Addis Ababa University
College of Natural Sciences
Department of Computer Science

WORD SENSE DISAMBIGUATION FOR AFAAN OROMO LANGUAGE

By: Tesfa Kebede Hundesa

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

Addis Ababa, Ethiopia
November, 2013

Addis Ababa University
College of Natural Sciences
Department of Computer Science

WORD SENSE DISAMBIGUATION FOR AFAAN OROMO LANGUAGE

By: Tesfa Kebede Hundesa

Advisor: Dida Midekso (PhD)

Signature of the Board of Examiners for Approval

Name	Signature
1. Dida Midekso (PhD), Advisor	_____
2. Dejene Ejigu (PhD), Examiner	_____
3. _____	_____
4. _____	_____
5. _____	_____

Dedicated to

My family, specially my brother Netsanet Kebede and My mother, Alemitu Bedasa, who have raised me to be the person I am today.

Acknowledgements

First of all, I would like to thank my GOD for all ups and downs, for every success in my life, for giving me the wisdom and the strength I need to discharge my duty. Secondly, I would like to thank my advisor **Dr. Dida Midekso**, for his critical comments on my work, for being my driving force throughout this thesis and his patience in helping me to complete my work. I would also thank my advisor for freedom he gave me to pursue my own interests.

I would like to thank Debela Tesfaye, for his source code on stemmer for Afaan Oromo texts. I would like to thank also students of Afaan Oromo department, Addis Ababa University, for their cooperation in the linguistic aspect during the design of the corpus.

My foremost gratitude goes to my class mate, Fiseha Berhanu, for being my good friend as project/group work partner during our graduate study, for initiating the research idea and giving me extremely useful technical assistance to complete this thesis. I would also thank my class mate and my life partner, Abebe Basazinew, for his encouragements and valuable support he gave me to complete my study.

My special thank goes to my brothers, Netsanet Kebede for his extreme support and encouragement in difficult times to become my life partner, and also, for passing all those hardships I had to go through easily. Really, I am lucky to have a brother like you. May God bless you for what you did and is doing for me and for the rest of our families. I would also thank my family for their encouragement and support specially my mother *Alemitu Bedasa*, thank you that you sent me to school.

I offer my regards and blessings to all of those who supported me in any respect during the completion of the thesis as well as expressing my apology that I could not mention all.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

ANC	American National Corpus
AI	Artificial Intelligence
AWI	Ambiguous Word Identifier Module
ATC	Automated Text Categorization
BNC	British National Corpus
CCT	Context Count Table
CCM	Context Counter Module
CEM	Context Extractor Module
CSA	Central Statistical Agency of Ethiopia
DT	Determiners
EM	Expectation Maximization
IA	Inter-annotation Agreement
IR	Information Retrieval
IE	Information Extraction
IDF	Inverse Document Frequency
JJ	Adjectives
KNN	K-Nearest Neighbor
LSA	Latent Semantic Analysis
LEXAS	LEXical Ambiguity-resolving System
LDOCE	Longman Dictionary of Contemporary English
MRD	Machine Readable Dictionary
MT	Machine Translation
ML	Machine Learning
NL	Natural Language
NLP	Natural Language Processing
NLG	Natural Language Generation
NN	Nouns
NP	Noun Phrase
NBC	Naive Bayes Classifier

OL	Ontology Learning
POS	Part-of-speech
SM	Semantic Mapping
SA	Semantic Annotation
SIM	Sense Identifier Module
SCM	Sense Couter Module
SR	Speech Recognition
SVD	Singular Value Decomposition
SCT	Sense Count Table
SVM	Support Vector Machines
VBD	Verbs
VP	Verb Phrase
WSD	Word Sense Disambiguation
WSJ	Wall Street Journal

ABSTRACT

This thesis 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.

Due to lack of sense annotated text to be able to do these types of studies; a total of 1240 Afaan Oromo sense examples were collected 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.

A standard approach to WSD is to consider the context of the ambiguous word and use the information from its neighboring or collocation words. 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 achieved result was encouraging, but further experiments for other ambiguous words and using different approaches will be needed for a better natural language understanding of Afaan Oromo language.

Keywords: Natural Language Processing, Word Sense Disambiguation, Supervised Learning Method, Naïve Baye's theorem.

Chapter One: Introduction

1.1. Background

Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications. There are in fact two distinct focuses of NLP: language processing and language generation. The first one refers to the analysis of language for the purpose of producing a meaningful representation, while the latter refers to the production of language from a representation [16]. The goal of NLP is to design and build software that will analyze, understand, and generate languages that humans use naturally. To achieve this, natural language system needs to acquire extensive knowledge about the world which is not easy to acquire.

Ambiguity is defined as the presence of two or more possible meanings in any sentence or passage. It also referred to as the property of being ambiguous, where a word, term, notation, sign, symbol, phrase, sentence, or any other form used for communication, is interpreted in more than one way. When language is capable of being understood in more than one way by a reasonable person, ambiguity exists [19]. For example, consider the following two sentences, each with a different sense of ambiguous word "***soquu***"

1. Caaltuun kitaba ishee bade ***soquu*** deemte.
2. Tolaan lafa irra marga ***soquu*** deeme.

The occurrences of the word "***soquu***" in the two sentences clearly denote different meanings. The meaning of the word "*soquu"* in the first sentence is: **looking carefully in order to find something missed or lost**, which is to mean "*Chaltu went to **find** her lost book*". However, the meaning "*soquu*" in the second sentence is: **cleaning or removing dirt or unwanted mater from the**

surface of something by rubbing it hard, which is to mean "*Tola went to* ***scour*** *grass from the land"*.

Ambiguity is inherent to human language. In particular, word sense ambiguity is predominant in all natural languages; with a large number of words in any given language carrying more than one meaning. For humans, resolving ambiguity is a routine task that hardly requires conscious effort. In addition to having a deep understanding of language and its use, humans possess a broad and conscious understanding of the real world, and this equips them with the knowledge that is relevant to make sense disambiguation decisions effortlessly, in most cases. However, successful solutions for automatic resolution of ambiguity in natural language often require large amounts of annotated data/knowledge resources, to achieve good levels of accuracy. These issues are clearly reflected in the performance of current word-sense disambiguation systems. When given a large amount of training data for a particular word with reasonably clear sense distinctions, existing systems perform fairly well [1].

Natural language presents many types of ambiguity, ranging from morphological ambiguity to pragmatic ambiguity passing through syntactic or semantic ambiguities. Thus, most efforts in Natural Language Processing are devoted to solve different types of ambiguities. The resolution of words syntactic ambiguity has largely been solved in language processing by part-of-speech taggers which predict the syntactic category of words in text with high levels of accuracy. The problem of resolving semantic ambiguity is generally known as word sense disambiguation and has proved to be more difficult than syntactic disambiguation [17].

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 approaches, corpus based approaches and hybrid approach. 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 [11]. With the availability of huge computer-readable text corpora and the corresponding development of statistical techniques for data mining, corpus-based methods have taken centre stage in the development of WSD solutions. These methods have been employed in the learning of probabilistic models for WSD from large collections of natural language texts. Probabilistic models for sense classification consist of feature variables, the class variable and a probability distribution that models the interactions among all the variables. The context of an ambiguous word is defined very simply and usually consists of linguistic information that can be easily extracted from the neighborhood of the ambiguous word. This information is captured in the model via the feature variables. The class variable on the other hand represents the various senses of a word or the semantic tags associated with it. The probability distribution is learned (estimated) from sense-tagged data, and is used to predict the most probable class (sense) for a given input.

1.2. Statement of the problem

Afaan Oromo is one of the major languages that are widely spoken in Ethiopia. Currently, it is an official language of Oromia national regional state. It is spoken by about 30 million Oromo's within Ethiopia [58]. In addition, the language is also spoken in Somalia, Kenya, Uganda, Tanzania and Djibouti. Like any other language, there are a number of ambiguous words in Afaan Oromo language. Hence, it is difficult to understand the meaning of those words in a given context. For instance, in the real world, when a person understands a sentence with an ambiguous word in it, that understanding is built on the basis of just one of the meanings. So, as some part of the human language understanding process, the appropriate meaning has been chosen from the range of possibilities. This would seem that WSD might be a well-defined task, undertaken by a particular module within the human language

processor. This module could then be modeled computationally in a WSD program to overcome the problem of understanding among multiple meanings of words in the language, and this program performs as one of the essential components of the human language processor. Hence, to have a clear understanding of ambiguous words in the language, WSD for Afaan Oromo language needs to be developed.

The development of word sense disambiguation is crucial for later development of Afaan Oromo natural language processing applications such as speech recognition, information retrieval, machine translation, text processing and others. Because it helps them to overcome the problem of ambiguity. For instance, WSD is one of the most difficult tasks in Machine Translation system, in which there may be several candidates in the target language associated with each lexical item of the source language. A single word can have many senses and each of those senses can be mapped into many target language words. So, selecting the correct target word which is the most suitable for the context is a challenging problem [6]. The absence of WSD in Afaan Oromo Natural Language Processing system limits, future efforts of making computer to understand Afaan Oromo Language. Hence, this research work tries to fill such a gap in the language.

1.3. Motivation

In the last decade, many methods have been developed in order to deal with word sense disambiguation, due to its importance in understanding semantics of natural languages processing [7]. However, none of the existing methods/systems are dealing with Afaan Oromo language. With the fact that the language is used in offices, schools and media, there is a huge electronic data available that encourages studies related to NLP tasks associated with the language. So, the development of WSD applications for this language is required to cope up with the current technologies of NLP. Besides these, this study is inspired by the contribution of word sense disambiguation tasks to

other NLP studies such as Information Retrieval, Information Extraction, and Machine Translation.

1.4. Objective of the study

1.4.1. General objective

The general objective of this research work is to develop Word Sense Disambiguation for Afaan Oromo language.

1.4.2. Specific objectives

The specific objectives of this research work are:

a) To study the general morphological structure of Afaan Oromo language.
b) To develop manually annotated corpus for selected ambiguous words of the language.
c) To review techniques of WSD adopted for other languages.
d) To develop a WSD algorithm and prototype which best fits for Afaan Oromo language.
e) To evaluate the performance of the prototype.

1.5. Scope of the study

The study:-

- deals only with textual information. That means the system will only accept data in text form but not accept data in voice or sound form.
- deals with only semantic level analysis. That means the system does not perform any kind of grammar and spelling correction. Because they are considered as lower level analysis.
- will limited to five ambiguous words, due to unavailability of sense annotated data and linguistic resources.

1.6. Methodology of the study

1.6.1. Literature Review

Various literatures that are considered to be relevant for the research work are reviewed to get better understanding of the area and to have detailed knowledge on the various techniques that are essential for WSD systems.

1.6.2. Discussion with Experts

For the linguistic knowledge acquisition, continuous discussion with linguistic professionals from Addis Ababa University, Department of Linguistic and others have been made, to refine the lexical semantic and compositional semantic of the language.

1.6.3. Development of the System

Based on the analysis of the language, a WSD algorithm has been developed. The developed system was cross checked with the linguists and an iterative improvement has been made on the system. Supervised approach has been employed to design the system. In this approach, the system uses information gathered from training corpus to assign senses to unseen examples. Hence, we developed Afaan Oromo corpus from the scratch. The corpus contains 1240 sense examples for 5 Afaan Oromo ambiguous words. Java programming language has been used to develop the prototype.

1.6.4. Evaluation Technique

Evaluation about the effectiveness of the algorithms on varieties of sentence from the language has been made. This evaluation was carried out using statistical technique called k-fold cross-validation with standard performance evaluation metrics. During this, two types of data sets have been prepared: training set and test set. The training set was used to train the system and the testing set was used to measure the performance of the developed system using selected Afaan Oromo sentences from different sources.

1.7. Significance of the study

Word sense disambiguation is useful in many NLP applications, and information retrieval is one of these applications. According to various researchers, to determine the impact of lexical ambiguity on information retrieval systems, word senses would help to separate relevant from non-relevant documents. The assumption is that if a retrieval system indexed documents by senses of the words they contain and the appropriate senses in the document query could be identified then irrelevant documents containing query words of a different sense would not be retrieved. In addition, documents should not be ranked based on words alone, rather the documents should be ranked based on word senses, or based on a combination of word senses and words. These results indicate the potential of word sense disambiguation to improve Information Retrieval [11,12]. In contrast, researchers in machine translation have consistently argued that effective word sense disambiguation procedures would revolutionize their field. As they stated, the construction of word sense disambiguation algorithm for machine translation system is needed to find the correct word in the target language [1].

Besides, WSD has a great contribution for parsing. The assumption is that if the semantics of each lexical item is known then this could aid a parser in constructing a phrase structure for that sentence [17]. Thus, WSD is critical for parsing accurately, by implication; it is significant for all those applications that depend on parsing. Moreover, WSD is required for the accurate analysis of text in many applications such as Information Extraction, Text Mining and Semantic Web [13]. It can also be used in language teaching, to identify the meaning of ambiguous words in a sentence.

1.8. Thesis Organization

The rest of this thesis is organized as follows. Chapter Two presents literature review. It mainly focuses on reviews made on different literatures regarding

Word Sense Disambiguation together with its approaches and different machine learning techniques. Chapter Three presents works related to word sense disambiguation system for another language. Chapter Four discusses Afaan Oromo writing system, morphological structure and ambiguities in the language. The Fifth Chapter discusses about the design and implementation of the system, which is composed of system requirement, corpus preparation, architecture of the system and its prototype. Evaluation results are presented in Chapter Six. This Chapter discusses the experimentation and its findings. Finally, Chapter Seven deals with the conclusion and the recommendations drawn from the findings of the study.

Chapter Two: Literature Review

2.1. Word Sense Disambiguation

Many words have more than one meaning in natural language, and each one of them is determined by its context. The automated process of recognizing word senses in context is known Word Sense Disambiguation (WSD). The algorithms used in WSD can be classified as knowledge based and corpus based. Corpus based algorithm can be further classified as supervised learning and unsupervised learning [22]. In knowledge based approach disambiguation is carried out using information from an explicit lexicon or knowledge base. The lexicon may be a machine readable dictionary, thesaurus or it may be hand-crafted. Supervised learning can be viewed as a classification task while unsupervised learning can be viewed as a clustering task [24].

We can distinguish two variants of the generic WSD task [21]:

1. **Lexical sample (or targeted WSD**) - where a system is required to disambiguate a restricted set of target words usually occurring one per sentence. Supervised systems are typically employed in this setting, as they can be trained using a number of hand-labeled instances (training set) and then applied to classify a set of unlabeled examples (test set).
2. **All-words WSD**- where systems are expected to disambiguate all open-class words in a text (i.e., nouns, verbs, adjectives, and adverbs). This task requires wide-coverage systems. Consequently, purely supervised systems can potentially suffer from the problem of data sparseness, as it is unlikely that a training set of adequate size is available which covers the full lexicon of the language of interest.

2.2. Knowledge Sources for WSD

Knowledge sources provide data which are essential to associate senses with words. It is one of the fundamental component of WSD. They can vary from

corpora of texts, either unlabeled or annotated with word senses, to machine-readable dictionaries, thesauri, glossaries, ontologies, etc. [22, 23]

2.2.1. Structured resources

Thesauri - is a reference work that lists words grouped together according to similarity of meaning (containing synonyms and sometimes antonyms), in contrast to a dictionary, which contains definitions and pronunciations [36]. It also provides information about relationships between words like synonymy and antonym. The most widely used thesaurus in the field of WSD is Roget's International Thesaurus [25].

Machine Readable Dictionary's (MRD) is a dictionary in an electronic form that can be loaded in a database and can be queried via application software [37]. It becomes a popular source of knowledge for natural language processing since the 1980s, when the first dictionaries were made available in electronic format. Examples of this dictionary include 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). LDOCE is one of the most widely used machine-readable dictionaries within the NLP research community, before the diffusion of Word-Net [26], presently the most utilized resource for word sense disambiguation in English.

Word-Net is a semantic lexicon for the English language. Word-Net was created and is being maintained at the Cognitive Science Laboratory of Princeton University [24]. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. Word-Net is organized as a network of lexicalized concepts, called synsets, that comprise set of synonyms. Each word meaning can be represented by a set of word-forms known as synonym sets or synsets. Synsets are created for content words, i.e., for noun, verb, adjective and

adverb. For example, the nouns {president, chairman, chair, chairperson} form a synset.

Ontology - represents knowledge as a set of concepts within a domain, and the relationships between pairs of concepts [38]. It specifies concept about a specific domain of interest [27], usually including taxonomy and a set of semantic relations.

2.2.2. Unstructured resources

Corpus - is a collection 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. Examples of raw corpora include the Brown corpus, the British National Corpus (BNC), the Wall Street Journal (WSJ) corpus, the American National Corpus and others. Examples of Sense Annotated Corpora include SemCor, MultiSemCor, the linehard- serve, the interest corpus, the Open Mind Word Expert data set and the likes [28].

Collocation resources - register the tendency for words to occur regularly with others. Examples of such resources include the Word Sketch Engine, Just The Word, The British National Corpus collocations, the Collins Cobuild Corpus Concordance and etc.

2.3. Representation of Context

As text is an unstructured source of information, to make it a suitable input to an automatic method it is usually transformed into a structured format. To this end, a preprocessing of the input text is usually performed, which typically (but not necessarily) includes the following steps:

- **Tokenization** - a normalization step, which splits up the text into a set of tokens usually words.

- ☞ **Part-of-speech tagging** - consisting of the assignment of a grammatical category to each word. For example "the/DT bar/NN was/VBD crowded/JJ," where DT, NN, VBD and JJ are tags for determiners, nouns, verbs, and adjectives, respectively.
- ☞ **Lemmatization** - is the reduction of morphological variants to their base form (e.g. was → be, bars → bar).
- ☞ **Chunking** - consists of dividing a text in to syntactically correlated parts (e.g., [the bar] NP [was crowded] VP, is respectively divided in to the noun phrase and the verb phrase).
- ☞ **Parsing** - identifies the syntactic structure of a sentence usually involving the generation of a parse tree of the sentence structure.

As a result of the preprocessing phase of a portion of text (e.g., a sentence, a paragraph, a full document, etc.), each word can be represented as a vector of features of different kinds or in more structured ways, for example, as a tree or a graph of the relations between words. The representation of a word in a context is the main support, together with additional knowledge resources, for allowing automatic methods to choose the appropriate sense from a reference inventory [21].

2.4. Approaches for WSD

Currently, there are three main methodological/approaches in the area of WSD: knowledge-based approach, corpus-based approach and hybrid approach [47]. Knowledge-based approach uses external knowledge resources, which defines explicit sense distinctions for assigning the correct sense of a word in a context. Corpus-based approach uses machine-learning techniques to induce models of word usages from large collections of text examples. The hybrid approach is a combination of both knowledge based and corpus based approach. The following sections provide a brief discussion on the three approaches of WSD.

2.4.1. Knowledge-based WSD

The objective of knowledge-based or dictionary-based WSD is to exploit knowledge resources such as dictionaries, thesauri, ontologies and collocations to infer the senses of words in a context. The work done earlier on WSD was theoretically interesting but practical only in extremely limited domains. Scaling up these works was the main difficulty at that time: the lack of large-scale computational resources prevented a proper evaluation, comparison and exploitation of those methods in end-to-end applications.

Since Lesk [31], the first WSD based on MRD, many researchers have used machine-readable dictionaries (MRDs) as a structured source of lexical knowledge to deal with WSD. These approaches, by exploiting the knowledge contained in the dictionaries, mainly seek to avoid the need for large amounts of training material. As stated by Agirre and Martinez [30], ten different types of information which is useful for WSD can be obtained from MRDs. This information includes part of speech, semantic word associations, syntactic cues, selectional preferences, and frequency of senses. In general, WSD techniques using pre-existing structured lexical knowledge resources differ in:

- The lexical resource used (monolingual and/or bilingual MRDs, thesauri, lexical knowledge base, etc.)
- The information contained in this resource, exploited by the method; and
- The property used to relate words and senses.

