ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

Morphological Analyzer for Afaan Oromoo Using Machine Learning

Moyka Degefa Mosa

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

Addis Ababa, Ethiopia

March, 2020

Addis Ababa University

College of Natural Sciences

Moyka Degefa Mosa

Advisor: Dida Midekso (PhD)

This is to certify that the thesis prepared by Moyka Degefa, titled: *Morphological Analyzer for Afaan Oromoo Using Machine Learning* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the examining committee:

Name	Signature	Date
Advisor: Dida Midekso (PhD)		
Examiner: Solomon Atnafu (PhD)		
Examiner: Solomon Gizaw (PhD)		

Abstract

This thesis describes the development of morphological analysis system for Afaan Oromoo. Morphological analysis is an analysis of words that aimed at segmenting words into their component morphemes and the assignment of grammatical information to grammatical categories. Many researches have been conducted in morphological analysis extensively for different languages, while this work is among a few works in Afaan Oromoo natural language processing applications. If there is a robust morphological analysis, there are many natural language processing applications that can be benefited from it. A new Afaan Oromoo morphological analysis is proposed based on memory-based learning. Memory-based learning techniques keep all training data available for classification and extrapolation without making any abstraction unlike eager learners. Because of its lazy property, memory-based learning achieved a higher accuracy than eager methods for many language processing tasks. The proposed morphological analyzer has two main components: training phase and analysis phase. The training phase comprises necessary components that are used in the process of training the learning component of memory-based learning. The analysis phase maps the input into output. A morphological database which consists of grammatical information of Afaan Oromoo nouns, verbs and adjectives has been developed. It contains 2270 annotated words. We performed an experiment in four scenarios to evaluate the system being developed based on memory-based algorithms: IB1 and IGTREE algorithms. We obtained the maximum generalization accuracy of 98.86% from IB1with interleaving and 94.36% from IGTREE with feature selection and interleaving. The result from our experiment shows that selecting the combination of features with highest accuracy plays a vital role on both default and optimal parameter settings. Examining the influence of feature justified that we used the best combination of features. Generally, the algorithms and techniques used in this research work obtained a good performance.

Key words: Morphological Analysis, Memory Based Learning, Afaan Oromoo

Dedication

I dedicate this work to those who sacrificed their life to develop qubee for qubee generation (including me) especially Dr. Haile Fida and Sheikh Bakri Saphalo.

Acknowledgments

First of all, I would like thank my Almighty God for his countless love and care to me. Almighty God, from the beginning to the end of this research work you were with me. I praise his name. This thesis has benefited from the support of many people. I would like to express my deepest gratitude to my advisor Dr. Dida Midekso for his continuous follow up from the beginning to this end. His constructive comments, suggestions, guidance and enlightening ideas were very essential. I would also like to thank Dr. Addunyaa Barkeessaa for his linguistic expert advice on many aspects of my research and providing me linguistic materials to develop my ideas. Dr. you really helped me so much to understand the linguistic properties of Afaan Oromoo.

I am grateful to Wakweya Olani for providing me supportive materials and for sharing his knowledge on many small but important details through email. I would also like to thank Dr.Gemechis, the Oromo research center deputy director and Mr.Bekele, the staff member of Oromo research center and PhD candidate at Addis Ababa University for their cooperation in allowing me to use the materials in the library of Oromo research center at any time.

My classmates Bahar Hussien and Gutu Merga and my friends Dame Kedir, Eshetu Deresu, Lami Garoma, Jabessa Olani and Dandi Kena encouraged me a lot to finish this research work. I really thank you guys for your encouragement, kindness and friendship.

My brother Tiksa Degefa supported and encouraged me. May God bless you and your family. Finally, I would like to thank my father Degefa Mosa and my mother Terfatu Adeba. You both didn’t get the opportunity to learn any formal education, but you gave me an opportunity. I have nothing to say, but baay’ee galatoomaa Dad and Mom.

Table of Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

10-FCV	10-Fold Cross Validation
IB1	Instance Base
IGTREE	Information Gain Tree
ILP	Inductive Logic Programming
K-NN	Nearest Neighbor
LOOCV	Leave One Out Cross Validation
LSV	Letter Successor Variety
MBL	Memory Based Learning
MBMA	Memory Based Morphological Analysis
MDL	Minimum Description Length
NLP	Natural Language Processing
OROLEX	Oromoo Lexica
SVM	Support Vector Machine
TiMBL	Tilburg Memory Based Learner

Chapter One: Introduction

1.1 Background

Language is one of the essential aspects of human behavior and crucial component of human lives [1]. It is a form of communication between humans. A language (spoken and written) is essential to all aspects of our interaction we make daily. In written form, it serves as a long-term record of knowledge from one generation to the next while in spoken form it serves as our primary means of coordinating our day-to-day interaction with others. Different academic disciplines such as linguists, psycholinguists, philosophers and computational linguists are involved in the study of a language [1]. Jurafsky and Martin [2] argue that engaging in complex language behavior requires various kinds of knowledge of language or levels of linguistic like:

- Phonetics and phonology: knowledge about linguistic sounds.
- Morphology: knowledge of the meaningful components of words.
- Syntax: knowledge of the structural relationships between words.
- Semantics: knowledge of meaning.
- Pragmatics: knowledge of the relationship of meaning to the goals and intentions of the speaker.
- Discourse: knowledge about linguistic units larger than a single utterance.

