for Amharic words, A Machine Learning Approach, u*npublished* Master's Thesis, Addis Ababa University, 2010.

[37] Solomon Assemu, Unsupervised machine learning approach for Word Sense Disambiguation to Amharic words, u*npublished* Master's Thesis, Addis Ababa University, 2011.

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

[39] Segid Hassen ,Amharic Word Sense Disambiguation Using WordNet, u*npublished* Master's Thesis, Department of Computer Science, Addis Ababa University, Addis Ababa, 2015.

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

[41] Kumar, R., Khanna, R.,(2011) "Natural Language Engineering: The Study of Word Sense Disambiguation in Punjabi", Research Cell: An International Journal of Engineering Sciences ISSN: 2229-6913 Issue July 2011, Vol. 1, pp. 230-238.

[42] M. Sinha, M. K. Reddy, P. Bhattacharyya, P. Pandey, and L. Kashyap, "Hindi word sense disambiguation," Master's thesis, Indian Institute of Technology Bombay, Mumbai, India, 2004.

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

[44] U. R. Dhungana and S. Shakya, "Word sense disambiguation in nepali language," in *The Fourth International Conference on Digital Information and Communication Technology and Its Application (DICTAP2014)*, Bangkok, Thailand, 2014, pp. 46–50.

[45] Udaya R Dhungana, Subarna Shakya, Kabita Baral and Bharat Sharma, "word sense disambiguation using wsd specific wordnet of polysemy words" International Journal on Natural Language Computing (IJNLC) Vol. 3, No.4, August 2014

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

[47] Kebede H ,Tesfaye F,. “Galmee Jechoota Afaan Oromo”, Wiirtuu Qo’annoofi Qoranno Afaanota Itoophiyaa,Yunversitii Finfinnee, Finfinnee,1999.

[48] Hinsene Mekuria, Bakkalcha,“ Oromo-English Dictionary”, Addis Ababa, Commercial Printing Enterprise ,1989.

Annexes

Annex A: Afaan Oromo Alphabet

Basic letters

Aa	Hh	Oo	Vv
Bb	Ii	Pp	Ww
Cc	Jj	Qq	Xx
Dd	Kk	Rr	Yy
Ee	Ll	Ss	Zz
Ff	Mm	Tt	
Gg	Nn	Uu	

Compound letters

CH	ch
DH	Dh
NY	Ny
PH	Ph
SH	Sh
TS	Ts
ZH	Zh

Annex B: stop word lists

akka	iseen	Simmoo
akkam	ituu	sun
Akkasumas	ituullee	Tanaafi
Akkuma	jechaan	tanaafuu
Ammo	Jechuun	Utuu
Booda	Kan	Waggaa
dura	Kanaaf	yoo
Eega	Kanaafi	yookiin
fi	odoo	yoom
Garuu	Ofii	Akum
Hanga	Oggaa	ani
Henna	oo	booddee
Hogguu	Osoo	eegana
Hoo	Otoo	eegasii
Illee	Otumallee	Ennaa
innaa	saniif	erga

Hoggaa	kun	tawullee
immoo	Malee	waan
ini	Moo	Woo
isaa	Otuu	Yammuu
Isaan	otuullee	Yemmuu
Itumallee	silaa	Yeroo
Jechuu	ta`ullee	Yommii
kanaafuu	Tahullee	Yommuu
Kee	Tanaaf	Yookaan
koo	tanaafuu	Yookinimoo

Annex C :Sample Polysemy words In Afaan Oromo Language

word id	Word	pos	Root	Senses	Definition
1	Aarsuu	adj	aars	1	Bobeessuu dhaan qilleensa gubataa kan gadi lakkisuu
				2	Dallansiisuu
2	Baasaa	n	baas	1	dhukkuba garaa
				2	arii'aa, mana gadi baasaa
3	Bituu	v	bit	1	waan gurguramu gatii kennanii fuudhuu.
				2	Bulchuu
4	Boqonnaa	n	boqonnaa	1	afuura baafacha, bayyanacha.
				2	iddoo itti afuura baafatan, bayyanatan.
				3	kutaa guguddaa kitaabni itti qoodamu.
5	Bulchaa	n	bul	1	hogganaa, ajajaa,nama biyya bulchu
				2	Nyaata
6	Buufachuu	v	buuf	1	Waraabbachuu
				2	Gad-fudhachuu
				3	Bu’aa argachuu
7	Caamsaa	n	caamsaa	1	aduun gar malee o'u, Horiin ganna beela'e hinbawu.
				2	maqaa ji' Eblaafi Waxabajjii gidu oolu.
8	Caffee	n	caffee	1	marga lafa jiidha qabutti margu;marga jiidha qabu.
				2	mana maree ummpta Oromoo. ; mana maree bakka bu'oota

					uummataa.
9	Daakuu	v	daak	1	caccabsuu, bulleessuu; Midhaan harkaan daakuun har'a hafaa jira; waan dhagaatti, baaburatti yookiin waan biraatti aakamee buddeen, marqaan, daabboon, w. k . f " , irrraa qophaawu ,
				2	Bishaan
10	Filuu	v	fil	1	rifeensa filaa dhaan wal qixxeessu.
				2	kan dhiyaate keessaa qobaatti baasuu, fo'uu.

Annex D: Sample test sentences

Word	Test sentences
Aars	1. akkeessama nama aarsuudhaaf waan inni jedhe jecha yookiin waan inni godhe gocha .
	2. nama arrabsuun amala ibsa nama aarsa.
	3. kasala aaraa isaa aarsii fixi.
	4. inni hoggayyuu sigaaraa aarsa.
	5. inni hoggayyuu sigaaraa aarsa.
Baas	6. waan isa dhukkubuuf amma amma baasaa.
	7. garaa kaasaa waan qabuuf isa baasisa
	8. gurba sana manaa baasaa.
	9. namni adabbii fixate mana hidhaa irraa bahuun barbaachisa dha
Bit	10. inni uffata bituuf gabaa deeme.
	11. mana bituu barbaadee maallaqni isa hanqate.
	12. biyyattin abbaa irree sirna soshaalizimii leellisuun bitama turte.
	13. biyyootni awurooppaa aardii afrikaa kolonii isaanii godhatanii humnaan bituuf baayyee carraaqan.
Boqonnaa	14. kitaaba kee boqonnaa sadaffaa banii dubbisi.
	15. hawwii kitaaba qulqulluu boqonnaa jalqabaa barsiisi .

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged.

Declared by:

Name: _____

Signature: _____

Date: _____

Confirmed by advisor:

Name: _____

Signature: _____

Date: _____

Confirmed by advisor:

Name: _____

Signature: _____

Date: _____