These methods usually have lower performance than their supervised alternatives, but they have the advantage of a wider coverage, thanks to the use of large-scale knowledge resources. In the following section, we overview two of the knowledge-based techniques, namely: the overlap of sense definitions and selectional preference.

2.4.1.1. Overlap of Sense Definitions

It is a simple knowledge-based approach which relies on the calculation of the word overlap between the sense definitions of two or more target words. This

approach is named gloss overlap or the Lesk algorithm after its author Lesk[31]. The original Lesk algorithm performs WSD by calculating the relative word overlap between the context usage of a target word, and the dictionary definition of each of its senses in a given MRD. The sense with the highest overlap is then assumed to be the correct one. Lesk algorithm identifies the sense of a word w whose textual definition has the highest overlap with the words in the context of w. Formally, given a target word w, the following score is computed for each sense S of w.

$$score_{LeskVar}(S) = |context(w) \cap gloss(S)|$$

where context(w) is the bag of all content words in a context window around the target word w and gloss(S) is the bag of words in the textual definition of sense S of w.

For example the two senses/meanings for the word ***sanyii*** are listed below and words which overlap with the following input sentence are marked in bold:

Tolaan sanyii loonii bifa tokko qaban bitate.

1. Sanyii - Waan gosa ykn ***bifa tokko*** ta'an, kan wal fakkaatan ykn kan firooma wal irra qaban.
2. Sanyii - Ija ykn firii midhaanii ykn biqiltu akka marguuf facaafamu ykn dhaabamu.

Sense 1 of ***sanyii*** has 2 overlaps, whereas the other senses have zero, so the first sense is selected.

Unfortunately, Lesk's approach is very sensitive to the exact wording of definitions, so the absence of a certain word can radically change the results. Further, the algorithm determines overlaps only among the glosses of the senses being considered. This is a significant limitation in that dictionary glosses tend to be fairly short and do not provide sufficient vocabulary to relate fine-grained sense distinctions.

Recently, Banerjee and Pedersen [32] introduced a measure of extended gloss overlap, which expands the glosses of the words being compared to include

glosses of concepts that are known to be related through explicit relations in the dictionary (e.g. hypernymy, meronymy, hyponyms, etc.). The range of relationships used to extend the glosses is a parameter, and can be chosen from any combination of Word-Net relations.
For each sense S of a target word w, we estimate its score as

$$score_{ExtLesk}(S) = \sum_{S':S \rightarrow rel\ S' or\ S \equiv S'} | context(w) \cap gloss(S') |$$

Where context (w) is the bag of all content words in a context window around the target word w and gloss(S') is the bag of words in the textual definition of a sense S' which is either S itself or related to S through a relation rel. The overlap scoring mechanism is also parameterized and can be adjusted to take into account gloss length (i.e. normalization) or to include function words. As stated by Banerjee and Pedersen disambiguation greatly benefits from the use of gloss information from related concepts which increase the accuracy from 18.3% for the original Lesk algorithm to 34.6% for extended Lesk. However, the approach does not lead to state-of-the-art performance compared to competing knowledge-based systems. This is due to the maximum accuracy obtained by this method is less than other knowledge based methods. For example, Agirre and Martinez [30], obtain accuracy of 86.7% for word-to-class, 97.3% for class-to-class methods using selectional preference approach.

2.4.1.2. Selectional Preferences

It is a type of knowledge-based algorithm which restricts the number of meanings of a target word occurring in a context. Selectional preferences or restrictions are constraint on the semantic type that a word sense imposes on the words with which it combines in sentences usually through grammatical relationships. For instance, the verb **eat** expects an animate entity as subject and an edible entity as its direct object. We can distinguish between selectional restrictions and preferences in that the former rule out senses that violate the

constraint, whereas the latter tend to select those senses which better satisfy the requirements.

The approach combines statistical and knowledge-based methods, but unlike many recent corpus-based approaches to sense disambiguation, it takes as its starting point the assumption that sense- annotated training text is not available. Motivating this assumption is not only the limited availability of such text at present, but skepticism that the situation will change any time soon.

The easiest way to learn selectional preferences is to determine the semantic appropriateness of the association provided by a word-to-word relation. The simplest measure of this kind is frequency count. Given a pair of words w1 and w2 and a syntactic relation R (e.g., subject-verb, verb-object, etc.), this method counts the number of instances (R, w1, w2) in a corpus of parsed text, obtaining a Count(R, w1, w2) [33]. Another estimation of the semantic appropriateness of a word-to-word relation is the conditional probability of word w1 given the other word w2 and the relation R [33]:

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

To provide word-to-class or class-to-class models, or to generalize the knowledge acquired to semantic classes and relieve the data sparseness problem, manually crafted taxonomies such as Word-Net can used to derive a mapping from words to conceptual classes. Several techniques have been employed from measures of selectional association such as Hidden Markov Models, class-based probability, Bayesian networks and etc. Almost all these approaches exploit large corpora and model the selectional preferences of predicates by combining observed frequencies with knowledge about the semantic classes of their arguments. Disambiguation is then performed with

different means based on the strength of a selectional preference towards a certain conceptual class (i.e., sense choice).

2.4.2. Corpus-based WSD

In the last fifteen years, empirical and statistical approaches have had a significantly increased impact on NLP. The types of NLP problems initially addressed by statistical and machine-learning techniques are those of language- ambiguity resolution, in which the correct interpretation should be selected from among a set of alternatives in a particular context. These techniques are particularly adequate for NLP because they can be regarded as classification problems, which have been studied extensively in the ML community [29].

We can broadly distinguish two main approaches to WSD based on statistical methods [21].

- ***Supervised WSD***: these approaches use machine-learning techniques to learn a classifier from labeled training sets, that is, sets of examples encoded in terms of a number of features together with their appropriate sense label (or class);
- ***Unsupervised WSD***: these methods are based on unlabeled corpora, and do not exploit any manually sense-tagged corpus to provide a sense choice for a word in a context. The following section explains the two methods in detail.

2.4.2.1. Supervised WSD

Supervised WSD uses machine learning techniques for inducing a classifier from semantically annotated corpora. Generally, supervised systems have obtained better results than unsupervised ones, a conclusion that is based on experimental work and international competitions. This approach uses semantically annotated corpora to

train machine learning (ML) algorithms to decide which word sense to choose in which contexts. The words in such annotated corpora are tagged manually using semantic classes taken from a particular lexical semantic resource (most commonly Word-Net). Corpus-based methods are called “supervised” when they learn from previously sense- annotated data, and therefore they usually require a large amount of human intervention to annotate the training data [35]. Although several attempts have been made, to overcome the knowledge acquisition bottleneck (too many languages, too many words, too many senses, too many examples per sense) is still an open problem that poses serious challenges to the supervised learning approach for WSD.

In supervised techniques words can be labeled with their senses [40]. For example in the following two sentences ***horii*** is an ambiguous word and it is tagged with sense *beelada (cattle)* and *qarshii (money)* respectively.

- Tolaan horii<beelada> qale nyaate.
- Tolaan horii<qarshii> isa mana baanki kaa’e.

Therefore supervised approaches can be seen as:
– accept a corpus tagged with senses
– define features that indicate one sense over another
– learn a model that predicts the correct sense given the features

In supervised approaches, a sense disambiguation system is learned from a representative set of labeled instances drawn from sense annotated corpus. Input instances to these approaches are features encoded along with their appropriate labels. The output of the system is a classifier system capable of assigning labels to new feature encoded inputs. The following section provides some of supervised techniques

namely NBC, Decision list, Neural Network, Examplar-based, SVM and Ensemble method.

Naive Bayes Classifier (NBC): A Naive Bayes classifier [21] is a simple statistical classifier based on the application of Bayes' theorem. It can predict class membership probabilities, such as the probability that a given sample belongs to a particular class. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases. Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is known as class conditional independence. However, The Naive Bayes assumption has two consequences. The first is that all the structure and linear ordering of words within the context is ignored. This is often referred to as a bug of words model. The other is that the presence of one word in the bag is independent of another. But, as in many other cases, the simplifying assumption makes it possible to adopt an elegant model that can be quite effective despite its shortcomings. Baye's Theorem is described as follow [55].

- Let X be a data sample whose class label is unknown
- Let h be a hypothesis that says X belongs to class C
- For classification problems, determine P(h/X): the probability that the hypothesis holds given the observed data sample X.
- P(h): prior probability of hypothesis h (i.e. the initial probability before we observe any data, reflects the background knowledge)
- P(X): probability that sample data is observed
- P(X|h) : probability of observing the sample X, given that the hypothesis holds
- Given training data X, posteriori probability of a hypothesis H, P(h|X) follows the Bayes theorem:

$$P(h|X)=\frac{P(X|h)P(h)}{P(X)}$$

Informally, this can be written as

Posterior = likelihood x prior/evidence

If all we want to do is choose the correct class, we can simplify the classification task by eliminating P(X) (which is constant for all senses and hence does not affect the maximum value). Given different possible set of Hypothesis H, maximum posteriori hypothesis is the one that maximizes the posteriori probability which is given as:

$$h_{map} \equiv \underset{h \in H}{\operatorname{argmax}}\ P(h|X) = \underset{h \in H}{\operatorname{argmax}}\ P(X|h)P(h).$$

Naive Bayesian classifier is one of the best methods for implementing supervised approach to WSD. That means the system calculates the prior probability and the likelihood based on Bayes Theorem. First, the conditional probability is calculated for each sense Si of a word w given the features f_j in the context. Then the sense S which maximizes the following formula is chosen as the most appropriate sense in the context:

$$\hat{S} = \text{argmax } P(S_i \mid f_1, \ldots, f_m) = \text{argmax } \frac{P(f_1, \ldots, f_m \mid S_i)P(S_i)}{P(f_1, \ldots, f_m)}$$

$$\hat{S} = \underset{S_i \,\varepsilon\, \text{sense}(w)}{\text{argmax}}\ P(S_i) \prod_{j=1}^{m} \left(\frac{fj}{Si}\right)$$

Where m is the number of features, and the last formula is obtained based on the naïve assumption that the features are conditionally independent given the sense (the denominator is also discarded as it does not influence the calculations). The probabilities P(Si) and $P(f_j / S_i)$ are estimated, respectively, as the relative occurrence frequencies in the training set of sense Si and feature f_j in the presence of sense S_i.

Decision Lists: These classifiers are equivalent to simple case statements in most programming languages. In decision list classifier, a sequence of tests is applied to each vector encoded input. If the test succeeds then the sense associated with that test is returned. If the test fails then next test in the sequence is applied. This continues until the end of the list, where a default test simply returns majority sense [40].

A neural network [41] is an interconnected group of artificial neurons that uses a computational model for processing data based on a connectionist approach. Pairs of (input feature, desired response) are input to the learning program. The aim is to use the input features to partition the training contexts into non overlapping sets corresponding to the desired responses. As new pairs are provided, link weights are progressively adjusted so that the output unit representing the desired response has a larger activation than any other output unit.

Neural networks are trained until the output of the unit corresponding to the desired response is greater than the output of any other unit for every training example. For testing, the classification determined by the network is given by the unit with the largest output. Weights in the network can be either positive or negative, thus enabling the accumulation of evidence in favour or against a sense choice.

Exemplar-Based or Instance-Based Learning

Exemplar-based (or instance-based, or memory-based) learning is a supervised algorithm in which the classification model is built from examples [35]. The model retains examples in memory as points in the feature space and, as new examples are subjected to classification, they

are progressively added to the model. One of this algorithm is the k-Nearest Neighbor (kNN) algorithm, which is one of the highest-performing methods in WSD [35].

Support Vector Machines (SVM)

This method [21] is based on the idea of learning a linear hyperplane from the training set that separates positive examples from negative examples. The hyperplane is located in that point of the hyperspace which maximizes the distance to the closest positive and negative examples (called support vectors). In other words, support vector machines (SVMs) tend at the same time to minimize the empirical classification error and maximize the geometric margin between positive and negative examples.

As SVM is a binary classifier, in order to be usable for WSD it must be adapted to multiclass classification (i.e., the senses of a target word). A simple possibility, for instance, is to reduce the multiclass classification problem to a number of binary classifications of the kind sense S_i versus all other senses. As a result, the sense with the highest confidence is selected. SVM has been applied to a number of problems in NLP, including text categorization, chunking, parsing, and WSD. SVM has been shown to achieve the best results in WSD compared to several supervised approaches [21].

Ensemble Methods

Sometimes different classifiers are available which we want to combine to improve the overall disambiguation accuracy. Combination strategies called ensemble methods [21] typically put together learning algorithms

of different nature, that is, with significantly different characteristics. In other words, features should be chosen so as to yield significantly different, possibly independent, views of the training data (e.g., lexical, grammatical, semantic features, etc.). Ensemble methods are becoming more and more popular as they allow one to overcome the weaknesses of single supervised approaches. Several systems participating in recent evaluation campaigns employed these methods.

Single classifiers can be combined with different strategies. Examples of this method include majority voting, probability mixture, rank-based combination, AdaBoost, weighted voting, maximum entropy combination, etc [21].

2.4.2.2. Unsupervised WSD

Unsupervised methods have the potential to overcome the knowledge acquisition bottleneck [42], that is, the lack of large-scale resources manually annotated with word senses. These approaches to WSD are based on the idea that the same sense of a word will have similar neighboring words. They are able to induce word senses from input text by clustering word occurrences, and then classifying new occurrences into the induced clusters. They do not rely on labeled training text and, in their purest version, do not make use of any machine–readable resources like dictionaries, thesauri, ontologies, etc. However, the main disadvantage of fully unsupervised systems is that, as they do not exploit any dictionary, they cannot rely on a shared reference inventory of senses. While WSD is typically identified as a sense labeling task, that is, the explicit assignment of a sense label to a target word, unsupervised WSD performs word sense discrimination, that is, it aims to divide “the occurrences of a word into a number of classes by determining for any

two occurrences whether they belong to the same sense or not" [43]. Consequently, these methods may not discover clusters equivalent to the traditional senses in a dictionary sense inventory. For this reason, their evaluation is usually more difficult: in order to assess the quality of a sense cluster we should ask humans to look at the members of each cluster and determine the nature of the relationship that they all share (e.g., via questionnaires), or employ the clusters in end-to-end applications, thus measuring the quality of the former based on the performance of the latter.

Admittedly, unsupervised WSD approaches have a different aim than supervised and knowledge-based methods, that is, that of identifying sense clusters compared to that of assigning sense labels. However, sense discrimination and sense labeling are both sub-problems of the word sense disambiguation task [43] and are strictly related, to the point that the clusters produced can be used at a later stage to sense tag word occurrences. Next, we present two of the approaches for unsupervised WSD, namely: methods based on word clustering, and co-occurrence graphs.

Word clustering

Word clustering is a method which aims at clustering words which are semantically similar and can thus convey a specific meaning. A well-known approach to word clustering [46] consists of the identification of words $W = (w_1 . . . W_k)$ similar (possibly synonymous) to a target word w_0. The similarity between w_0 and w_i is determined based on the information content of their single features, given by the syntactic dependencies which occur in a corpus (such as:- subject-verb, verb-object, adjective-

noun, etc.). The more dependencies the two words share, the higher the information content. However, as for context vectors, the words in W will cover all senses of w_0. To discriminate between the senses, a word clustering algorithm is applied. Let W be the list of similar words ordered by degree of similarity to w_0. A similarity tree T is initially created which consists of a single node w_0. Next, for each $i \in \{1, \ldots, k\}$, $w_i \in W$ is added as a child of w_j in the tree T such that w_j is the most similar word to w_i among $\{w_0, \ldots, w_{i-1}\}$. After a pruning step, each sub-tree rooted at w_0 is considered as a distinct sense of w_0.

Co-occurrence Graphs

A different view of word sense discrimination is provided by graph-based approaches. These approaches are based on the notion of a co-occurrence graph, that is, a graph G = (V, E) whose vertices V correspond to words in a text and edges E connect pairs of words which co-occur in a syntactic relation, in the same paragraph, or in a larger context.

The construction of a co-occurrence graph based on grammatical relations between words in context was described by Widdows and Dorow [47]. Given a target ambiguous word w, a local graph G_w is built around w. By normalizing the adjacency matrix associated with G_w, we can interpret the graph as a Markov chain. The Markov clustering algorithm is then applied to determine word senses based on an expansion and an inflation step, aiming, respectively, at inspecting new more distant neighbors and supporting more popular nodes.

2.4.3. Hybrid Approaches

Hybrid approaches obtain disambiguation information from both corpora and explicit knowledge-bases. Hybrid systems aim to use the strengths of

both approaches to overcome the specific limitations associated with a particular approach and improve WSD accuracy. For example, Yarowsky [21] used bootstrapping approaches where initial data comes from an explicit knowledge source which is then improved with information derived from corpora. He defines a small number of seed definitions for each of the senses of a word. Then the seed definitions are used to classify the obvious cases in a corpus.

2.5. Summary

This Chapter discussed about types of Knowledge source used for disambiguation as well as components used for preprocessing of the input text. Survey of the major approaches to WSD was also presented. The next Chapter discusses related works in the area of WSD that have been done for different languages.

Chapter Three: Related Work

3.1. Introduction

Word sense disambiguation (WSD) is one of the most critical and widely studied Natural Language Processing tasks, which is used in order to increase the success rates of NLP applications like machine translation, information retrieval, natural language understanding, language study and etc [48]. Three main approaches have been applied in the area of WSD field. These are knowledge-based approaches, corpus based approaches and hybrid approach. All these approaches have been used by different researchers for different languages. Among them, supervised machine learning approach is one of the successful lines of research on WSD in which information is drawn from annotated corpora. The next sections briefly review the work done on WSD.

3.2. WSD for English

Many researchers have been working for word sense disambiguation in the English Language. In the following paragraphs, we discuss briefly two of the related work in the area of Word Sense Disambiguation.

LEXAS - (LEXical Ambiguity-resolving System) is a word sense disambiguation (WSD) program for the English language using an exemplar-based learning algorithm which is implemented by Hwee Tou Ng and Hian Beng Lee [49]. This program integrates a diverse set of knowledge sources to disambiguate word sense including part of speech of neighboring words, morphological form of words in the sentence, the unordered set of surrounding words, local collocations, and verb-object syntactic relation. LEXAS takes an input consisting of unrestricted real-world English sentences. In the output, each word occurrence w is tagged with its correct sense according to the context. It uses the sense definitions as given in Word-net.

LEXAS performs WSD by first learning from a training corpus of sentences in which words have been pre-tagged with their correct senses. That is, it uses supervised learning, in particular exemplar-based learning, to achieve WSD. It operates in two phases: training phase and test phase. In the training phase, LEXAS is given a set S of sentences in the training corpus in which sense-tagged occurrences of w appear. For each training sentence with an occurrence of w, LEXAS extracts the parts of speech (POS) of words surrounding w, the morphological form of w, the words that frequently co-occur with w in the same sentence, and the local collocations containing w; where local collocations are common expressions containing the word to be disambiguated. For disambiguating a noun w, the verb which takes the current noun w as the object is also identified.

This set of values form the features of an example, with one training sentence contributing one training example. Subsequently, in the test phase, LEXAS is given new, previously unseen sentences. For a new sentence containing the word w, LEXAS extracts from the new sentence the values for the same set of features, including parts of speech of words surrounding w, the morphological form of w, the frequently co-occurring words surrounding w, the local collocations containing w, and the verb that takes w as an object (for the case when w is a noun). These values form the features of a test example.

This test example is then compared to every training example. LEXAS determines the closest matching training example as the one with the minimum distance to the test example. Distance between two symbolic values v_1 and v_2 of a feature f can be measured using:

$$d(v_1, v_2) = \sum_{i=1}^{n} \left| \frac{C_{1,i}}{C_1} - \frac{C_{2,i}}{C_2} \right|$$

Where:-

- $C_{1,i}$:-is the number of training examples with value v1 for feature f that is classified as sense i in the training corpus, and
- C_1:- is the number of training examples with value v_1 for feature f in any sense.
- $C_{2,i}$ and C_2 :-denote similar quantities for value v_2 of feature f.
- n :- is the total number of senses for a word w.