Natural language processing (NLP) plays a major role in solving the problems of natural (human) language facing human being and making the life of human being easy by doing complicated works. Natural language processing (NLP) is a field of computer science and computational linguistics concerned with the interactions between computers and human (natural) languages. The goal of NLP is to get computers to perform useful tasks involving human language, tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech [2]. Natural language processing deals with analyzing, generating and understanding the languages that human generally use. So, language technology, or language engineering, uses the formalisms and theories developed within NLP in applications ranging from spelling error correction to machine translation and automatic extraction of knowledge from text [3].

Morphology is the field of linguistics that studies the internal structure of the word and how words can be formed [4]. Morphological analysis is the process of segmenting words into morphemes and analyzing the word formation [5]. Morphological analysis is concerned with retrieving the syntactic and morphological properties or the meaning of a morphologically complex word [6]. Morphological analyzer is a tool which identifies and analyzes the internal structure of a given word and also gives the morphological and grammatical information associated with the given word. Morphological analysis plays an important role in the development of all-natural language processing applications. The development of natural language processing applications such as search engines, speech synthesizer, speech recognizer, lemmatization, noun compounding, spell and grammar checker, machine translation, etc. highly rely on a good morphological analysis [5, 7].

There are three classifications of morphological structures and they are isolating languages (eg. Chinese), agglutinative languages (eg. Turkish) and inflectional languages (eg. Latin) [8]. The isolating languages are analytical language which do not allow inflectional and derivational affixation at all [9]. The agglutinative languages like Afaan Oromoo is rich in inflection, which requires complex procedures to extract its inflections and grammatical information. The inflectional languages are languages that use affixation, often fuse grammatical categories into one affix, this fusion may be accompanied by phonological alternations [9].

In order to solve the problem of morphological analysis, there are two broad categories of approaches in computational morphology: rule-based and corpus-based (machine learning). The rule-based approach is used to incorporate domain knowledge into the linguistic knowledge which provides highly accurate results [10]. Machine learning is a branch of artificial intelligence concerned with the design of algorithms that learn from examples. Machine learning algorithms can be classified into supervised, unsupervised and semi-supervised. The supervised learning approach learns by example and includes inductive logic programming (ILP), support vector machine (SVM), hidden Markov model (HMM) and memory-based learning (MBL), while the unsupervised learning approach is learning by patterns [4, 7, 8]. The semi-supervised learning approach is the combination of both supervised and unsupervised learning approach.

The morphological properties, patterns, and grammatical structure of Afaan Oromoo are different from other languages, in that it needs its own morphological analyzer with different approaches in order to pick the most efficient one.

Thus, the development of morphological analyzer for Afaan Oromoo is crucial for easing the development of higher-level linguistic analysis of Afaan Oromoo natural language processing applications and other tasks.

1.2 Motivation

Afaan Oromoo is one of the Cushitic family languages and most widely spoken languages in Ethiopia and neighboring countries like Kenya and Somalia [11]. With the fact that the language is used in offices, schools and media, there is an electronic data available that encourages studies related to NLP tasks associated with the language.

Morphological analyzer is an important component in Afaan Oromoo natural language processing, which plays a positive role for the development of NLP tasks. It has a significant importance in the development of the higher linguistic levels and different natural language processing applications such as information retrieval, machine translation, search engine, anaphora resolution and etc. So, developing a novel morphological analyzer for Afaan Oromoo contributes to the development of Afaan Oromoo natural language processing applications. This is what motivates us to conduct the research work.

1.3 Statement of the Problem

Morphological analyzer for different languages has been developed by different researchers using different approaches (techniques), such as Malayalam language [5, 8], Kokborok language [6], Amharic language [7], Tamil language [4], Arabic language [12] and, etc. Even if morphological analyzer is developed for different languages around the world using different approaches, it cannot be used for Afaan Oromoo language because each language has its own pattern, properties and grammatical structures. For instance, the number of affixes and the way affixes attached to a root is different from language to language. Morphology is language specific.

In order to develop a higher-level linguistic analysis, at least there should be a novel work of morphological analyzer. Morphological analyzer has important significance in the Internet search engines and is an important task in many of the language engineering applications such

as machine translation, question-answering systems, information retrieval and automatic summarization and others. These facts show that morphological analyzer is the basic and relevant component for the development of most of NLP related applications. In order to select the best performing morphological analyzer for Afaan Oromoo language, developing morphological analyzer using all available approach is important.