The distance between two examples is the sum of the distances between the values of all the features of the two examples. Then the sense of w in the test example is the sense of w in this closest matching training example.

Finally, the developers of LEXAS, tested the program against a common data set used in previous work done by Bruce and Wiebe [56], as well as on a large sense-tagged corpus that they separately constructed. LEXAS achieves a higher accuracy on the common data set, and performs better than the most frequent heuristic on the highly ambiguous words in the large corpus tagged with the refined senses of Word-NET.

The other WSD scheme using English language has been done by A. R. Rezapour, S. M. Fakhrahmad and M. H. Sadreddini in London, UK [50]. It has been developed using a supervised learning method based on K-Nearest Neighbor algorithm. In this work, they first extracted the set of words that have co-occurred with the ambiguous word in the text frequently, and the set of words surrounding the ambiguous word. Then by comparing the context where the ambiguous word has occurred in and the texts existing in the training corpus they assigned a sense to an ambiguous word using K-NN algorithm.

According to the researchers, K-NN is a supervised learning algorithm in which the classification is accomplished by comparing a given test vector with training vectors that are similar to it. When an unknown vector is introduced, K-NN classifier finds k most similar training vectors (i.e. nearest neighbors) that are closest to the unknown vector. Then the unknown vectors are simply

assigned to the class of its nearest neighbor, otherwise it is classified by the majority vote of its neighbors. The distance between a test vector and the training vectors in K-NN classifier is computed based on the Euclidean distance.

Finally, they evaluated the system using TWA [57]. TWA is a sense tagged corpus developed at University of North Texas, which focuses on six words each having two different senses (including " bass", " crane", "motion", " palm", " plant" and " tank"). Then, using 5 fold cross validation approach, the dataset was divided into training and test parts for a k-NN classifier. In order to improve the classification accuracy of K-NN, they proposed and evaluated a feature weighting strategy. And they proved that, the effect of the weighting scheme was encouraging and led to promising improvements in most cases.

3.3. Amharic WSD

The first WSD for the Amharic language has been done by Teshome Kassie [51]. He has studied how linguistic disambiguation can improve the effectiveness of an Amharic document query retrieval algorithm. During his study, he developed Amharic disambiguation algorithm based on the principles of semantic vectors analysis and implemented in Java. He used the Ethiopian Penal Code which is composed of 865 Articles as a corpus. The disambiguation algorithm was then used to develop a document search engine.

He developed his own algorithm based on distributional hypothesis stating that words with similar meanings tend to occur in similar contexts. For disambiguation of a given word, he computed the context vector of each occurrence of the words. The context vector was derived from the sum of the thesaurus vectors of the context words. He constructed the thesaurus by associating each word with its nearest neighbors.

For evaluating WSD, he used pseudo words which are artificial words rather than real sense tagged words reasoning that it is costly to prepare sense annotation data. He compared his algorithm with Lucene algorithm and reported that the algorithm is superior over the Lucene's one.

The second attempted work was by Solomon M. [52]. He used corpus based, supervised machine learning approach using Naive Bayes algorithm for Amharic WSD, to check standard optimal context window size which refers to the number of surrounding words sufficient for extracting useful disambiguation. Based on Native Bayes algorithms, experiment found that three-word window on other side of the ambiguous word is enough for disambiguation. He used a monolingual corpus of English language to acquire sense examples and the sense examples are translated back to Amharic which is one approach of tackling the knowledge acquisition bottleneck. Based on Naive Bayes algorithm, experiments were conducted on Weka 3.6.2 package concluded that, Naive Bayes methods achieve higher accuracy on the task of WSD for selected ambiguous word, provided that the quality of the labeled data set. He achieved accuracy within the range of 70% to 83% for all classifiers. This is an impressive accuracy for supervised WSD but it suffers from knowledge acquisition bottleneck.

The other WSD for Amharic language has been implemented by Solomon A. [53]. He used a corpus based approach to word sense disambiguation that only requires information that can be automatically extracted from untagged text. Unsupervised machine learning technique was applied to address the problem of automatically deciding the correct sense of an ambiguous word. He used corpus of Amharic sentences, based on five selected ambiguous words, 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 which is one approach of tackling knowledge acquisition bottleneck.

He 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. Based on the selected algorithms, he concluded that simple k means and EM clustering algorithms achieved higher accuracy on the task of WSD for selected ambiguous word, provided with balanced sense distribution in corpus. He achieved accuracy within the range of 65.1 to 79.4 % for simple k means, 67.9 to 76.9 for EM and 54.4 to 71.1 for complete link clustering algorithms for the five ambiguous words.

3.4. Turkish WSD

Turkish is the most widely-spoken of the Turkic languages, with over 63 million native speakers which are located predominantly in Turkey. WSD for Turkish has been developed by a number of researchers. Vildan Ozdemir [54] is one of them who developed WSD for Turkish using supervised machine learning approach at Graduate Institute of Sciences and Engineering of Fatih University. The researcher uses four example words which have more than one sense and the WSD study was performed for these words. During the selection of words, rich verb structure of Turkish was taken into consideration.

Due to lack of sense annotated text to be able to do these types of studies, the researcher first collected the data composed of sentences containing the sample words chosen. Four sample words were chosen as ambiguous word and the average number of sense per word was two. A total of 1008 sentences were collected for the sample words. These sentences were selected from different sources in accordance with senses for the target word. Then, the features that distinguish the sense of the word were identified. According to the researcher, Turkish is a language with an agglutinative structure, there are several

features affecting the word sense. Hence, structural features like the suffixes of target words, the type of words that are used with them and the suffixes they get have been examined and based upon this correct sense of the sentence was identified.

Supervised Learning algorithms were applied to the data by the researcher, and the results obtained using evaluation methods have been interpreted. For sense disambiguation, Naïve Bayes, K-Star, Simple Cart and Bagging algorithms have been used in the test processes performed. The data were evaluated separately as test and train so as to measure the effects of different evaluation methods, and also evaluations were made with Cross Validation (CV) method. The best results were obtained with Naïve Bayes algorithms. Furthermore, the features for the words were identified that are believed to be effective in the study. The researcher observed that the most effective feature was the type of the word, and this was followed by the suffix on the target word, the preceding and succeeding words' types and their suffixes.

3.5. WSD for Myanmar language

Myanmar language is the official language of the Union of Myanmar. It is written from left to right and no spaces between words, although informal writing often contains spaces after each clause. It is syllabic alphabet and written in circular shape. The first WSD for Myanmar language has been done by three researchers in coordination (i.e. Nyein Thwet Thwet Aung, Khin Mar Soe, Ni Lar Thein)[55]. A supervised machine learning approach using Naïve Bayesian Classifier was used for disambiguation that assumes a corpus where each use of ambiguous words is labeled with its correct sense. They developed WSD module to improve Myanmar-English machine translation system.

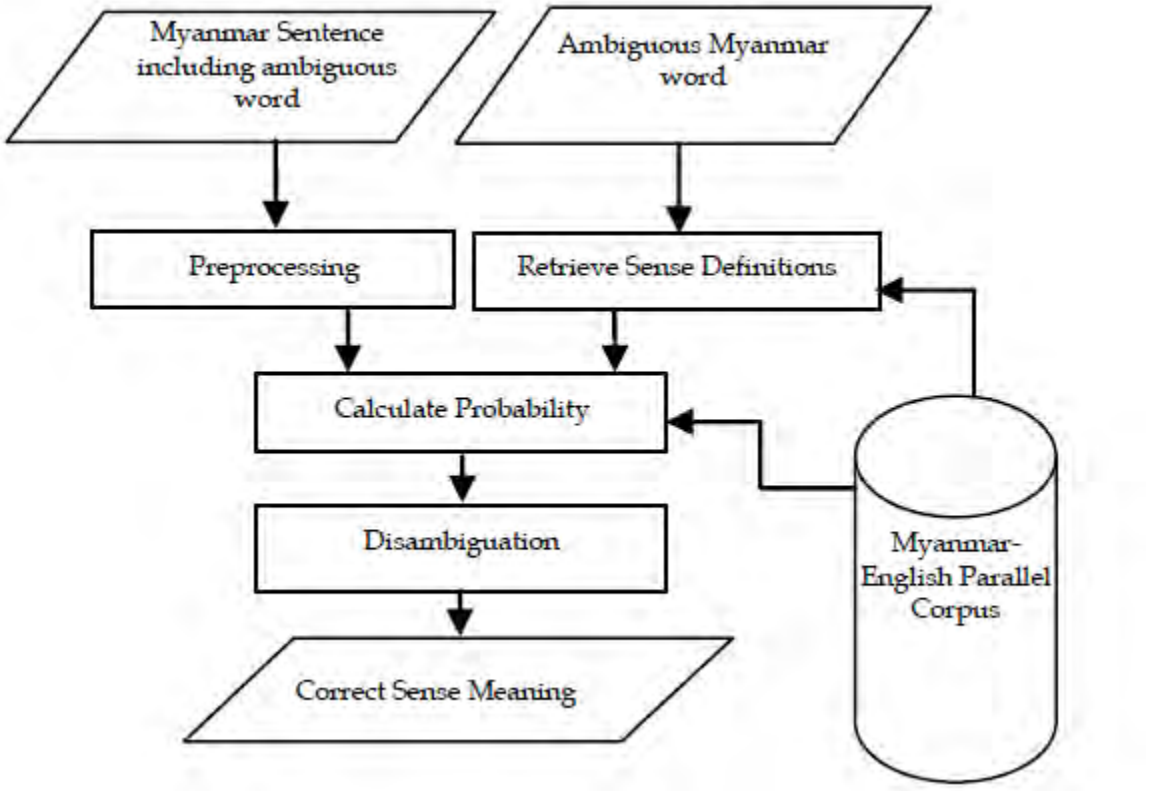

Figure 3.1. Architecture for Myanmar language WSD system

The WSD system for Myanmar language consists of four main parts:-

- Preprocessing
- Multi-Senses look-up on Corpus
- Calculating Probability based on Bayes Theorem
- Disambiguation -Calculating maximum scores using Bayes decision rule.

Firstly, the system takes the Myanmar sentence including ambiguous words and the ambiguous Myanmar word that want to be disambiguated as input. In preprocessing stage, it segments the input sentence by using Myanmar word segmenter and removes all stop words which are a list of common or general terms from the input sentence. After gathering information in the

preprocessing step, the system uses the remaining words in the input sentence as features. Secondly, the system retrieves the possible English sense definitions of the ambiguous word from the corpus. Thirdly, the system calculates the prior probability and the likelihood based on Bayes Theorem. Finally, disambiguation process is performed using Bayes decision rule. The disambiguation process computes the score of each sense of ambiguous word and decides the most appropriate sense for a given word in the test sentence.

They conduct the experiment using data drawn from Myanmar- English parallel corpus, which contains sentences used in various domains. It contains various sense meanings of ambiguous Myanmar word. The training set consists of 1000 sentence pairs. They collected 60 ambiguous nouns and 100 ambiguous verbs for experiment. They used only the pure text data, and not the speech transcriptions. The sense of the ambiguous words was obtained from the Myanmar-English dictionary. The number of senses per test word ranged from 2 to 9 and the average was 4.

They also evaluated the system using the most common evaluation techniques, which select a small sample of words from Myanmr-English parallel corpus and compare the results of the system with a human judge. As a result, the system improves the accuracy of Myanmar to English language translation which achieves 89% precision.

3.6. Summary

In this Chapter we presented WSD work on four languages: English, Amharic, Turkish and Myanmar. While reviewing, we focused on the technique they used to resolve ambiguity, corpus size used and performance evaluation of those works. The general working principles of all WSD are almost the same. All of them incorporate preprocessing phase, feature extraction and classification technique for disambiguation. But differences are observed on feature extraction and techniques applied for disambiguation. Even if all of

them are considered words around the ambiguous word as a feature; some of them were considered additional information like POS of words surrounding the ambiguous word, morphological form of the ambiguous word and the likes. Different type's algorithms were also applied for disambiguation such as Naïve bayes, KNN, simple cart and the like. The best result was obtained with naïve bayes algorithm. Taking what we obtained from the review of the related work we propose a solution to WSD for Afaan Oromo language. Hence, we selected supervised machine learning approach using naïve bayes technique to develop WSD for Afaan Oromo language. The next Chapter deals with Afaan Oromo language.

Chapter Four: Afaan Oromo Language

4.1. Introduction

Afaan Oromo is a Cushitic language spoken by about 30 million people in Ethiopia, Kenya, Somalia and Egypt and is the 3rd largest language in Africa [58]. Currently, it is the official language of Oromia Regional State which is the largest regional state among the current Federal States in Ethiopia. It is used by Oromo people, who are the largest ethnic group in Ethiopia and account for more than 40% of the population [58]. Afaan Oromo is the instructional medium for primary and junior secondary schools throughout the region. Moreover, a number of literatures, newspapers, magazines, educational resources, official documents and religious writings are written and published in Afaan Oromo [59, 60]. With regard to the writing system, Qubee (a Latin-based alphabet) has been adopted and become the official script of Afaan Oromo [60].

4.2. Afaan Oromo Alphabet and Writing System

The writing system of Afaan Oromo language is straightforward which is designed based on the Latin script. Thus, letters in the English language are also in Oromo except the way it is written. Afaan Oromo text is written from left to right and spaces between words use as demarcation [61].

4.3. Consonant and Vowel phonemes

Like most other Ethiopian languages, Afaan Oromo has a set of ejective consonants, that is, voiceless stops or affricates that are accompanied by glottalization and an explosive burst of air. Afaan Oromo has another glottalized phone that is more unusual, an implosive retroflex stop, "dh" in Afaan Oromo orthography, a sound that is like an English "d" produced with the tongue curled back slightly and with the air drawn in so that a glottal stop is heard before the following vowel begins [62, 63]. Afaan Oromo has the typical Southern Cushitic set of five short (a, e, i, o, u) and five long vowels, indicated

in the orthography by doubling the five vowel letters (aa, ee, ii, oo, uu). The difference in length of vowels results in change of meaning.

For Example:

Afaan Oromo	**English**
hara	lake
haaraa	new

Gemination (doubling a consonant) is also significant in Afaan Oromo. That is, consonant length can distinguish words from one another.

For Example:

Afaan Oromo	**English**
Badaa	bad
Baddaa	highland

In Afaan Oromo alphabet, a letter consists either of a single symbol or a digraph (ch, dh, ny, ph, sh). Gemination is not obligatorily marked for the digraphs [62].

4.4. Punctuation Marks in Afaan Oromo

Words in Afaan Oromo sentences are separated by white spaces the same way as it is used in English. Different Afaan Oromo punctuation marks follow the same punctuation pattern used in English and other languages that follow Latin writing system. For example, comma (,) is used to separate listing of ideas, concepts, names, items, etc and the full stop (.) in statement, the question mark (?) in interrogative and the exclamation mark (!) in command and exclamatory sentences mark the end of a sentence[64].

4.5. Afaan Oromo Morphology

Like in a number of other African and Ethiopian languages, Afaan Oromo has a very complex and rich morphology [60]. 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 [61]. Afaan Oromo nouns have also a number of different cases and gender suffixes depending on the grammatical level and classification system used to analyze them. Frequent gender markers in Afaan Oromo include ***-eessa/-eettii***, ***-a/-ttii*** or ***–aa/tuu***.

Example:

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

Likewise, 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 [63, 64].

The extensive inflectional and derivational features of Afaan Oromo are presenting various challenges for a number of NLP tasks in the language.

Usually, WSD systems do not consider morphological variations of the context words. While this might not have any serious consequences for the performance of the algorithms for 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 [65]. A morphological-analyzer reduces the different forms of an ambiguous word into their root forms and plays an important role in this regard.

4.6. Ambiguities in Afaan Oromo

We try to identify different types of ambiguities that exist in Afaan Oromo based on types of ambiguity identified by Getahun [66]. Getahun identifies different types of ambiguity in Amharic language such as Phonological, Lexical, Structural, Referential and Semantic ambiguity. We now summarize each type of ambiguity with example in the following sub sections.

4.6.1. Phonological Ambiguity

Phonological ambiguity is a result 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 above sentence, '+' sign shows the place where the pause is 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*".

4.6.2. Lexical Ambiguity

Lexical ambiguity refers to a case in which either a lexical unit belongs to different part-of-speech categories with different senses, or to a lexical unit for

which there is more than one sense, while these different senses fall into the same part-of-speech category [67]. There are different factors that can cause lexical ambiguity such as Categorical Ambiguity, Homonymy and others.

Categorical Ambiguity

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

Barsiisan kutaa ***seena*** jira.

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

I. The teacher is getting into the class room. [With nominal meaning]
II. 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 above example the word "***ulfina***" is an ambiguous word having the following two different senses:

I. Tolaa has a huge weight
II. Tolaa is a respected person

4.6.3. Structural Ambiguity

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

Barsiisa seena Ferensay

The above sentence can have two different interpretations:

I. A French man who teaches History.
II. A person who teaches French History.

The structural organization of the constituent words in the above sentence is:

Barsiisa[N] seena[N] Ferensay[N]

4.6.4. 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 [68]. For example, ***Tolaan nama gudda dha (tolaa is a big man)*** you will 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 above sentence has two different meanings:

I. Gadisa was pleased because he graduated.
II. Somebody was pleased because Gaadisa graduated
III. Gadisa was pleased because he offered blessing.

Referential ambiguities are usually easy to spot and once recognized are easily avoided [68].

4.6.5. 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 above sentence has two interpretations:

I. The flower has grown.
II. Lalise's(name of a person)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 above example is "*that is my blood*" but the idiomatic expression refers to "that is my relative".

4.7. Summary

This Chapter, discussed about Afaan Oromo language such as Afaan Oromo alphabet and writing system, consonant and vowel phonemes, and Afaan Oromo punctuation marks. It also summarizes different type of Afaan Oromo ambiguities with examples. For this study we focused on lexical ambiguity which was believed to be resolved by word sense disambiguation. The next Chapter discusses the design and implementation of Afaan Oromo WSD.

Chapter Five: Design and Implementation of Afaan Oromo WSD

This Chapter is devoted to describing the design and implementation of WSD for Afaan Oromo language. It mainly focuses on design requirement, corpus preparation, architecture of Afaan Oromo WSD and prototype. In addition to this, the detail description of components on the architecture and their algorithms are also presented.

5.1. Design Requirements

The design and realization of every WSD must consider the language feature that it is intended for. In designing WSD for Afaan Oromo, typical features of the language play a pivotal role. In addition to this, four main elements are required in designing every WSD system: the selection of word senses, the use of knowledge sources, the representation of context, and the selection of an automatic classification approach [21].

5.1.1. Selection of Word Senses

A *word sense* is a commonly accepted meaning of a word. For instance, consider the following two sentences:

- Tolaan mana baankiti ***horii*** baayyee qaba.
- Qonnaan bultoonni hedduun ***horii*** horsiisuun galii argatu.

The word "***horii***" is used in the above sentences with two different senses: **qarshii (money)** in the 1st sentence and **beelada (cattle)** in the 2nd sentence. The example makes it clear that determining the sense inventory of a word is a key problem in word sense disambiguation. A *sense inventory* partitions the range of meaning of a word into its senses. Word senses cannot be easily discretized, that is, reduced to a finite discrete set of entries, fact that the language is inherently subject to change and interpretation [21].

5.1.2. Knowledge sources

Knowledge source provides data which are essential to associate senses with words. Knowledge sources used for WSD are either lexical knowledge released to the public, or world knowledge learned from a training corpus [21]. Lexical knowledge is usually released with a dictionary. World knowledge is too complex to be verbalized completely. So, it is a smart strategy to automatically acquire world knowledge from the context of training corpora on demand by machine learning techniques. For this study, we use training corpus as knowledge source.