There were attempts made to developing morphological analysis and synthesis for Afaan Oromoo language by Gasser [13] and Abebe Abeshu [14]. Both works have been attempted by using the traditional and hand-engineered approach, rule-based approach. The work made to developing morphological analysis for Afaan Oromoo was limited to verb word class only. However, rule-based approach is not sufficiently enough to tackle the problem of morphological analyzer because of some drawbacks. The first is, creating rules by hand is an arduous and time-consuming task especially for a complex language like Afaan Oromoo and difficult to debug, modify, or adapt to other similar languages [15]. The second is, if one rule fails, it affects the entire rule that follows [4].

For these strong drawbacks of rule-based approach, there is a much considerable interest in robust machine learning approaches to morphology learning, which extracts linguistic knowledge automatically from an annotated or un-annotated corpus [15]. Machine learning approaches don't require any hand coded morphological rules. It needs only corpora with linguistic information. The rules are learned automatically from data, uses learning and classification algorithms to learn models and make predictions. These morphological or linguistic rules are automatically extracted from the annotated corpora [4, 7, 8]. The current research focus is on memory-based learning approach. It is a lazy learner which keeps all training data available for classification and extrapolation without making any abstraction unlike eager learners. Memory-based learning has been shown to achieve higher accuracy than eager techniques for many language processing tasks.

As far as the knowledge of the researcher is concerned, there was no prior study conducted on developing an automatic morphological analyzer for Afaan Oromoo using memory-based learning, machine learning approach.

1.4 Objectives

General Objective

The general objective of this research work is to design and develop an automatic morphological analyzer for Afaan Oromoo using machine learning approach.

Specific Objectives

The specific objectives of this research work are as follows:

- To review literatures to gain a better understanding in the area.
- To study morphological property of Afaan Oromoo language, especially verbs, nouns and adjectives for better understanding.
- To collect inflected and derivation of Afaan Oromoo verb, noun and adjective words.
- Designing morphological model for Afaan Oromoo.
- To organize training and test data from the corpus data.
- To develop a prototype of the developed analyzer.
- Finally, to evaluate the developed prototype.

1.5 Methods

In order to accomplish the primary objective of the research, the following methods and procedures will be employed.

i. Literature Review

The literature review deals with reviewing the literatures like books, journals, proceeding, thesis work and others on the area related to the present research work, in our case, morphological analysis. A thorough literature review will be done on morphological analysis methods in general and morphological analyzer using machine learning approach in particular with regard to techniques used in each approach. Finally, the linguistic behavior and the formation of Afaan Oromoo word classes, such as noun, verb and adjective will be intensively reviewed.

ii. Data Collection

Afaan Oromoo does not have publicly available annotated corpus text for any NLP task including morphological database so far. As a result, the data to be used for the corpus will be collected from various sources of Afaan Oromoo documents such as websites, dictionaries,

newspapers and textbooks. Specifically, the inflected words of three major word classes like nouns, verbs and adjectives will be collected manually from the sources listed. The collected words need further editing manually in order to be used wisely. In addition, the words will be annotated manually by following the language procedure. Hence, the training and the test data will be extracted from the manually annotated dataset to make suitable for memory-based learning algorithms we use.

iii. Tools

For the development of afaan Oromoo morphological analyzer, TiMBL will be selected. TiMBL is an open source software package implementing memory-based learning algorithms and developed using C and C++ programming languages. Java programming language will be also used to extract instances from the dataset annotated manually.

iv. Prototype Development

In order to evaluate the performance of the model, a prototype will be developed for the Afaan Oromoo morphological analyzer that can return morphemes.

v. Evaluation

Experiments will be conducted, after formal system will be developed to test and evaluate the functionality of the system with different parameters. The performance of the system will be evaluated in terms of evaluation metrics such as, 10-fold cross validation, and precision, recall and F-score which can be computed from the confusion matrix.

1.6 Scope and Limitations

The scope of this study is limited to developing an automatic morphological analyzer for Afaan Oromoo using memory-based learning particularly for noun, verb and adjective word classes.

1.7 Application of Results

The development of morphological analyzer for Afaan Oromoo will help the development of other NLP applications for Afaan Oromoo as an input. As an application of results, the proposed morphological analyzer for Afaan Oromoo can be applicable for:

- The development of the next higher-level NLP applications such as, search engines, speech synthesizer, speech recognizer, lemmatization, noun decompounding, spell and grammar checker and machine translation.
- The development of full-fledged morphological analyzer for Afaan Oromoo.
- Teaching and Learning Afaan Oromoo.
- Building a morphological dictionary(lexeme) for Afaan Oromoo.

1.8 Organization of the Thesis

The remaining part of the thesis is organized as follows. Chapter 2 deals with the literature review. This chapter will try to address the linguistic structure of Afaan Oromoo, state of the art of morphological analyzer and computational morphology. Chapter 3 presents the related work that is highly relevant and more related to our study. Chapter 4 presents the general architecture of the system with its basic components and the discussion of the components and their interaction in the system. Chapter 5 deals with the experimentation of the system. Finally, Chapter 6 presents conclusion of our work and shows the feature work that needs to be included to the study in order to enhance the system.

Chapter Two: Literature Review

2.1 Introduction

This chapter basically explores the fundamental background of natural language processing particularly morphological analysis, the approaches used to develop morphological analysis and the fundamental background of Afaan Oromoo language. The second section, Section 2.2, explains natural language processing and its applications in computer science and other discipline. Section 2.3 describes morphology. Section 2.4 describes the smallest unit of a word, morpheme. Section 2.5 describes the variant pronunciations of a morpheme. The general concept of morphological analysis is discussed in Section 2.6. In Section 2.7, the different approaches to solve linguistic morphological problems ranging from hand-crafted to machine learning are discussed briefly. Finally, Section 2.7 describes the background of Afaan Oromoo and its word formation concepts.

2.2 Overview of Natural Language Processing

The arena of natural language processing has a deep and diverse concept. The basic foundations of natural language processing lie in a number of disciplines, such as, computer science, information science, linguistics, mathematics, electrical and electronic engineering, artificial intelligence, robotics, and psychology [16]. Natural language processing is concerned with the knowledge representation and problem-solving algorithms involved in learning, producing, and understanding language. The formalisms and theories developed within NLP in applications ranging from spelling error correction to machine translation and automatic extraction of knowledge from text are used in language technology, or language engineering [3]. 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. It is a subfield of artificial intelligence and linguistic, devoted to make computers understand the statements or words written in human (natural) languages. It is also a collection of techniques used to extract grammatical structure and meaning from input in order to perform a useful task as a result, natural language generation builds output based on the rules of the target language and the task at hand [17]. It involves the development of computer programs which can analyze natural language and act appropriately on the information contained in the text or utterance. It is concerned with the

study of mathematical and computational models of the structure and function of language, its use, and its acquisition and the design, development, and implementation of a wide range of systems [18].

Natural language processing is a field of computer science and computational linguistics with human-machine interaction having two major parts: natural language understanding systems to convert natural language to information and natural language generation systems to convert information from computer to natural language [19]. The aim of NLP is studying problems in the automatic generation and understanding of natural languages. It is successfully applied to a number of fields of study, such as machine translation, natural language text processing and summarization, user interfaces, multilingual and cross-language information retrieval, speech recognition, artificial intelligence, and expert systems [16].

2.3 Morphology (Xinjecha)

The term morphology is the study of how words can be broken down into meaningful pieces, which has been taken over from biology concepts [20]. Its first recorded use is in writings by the German poet and writer Goethe in 1796 and it was first introduced in 1859 by the German linguist August Schleicher for linguistic purposes, to refer to the study of the form of words [21]. In present-day linguistics, the term 'morphology' refers to the study of the internal structure of words, and of the systematic form–meaning correspondences between words [21]. Morphology refers to the study of how various sub-word units, called morphemes combine together to form new words through a sequence of rule applications [22]. The study of morphology deals with the construction of words from more basic components corresponding roughly to meaning units. Morphology is the branch of computational linguistics that studies the internal structure of words, their forms in different uses and the way words are built up from smaller meaning-bearing units called morpheme [2, 23, 24]. It is the study, identification, analysis and description of the minimal meaning bearing units that constitute a word [25]. Basically, there are two ways through which words can be formed, such as inflectional morphology and derivational morphology.

Inflectional morphology is the combination of a word stem with a grammatical morpheme, usually resulting in a word of the same class as the original stem, and usually filling some syntactic function like agreement [2]. It is the process of adding inflectional morphemes to a word and the inflectional morpheme adds some type of grammatical information, i.e., case,

number, person, gender, mood, mode, tense, aspect, etc. [23]. It also comprises the processes by which numerous inflectional forms are formed from a lexical stem [24].

Derivational morphology is the combination of a word stem with a grammatical morpheme, usually resulting in a word of a different grammatical category like turning a verb to an adjective or the same grammatical category, often with a meaning hard to predict exactly [2]. It involves the processes by which new lexemes are built from present ones mainly through the addition of affixes [24].

John A. Goldsmith [26] claims that morphologies are inspired by four considerations:

1. The discovery of regularities and redundancies in the lexicon of a language (such as the pattern in *deeme:deemte:deemani :: yaase:yaaste:yaasani*).
2. The need to make explicit the relationship between grammatical features (such as nominal number or verbal tense) and the affixes whose function it is to express these features.
3. The need to predict the occurrences of words not found in a training corpus.
4. The usefulness of breaking words into parts in order to achieve better models for statistical translation, information retrieval, and other tasks that are sensitive to the meaning of a text.

Morphology conveys information but exactly how much information and what kind of information is conveyed morphologically differs greatly from language to language [27]. Morphology is not equally prominent in all spoken languages. What one language expresses morphologically may be expressed by a separate word or left implicit in another language. For instance, Afaan Oromoo plural nouns are expressed by means of morphology (Example: *nama/namoota 'man/men', durba/durboota 'girl/girls'*, and so on), but Yoruba language utilizes a separate word for expressing the same meaning.

Generally, the field of morphology in practice, is the study of four relatively autonomous aspects of natural language namely, the identification of the lexicon of a language, morphophonology, morphosyntax, and morphological decomposition, or the study of word-internal structure [26].