5.1.3. Representation of Context

A standard approach to WSD is to consider the context of the ambiguous word and use the information from its neighboring or collocation words. This information is gathered from text representation of knowledge source (i.e. corpus) which is an unstructured source of information. To make it a suitable input to WSD, it is usually transformed into a structured format. To this end, a preprocessing of the input text is usually performed, which typically includes tokenizer, stop-word remover, stemmer and named entity recognizer.

5.1.4. Choice of a Classification Approach

Three main approaches have been applied in the field of WSD. These are knowledge based approaches, corpus based approaches and hybrid approach. Knowledge based approaches use Machine Readable Dictionaries (MRD). It relies on information provided by MRD. Corpus based approaches can be divided into two types, supervised and unsupervised learning approaches. Supervised learning approaches use information gathered from training on a corpus that has sense tagged for semantic disambiguation. Unsupervised leaning approaches determine the class membership of each object to be classified in a sample without using sense tagged training examples. Hybrid

approach combines aspects of fore mentioned methodologies [69]. For this study we use supervised learning approach.

5.2. Design of the Corpus

A corpus (plural corpora) is a collection of texts used for linguistic analyses, usually stored in an electronic database so that the data can be accessed easily by means of a computer. Corpus texts usually consist of thousands or millions of words and are not made up of the linguist's or a native speaker's invented examples but on authentic (naturally occurring) spoken and written language [73]. According to [70], corpus is expected to have the following features:

- Sampling and representativeness - many natural languages have large number of words and it is difficult to prepare the corpus that constitutes all the words in the language. Sample words are taken and used which can be representative of the other words. The sample has to represent variety of the words and their morphological and structural variation.
- Machine readable - nowadays many corpora are also expected to be machine readable even though it is not always true.

A corpus may exist in two different forms: unannotated and annotated corpus. Annotated corpus is a collection of texts that contains grammatical or linguistic information. Whereas unannotated corpus is a collection of text without linguistic information. Annotated corpus can be used for various purposes. In linguistics, properly annotated (tagged) corpus can be used to study linguistic features such as morphology and phonology of a language. It can also be used for part of speech taggers and WSD. The corpus will be provided to the system as training data so that the system can learn/adapt some pattern from the corpus for each word or sentence.

The size of the corpus affects the learning tendency of the system. Larger size of corpus provides greater learning tendency for the system. As a result, accuracy of the system will be better to automatically assign a meaning to ambiguous word. However, there is no such large size corpus which is already

prepared for Afaan Oromo language for disambiguation purpose. Preparation of this large size corpus is expensive and time consuming task. As a result of this, we created Afaan Oromo corpus that contains ambiguous word manually. The corpus was collected from different magazines, bulletins and Oromo news papers. As it is discussed in [71] bulletins, magazines and newspapers contain many social, economical, technological and political affairs of a certain society. Hence, they are good source for collected representative corpus for natural language processing.

The process of collecting such data took place in several ways. First, we selected words having more than two meanings from Afaan Oromo dictionary as an ambiguous word. In determining the words to be used in WSD, the most common words and their senses were chosen with care. Then the target word was searched in the document and sentences including this word were examined and used. Second, in order to acquire all the possible senses of the ambiguous word we collected additional sentences from different sources such as Afaan Oromo books, internet and others. For example, the Afaan Oromo ambiguous word "***soquu***" has two senses that are *"barbaadu (find)" and "qulquleessu (scour)".* Using these two senses, sense example sentences are acquired and used in a corpus.

In addition to this, ambiguous words in the corpus have been manually annotated with word senses by the help of Afaan Oromo experts and Afaan Oromo dictionary. For instance

1. Caaltuun kitaba ishee bade ***soquu*** deemte.
2. Tolaan lafa irra marga ***soquu*** deeme.

The Word "***soquu***" in the above two sentence is annotated with sense <***barbadu***> and <***qulquleessu***> respectively. During annotation of an ambiguous word, we ignore tagging of the ambiguous word with full definition rather we describe using a single word (statement) found in a dictionary. Those words are brief statements which are defined as "a unit of language that native speakers can easily identify them".

For example, the meaning (definition) of the word “soquu” in the first sentence is: “**want bade tokko barbadaani argachu (looking carefully in order to find something missed or lost**)” which is to mean **barbaadu (find)**” using a single word.

The meaning (definition) of the word “**soquu”** in the second sentence is: “**waan xuraa’e tokko qulqulessu** (**cleaning or removing dirt or unwanted matter from the surface of something by rubbing it hard)”** which is to mean **qulqullessu (scour)** using a single word. Therefore the above two sentences are annotated in the corpus using a single word i.e “barbaadu” and “qulqullessu” respectively as follows:

1. Caaltuun kitaba ishee bade ***soquu<barbadu>*** deemte.
2. Tolaan lafa irra marga ***soquu<qulqullessu>*** deeme.

As stated by Agirre & Martinez [72], the accuracy of machine learning algorithms degrade significantly when the training and testing samples have different distributions for the senses. In this study, we tried to use a balanced distribution of senses for the ambiguous words to maximize performance when enough sense examples are available. Therefore, a total of 1240 sense examples for 5 Afaan Oromo ambiguous words were used for this study as a corpus. Table 5.1 shows each ambiguous word with their corresponding data set count

Table 5.1: Ambiguous words and their sense example count in a corpus

No	Ambiguous word name	Sense Name	Translation	Data Set Count	
1	Sanyii	Ija midhaani ykn biqiltu	Seed	90	180
		Gosa	Type or Kind	90	
2	Horii	Qarshii	Money	130	260
		Beelada	Cattle	130	
3	Karaa	Daandi	Road	210	420
		Akkaata ykn kallatti	Way or via	210	
4	Sirna	Qophi	Event	90	180
		Seera	System or Procedure	90	
5	Qophii	Haala mijeessu	Preparation	100	200
		Sagantaa	Event/program	100	`
Grand Total					**1240**

5.3. Architecture for Afaan Oromo WSD

Considering the behavior of Afaan Oromo language, the proposed architecture for Afaan Oromo word sense disambiguation is shown in Figure 5.1. It has two main phases preprocessing phase and disambiguation phase. Preprocessing phase is required to prepare the data for further processing, which includes tokenizer, stop word remover and stemmer. Disambiguation phase involves a means to assign the appropriate sense to ambiguous word. To accomplish this, it incorporates various modules such as ambiguous word identifier, sense identifier, context extracter, sense counter, context counter and disambiguater module. The detail description of each of the components is given under section 5.3.1 up to 5.3.9.

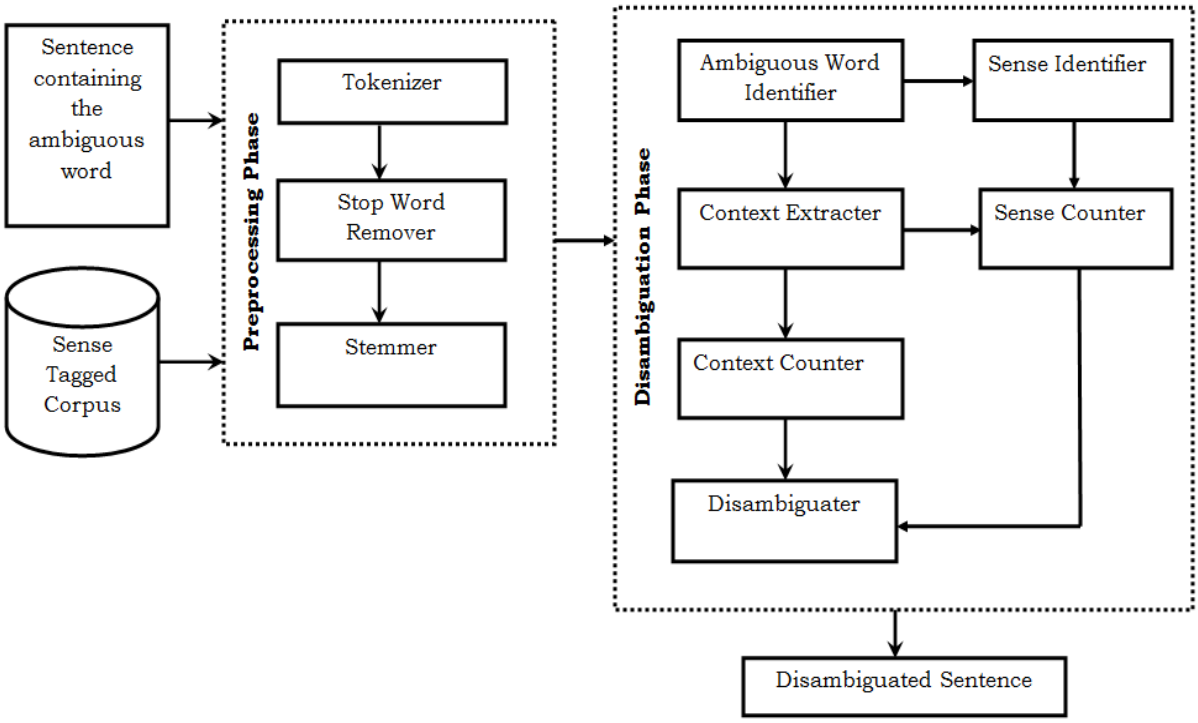

Figure 5.1: Architecture for Afaan Oromo word sense disambiguation

5.3.1. Tokenizer Module

Tokenizer is a module which splits up the text into a set of tokens usually words, based on the boundaries of a written text. Tokenizing of a given text depends on the characteristics of the language of the text in which it is written. Word demarcation in Afaan Oromo is handled following white space. Thus, Afaan Oromo tokenizer parses text into its constituent words usually by considering the white space and punctuation mark. Punctuation mark usage in Afaan Oromo is similar to that of English which include 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.

5.3.2. Stop Word Remover Module

Stop word remover 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. These words 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. This technique is much easier for well studied languages that have standard list of such words. As a result of this, dictionary lookup was employed for this study. 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 [39] is used. The algorithm is described in Figure 5.2.

1. *Open stop word list*
2. *Read a sentence*
   For each word in the sentence
      If a word is in stop word list then
         Remove a word
      End if
   End for
3. *Return the remaining words*
4. *Stop processing*

Figure 5.2: Algorithm for stop word remover

5.3.3. Stemmer Module

Stemmer is a module that reduces morphological variants of words into base or root form. In morphologically complex languages like Afaan Oromo, a stemmer will lead to significant improvements in WSD systems. According to [20], morphologies of a word, specially suffixes, can be composed of attached,

derivational, and inflectional suffixes. Afaan Oromo attached suffixes are particles or postpositions. Derivational suffixes are mainly used for the formation of new words in the language from stem or base form of a word. Inflectional suffixes of a word may indicate tense, case, plurality (number), and gender differences. The most common order/sequence of Afaan Oromo suffixes (within a given word) is: <stem> <derivational suffixes> <inflectional suffixes> <attached suffixes>. Thus, Afaan Oromo stemmer is expected to remove (from the right end of a given word) first all the possible attached suffixes, then inflectional suffixes and finally derivational suffixes step by step. For example, the word *barattootarratti* (on the students) is composed of *itti, irra* (attached suffixes), *oota* (inflectional suffix), *at* (derivational suffix), and *bar* (the stem). Therefore firs *-tti*, then *-rra,* then *-oota* and finally *-at* is removed to get the root "*bar-*". Affixes that are formed out of this sequence can also be removed though special condition. A stemmer for Afaan Oromo text developed by Debela Tesfaye [39] is adapted for this research work with some adjustment. That means we adjust the stemmer to work for a single word at a time. The algorithm is described in Figure 5.3.

1. *Read a word to be stemmed*
2. *If word matches with one of the rules*
   Remove the suffix and do the necessary adjustments
   Go to 3
   Else
   Go to 4
   End if
3. *Return the stemmed word to the calling portion*
4. *IF there is no applicable condition and action exist*
   Return the word as it is
5. *Stop processing*

Figure 5.3: Algorithm for Afaan Oromo stemmer

5.3.4. Ambiguous Word Identifier Module

It is a module used to identify the ambiguous word from the given sentence based on the information in the corpus. In order to perform WSD task, detecting the ambiguous word from the given sentence is the first step. Most of the time, a sense inventory is used for identifying ambiguous word. A sense inventory is a list of senses of a given word, which is nothing more than what is available in traditional dictionaries. As we mentioned earlier, semantically tagged corpus is used as a knowledge source for this study. Based on the data in the corpus, we identify the ambiguous word from the given input sentence using AWI module.

To detect an ambiguous word in the input sentence, first preprocessing step is done on the input sentence. Then each word in the input sentence is checked whether it is semantically tagged in the corpus or not. As we stated before, each sense examples in the corpus is manually tagged with their senses. That means all ambiguous words in the corpus were tagged with their correct senses according to the context in which they appear. Hence, each word in the input sentence was compared against manually tagged words in the corpus. When the mach is found, that word is detected as an ambiguous word. For example: assume that the following sentences were sense examples in the corpus.

- Caalan ***kutaa<mana>*** keessa gale. (Chala got into the room)
- Tolaan minda isa irra persenti kudhaan ji'atti ***kutaa<hirrisa>*** jira. (Tola is reducing ten percent of his salary per month)

And if the input sentence is:

- Caaltuun ***kutaa*** minda jirti. (Chaltu is in the salary room)

First, the input sentence and instances in the corpus are preprocessed. After preprocessing only three words will be left (i.e kut, mind and jirt) in the input sentence. Then each word in the input sentence is compared against manually

tagged words in the corpus. In our case ***"kutaa"*** is manually tagged as "***room and reducing***. So that, "**kutaa**" will be detected as ambiguous word in the input sentence. The algorithm is described in Figure 5.4.

1. *Read stemmed list of words in the input sentence*
2. *Open a corpus*
3. *Read a sentence from the corpus*
4. *Tokenize the sentence*
5. *For each token in the sentence*
    If token manually tagged
        Extract the substring up to '<' character
        Stem the substring
        For each stemmed list of words in the input sentence
            If a stemmed word in the input sentence match with - stemmed substring on the corpus
                Go to 6
            Else
                Go to 7
            End if
        End for
    End if
    End for
6. *Return a word as ambiguous word*
7. *IF end of corpus not reached*
        Go back to 3
    ELSE
        Return "no ambiguous word was detected"
        Go to 8
    End if
8. *Stop process.*

Figure 5.4: Algorithm for ambiguous word identifier

5.3.5. Sense Identifier Module

Sense identifier is a module used to identify the possible senses of ambiguous word in the given input sentence. A word sense is a commonly accepted meaning of a word. The identification was done based on the ambiguous word given by AWI module. During this task, all sense examples for a given ambiguous word were examined in the entire corpus. Finally all senses given for that ambiguous word were extracted and used as possible senses of a given ambiguous word. For instance, ***"marga" (grass)*** is the sense given for ambiguous word ***"citaa"*** in the following sense example.

- Mana citaa<margaa> ijaare. (He builds a house by grass)

To identify "***margaa***" as a sense of ambiguous word "***citaa***": First, manually tagged words in the entire corpus were selected for a specific ambiguous word. In our case *citaa<marga>* was selected. This selection was done based on the ambiguous word given by AWI module. Then, sense given for that ambiguous word was extracted. The extraction was done by removing the substring between the character "<" and ">". As such, we examine and extract all the possible senses of the ambiguous word in the entire corpus and fill into sense list. A sense list is an array list that holds all senses of a given ambiguous word. The algorithm is described in Figure 5.5.

1. *Read ambiguous words in the input sentence*
2. *Open a corpus*
3. *Read a sentence from the corpus*
4. *Tokenize the sentence*
5. *For each token in the sentence*
   If token contains ambiguous word
   Extract the substring starting from "<" up to '>' character
   Add the substring to sense list
   End if
   End for
6. *IF end of corpus not reached*
   Go back to 3
   Else
   Go to 7
   End if
7. *Return sense list*
8. *Stop process*

Figure 5.5: Algorithm for sense identifier

5.3.6. Context Extracter Module

Context extracter is a module used to select the context (sense example) that contains the ambiguous word from the corpus. All disambiguation works involve matching the context of the word to be disambiguated either with information from external knowledge source like Word-Net or with contexts of previously disambiguated instance of the word (i.e. semantically tagged corpus). In other words, we can say that context uniquely identifies meaning of the sentence. Based on this interpretation, the ambiguity of word known as lexical ambiguity is disambiguated. Specifically, in supervised approach to WSD the correct sense of ambiguous word is identified with the help of surrounding words in a sentence. That means the correct sense of a word is obtained from the context of the sentence. A different meaning of the single

word is associated with each sentence based on the context, the remaining sentence gives us. Thus, we use the context extraction module as a basic building block of our WSD. We did this extraction based on the ambiguous word identified by AWI module. During this operation, we discover the ambiguous word within all instances of the corpus. When a sense example containing the ambiguous word is found, we extract it and load into array list. The algorithm is described in Figure 5.6.

1. *Read ambiguous words in the input sentence*
2. *Open a corpus*
3. *Read a sentence from the corpus*
4. *Tokenize the sentence*
5. *For each token in the sentence*
   If token contains ambiguous word
   Add the sentence to context list
   End if
   End for
6. *IF end of corpus not reached*
   Go back to 3
   Else
   Go to 7
   End if
7. *Return list of sense examples*
8. *Stop process*

Figure 5.6: Algorithm for context extractor

The Feature Set

Feature extraction is a very important step in developing WSD system, which will then have a high effect on the system performance. A wide range of

features can be used in WSD. Specifically, in supervised approach to WSD a classifier is learned which is then used to assign senses to unseen examples. In these approaches, the initial input consists of the word to be disambiguated (target) word, along with text in which it is embedded which is called as context. This initial input is processed using part-of-speech tagging or any morphological processing. After this initial processing, fixed set of linguistic features are extracted relevant to learning task. These features are of two classes: collocation and co-occurrence features [21].

- Collocation features encode information about words of specific positions that are located to the left or right of the target word. Usually, this is set to a pre-defined window of two, sometimes three words on each side. Typical features include the word, the root form of word, and the word's part-of-speech.
- Co-occurrence features consist of data about neighboring words. In this approach words themselves serve as features. The value of feature is the number of times the word occurs in the region surrounding the target word. The region is often a fixed window with target word as center.

The contextual feature used in this thesis is co-occurrence feature which indicate word occurrence within some number of words to the left or right of the ambiguous word. Where the number of words to the left or right of the ambiguous word is determined by windows size. A windows size refers to the number of words need to considered, to the left and to the right of the ambiguous word, for the purpose of disambiguation. No part of speech tagging information is considered in these features since there is no full flagged POS tagger for Afaan Oromo language for public use. Punctuation and stop words are removed from the windows of context. In addition to this, stemming is performed for all lexical items included in the context window.

Niu, [14] proved in his experiments that Naive Bayes classifier achieved best disambiguation accuracy with small context window size (< 10 words). We follow their method and set the contextual window size to 4 as default widow's

size in our system, after performing an experiment on different windows size. That means four words to the right and four words to the left of the ambiguous word were considered during context extraction.

5.3.7. Sense Counter Module

Sense counter is a module which is used to count the occurrence of each unique sense, given for ambiguous word, in the extracted sense examples. This is done based on sense examples extracted for a single ambiguous word by the context extracter module. One ambiguous word can have a number of possible senses and each sense can have a number of sense examples in the corpus. So that, the purpose of this module is to count the occurrence of each unique sense given for the ambiguous word in the extracted sense examples. Finally, this sense count will be stored in sense count table for later use by disambiguater module. Sense count table is a table used to store sense list and its frequency count obtained from extracted sense examples. The implementation of Sense count table is similar to that of context count table which is discussed in section 5.3.8. Their difference is, SCT is used to maintain sense count list where as CCT is used to store context count list. For instance, if 25 sense examples are extracted for a given ambiguous word "*soquu*" by CE Module and if there are two senses for that ambiguous word (i.e. *barbaadu and qulqulessu*), And if the count of sense "*barbaadu" (find)* is 10 and the count of sense "*qulqullessu*" (scour) is 15, then, the sense count will be stored in SCT as shown in a Table 5.2 and returned to a calling portion of a system (i.e. disambiguater module). The algorithm is described in Figure 5.7.