2.4 Morpheme (Dhamjecha)

Morphemes, sub-word units, are the smallest building blocks and minimal meaning bearing unit that make up words in a natural language [20]. For example, the word *nama 'man'* is

made up of '*nam-', and '-a'.* Both sub units *'nam-'* and *'-a'* are called morphemes. Morpheme expresses concepts or relationships, which could be meaningful full words or could be sequence of characters which are not meaningful until joined with another morpheme or word. It is usually identified as a string of phonemes carrying meaning on its own; a special class of morphemes, affixes, does not carry meaning on its own, but instead affixes have the ability to add or change some aspect of meaning when attached to a morpheme or string of morphemes [3].

Addunyaa Barkessaa [28] argues that, based on their forms, Afaan Oromoo morphemes can be classified into two major parts, such as free morpheme (*dhamjecha walaba*) and bound morpheme (*dhamjecha hirkataa*). Free morphemes (*Dhamjecha walaba*) are those that can stand on their own as individual words, like *muka, farda, nama*. Bound morphemes (*dhamjecha hirkataa*) are those that must be attached to some kind of host morphemes to be realized as individual word. As its name suggests, this morpheme, bound morpheme cannot stand by itself independently without the supplement of other types of morphemes. This type of morpheme can be further divided into two, such as, bound root (*hundee hirkataa*) and affix (*fufii*).

Bound root is a part of a word that can be left after affixes are eliminated from it. For example, *bare, barte, barsiise, barumsa*. The root of these words is 'bar-'. So, in Afaan Oromoo roots are classified under bound morphemes. Affixes are bound morphemes that cannot standalone, but when attached to a body of words, they may change grammatical information or classes of a word. Based on their function, affixes can be classified into derivational affixes (*fufiilee yaasaa*) and inflectional affixes (*fufiilee hortee*). The former one, derivational affixes are the type of affixes attached to the body of words and which may change the categories and grammatical information of a word. Example, *-ummaa, -maata, -siis-, -sis-* and others. The later one, the inflectional affixes are the type of bound morphemes which indicate grammatical information like gender, person, case, aspect, number and other grammatical information.
Furthermore, affixes can be classified into prefix (*fufiii duree*), curcumfix (*fufii naannee*) and suffix (*fufii duubee*) based on their occurrences in a word. Prefix (*fufiii duree*) primarily focuses on changing the meaning of words rather than changing the class of words. For example, *hindeemne, haadeemu, nideema*. Those underlined morphemes are prefixes. In Afaan Oromoo, this type of affix is very small when compared to suffix. Circumfixes (*fufii*

naannee) are a special of a more general phenomenon whereby two or more affixes act together to provide a meaning which neither can have in isolation. Example, *hin-dhuf-n-e, ni-deem-a.* This type of affix is known as discontinuous affix. Suffix (*fufii duubee*) is affixes that can be attached to root words. They are large in number when compared to other types of affixes. Example, *qal<u>ama</u>, gal<u>e</u>, mur<u>siise</u>, qab<u>atame</u>*. The underline shows suffixes.

2.5 Allomorphs (Firjecha)

Allomorphs are the physical realization and variant pronunciations of a morpheme [28]. There are different morphs of allomorphic variation occurring in complementary distribution. The morphemes used to express plurality in Afaan Oromoo has basically two major allomorphs. They are *"-oota"* which appears after a word when penultimate syllable contains a short vowel sound and "*-ota"* which appears after a word when penultimate syllable contains a long vowel sound. For example, the Table 2.1 shows the allomorphic variation of Afaan Oromoo noun plurality.

Table 2.1: Allomorphic variation of Afaan Oromoo nouns plurality

Base forms	**-oota and -ota allomorphs**	**Gloss**
ganda	gand-oota	'kebeles'
durba	durb-oota	'girls'
leenca	leenc-ota	'lions'
diina	diin-ota	'enemies'

2.6 Morphological Analysis

At its heart, morphological analysis deals with the process of separating existing words into their elements, called morphemes, and describing how words are built by combining these morphemes. The development of a morphological analysis for any language requires an in-depth study of the nature of the respective language at the lexicon level. Since the use of the correct word form is the basis for grammar of any language, the morphological analysis is of great importance to all the steps in natural language processing. Incorporating a robust and novel morphological analysis in speech and natural language processing applications has a crucial role in improving and giving an encouragement result.

A morphological analysis for text corpora is a prerequisite for many text analytics applications, which has attracted many researchers from different disciplines such as

linguistics (computational and corpus linguistics), artificial intelligence, and natural language processing, to morpho syntactically analyze text of different languages [25]. Morphological analysis is the segmentation of words into their component morphemes and (usually) the assignment of grammatical information to grammatical categories and the assignment of the lexical information to a particular lexeme or lemma. Morphological analysis consists of the identification of parts of the words, or more technically, constituents of the words.