Table 5.2: Sense Count Table for ambiguous word *soquu*

Word sense Name	*Frequency Count*
Barbaadu	*10*
Qulqullessu	*15*

1. *Read the word sense list identified by SI module*
2. *Read sentence list extracted from the corpus by CE module*
3. *For each word sense in the word sense list*
    For each sentence extracted from the corpus
        If the sentence contains word sense
            Increase the counter by one
        End if
    End for
    Add word sense and its count to SCT
   End for
4. *Return a SCT*
5. *Stop process*

Figure 5.7: Algorithm for sense counter

5.3.8. Context Counter Module

It is a module used to count the occurrence of each context with respect to its sense in the extracted sense examples. Context refers to words surrounding the ambiguous word. This kind of analysis is important in order to understand the exact meaning of the ambiguous word. Almost all supervised approach to WSD rely on local contexts (i.e., surrounding words), for disambiguation. In line to this, in our study we considered all surrounding words of the ambiguous word, except the stop words, as co-occurrence feature. Their frequencies are counted in the entire corpus with respect to each sense of an ambiguous word. This frequency indicates the number of times words around the target word is occurring in the extracted sense examples. We used context count table (CCT) for keeping co-occurrence of word w within a sentence s. For instance, if we observe a given context word three times in the entire corpus, we enter 3 as a count value to a respective context word.

A context count table is a table that stores frequency count of a context word in the extracted sense examples. The number of columns in the context count table is fixed to two which holds a word surrounding the ambiguous word and their frequency count. The number of rows in CCT is determined by the number of context surrounding the target word. The number of rows in the table shows words that are considered for disambiguation. For implementing context count table we use hash map in java.

The frequency of context word is counted based on the information we gather from context extractor module and sense identifier module. Both modules provide the necessary information for the contextual analysis. The context extractor module provides all the context word within a specific distance from the ambiguous word where the distance is determined by windows size. The sense identifier module provides all the possible sense detected for the ambiguous word. Based on this information, context counter module will count and record the frequency of each context word in respect to its sense within the extracted sense examples. For instance, suppose that we want to count the occurrence of context word surrounding the ambiguous word "***soqaa"*** in the following sentence.

Tolan marga lafa irra ***soqaa*** oolee dadhabee dhufe.

First, the frequency of each context word is counted in the corpus with respect to their senses. For instance, Table 5.3 shows the frequency count of the context word in the above sentence (i.e. *marga, lafa, irra, oolee, dadhabee* and *dhufe*) with sense "***barbadu***" and "***qulqullessu***" in the corpus.

Table 5.3. Sense example observed from the corpus for ambiguous word *soquu*.

	Marga	Lafa	Irra	Oole	Dadhabe	Dhufe
Barbaadu	0	0	3	6	5	6
Qulqullessu	3	12	4	2	5	6

Then, based on information observed from the corpus, CCT is filled for the two senses as shown in Table 5.4 and Table 5.5. In our case CCT is filled based on information in Table 5.3.

Table 5.4. Context Count Table for sense *barbaadu*

Context	***Frequency Count***
Marga	*0*
Lafa	*0*
Irra	*3*
Oole	*6*
Dadhabe	*5*
Dhufe	*6*

Table 5.5. Context Count Table for sense *qulqullessu*

Context	***Frequency Count***
Marga	*3*
Lafa	*12*
Irra	*4*
Oole	*2*
Dadhabe	*5*
Dhufe	*6*

Finally, this information is sent to disambiguater module for selecting the appropriate sense from a number of senses. The algorithm is described in Figure 5.8.

1. *Read list of context word in the input sentence*
2. *Read the word sense identified by SI module*
3. *Read sense example extracted from the corpus by CE module*
4. *For each context word in the input sentence*
    For each sense example extracted from the corpus
        If a sense example contains context word and word sense
            Increment the counter by one
        End if
    End for
    Add context word and its count in to CCT
   End for
5. *Return a CCT*
6. *Stop process*

Figure 5.8: Algorithm for context counter

5.3.9. Disambiguater Module

Disambiguater is a module used to calculate the occurrence probability of each sense of ambiguous word in a given sentence. This probability is calculated based on information provided by context counter and sense counter module. As we have mentioned in Chapter one, we used corpus based approach and Naïve Bayesian classifier for disambiguation. Hence, this module computes the score of each sense of ambiguous word and decides the most appropriate sense based on Naïve Bayesian classification technique.

5.3.9.1. Naive Bayesian Classification for WSD

Naive Bayesian is a learning algorithm. A learning algorithm is the forms of concept descriptions from example data. Concept descriptions are often referred to as the knowledge or model that the learning algorithm has induced from the data. Knowledge may be represented differently from one algorithm to another. For example, C4.5 represented knowledge as a decision tree; Naive Bayes represented knowledge in the form of probabilistic summaries. Experience shows that the NBC approach is effective and gives relatively good classification accuracy in comparison with other learning methods. This method was first used for WSD by Gale [42].

Naive Bayesian (NB) classifier for word sense disambiguation looks at the words around an ambiguous word in a large context window. For our study we use co-occurrence feature vector for representing content word. Each content word contributes potentially useful information about which sense of the ambiguous word is likely to be used with it. As we mentioned in other section, supervised approach was used for this study, which assumes semantically tagged corpus where each use of ambiguous words is labeled with its correct sense. Based on this assumption all features representing the problem are conditionally independent given the value of classification variables (i.e word sense).

For a word sense disambiguation tasks, giving a word w, candidate classification variables S= that represent the sense of the ambiguous word, and the feature F= that describe the context in which an ambiguous word occurs, the Naive Bayesian finds the proper sense s_i for the ambiguous word w by selecting the sense that maximizes the conditional probability P (si/F).

Suppose C is the context of the target word w, and F= is the set of features extracted from context C, to find the right sense of w given context C, we have:

$$S' = \arg\max P(s_i/F)$$

$$= arg\, max \frac{p(F/si)}{P(F)} P(si)$$

$$= \arg\max P(F/s_i)P(s_i)$$

The NB classifier works with the assumption that the features are conditional independent, so that we have:

$$S' = \arg\max \prod P(f_i/s_i)P(s_i)$$

The features for WSD using a NB algorithm are words which are extracted from the context of the ambiguous word. The probability of sense s_i, $P(s_i)$, and the conditional probability of feature f_j with observation of sense si, P(fj/si), are computed via Maximum-Likelihood Estimation:

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

$$P(f_j/s_i) = C(f_j, s_i)C(s_i)$$

Where $C(f_j, s_i)$ is the number of occurrences of f_j in a context of sense s_i in the training corpus, $C(s_i)$ is the number of occurrences of s_i in the training corpus, and N is the total number of occurrences of the ambiguous word w or the size of the training dataset. To avoid the effects of zero counts when estimating the conditional probabilities of the model, when meeting a new feature f_j in a context of the test dataset, for each sense s_i, we set $P(f_j/s_i)$ equal 1/N.

For instance, suppose that we want to classify the occurrence of "***horii" (money)*** in the sentence:-

- *mana bankiiti **horii** baayyee qaba (He has a lot of money in the bank).*

Given the features: *{w−2 =mana, w-1 = baankiiti, w+1=baayyee, w+2=qaba}*. Suppose that context count table for this feature is listed in the Table 5.6 and 5.7. And sense count for the two possible sense of the word "***horii***" is also listed in sense count table 5.8. CCT shows the number of occurrences of feature f_i in a context of sense s_i in the training corpus. Whereas SCT shows the number of occurrences of sense s_i in the training corpus for the ambiguous word "***horii"***.

Table 5.6. Context Count Table for sense *qarshii*

Context name	**Frequency count**
Mana	4
Baankiti	7
Baayyee	9
Qaba	12

Table 5.7. Context Count Table for sense *beelada*

Context name	**Frequency**
Mana	3
Baankiti	0
Baayyee	10
Qaba	14

Table 5.8. Sense Count Table for ambiguous word *horii*

Word sense name	Frequency count
Money/qarshii	20
Cattle/beelada	20

Note that context count and sense count table are returned by CCM and SCM respectively. Based on the above information, the occurrence probabilities of these four features given **"qarshi**" (**money)** sense of "**horii"** are calculated as follows:

$P(w-2 = mana / qarshii) = 4/20=0.2$

$P(w-1 = baankiti / qarshi) = 7/20=0.35$

$P(w+1 = baayyee / qarshi) = 9/20=0.45$

$P(w+2 = qaba / qarshi) = 12/20=0.6$

Also, we calculate the probability of occurrence of $P(qarshi) = 20/40=0.5$. Therefore, the probability of "***qarshii" (money***) sense of "**horii"** is calculated as:

$p(qarshii) = (0.2*0.35*0.45*0.6) *0.5= 0.00945$

Likewise the occurrence probability of the four features given "***beeledaa" (cattle)*** sense of "***horii***" is also calculated as:

$P(w-1 = mana / beelada) = 3/20=0.15$

$P(w+1 = baankiti / beelada) = 1/40=0.025$

$P(w+2 = baayyee / beelada) = 10/20=0.5$

$P(w+3 = qaba / beelada) = 14/20=0.7$

In order to avoid the effects of zero counts when estimating the conditional probabilities of $P(w+1 = baankiti / beelada)$ we set it equal to 1/N, where N is the total number of occurrences of the ambiguous word in the training dataset. Also, we calculate the probability of occurrence of $P(beeleda) = 20/40=0.5$. Therefore, the probability of "***beelada" (cattle)*** sense of "***horii***" is calculated as:

$p(beelada) = (0.15*0.025*0.5*0.7)*0.5= 0.00065625$

Finally, the disambiguater module selects a word sense with the highest probability, to find the right sense of ambiguous word "***horii***" in the given context. As a result of this, ***qarshii (money)*** is selected as the appropriate sense of ambiguous word "***horii***". The algorithm is described in Figure 5.9.

1. *Read the SCT created by SCM*
2. *Initialize a variable score to one*
3. *For each word sense and sense count in the SCT*
   Add sense count to Total count
   End for
4. *For each word sense and sense count in the SCT*
   Read context word and its count in the CCT
   For each context word in CCT
   *Calculate the occurrence probability as score=score * p(context count/sense count)*
   End for
   *Calculate probability=Score * p(sense count/Total Count)*
   If the probability is the largest probability
   Keep the word sense on some variable
   End if
   End for
5. *Display a word sense as the final result of disambiguation*
6. *Stop process*

Figure 5.9: Algorithm for Disambiguater Module

5.4. The Prototype

Developing a prototype to demonstrate the usability of the proposed Afaan Oromo Word Sense Disambiguation is one of the objectives of this study. Hence, we developed the prototype of Afaan Oromo WSD using Java programming language. The main screen of the system is depicted in Figure 5.10.

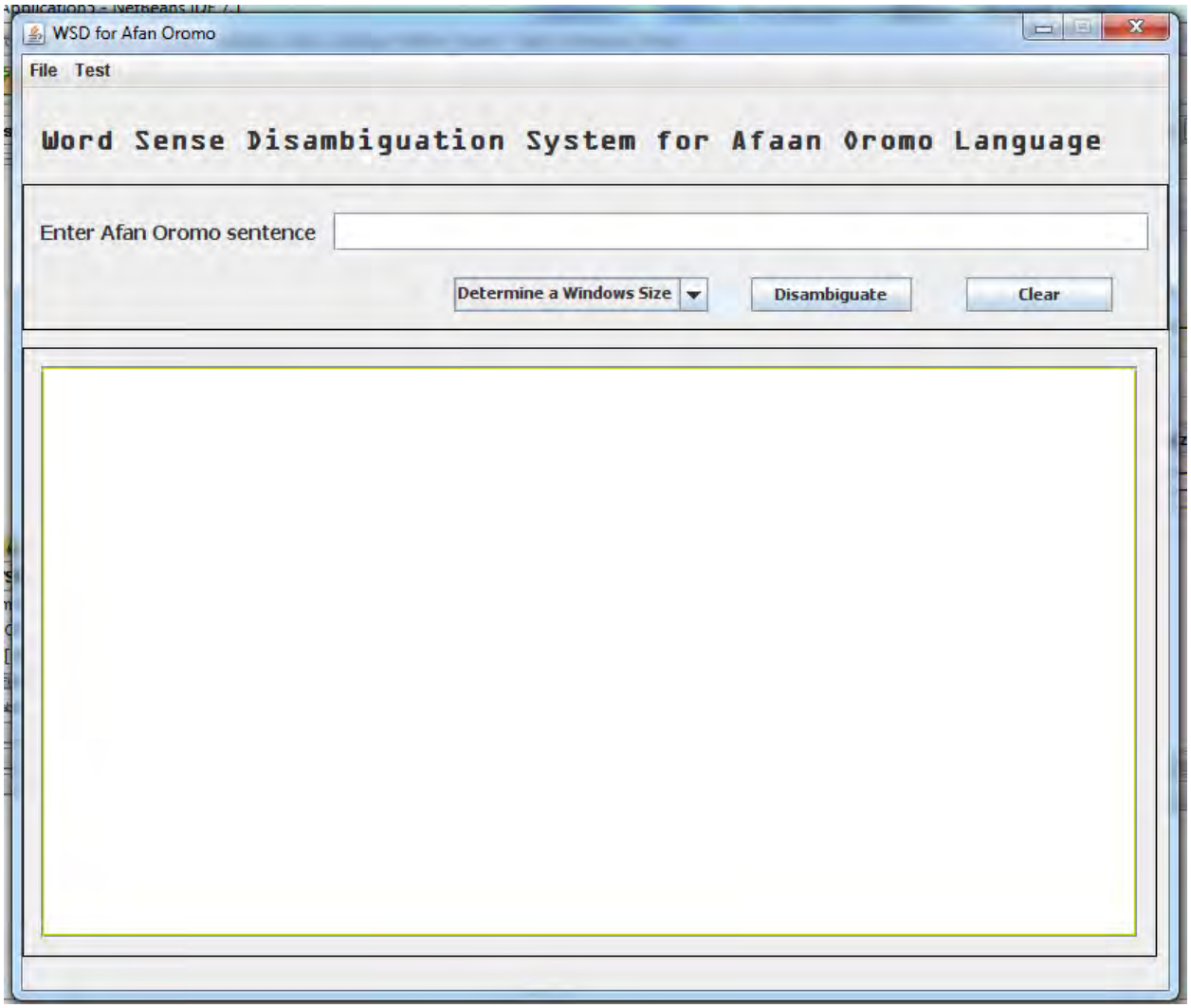

Figure 5.10: Screen shoot for the main screen of Afaan Oromo WSD

The input for the prototype is either a sentence or text file. The text file can be browsed from its location. The system will disambiguate the given sentence automatically after the disambiguate button is pressed. A word that the system believes to be ambiguous and the exact meaning of the ambiguous word according to the given sentence will be displayed as an output on the white screen. If there is no ambiguous word in the input sentence the system display a message box and allow the user to enter another sentence. Determine a windows size combo box is an option which used to limit the number of words

to be considered to the write and to the left of ambiguous word during disambiguation.

Figure 5.11 shows a sample output for a sentence "*Horii qalee nyaate".* The output screen shows the disambiguation result as well as the detail process of disambiguation. As we have mentioned earlier, a number of modules (components) are used in Afaan Oromo WSD system such as tokenizer, stemmer, AWI, SSM, CCM SCM and CEM. All these modules have their own output. Hence, the outputs of all these modules are also displayed on the output screen.

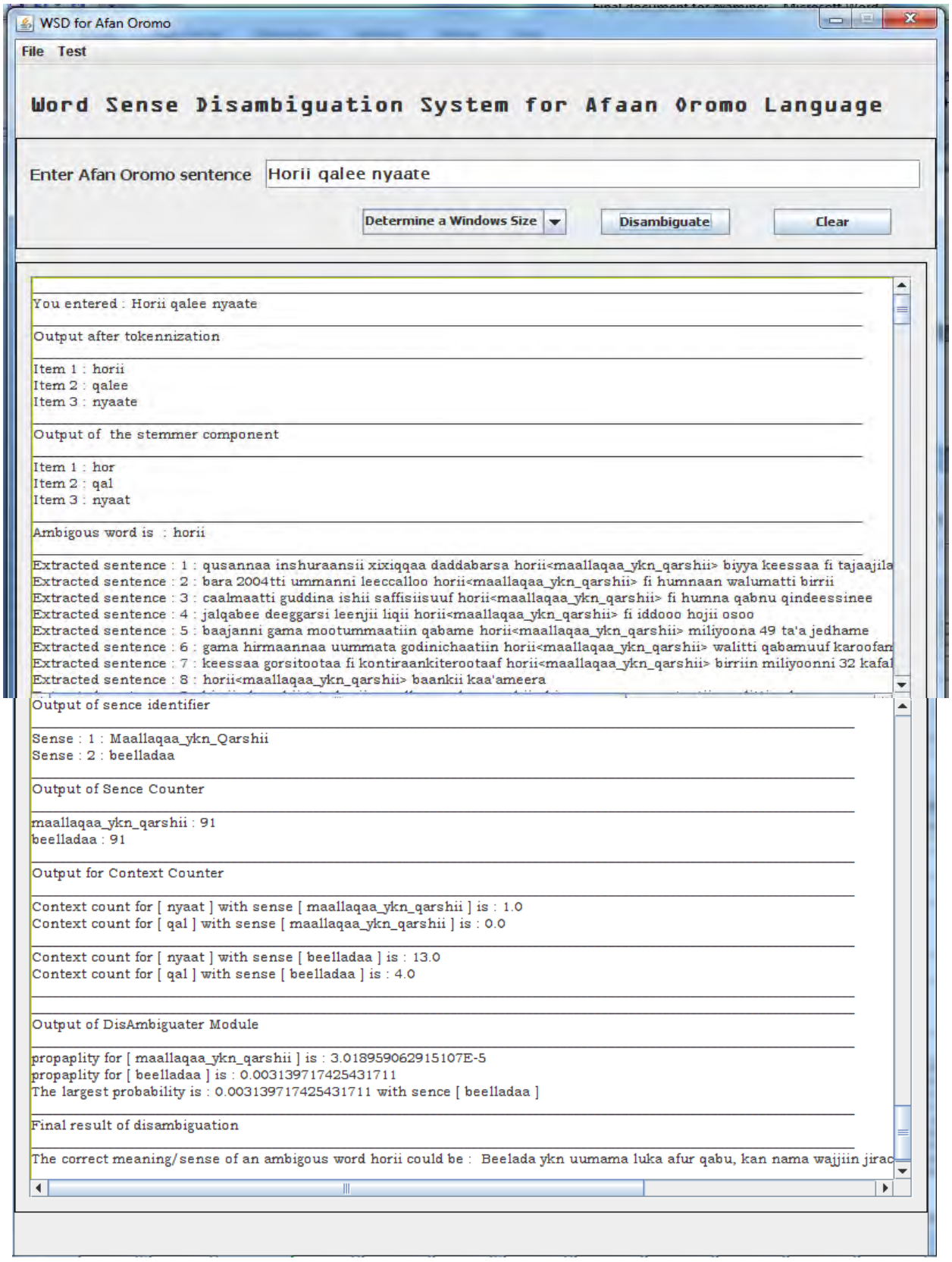

Figure 5.11: Screen shoot for the result of disambiguation

If the user wants to disambiguate more than one sentence at one time, they can use test menu from the menu bar. The input for the test menu is a text file. These file can be accessed through open file sub menu. Then, the user can use test button for disambiguation. We use this screen for testing the performance of our system. The screen shoot for testing is depicted on figure 5.12.

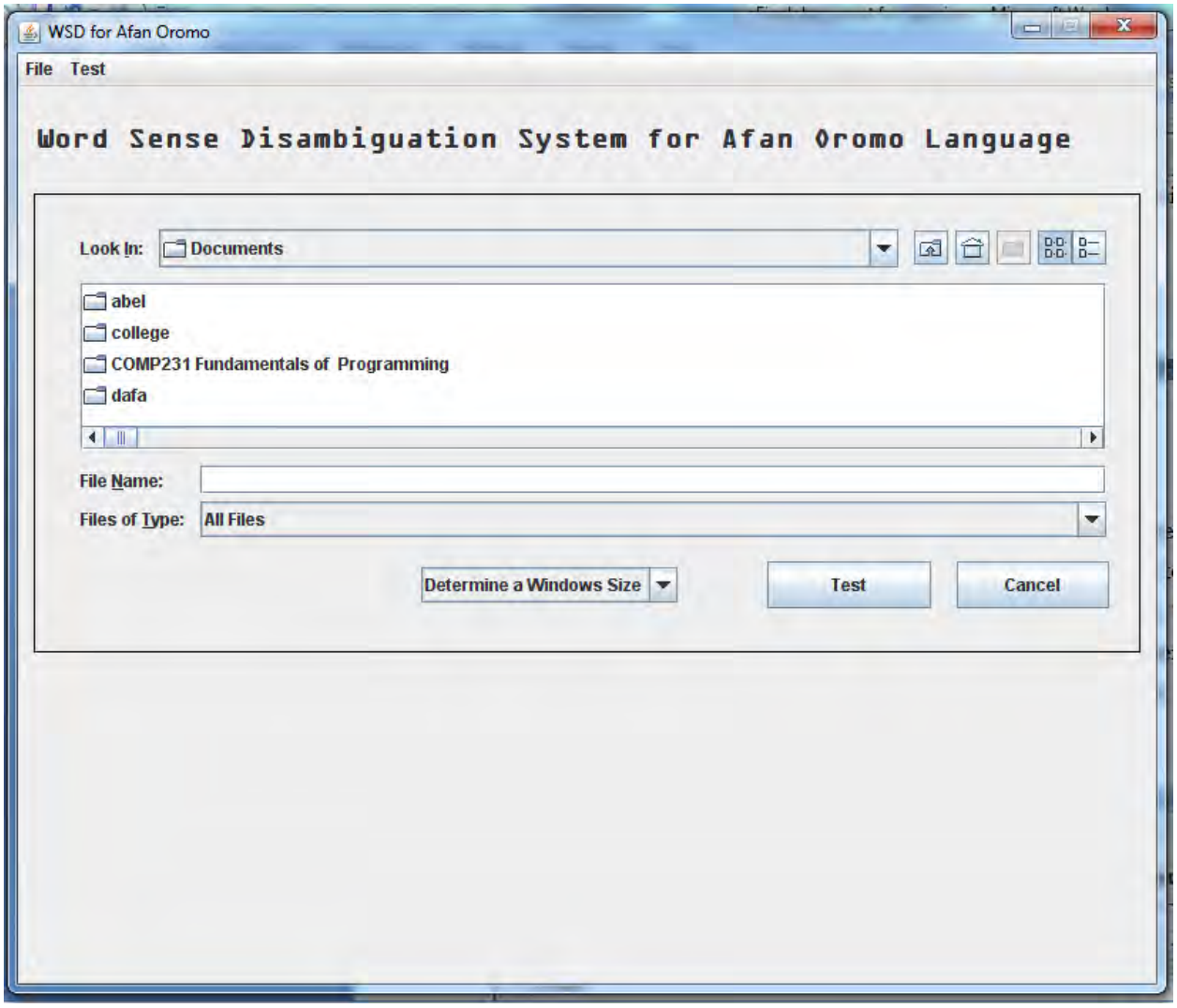

Figure 5.12: Screen shoot for disambiguating more than one sentence at a time

5.5. Summary

In this Chapter, the design and implementation of Afaan Oromo WSD was presented. It includes design requirement, corpus preparation, architecture of the system and its prototype. The components used in developing WSD system and their explanation are also given in detail. These components were organized under preprocessing phase which is required to prepare the data for further processing, and disambiguation phase that involve a means to assign the appropriate sense to ambiguous word. To find the likelihood ratio of the sense in the given context the system employs Naive Baye's theorem. In addition to this, algorithm developed by the researcher for all the components are also described. Finally the prototype of the system was presented. Java programming language was used to develop the prototype. The next Chapter presents the experimentation and discussion on the results of the experiment.

Chapter Six: Experimentation and Result

6.1. Introduction

Evaluation plays an important role to determine the accuracy of any learning method. Machine learning method needs data for training and testing, in order to evaluate the performance of the system. There are several ways of doing this evaluation and the most common is to split data into two sets, training set and test set. Although this distribution is commonly used for large datasets, it presents a challenge for smaller datasets and it might lead to problem of representativeness of the training or testing data. In evaluation of our system we conduct learning curve analysis that best suit to detect goodness of the size of the dataset. Learning curve is one of the most important tools to indicate whether the data set is sufficiently large or not. Based on the analysis result we conclude that our data set is not sufficiently large. Hence a statistical technique called k-fold cross-validation is applied for evaluation. We use this technique to avoid inaccuracy of results due to data splitting.

In k-fold cross-validation, the initial data are randomly partitioned into k mutually exclusive subsets or "folds," D_1, D_2, … , D_k, each of approximately equal size [15]. Training and testing is performed k times. In iteration i, partition D_i is reserved as the test set, and the remaining partitions are collectively used to train the system. That is, in the first iteration, subsets D_2, … , D_k collectively serve as the training set which is tested on D_1; the second iteration is trained on subsets D_1, D_3, … , D_k and tested on D_2; and so on. Hence, different combinations of training and testing data are used, i.e. the data set is divided in different ways in such a way that different combination of sense examples are available for training and testing each time. For these experiments, 10-fold cross validation is used which divides the data set into ten sets, each set containing 10% of the total data.

In the ideal setup, formal evaluation of a WSD system would require a sizeable hand annotated corpus containing several ambiguous words that would provide a gold standard for evaluation. In addition, performance figures for other systems on the same task and evaluated against the same gold standard would be required, in order to benchmark the performance of the developed system [5]. However, there is not any cited work for resolving ambiguity of words in Afaan Oromo language. Hence, we didn't evaluate the developed system with any other related system. In addition, to alleviating the need for a sizeable hand annotated corpus, a small manually tagged corpus was prepared for evaluation. This corpus is made up of 1240 Afaan Oromo sentences. It contains 5 ambiguous words namely ***sirna, karaa, sanyii, qophii*** **and** ***horii*** which are defined in Chapter Five and each having two major senses and whose distribution in the corpus is not skewed to a particular sense, i.e. both senses appear with comparable (equal) frequencies. The contextual features used in this experiment were co-occurrence features which indicate a word occurs within some number of words to the left or right of the ambiguous word.

6.2. Evaluation Metrics

To measure the rate of disambiguation of our system, we use the most common evaluation techniques, which select a small sample of words and compare the results of the system with a human judge. We use the metrics such as precision P, recall R, F_1-measure and accuracy.

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

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

$$\mathrm{F}_1 \text{ Measure} = \frac{2*P*R}{P+R}$$

$$\text{Accuracy (Acc)} = \frac{TP+TN}{Pt+Nt}$$

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 as used in these definitions are associated with membership to one of the two semantic classes (senses) involved in the classification. For example, where disambiguation involves the classes BEELADA (CATTLE) and QARSHII (MONEY), Pt and Nt refer to the total number of test occurrences belonging to class BEELADA and QARSHII respectively, while TP (TN) refers to the BEELADA (QARSHII) test occurrences correctly classified as such by the system. Likewise, FP (FN) refers to those QARSHII (BEELADA) test occurrences that have been misclassified by the system as belonging to class BEELADA (QARSHII).

Due to the performance trade-off between precision and recall, the F1 measure, computed as a harmonic mean between these two values, yields a single number by which performance can be measured. This provides a convenient way to compare the performance of two or more classifiers on the same problem, ranking them in order of quality of prediction.

The other evaluation metric used in this study is accuracy. Accuracy is the easiest and most common way of reporting the performance of machine learning methods. However, for some classification tasks, especially those involving highly imbalanced data, more precise metrics should be adopted in order to evaluate results more clearly. The accuracy value enables comparison of a classifier's performance against a given base line such as the majority

classifier which acts as the lower bound for the performance of probabilistic classifiers.

6.3. Experimentation procedure

In this study, a total of two experiments were conducted. The first experiment was conducted to evaluate the performance of our algorithm. The second experiment sought to investigate the effect of different context sizes on disambiguation accuracy for Afaan Oromo ambiguous word, and to find out, if the standard two-word window applicable for other languages and especially English [18] holds for Afaan Oromo. In this regard, different training and test data sets were prepared for each ambiguous word, where the contextual information was obtained from 1-left and 1-right to 10-left and 10-right consequent surrounding words.

6.4. Discussion of Results

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

Experiment I: Evaluating performance of the algorithm

As we stated earlier, to evaluate the performance of the algorithm, 10-fold cross-validation evaluation technique is used in our experiment. In this technique, first the total data set is divided into 10 mutually disjoint folds approximately of equal size using stratified sampling mechanism. In stratified sampling, the folds are stratified so that the class distribution of the tuples in each fold is approximately the same as that in the initial data. We have a total of 1240 manually tagged sense examples which is divided into 10 approximately of equal size. As a result of this each fold of a data set contains 124 sense examples with balanced distribution number of senses per fold. It is described in Table 6.1.

Table 6.1: Balanced distribution of sense examples in each fold

No	**Ambiguous word name**	**Sense Name**	**Class label**	**Count in each fold**	**Total**
1	Sanyii	Ija midhaani ykn biqiltu	Positive	9	18
		Gosa	Negative	9	
2	Horii	Qarshii	Positive	13	26
		Beelada	Negative	13	
3	Karaa	Daandi	Positive	21	42
		Akkaata ykn kallatti	Negative	21	
4	Sirna	Qophii	Positive	9	18
		Seera	Negative	9	
5	Qophii	Haala Mijeessu	Positive	10	20
		Saganta	Negative	10	
Grand Total					**124**

Second, the training set and testing set was identified and separated from the total data set. In order to check the result using the developed system, we remove manually tagged sense examples from test set. Before doing the actual experiment, pre-test has been done by the researchers using sense examples in test set and comparing the result with manually tagged test set. The pre-test has been conducted iteratively to increase prototype's performance. The errors encountered during this experimentation have been corrected and the experiment has been done iteratively until the result is found to be satisfactory.

Finally, the actual test was conducted using sense examples in test set. During this process nine fold were used for training the developed system whereas the remaining tenth fold was used for testing the system that was trained on the previous nine folds. The process was repeated ten times by taking other nine as training and tenth one as testing. After each training phase, the system was tested on average of 124 Afaan Oromo sentence. Each of the corresponding training set contains an average of 1116 sentences. The result on test data set was obtained by comparing the result returned by the system with the corresponding test set which was manually tagged. Then, the class labels (sense) assigned by the system are counted for each test data set accordingly. Positive tuples that were correctly labeled by the system were counted as true positive, while negative tuples that were correctly labeled by the system are counted as true negative. Similarly, negative tuples that were incorrectly labeled by the system are counted as false positives and positive tuples that were incorrectly labeled by the system are counted as false negative. Based on this information, precision, recall, f-measure and accuracy of the system were measured. Summary of the evaluation result obtained for each data set is presented on appendix B.

Table 6.2: The average evaluation result of the system

Data Set Number	**Precession**	**Recall**	**F-measure**	**Accuracy**
Data Set 1	0.85	0.91	0.88	0.86
Data Set 2	0.74	0.88	0.80	0.77
Data Set 3	0.74	0.89	0.81	0.78
Data Set 4	0.78	0.89	0.83	0.82
Data Set 5	0.78	0.86	0.82	0.79
Data Set 6	0.74	0.91	0.82	0.78
Data Set 7	0.69	0.88	0.77	0.75
Data Set 8	0.70	0.92	0.79	0.76
Data Set 9	0.76	0.82	0.78	0.77
Data Set 10	0.77	0.88	0.82	0.80
Total Average	**0.76**	**0.88**	**0.81**	**0.79**

In order to measure our system in terms of precision, recall and accuracy, Table 6.2 presents the results based on the average scores obtained from the ten data sets. During this evaluation first precision, recall and accuracy of each ambiguous word for each data set was calculated based on count obtained for TP, TN, FP and FN values (i.e. it is summarized on Appendix B). Then average score for each data set was calculated based on evaluation metrics presented in section 6.2. Finally, as shown in the table 6.2, the total average score of the ten data sets was calculated to get the final result. Hence, our system gets 76% precision, 88% recall and 81% F1-measure and 79% accuracy. As this research was the first attempt to create a word sense disambiguation system for Afaan Oromo Language we got a satisfactory result in terms of accuracy. Therefore, the system can improve the accuracy of Afaan Oromo NLP application such as

speech recognition, information retrieval, machine translation, text processing, computational advertising and others.

Table 6.3 presents the accuracy obtained for each of the five ambiguous words along with the respective precision, recall, F1-measure and accuracy value. The system performed better for all ambiguous words in terms of accuracy. The best accuracy results were obtained for the word *sanyii and karaa.* This is due to the contextual future considered for disambiguation for the two words was minimum compared to others. For example *ija_midhaani ykn_biqiltu* (seed) sense of ambiguous word *sanyii* has a minimum number of context words that goes with it than others. The second and the third best accuracy results were obtained for a word *horii* and *sirna* respectively.

Table 6.3 : Summary of evaluation result for individual ambiguous word

	Precision	**Recall**	**F1-Measure**	**Accuracy**
Sanyii	0.79	0.88	0.83	0.81
Horii	0.78	0.85	0.81	0.79
Karaa	0.82	0.78	0.79	0.81
Sirna	0.72	0.92	0.81	0.77
Qophii	0.68	0.98	0.80	0.76

The accuracy of WSD depends on the size of the training data. For example consider the case of ambiguous words *karaa* and *horii* with average number of training instances 260 and 420 respectively. And average number of training instances for the words *sirna* and *qophii* are 180 and 200 respectively. When we compare the accuracy result obtained for the words *karaa* and *horii* with the accuracy obtained for the words *sirna* and *qophii,* the accuracy is higher for the words *karaa* and *horii* as they have more number of training instances than the words *sirna* and *qophii*. Hence, the number of training instances used for an ambiguous word affects the accuracy of the system.

The best recall value was scored with the word *qophii* which shows how complete the classification is on the positive class. A positive class for the word *qophii* was *saganta* and a negative class was *mijessu* (i.e. *saganta* and *mijessu* are the two senses of ambiguous word *qophii*). The performance evaluation of our system considers those senses assigned as positive class to calculate precision and recall values. The word *qophii* was complete on the positive class. That means *saganta* sense (i.e. positive class) of a word *qophii* performs better than *mijeessu* sense (i.e. negative class) of a word *qophii*. However, the lowest precision value was scored with the word *qophii* which is 0.68. From this result one can observe that the same recall result cannot be found with negative class of *qophii*. Hence, the ambiguous word *qophii* is not complete on the negative class. Hence, due to the performance trade-off between precision and recall, the F1 measure, was computed as a harmonic mean between these two values, which yields a single number by which performance can be measured. The highest F1 measure was scored with the word *sanyii* and the lowest value was scored with the word *qophii*. Hence, *qophii* was the least performing word among all.

Experiment II: Determining optimal context window

Windows size refers to the number of words needed to be considered, to the left and to the right of the ambiguous word, for the purpose of disambiguation. In English, a standard two-word window on either side of the ambiguous word is found to be enough for disambiguation [18]. For Amharic supervised WSD, three window size on both sides for the ambiguous word is found to be enough [52]. To determine optimal window size for Afaan Oromo WSD, experiments were carried out ten times from one-one window to ten-ten window on both side of the ambiguous word.

Table 6.4 : Summary of window size experimentation for the system

WS	Sanyii	Horii	Karaa	Sirna	Qophii	average
1-1	0.67	0.74	0.82	0.68	0.58	0.69
2-2	0.76	0.79	**0.86**	0.74	0.58	0.75
3-3	0.74	0.79	0.85	0.79	0.62	0.76
4-4	**0.83**	**0.81**	**0.86**	0.85	0.63	**0.79**
5-5	0.79	0.79	0.85	0.83	0.6	0.77
6-6	**0.83**	0.79	0.82	0.85	**0.65**	0.78
7-7	**0.83**	0.78	0.81	0.87	**0.65**	0.78
8-8	0.79	0.72	0.82	0.87	0.6	0.76
9-9	0.79	0.72	0.84	**0.89**	0.61	0.77
10-10	0.79	0.76	0.84	0.88	0.63	0.78

As shown in Table 6.4 for the ambiguous word *sanyii,* the maximum accuracy was achieved on four-four, six-six and seven-seven word window size. Whereas, for ambiguous word *horii* the highest accuracy was attained on four-four word window size and for ambiguous word *karaa* the highest accuracy was attained on two-two and four-four word window size. Among all, the maximum accuracy was achieved for ambiguous word *sirna* with nine-nine windows size. The lowest accuracy result was scored for ambiguous word *qophii*. This is due to the minimum number of contextual feature for the word *qophii* and minimum number of training data used for disambiguation.

Even if the maximum accuracy result for the word *sirna* is achieved with windows size nine-nine; all the other words achieve the highest accuracy result between two-two windows size up to seven-seven windows size. Hence, the result agreed with the findings in other language that the nearest words surrounding the ambiguous word give more disambiguation information than

words far from the ambiguous word [18]. For this study, since the average accuracy result for windows size four-four is larger than all the other windows, window size four-four was considered to be effective for Afaan Oromo Word Sense Disambiguation.

Chapter Seven: Conclusion and Recommendations

7.1. Conclusion

Many words have more than one meaning in natural language, and each one of them is determined by its context. The automated process of recognizing word senses in context is known Word Sense Disambiguation (WSD). The problem of distinguishing between multiple possible senses of a word is an important subtask in many NLP applications such as machine translation (MT), semantic mapping (SM), semantic annotation (SA), and ontology learning (OL). It is also believed to be helpful in improving the performance of many applications such as information retrieval (IR), information extraction (IE), speech recognition (SR) and others [4].

There are three main approaches used for assigning senses to words in a context. These are knowledge-based approaches, corpus based approaches and hybrid approach. Knowledge based approaches uses information provided by Machine Readable Dictionaries (MRD), Corpus based approaches uses information gathered from training corpus and Hybrid approach combines aspects of the two methodologies [11]. Corpus based approach further divided as supervised and unsupervised approach based on whether sense examples in a corpus is manually tagged with their sense or not. We used supervised based approach to develop WSD for Afaan Oromo language.

This research work is the first attempt to develop a word sense disambiguation system for Afaan Oromo Language. As there is no large size corpus which is already prepared for Afaan Oromo language for WSD purpose, we prepared Afaan Oromo corpus manually for this study. During this preparation we have selected 5 ambiguous words having two senses on average. Based on the 5 words, we extracted 1240 sentences from Afaan Oromo news paper as our training and test set which is then manually tagged by the help of Afaan Oromo experts.

The architecture of the system includes two main phases: preprocessing and disambiguation phase. The system takes Afaan Oromo sentence including ambiguous words as an input. In the preprocessing stage, it segments the input sentence by using Afaan Oromo word segmenter and removes all words that can be stop words from the input sentence. By considering the morphological variants of the language, stemming is also applied in the preprocessing stage. After gathering information in the preprocessing step, the system uses the remaining words in the input sentence as features which are called co-occurrence feature in this thesis. The system also identifies ambiguous word and its sense based on the information in the corpus. Then, the system calculates the occurrence probability of each context word based on Bayes Theorem. Finally, the disambiguation process computes the score of each sense of ambiguous word and decides the most appropriate sense for a given ambiguous word in the input sentence.

We have conducted two experiments; one for evaluating the performance of the prototype and the other for determining an optimal windows size for Afaan Oromo WSD. For the first experiment we have achieved 79% accuracy. For the second experiment we have found that four-word window on each side of the ambiguous word is enough for Afaan Oromo WSD.