The task of performing a full morphological analysis of a word form is usually taken as a segmentation of the word into morphemes, combined with an analysis of the interaction of those morphemes that determine the syntactic class of the word form as a whole. The complexity of morphology varies widely among the world's languages, but is regarded as non-trivial even in the relatively simple cases, such as English and Chinese Mandarin. Morphology of a language contains ambiguities in its morphological composition that can be quite complex. There are different classes of linguistic knowledge that are usually assumed to play a crucial role in this disambiguation process such as knowledge of the morphemes of a language, the morphotactic-constraints on how morphemes are allowed to be combined, and the orthography, the model of how spelling changes in the resulting word form that can occur due to morpheme attachment [3].

The morphological analysis can discover different morphological characteristics of each word in the input text, and then extract the syntactic characteristics of each individual word based on built-in grammatical forms and the context.

Goldsmith [29] argues that the work in automatic morphological analysis can be usefully divided into four major approaches. The first approach proposes to identify morpheme boundaries first, and thus, indirectly to identify morphemes, on the basis of the degree of predictability of the n+1st letter given the first n letters (or the mirror-image measure). This approach was proposed by Harris [30], and Hafer and Weiss [31]. The second approach seeks to identify bigrams (and trigrams) that have a high likelihood of being morpheme internal. The third approach focuses on the discovery of patterns of rules of phonological relationships between pairs of related words. The fourth approach is top-down, and seeks an analysis that is globally most concise.

2.7 Approaches to Morphological Analysis

This section looks at the computational approaches to the study of morphology. Computational morphology can be viewed as having three separate subtasks such as, segmentation, clustering related words, and labelling. There are a number of approaches which are used for each of these tasks, ranging from rule-based approaches to various unsupervised, semi supervised and fully-supervised techniques which fall under machine learning or corpus-based approach which would generally deal with one or two of the subtasks [32]. In this section, we present the description of each approach employed for morphological analysis problem.

2.7.1 Rule Based Approach

The typical traditional approach to morphological analysis builds a description of the general rules governing mapping the segmentation of words to morphemes, describe additional sub regularities, and list the remaining exceptions to the rules and sub regularities. Rule based approach is based on a set of rules and dictionary for root and morphemes. It presupposes three components: a morpheme lexicon, a set of spelling rules and morphological rules to discover possible analyses of morphologically complex words, and prioritizing heuristics to choose the most probable analysis from sets of possible analyses [33]. Generally speaking, the acquisition of this knowledge is labor-intensive and expensive [3]. In rule-based approach, if a complex word is given as an input to the morphological analyzer and its corresponding root word doesn't exist in the dictionary then the rule-based system fails. It can offer extremely high accuracy for predicting inflections, but laborious to construct and does not exist with full lexical coverage in all languages [34].

2.7.2 Machine Learning

Machine learning has been around at least for more than five decades solving different categories of real-world problems. The ever increasing of huge amount of digitalized data and the problem of hand engineered led to the emergency and development of machine learning approach. Machine learning is the fast-growing areas of computer science and also considered as the subfield of artificial intelligence with far-reaching applications which is used to instruct computers to use the data or past experience to solve a given problem [35, 36]. The fundamental concept behind machine learning is that a computer learns to perform a task by

studying a training set of examples and the computer then performs the same task with the data it has not encountered previously [37]. Machine learning has got its inspiration from different academic disciplines such as computer science, statistics, biology, and psychology [35]. It permits system with the ability to learn automatically and get better with experience without being explicitly programmed.

Nowadays, the impact that machine learning has made so far became the major success factor in the ongoing digital transformation across the giant industries and AI research institute. Machine learning methods are applied to a wide variety of domains namely: robotics, security heuristics, image analysis, virtual personal assistants, data mining, computer games, pattern recognition, natural language processing, traffic prediction, online transportation network, product recommendation, share market prediction, medical diagnosis, agriculture advisory, search engine result refining, online customer support and many more [36, 38]. Recently, natural language processing became the most significant application area of machine learning among the applications of machine learning. Hence, machine learning has been shown to be quite adequate and directly applied to most natural language processing applications namely: morphological analysis and generation, named entity recognition, document classification and machine translation. It tackles the problem of those natural language processing applications in different approaches. From the natural language processing applications problem, the problem of learning morphology could be tackled by supervised machine learning [34, 39, 40], unsupervised machine learning [29, 30, 41] and semi supervised machine learning [42] depending on the task they are expected to perform, the type of data they deal with and the supervision they need. All approaches of machine learning to morphology learning is discussed as supervised morphology learning, unsupervised morphology learning and semi-supervised morphology learning.

i. Supervised Morphology Learning

Symbolic supervised machine learning methods have recently taken a front stage in language technology specially in morphology learning. Supervised machine learning is the construction of algorithms that are able to produce general patterns and hypotheses by using externally supplied instances to predict the fate of future instances. In supervised machine learning, the learning problems can be grouped into regression problems and classification problems [43].