7.2. Recommendations

We have the following recommendations which shows interesting research directions that if undertaken, would further improve the developed WSD for Afaan Oromo

- ☞ Researches for WSD in other languages use linguistic resources like Thesaurus, Word-Net, Machine Readable Dictionaries and machine translation software. Regarding this knowledge sources adopted by WSD

systems, in recent years, the results of many research efforts for the construction of online lexical knowledge repositories, ontologies and glossaries became available creating new opportunities for knowledge-based sense disambiguation methods. We recommend the development of these resources to enhance WSD for Afaan Oromo.

☞ For other languages standard sense annotated corpora is available for WSD research and also for testing a WSD system. We don't have such data for Afaan Oromo language which makes the study to be limited for five ambiguous words. So, large size Afaan Oromo corpus for WSD research is necessary.

☞ A wide range of features can be used in WSD. The contextual features used in this thesis were co-occurrence feature which indicate word occurs within some number of words to the left or right of the ambiguous word. But a number of features can be used in WSD including POS tagging information of neighboring words. We recommend those features to be included in future work.

☞ The other recommendation is extending this experimentation using both supervised and unsupervised WSD including other ambiguous words in addition to those covered in the research. In addition to corpus based approach, there are also knowledge based and hybrid approach (combination of knowledge base and corpus based approach) which are used for WSD for other languages and produced a good result. These approaches need to be investigated for Afaan Oromo language as well.

☞ As unsupervised WSDs are based on unlabeled corpora, and do not exploit any manually sense-tagged corpus to provide a sense choice for a word in context, we recommend future work to use this approach to avoid the problem of knowledge acquisition bottleneck, that is, lack of large-scale resources manually annotated with word senses

- ☞ The system developed in this research work is just a prototype. Any interested body can do a project to make a full-fledged Afaan Oromo WSD that can be easily integrated into different Afaan Oromo NLP works such as machine translation, information retrieval, information extraction, speech recognition and the like.

- ☞ There are a lot of holes in the linguistic study of the language in general, and in morphology of the language in particular. Since morphological inflections affect the system performance, Linguists should give appropriate consideration to intensively study the language structure and make it available for use in developing computational models.

Reference

[1] David Vickrey, Luke Biewald, Marc Teyssier, Daphne Koller, Word-Sense Disambiguation for Machine Translation, Department of Computer Science, Stanford University, Stanford, The Association for Computational Linguistics, 2005.

[2] Jose Maria Gomez Hidalgol, Manuel de Buenaga Rodrıguez1, Jose Carlos Cortizo Perez, The Role of Word Sense Disambiguation in Automated Text Categorization, University Europea Madrid, Spain, Appeared in the 15th NLDB Conference, 2005.

[3] EshaPalta, Kanwal, Rekhi, word sense disambiguation, Master's Thesis, School of Information Technology Indian Institute of Technology, Powai, Mumbai, 2006-2007.

[4] Xiaohua Zhou, Hyoil Han, Survey of Word Sense Disambiguation Approaches, College of Information Science & Technology, Drexel University, Appeared in The 18th FLAIRS Conference, Clearwater Beach, Florida, 2005.

[5] Wanjiku Ng'ang'a, Word Sense Disambiguation of Swahili: Extending Swahili Language Technology with Machine Learning, University of Helsinki, Finland, [http://ethesis.helsinki.fi], 2005.

[6] Dorr, Bonnie, Machine translation divergences: A formal description and proposed solution, Computational Linguistics, 597-633, 1994,

[7] Ping Chen, Chris Bowes, David Brown, Wei Ding, A Fully Unsupervised Word Sense Disambiguation Method Using Dependency Knowledge, University of Houston and University of Massachusetts-Boston, Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, 28–36, 2009.

[8] Banerjee S., Pedersen T., Extended gloss overlaps as a measure of semantic relatedness, In Proc. of IJCAI-03, 2003.

[9] Ariel Raviv, Shaul Markovitch, Concept-Based Approach to Word-Sense Disambiguation, Computer Science Department, Israel Institute of Technology, Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, 807-813 Toronto, Canada, 2012.

[10] Ying Liu, Peter Scheuermann, Xingsen Li, Xingquan Zhu, Using WordNet to Disambiguate Word Senses for Text Classification, Data Technology and Knowledge Economy Research Center, Chinese Academy of Sciences Graduate University of Chinese Academy of Sciences, Beijing, China , ICCS 2007 Part III- LNCS 4489, 2007.

[11] Krovetz R., W. B. Croft, Lexical Ambiguity and Information Retrieval, ACM Transactions on Information Systems, 115–141, 1992.

[12] Krovetz R, Homonymy, Polysemy in Information Retrieval, Proceedings of the 35th Meeting of the Association for Computational Linguistics, 72–79, 1997.

[13] Dill, Stephen, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, Jason Y. Zien. SemTag, Seeker, Bootstrapping the Semantic Web via automated semantic annotation, Proceedings of the Twelfth International Conference on World Wide Web (WWW-2003), Budapest, Hungary, 178–186, 2003.

[14] Zheng-Yu Niu, Dong-Hong Ji, Chew Lim Tan, Optimizing Feature Set for Chinese Word Sense Disambiguation, To appear in Proceedings of the 3rd International Workshop on the Evaluation of Systems, 2004.

[15] Jiawei Han, Micheline Kamber, A book called Data Mining: Concepts and Techniques, Second Edition, University of Illinois at Urbana-Champaign, 2006.

[16] Liddy E. A book called: In Encyclopedia of Library and Information Science, 2nd Edition, 2011.

[17] http://www.ilc.cnr.it/EAGLES96/rep2/node39.html, accessed Oct 28, 2012.

[18] Kaplan A., An experimental study of ambiguity and context, Mechanical Translation, vol.2 no.2, 1955.

[19] Rada Mihalcea, Ted Pedersen, Advances in Word Sense Disambiguation, Tutorial at ACL Conference, 2005.

[20] Kula Kekeba Tune, Vasudeva Varma, Oromo-English Information Retrieval Experiments at CLEF 2007, Working Notes of CLEF Workshop, Spain, 2007.

[21] Navigli R., Word sense disambiguation: A survey, ACM Computing Surveys, Vol. 41, No. 2, Article 10, 2009.

[22] Litkowski, K. C., Computational lexicons and dictionaries, In Encyclopedia of Language and Linguistics (2nd ed.), K. R. Brown, Ed. Elsevier Publishers, Oxford, U.K., 753–761, 2005.

[23] Agirre, E., Stevenson, M., Knowledge sources for WSD, In Word Sense Disambiguation: Algorithms and Applications, E. Agirre and P. Edmonds, Eds. Springer, New York, NY, 217–251, 2006.

[24] Doina Tatar, Gabriela Serban, A New Algorithm for Word Sense Disambiguation, Studia Universitatis "Babes-Bolyai", Seria Informatica, Volume-XLVI, 2001.

[25] Kilgarriff, A. , Yallop, C, What's in a thesaurus? In Proceedings of the 2nd Conference on Language Resources and Evaluation , 1371–1379, 2000.

[26] Miller, G. A., Beckwith, R., Fellbaum, C. D., Gross, D., Miller, K., WordNet: An online lexical database. Int. J. Lexicograph. 3, 4, 235–244, 1990.

[27] Gruber, T. R., Toward principles for the design of ontologies used for knowledge sharing, In Proceedings of the International Workshop on Formal Ontology, 1993.

[28] Miller, G. A., Leacock, C., Tengi, R., Bunker, R. T., A semantic concordance, In Proceedings of the ARPA Workshop on Human Language Technology. 303–308, 1993.

[29] Andres M., Armando S., German R. and Manuel P., Combining Knowledge- and Corpus-based Word-Sense-Disambiguation Methods, Journal of Artificial Intelligence Research 23 (2005) 299-330, 2005.

[30] Agirre, E. Andmartinez, D., Learning class-to-class selectional preferences, In Proceedings of the 5th Conference on Computational Natural Language Learning . 15–22, 2001.

[31] Lesk, M., Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone, In Proceedings of the 5th SIGDOC . 24–26, 1986.

[32] Banerjee, S., Pedersen, T., Extended gloss overlaps as a measure of semantic relatedness, In Proceedings of the 18th International Joint Conference on Artificial Intelligence. 805–810, 2003.

[33] Hindle, D., Rooth,M., Structural ambiguity and lexical relations, Computat. Ling. 19, 1, 103–120, 1993.

[34] Fellbaum, C., WordNet. An Electronic Lexical Database, MIT Press, 1998.

[35] Ng, H., Exemplar-Base Word Sense Disambiguation: Some Recent Improvements, In Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, EMNLP, 1997.

[36] http://en.wikipedia.org/wiki/Thesaurus, accessed Feb 20, 2013.

[37] http://en.wikipedia.org/wiki/Machine-readable_dictionary, accessed Feb 22, 2013.

[38] http://en.wikipedia.org/wiki/Ontology_(information_science), accessed Feb 21, 2013.

[39] Debela Tesfaye, Designing a Stemmer for Afaan Oromo Text: Hybrid Approach, Master's thesis, Addis Ababa University, Department of Information Science, 2010.

[40] Esha Palta, Kanwal Rekhi, Word Sense Disambiguation, Master's Thesis, School of Information Technology Indian Institute of Technology, Powai, Mumbai, 2006-2007.

[41] Mcculloch, W., Pitts, W., A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 5, 115–133, 1943.

[42] Gale, W. A., Church, K., Yarowsky, D., A method for disambiguating word senses in a corpus. Comput. Human. 26, 415–439, 1992.

[43] Schutze, H., Automatic word sense discrimination. Computat. Ling. 24, 1, 97–124, 1998.

[44] Schutze, H., Dimensions of meaning, In Supercomputing '92: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing. IEEE Computer Society Press, Los Alamitos, CA. 787–796, 1992.

[45] Golub, G. H. , Van loan, C. F., Matrix Computations, The John Hopkins University Press, Baltimore, MD, 1989.

[46] Lin,D., Automatic retrieval and clustering of similar words, In Proceedings of the 17th International Conference on Computational linguistics. 768–774, 1998.

[47] Widdows, D., Dorow, B., A graph model for unsupervised lexical acquisition, In Proceedings of the 19th International Conference on Computational Linguistics.1–7, 2002.

[48] N.Ide, J.Veronis, Word Sense Disambiguation, In Proceedings of the 19th International Conference on Computational Linguistic, 1-42, 1998.

[49] Hwee Tou Ng, Hian Beng Lee, Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach, Proceedings of the 34th annual meeting on Association for Computational Linguistics, pp 40-47, 1996.

[50] A. R. Rezapour, S. M. Fakhrahmad, M. H. Sadreddini, Applying Weighted KNN to Word Sense Disambiguation, Proceedings of the World Congress on Engineering , London, U.K., 2011.

[51] Teshome, K., Word Sense disambiguation for Amharic text retrieval: a case study for legal documents. Master Thesis, Addis Ababa University, 1999.

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

[53] Solomon A., Unsupervised machine learning approach for Word Sense Disambiguation to amharic words, Master's Thesis, Addis Abeba University, 2011.

[54] Vildan Ozdemir, Word Sense Disambiguation for Turkish Lexical sample, Master's Thesis, Fatih University, 2009.

[55] Nyein Thwet, Thwet Aung, Khin Mar Soe, Ni Lar Thein, A Word Sense Disambiguation System Using Naïve Bayesian Algorithm for Myanmar Language, International Journal of Scientific & Engineering Research, ISSN 2229-5518, 2011.

[56] Rebecca Bruce, Janyce Wiebe, Word sense disambiguation using decomposable models, In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico. 1994.

[57] WWW.cse.unt.edu/~rada/downloads.html, accessed, march 15,2013.

[58] Omnigton "the online wncyclopedia of writing systems and language" accessed from http://www.omniglot.com/writing/oromo.htm, May 10, 2013.

[59] Oromo language. Http://en.wikipedia.org/wiki/oromo_language; last accessed on May 12, 2013.

[60] Kula Kekeba Tune, Vasudeva Varma, Prasad Pingali, Evaluation of Oromo-English Cross-language Information Retrieval, ijcai 2007 workshop on clia, hyderabad, india, 2007.

[61] Wakshum Mekonnen, Development of stemming algorithm for Oromo texts, Master's Thesis, 2000.

[62] Oromo language: encyclopedia, http://en.allexperts.com/e/o/or/oromo_language.htm, last accessed on May 15, 2013.

[63] C. Griefenew-mewis, A Grammatical sketch of written Oromo, druckerei franz hansen, bergisch gladbach, germany, 2001.

[64] Tesfaye Guta, Afaan Oromo search engine, Master's Thesis, Addis Ababa University, Departement of Computer Science, 2010.

[65] 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.

[66] Getahun A., The analysis of ambiguity in Amharic, Journal of Ethiopian Studies, Volume 34#2, 2001.

[67] Solomon Assemu, Unsupervised machine learning approach for word sense disambiguation to Amharic words, Master's Thesis, Addis Ababa university school of Information Science, 2011.

[68] A book called "critical thinking, fourth edition: an introduction to the basic skills" by William Hughes and Jonathan Lavery, 2004.

[69] S. Pongpinigpinyo, W. Rivepiboon, Distributional Semantics Approach to Thai Word Sense Disambiguation, In Proceedings of the International Journal of Computational Intelligence, 2006.

[70] Tony McEnery, Andrew Wilson, Corpus Linguistic, Edinburgh University, published by Edinburgh University Press, 2001.

[71] Getachew Mamo, Automatic Part Of Speech Tagging for Afaan Oromo Language, Master's Thesis, School of Graduate studies, Addis Ababa University, 2009.

[72] Agirre E., Martinez D., Exploring automatic word sense disambiguation with decision lists and the web, in Proceedings of the coling 2000 Workshop on Semantic Annotation and Intelligent Content, 2000.

[73] http://www.anglistik.uni-freiburg.de/seminar/abteilungen/sprachwissenschaft/ls_mair/corpus-linguistics, accessed April 20, 2013

Appendix A: List of Afaan Oromo stop words

akka	hanga	jechuun	ol	waan
akkam	henna	kan	oliif	waggaa
akkasumas	hoggaa	kanaaf	oliin	woo
akkum	hogguu	kanaafi	yammuu	
akkuma	hoo	kanaafuu	osoo	yemmuu
ammo	illee	kee	otoo	yeroo
an	immoo	keenya	otumallee	ykn
ani	innaa	keenyaa	otuu	yommii
booda	inni	keeti	otuullee	yommuu
boooddee	isaa	keetii	saniif	yoo
dura	isaan	koo	silaa	yookaan
eega	isee	kun	simmoo	yookiin
eegana	iseen	malee	sun	yookinimoo
eegasii	ishee	moo	ta`ullee	yoom
ennaa	isheen	nu	tahullee	garuu
erga	itumallee	nuti	tanaaf	Jechuu
fakkeenyaaf	ituu	nuyi	tanaafi	oggaa
fi	ituullee	odoo	tanaafuu	utuu
fkn	Jechaan	ofii	tawullee	

Appendix B: Evaluation result obtained for each data set

	Ambiguous word	**TP**	**TN**	**FP**	**FN**	**P**	**R**	**Acc**
Data Set 1	Sanyii	8	9	0	1	1	0.89	0.94
	Horii	13	11	2	0	0.87	1	0.92
	Karaa	18	20	1	3	0.93	0.86	0.90
	Sirna	7	7	2	2	0.78	0.78	0.78
	Qophii	10	5	5	0	0.67	1	0.75
Average						**0.85**	**0.91**	**0.86**

	Ambiguous word	**TP**	**TN**	**FP**	**FN**	**P**	**R**	**Acc**
Data Set 2	Sanyii	8	7	2	1	0.80	0.89	0.83
	Horii	11	9	4	2	0.73	0.85	0.77
	Karaa	19	19	2	2	0.90	0.90	0.90
	Sirna	7	6	3	2	0.7	0.78	0.72
	Qophii	10	3	7	0	0.59	1	0.65
Average						**0.74**	**0.88**	**0.77**

	Ambiguous word	**TP**	**TN**	**FP**	**FN**	**P**	**R**	**Acc**
Data Set 3	Sanyii	8	7	2	1	0.80	0.89	0.83
	Horii	11	6	7	2	0.61	0.85	0.65
	Karaa	17	18	3	4	0.85	0.81	0.83
	Sirna	9	7	2	0	0.81	1	0.89
	Qophii	9	5	5	1	0.64	0.90	0.70
Average						**0.74**	**0.89**	**0.78**

	Ambiguous word	**TP**	**TN**	**FP**	**FN**	**P**	**R**	**Acc**
Data Set 4	Sanyii	9	7	2	0	0.82	1	0.89
	Horii	11	10	3	2	0.76	0.85	0.81
	Karaa	15	19	2	6	0.88	0.71	0.81
	Sirna	8	6	3	1	0.73	0.89	0.78
	Qophii	10	6	4	0	0.71	1	0.80
Average						**0.78**	**0.89**	**0.82**

	Ambiguous word	**TP**	**TN**	**FP**	**FN**	**P**	**R**	**Acc**
Data Set 5	Sanyii	6	8	1	3	0.86	0.67	0.78
	Horii	12	12	1	1	0.92	0.92	0.92
	Karaa	17	16	5	4	0.77	0.81	0.79
	Sirna	8	5	4	1	0.67	0.89	0.72
	Qophii	10	5	5	0	0.67	1	0.75
Average						**0.78**	**0.86**	**0.79**

	Ambiguous word	**TP**	**TN**	**FP**	**FN**	**P**	**R**	**Acc**
Data Set 6	Sanyii	8	7	2	1	0.80	0.89	0.83
	Horii	11	9	4	2	0.73	0.85	0.77
	Karaa	17	20	1	4	0.94	0.81	0.88
	Sirna	9	3	6	0	0.6	1	0.67
	Qophii	10	5	5	0	0.67	1	0.75
Average						**0.74**	**0.91**	**0.78**

	Ambiguous word	**TP**	**TN**	**FP**	**FN**	**P**	**R**	**Acc**
Data Set 7	Sanyii	8	5	4	1	0.67	0.89	0.72
	Horii	12	9	4	1	0.75	0.92	0.81
	Karaa	12	14	7	9	0.63	0.57	0.62
	Sirna	9	6	3	0	0.75	1	0.83
	Qophii	10	5	5	0	0.67	1	0.75
Average						**0.69**	**0.88**	**0.75**

	Ambiguous word	**TP**	**TN**	**FP**	**FN**	**P**	**R**	**Acc**
Data Set 8	Sanyii	8	6	3	1	0.73	0.89	0.78
	Horii	12	7	6	1	0.67	0.92	0.73
	Karaa	19	14	7	2	0.73	0.90	0.79
	Sirna	9	4	5	0	0.64	1	0.72
	Qophii	9	7	3	1	0.75	0.90	0.80
Average						**0.70**	**0.92**	**0.76**

	Ambiguous word	**TP**	**TN**	**FP**	**FN**	**P**	**R**	**Acc**
Data Set 9	Sanyii	7	6	3	2	0.70	0.78	0.72
	Horii	9	12	1	4	0.90	0.69	0.81
	Karaa	13	17	4	8	0.76	0.62	0.71
	Sirna	9	6	3	0	0.75	1	0.83
	Qophii	10	6	4	0	0.71	1	0.80
Average						**0.76**	**0.82**	**0.77**

	Ambiguous word	**TP**	**TN**	**FP**	**FN**	**P**	**R**	**Acc**
Data Set 10	Sanyii	9	5	4	0	0.69	1	0.78
	Horii	9	11	2	4	0.82	0.69	0.77
	Karaa	17	18	3	4	0.85	0.81	0.83
	Sirna	8	6	3	1	0.73	0.89	0.78
	Qophii	10	7	3	0	0.77	1	0.85
Average						**0.77**	**0.88**	**0.80**

Appendix C: Sample list of Afaan Oromo sense examples used in the corpus

- Dhaabbatichi hanga kurmaana 2ffaa bara 2005tti hojiiwwan misooma bosonaa deggeran; hojii sanyii<Ija_mukaa_ykn_midhaani> qopheessuu fi biqiltuu qopheessuu kan raawwate yoo ta'u lafa hektaara 5389 irrattis hojiin biqiltuu kunuunsuu raawwatamuusaa obbo Didhaa Dirribaa ibsaniiru.
- Qamoolee badhaasa kanarratti hirmaatan keessaa qonnaan bultoota adda duree, waldalee hojii gamtaa, jiduugaleessaa qorannoo qonnaa fi qoratoota, waldalee garaagaraa sanyiiwwaan<Ija_mukaa_ykn_midhaani> adda addaa baayi'suu irraatti bobba'aan akka badhaafamanis dubbataniiru.
- Ejensichi qonnaan bulaa fi mootummaa fayyadamaa gochuuf hojii sanyii<Gosa> loonii fooyyessuuf oolan babal'isaa jira jedhan.
- Haaluma wal fakkaatuun Sanyiin<Gosa> kormaa bara 2004 hanga kurmaana lammaffaatti qonnaan bulaaf dhiyaate doozii 41,754 ture bara kana hanga ji'a jahaattii doozii 70,775 ga'eera; faayidaa isaa hubachuu irraa kan ka'es fedhii qonnaan bulaan sanyii loonsaa fooyyessuuf qabu daran dabalaa waan jiruuf giddu galoonni tajaajila kana kennaa jiran gahaadha nama hin jechisiisu.
- Teekinooloojii omishaa duraa lafa qopheessu, lafa qotuu, biyyee bulleessuu, sanyii<Ija_mukaa_ykn_midhaani> facaasuu, biqiltuu midhaanii babbaquu, teekinoolojiiwwan jal'isii fi bishaan harkisu ta'a.
- Hanqina Sanyii<Ija_mukaa_ykn_midhaani> filatamaa jiru furuuf hojii hojecha jiru.
- Xaa'oo fi sanyii<Ija_mukaa_ykn_midhaani> filatamaa yeroon qonnaan bulaa biraan gahuufis ta'e omisha isaa gabaatti geeffachuu akka danda'u daandiin bu'uura misoomaa hundaa ol murteessaadhas jedhaniiru.
- Haaluma kanaan guutuu Oromiyaa aanaalee sagantaa fooyya'iinsa sanyii<Gosa> looniitiin hammataman 75 keessatti loowwan kuma 45 ol tibbana bifa duulaan guraandhala 30 irraa eegalee diqaalomfamaa akka jiran Ejensichatti Abbaan Adeemsa Hojii Dhimmoota Kominikeeshinii Mootummaa obbo Mulaatuu Haayilee himaniiru.
- Kaayyoon bifa duulaatiin loowwan diqaalomsuu kunis, loowwan hedduu yeroo tokkoon sanyii kormaa qabsiisuun yeroo gabaabaa keessatti loon sanyii<Gosa> filatamoo horachuun oomishaa fi oomishtummaa misooma horii dabaluudhaan fedhii ummatni aannanii fi bu'aa horiirraa argatu akka guutamu taasisudha jedhaniiru.

- Sagantaa fooyya'iinsa sanyii<Gosa> looniitiin barana loowwan kuma 96 fi 800 diqaalomsuuf karoorfame keessaa hanga ammaatti loowwan kuma 50 ta'an diqaalomsamuusaanii obbo Mulaatuun eeranii, hojii kanarratti qonnaan bultoonni kuma 45 ol ni hiirmaatu jedhamee ni tilmaamamas jedhaniiru.
- Waldaan Aksiyoona Liiqii fi Qusannaa Oromiyaa imaammataa fi tarsiimoo mootummaan baase deeggaruun jiruu fi jireenya ummataa fooyyeessuuf waggaa 15 dura hundaa'uun tajaajila liqaa, qusannaa, inshuraansii xixiqqaa, daddabarsa horii<Maallaqaa_ykn_Qarshii> biyya keessaa fi tajaajila gorsaa maamiltootaaf kennaa jira.
- Magaalaa keenya caalmaatti guddina ishii saffisiisuuf horii<Maallaqaa_ykn_Qarshii> fi humna qabnu qindeessinee investimentii magaalichaa keessa yoo galle dha.
- Waldaalee kanaafis gurmii irraa jalqabee deeggarsi leenjii, liqii horii<Maallaqaa_ykn_Qarshii> fi iddooo hojii osoo walirraa hin-citin kennamaa tureera.
- Gama baajataatiin immoo, akka godina Shawaa Kaabaatti, baajanni gama mootummaatiin qabame horii<Maallaqaa_ykn_Qarshii> miliyoona 49 ta'a jedhame yaadama.
- Gama hirmaannaa uummata godinichaatiin horii<Maallaqaa_ykn_Qarshii> walitti qabamuuf karoofamee, birriin miliyoonni 14 fi kumni 84 fi 804 walitti qabameera.
- Birriin baankii ta'e horii<Maallaqaa_ykn_Qarshii> hirmaannaa ummataatiin walitti qabame dha.
- Hojiiwwaan daandii aanichaa hojjachuuf baajata hirmaannaa uummataatiin horii<Maallaqaa_ykn_Qarshii> kumni 800 walitti qabameera.
- Yuuniyeenichi, rakkoolee hawaasa naannoo hiikuudhaafis, kilinika fayyaa horii<beelladaa> fi keellaa fayyaa namaa tokko hojjechuun faayidaa irra oolcheera.
- Keellaa fayyaa horii<beelladaa> hojjechiise tajaajilaaf oolcheera.
- Marga gargaaramuun horii<beelladaa> furdisuun ni danda'ama.
- Hojiilee qabeenya uumamaa hojjechuun bakka turetti deebisuu fi sanyii horii<beelladaa> horsiisee bulaa fooyyessuu hojjetamaa jiru.
- Rakkoowwan bishaan dhugaatii namaa fi horii<beelladaa> furuudhaaf piroojektoota gurguddoo ta'an waliin ta'iinsa mootummaa fi mit-mootummaatiin hojjetamaa jiru.
- Waldoonni afur ta'nis hojii horii<beelladaa> furdisuu irratti bobba'aniiru.

- Har'a jireenyi keenya fooyya'uusaatiin badhaasaaf geenyeerra jedhanii, hojii misooma horii<beelladaa> fooyya'aa fi misooma qonnaa teekinooloojiin deeggarametti fayyadamuun jireenyi isaanii akka fooyya'es himaniiru.
- Haaluma kanaan bulchiinsa magaalaa sabbataatti ji'oottan jahan darban keessa hojiileen ibsaa, karaa<daandi> keessaa keessaa cirrachaa fi dhagaa koobiliitiin hojjechuu kiiloometira 53, diichiin lolaa kiiloometira 4, boononn bishaanii 6, riqichi guddana tokkoo fi kkf baasii qarshii miiliyoona 19.2n kan raawwataman ta'uu kanatiibaan bulchiisa magaalattii Obbo Yemaanee yiggazuu ibsaniiru.
- Tolaan meeshaa karaa<daandi> ittin hojjetaan ergifatee deebise.
- Bu'uuraaleen misoomaa kanneen hafan kan akka karaa<daandi>, ibsaa, bilbila, bishaan dhugaatii qulqulluu waliin gahuuf hojii hojjetamaa turee fi hojjetamaa jiruun ummanni naannoo keenyaa sadarkaa sadarkaan irraa fayyadamaa ta'aa jira.
- Godina Oromiyaa garaa garaa aanaalee sadii keessatti karaan<daandi> bonaa fi ganna tajaajilu kiloomeetirri 197 ol bajata mootummaan rammadee fi hirmaannaa ummataa qarshii miliyoona 87 oliin hojjetamaa akka jiru Waajjirri Abbaa Taayitaa Daandiiwwan aanaalee kanneenni beeksisan.
- Godina Wallaggaa Lixaa aanaa Sayyoo Nooleetti karaan<daandi> bonaa fi ganna tajaajilu kiloomeetirri 49 bajata mootummaan ramadee fi hirmaannaa ummataa qarshii miliyoona 22 oliin hojjettamaa akka jiru Abbaan Taayitaa Daandiiwwan aanichaa beeksise.
- Ogeessi Teekinishaanii karaa<daandi> aanichaa Obbo Masgabuu Abdiisaa akka jedhanitti karaan kun bara darbe keessa bajata mootummaa fi hirmaannaa ummataatiin akka eegalamu taasifameera.
- Karaan<daandi> kun yeroo tajaajila kennuu eegaluutti gandoota aanichaa 13 kan walqunnamsiisu ta'uus himaniiru.
- Haaluma walfakkaatuun, godina Harargee Bahaa aanaa Baabbilee fi Gursumitti karaan<daandi> kiloomeetirri 148 ol hirmaanaa ummataa fi bajata mootummaan ramade qarshii miliyoona 65 oliin hojjetamaa akka jiru Waajjirri Abbaa Taayitaa Daandiiwwan aanaalee kanneenii ibsan.
- Aanaa Baabileetti karaan<daandi> kiloomeetira 57 ta'u hojjettamaa jiru keessaa daandiin kiloomeetira 19 tahuu xumuramuun tajaajila kennaa akka jiru itti gaafatamtuun Abbaa Taayitaa Daandiwwan aanichaa Aadde Immabeet Boggaalaa dubbataniiru.
- Pirojektiin karaa<daandi> Asfaaltii Baalee kiiloometira 132 haguugu birr.mili 300 fi mil 80 oliin ijaarame tibbana sirna ho'aan eebbifame.

- Tajaajilawwan fayyaa, barnootaa, bishaan dhugaatii, ekisteenshinii fi misooma adda addaa ummata magaalaa fi baadiyyaan ga'uufis karaan<daandi> ga'ee ol'aanaa taphata.
- Gama biraatiin Koreen kun kan ilaale, aadaa gaarii ummanni rakkoowwan mudatan karaa<akkaata_kallatti> nagaatiin hiikkachuuf tumsi inni yeroo garaa garaa taasisaa jiru ammas cimee akka itti fufuu fi jajjabaachuu qabas jedheera.
- Galmi kun ammoo milkaa'uu kan danda'u hawaasni faayidaa barnootaa karaa<akkaata_kallatti> guutu ta'een hubachuun qooda fudhannaan inni dhimma mana barumsaarratti qabu yoo dabalee dha.
- Yeroo ammaa karaa<akkaata_kallatti> marii sadarkaa raayyaa dubartootaatti, sagantaalee paakeejii fayyaa 16n keessaa tokko kan ta'e karoora maatiirratti mar'achuun, daa'imman meeqaafi akkamiin akka godhachuu qabanirratti ni mari'atu.
- Bulchiinsi mootummaa fi Ummataas karaa<akkaata_kallatti> seeraatiin akka raawwatamu ciminaan kan qabsaa'ee fi biyya olaantumman seeraa itti mirkanaa'e uumuuf halkanii fi guyyaa tattaafachaa kan ture hogganaa bilchaataa ture.
- Namoonni dogongora isaanii yoo sirreeffatanii fi karaa<akkaata_kallatti> seeraafi nagaatiin yoo sochoo'an jedhee kan amanu hogganaa garaa bal'atu ture.
- Hogganoonni dhaabbiilee mormitootaa karaa<akkaata_kallatti> seera fi nagaatiin mormii isaanii akka tarkanfachiisan bilisummaa kan gonfachiisee fi ejjannoo dimokraatawaan paartiilee kanneen wajjin hojjachuuf fedhii agarsiise qabatamaan mirkaneessuuf hoggansa bilchaataa kenneera.
- Kanaan ala, karaa<akkaata_kallatti> kamiiniyyuu aangoo argachuunis ta'e aangoo mirkaneeffachuun akka hin danda'amne heerri mootummaa keenya ifatti kaa'eera.
- Qophii keenya kanaanis tarsiimoo misooma barnootaa karaa<akkaata_kallatti> fooyyee qabuun hojiirra oolchuun qabsoo hiyyummaa waliin godhamu ariifachiisaa warra jiran irratti xiyyeeffanneerra.
- Sagantaaleen kunneen karaa<akkaata_kallatti> guutuu ta'een hojiirra oolanii qulqullinni barnootaa mirkanaa'uu kan danda'u, sochii barsiisoonni, barattoonnii fi hoggansi barnootaa taasisuun qofa miti.

- Mohaammad mootummaan naannoo Oromiyaa sagantaa fooyya'iinsa sirna<seera_ykn_aadaa> haqaa hojiirra oolchuun olaantummaan seeraa fi bulchiinsi gaariin akka mirkanaa'uuf hojjechaa jira jedhaniiru.
- Daldalaan karaa haqa qabeessaan gabaa keessatti dorgomuun ofis fayyadee guddina dinagdee biyyatti keessatti qooda isaa akka bahuf mootummaan haala mijataa uumuuf imaammata gabaa bilisaa baasee hojiirra oolchuusaatiin sirna<seera_ykn_aadaa> daldalaarraa bu'aa guddaan argamuu danda'eera.
- Haata'u malee sirna<seera_ykn_aadaa> gabaa bilisaa kana karaa sirrii fi haala imaammatichi jedhuun hojiitti hiikuurratti rakkoo guddaatu mul'ata .
- Karoora waggoota shaniif (2003-2007) qophaa'e keessattis daa'imman fedhii addaa qaban qaama miidhamtoota ta'an bara barnoota darban keessatti carraa barnootaa dhabanii turan ilaalchii fi xiyyeeffannaan addaa itti kennamee sirna<seera_ykn_aadaa> barnoota hunda hammatootiin manneen barnootaa hunda keessatti hiriyootasaanii waliin akka baratan gochuuf karoorfameera.
- Humna raawwachiisummaa jechuun hojjetoota Inistiitiyuutichaa fi mamiltoota rakkoo gama ilaalchaa,dandeetii fi hordooffii degarsaa qaban lenjiiwwan garaa garaatiin cimsuudhaan raayyaa jijjiiramaa tokko ta'uudhaan sirna<seera_ykn_aadaa> diriirsuun kan raawwatamu ta'a.
- Filannoo, qacarraa fi ramaddii hojii akkasumas haala sirna<seera_ykn_aadaa> hojii hunda irratti loogiin Qaama Miidhamummaa bu'uurefachuun qaama Miidhamtoota irratti taasifamu dhorkaa ta'uun isaa keewwata 27(a) jalatti ifatti tumamee jira.
- Ummanni Oromoo ilma fuudhaaf gaheef intala heerumaaf geesse sirna<qophii_ykn_sagaanta> walitti fiduu mataasaa danda'e qaba.
- Adeemsa sirna<qophii_ykn_sagaanta> kanaa keessatti dursa mucaan fuudhaaf gahe mucayyoo fuuchuuf barbaadu yeroo bishaan buutu, yeroo qoraan cabsituuf erga adda addaa warra isheetii ergamtee deemtu ilaallachuun fedhiisaa warrasaa gurra buusa.
- sirna<qophii_ykn_sagaanta> cufiinsa shaampiyoonaa Ispoortii yeroo sadaffaaf akka godina addaa Oromiyaa Naannawa Finfinneetti adeemsifamaa ture irratti itti aanaan bulchaa godinichaa obbo Admaasuu Tashoomaa haasawa taasisaniin kaayyoon shaampiyoonaa ispoorti kanaas ispoortessitoota ciccimoo dandeettifi gahumsa qaban baay'inaan horachuun dorgommiilee adda addaarraatti godinicha bakka bu'uu danda'an filachuuf akka ta'e himaniiru.

- sirna<qophii_ykn_sagaanta> eebbaarratti haasaa kan taasisan pirezidanti Alamaayyoon tajaajila fayyaa gahaa ta'e saffisaaf qulqullina qabu ummataan ga'uudhaan dhukkuboota daddarboo ittisuufi waldhaanuudhaan du'aafi dhibama sadarkaa sadarkaan xiqqeessuun hawaasa fayya buleessa ijaaruun omishtummaa ummataa guddisuuf hojjetamaa jira.
- sirna<qophii_ykn_sagaanta> kenniinsa boondii godina Addaa Naannawaa Finfinnee magaalaa Sabbataatti raawwatamerratti argamuun haasaa kan taasisan hogganaan biirichaa Obbo Siiraj Kadir akka jedhanitti biyyi keenya guddina saffisaa hawaasni sadarkaan irraa fayyadamu galmeesisuutti argamti.
- Bara kanas sagantaa badhaasa gootota misoomaa marsaa 7ffaa haala ho'aa ta'een geggeessuuf qophii<Haala_mijeessu> barbaachisaan xumurameera.
- Tooftaa safarrii lafaa sadarkaa lammaffaatiinis naannolee garaa garaa keessatti bara kana lafa hektaara miliyoona 50 ta'u safaruun kenni waraqaa raga qabiyyee lafaa mirkaneessu akka raawwatu gochuuf qophiin<Haala_mijeessu> xumurameera jedhan.
- Keessumattuu murtii haqaafi qulqullina qabu kennisiisuu irratti dhiibbaa fidaa kan jiru ragaa sobaa irratti qorannoon kan gaggeeffame yoo ta'u, qulqullina qorannoo yakkaa fooyyessuuf giddu-galeessa qorannoo fooreensikii naannichatti hundeessuun qorannoo xumuramee humna namaa, meeshaafi baajanni barbaachisu adda bahee mootummaaf dhiyaachuuf qophii<Haala_mijeessu> irratti argama.
- Dargommii tapha shaampiyoonaa isopoortii manneen barnoota 2ffaa naannoo Oromiyaa Guraandhala 3 -17 bara 2005 magaalota Adaamaafi Asallaa geggeessuuf qophiin<Haala_mijeessu> barbaachisu xumurameera.
- Haaluma kanaan milkaa'ina dorgommii kanaaf qophii<Haala_mijeessu> guutuun taasifameera.
- Dorgommichi jalqabaa hagaa xumuraatti naga-qabeessa akka ta'u gochuuf qophii<Haala_mijeessu> gahaan yoo taasiifameyyuu, hundaa ol hirmaannaan hawaasaafi ispoortessitootaa ga'ee bakka hin bu'amne qaba.
- Gareen kubbaa millaa biyyaalessa Itoophiyaa qophii<saganta> tapha waancaa kubbaa millaa Afrikaa 29ffaaf taasisaa jiruun tapha wiixata darbe garee biyyaaleessa Tuuniziyaa waliin taasiseen 1fi 1n walqixa xumureera.
- Kutaa jalqabaa qophii<saganta> keenyaa maxxansa darbeen ilaalleerra.

- Tumsa ummanni dhugeeffannaadhaan taasisuun, seenaa kana caalu hojjechuu ni danda'a kan jedhu ammoo dhaamsa qophii<saganta> keenya ittii goolabnu ta'a.
- qophii<saganta> barreeffama ejjannoo mootummaa naannoo Oromiyaa balballoomsuu ilaalchisee dhimmootni wayitaawoon, imaammataaf qabxiiwwan iftoomina barbaadan irratti ejjennoo mootummaan naannichaa dhimmicha irratti qabu ifa taasisuun ummanni hubannoo akka argatuuf miidiyaafis ka'umsa ta'ee haala itti fufinsa
- Cumboon kan qophaa'us yeroo mara osoo hin ta'in, yeroo qophii<saganta> aadaa fi ayyaanonni garaa garaa kabajamanittifi keessummaan kabajaa kanneen akka Soddaa argamanitti nyaataf kan qophaa'udha.
- Waajjirichatti ogeessa qulqullina bunaa kan ta'an Obbo Immiruu Tesammaa akka dubbatanitti dhaabbii bara 2005/6f buufataalee biqiltuu mootummaa afuriifi kan dhuunfaa 720 irratti biqiltuuleen bunaa dhukkuba dandamachuu dandaa'an miliyoonni 3,740,050 kan qopheeffaman yemmuu ta'u, biqiltuulee kanneen keessaas miliyoonni 3,181,800 buufata dhuunfaarratti akka ta'eefi qophii<saganta> kanarrattis qonnaan bultoonni 720 hirmaachuusaanii beeksisaniiru.

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: _____

Place and date of submission: Addis Ababa, November 2013.