Supervised machine learning sees most linguistic problems as context-sensitive mappings from one representation to another (e.g., from text to speech; from a sequence of spelling words to a parse tree; from a parse tree to logical form, from source language to target language, etc.) [44]. Accordingly, the problem of morphology learning can be formulated as a classification problem. Supervised morphology learning methods need a set of annotated datasets (e.g., annotated words) in order to extract regularities from them. Supervised morphology learning, a part of inductive machine learning is the process of learning a set of rules from instances (examples in a training set) extracted from a collection of annotated words, or more generally speaking, creating a classifier that can be used to generalize from new instances [45]. Supervised machine learning classification aims at categorizing data from prior information. Therefore, it is usually subdivided into classification problems, where given a set of instances and their class label, the task of the learning algorithm is to correctly predict the class of new unlabeled instances.

There are various successful algorithms of supervised machine learning used to tackle most linguistic classification problems particularly morphology learning problem. They are memory-based learning, inductive logic programming and support vector machines.

A. Memory Based Learning

Memory based learning (MBL) is a supervised inductive learning algorithm used for learning classification tasks based on the k-nearest neighbor classifier [46, 47]. It is inherited from the classical k-nearest neighbor (K-NN) approach to classification. Before diving into the concept of memory-based learning, let's discuss about the classical k-nearest neighbor.

K-NN is named as one of the most popular and extensively used classification algorithm [48]. The main purpose of this algorithm is to classify new objects (instances) based on the attributes and training data when there is little or no prior knowledge about the distribution of the data [35, 48]. It is the simplest and easiest classification algorithm which makes the classification task by getting votes of the k-nearest neighbor. The k- nearest neighbor (K-NN) classifier, which falls in the category of a lazy learner stores the given training instances and does nothing and waits for a test instance [49, 50]. In the K-NN classification algorithm, all training instances are stored in a fixed-length vector of n feature-value pairs. When an instance from the test instances is given to the classifier, it searches the k training instances which are

closest to the unknown instance. These selected k instances are k nearest neighbor of the unknown instance. To classify an unknown instance, the distance between other training instances are computed. Given a distance or similarity metric, K-NN first retrieves the k-nearest neighbors of the target instance, then assigns the class label that has the highest frequency in k-nearest neighbors to the target instance. While K-NN is simple, it is widely applied in numerous analytical tasks such as pattern recognition, text analysis, object recognition, web mining, and medical applications [50].

Memory-based learning is based on the assumption that in learning a cognitive task from experience, people do not extract rules or other abstract representations from their experience but reuse their memory of that experience directly [3]. Practically, the assumption that memory based learning has built on has been demonstrated successfully in a variety of natural language application domains such as morphological analysis [3], semantic role labeling [51], grapheme-phoneme conversion, word stress assignment, part of speech tagging, word pronunciation [52], and many other pattern recognition and machine learning applications. Memory based learning also known as instance based, example based, or lazy learning learns by storing examples of a particular task in memory [44]. Bosch and Daelemans [44] argue that memory based learning and problem solving incorporates two principles: learning is the simple storage of a representation of experiences in memory, and solving a new problem is achieved by reusing solutions from similar previously solved problems. Memory-based learning stores feature representations of training instances in memory without abstraction and classifies new instances (test instances) by matching their feature representation to all instances in memory, finding the most similar instances [51]. MBL treats a set of labeled (pre-classified) training instances as points in a multidimensional feature space and stores them as such in an instance base in memory. Therefore, in comparison with most other inductive machine learning algorithms, it performs no abstraction, which naturally allows it to deal with productive but low frequency exceptions. Bosch and Daelemans [44] also argue that the memory-based learning approach applied to a reformulation of the problem as a classification task of the segmentation type has a number of advantages when compared to rule-based approach. Those advantages are:

- It presupposes no more linguistic knowledge than explicitly present in the corpus used for training, i.e., it avoids a knowledge-acquisition bottleneck;

- It is language-independent, as it functions on any morphologically analyzed corpus in any language;
- Learning is automatic and fast;
- Processing is deterministic, non-recurrent (i.e., it does not retry analysis generation) and fast, and is only linearly related to the length of the wordform being processed.

Memory-based learning system has two components: a learning component which is memory based and a performance component which is similarity based [3]. The leaning component is used to store examples in memory while the performance component is used to map input to output. The general architecture of MBL system is shown in Figure 2.1 [53].

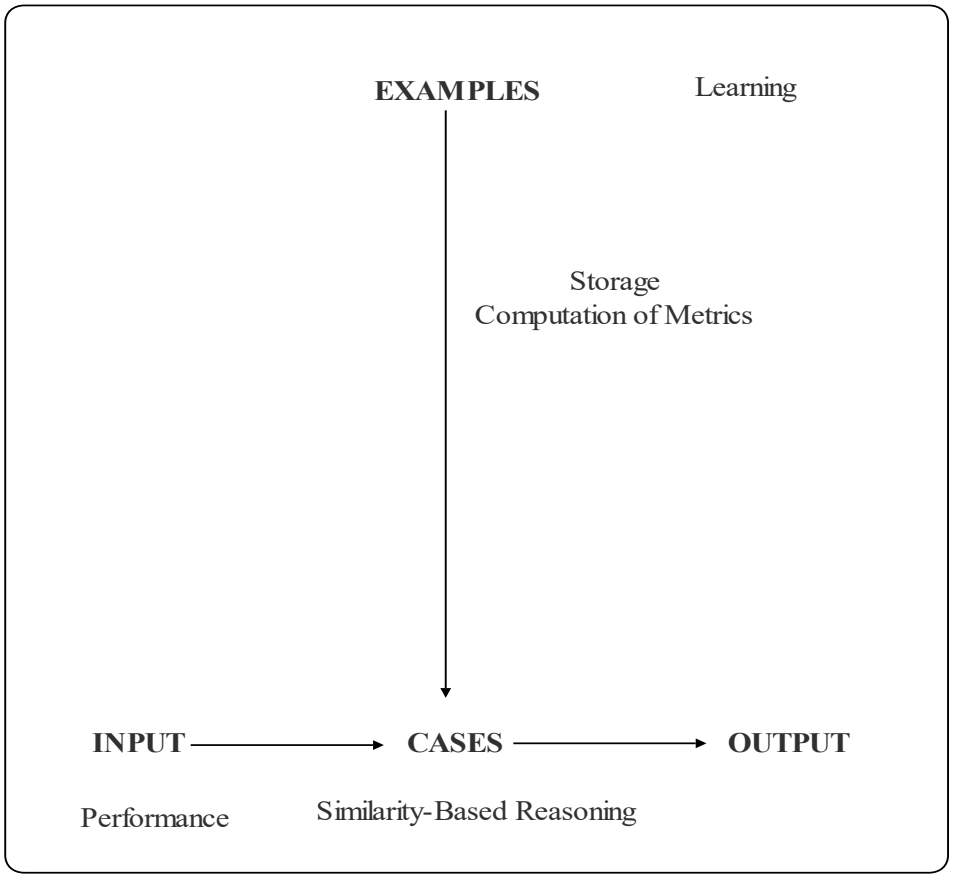

Figure 2.1: The general architecture of MBL system

There are two major memory-based learning algorithms namely: IB1 and IGTREE algorithms. These algorithms had achieved an immense accuracy result on most of natural language processing applications problem expressed as classification problem. However, both algorithms differ in the way the similarity between test instances and the instances in the

memory are calculated, the way instances are stored in the memory and the way the search of instances through the memory is conducted.

IB1 is one of the memory-based learning algorithms which constructs a database of instances called instance base that can be stored in memory during learning [33, 53]. The instances that are stored in a memory to construct instance base consists of a fixed-length vector of n feature-value pairs and additional information field containing the classes of that particular feature-value vector. When a class has more than one associated feature-value vector, the occurrences of the different classifications in the learning material are counted and stored with the instance. After the instance base is built, test instances are classified by IB1 by matching them to all instances in the memory, and calculating with each match the distance between the test instance X and the memory instance Y, Δ(X, Y), using the function in equation (1).

$$\Delta(X, Y)=\sum_{i=1}^{n} \delta(xi, yi) \quad (1)$$

Where:

$$\delta(xi, yi)=\begin{cases}0 & if\ xi = yi, \\ 1 & if\ xi \neq yi\end{cases}$$

n is the number of characters of instances

The major difference between the originally developed IB1 [54] and the one described here is, that the value of k refers to k-nearest distances rather than k-nearest examples.

IGTREE is a memory-based learning algorithm developed based on the positive effect of using information gain and decision tree [55, 56]. IGTREE constructs decision trees of instances and retrieves classification information from that tree by combining two algorithms. Instances are stored in the tree as a path of connected nodes during the construction of IGTREE decision trees. Nodes are connected via arcs representing feature values. Information gain is used in IGTREE to determine the order in which instance feature values are added as arcs to the tree. The reasoning behind this reorganization (compression) is that when the computation of information gain points to one feature clearly being the most important in classification, search can be restricted to matching a test instance to those memory instances that have the same feature value as the test instance at that feature. Instead of indexing all memory instances only once on this feature, the instance memory can then be optimized

further by examining the second most important feature, followed by the third most important feature, etc. A considerable compression is obtained as similar instances share partial paths. The tree structure is compressed even more by restricting the paths to those input feature values that disambiguate the classification from all other instances in the training material. The idea is that it is not necessary to fully store an instance as a path when only a few feature values of the instance make the instance classification unique.

IGTREE also stores with each non-terminal node information concerning the most probable or default classification given the path thus far, according to the classification bookkeeping information maintained by the tree construction algorithm. This extra information is essential when processing new instances. Processing a new instance involves traversing the tree (i.e., matching all feature-values of the test instance with arcs in the order of the overall feature information gain), and either retrieving a classification when a leaf is reached (i.e., an exact match was found), or using the default classification on the last matching nonterminal node if an exact match fails.

In memory-based learning algorithms, there are no universal rules of thumb that exist for setting parameters such as the k in the k-NN classification rule, the distance similarity metric, or the feature weighting metric, or the distance weighting metric [3]. They interact in unpredictable ways. They can seriously change generalization performance on unseen data. The most widely used similarity metrics in memory-based learning algorithm (IB1) are overlap, modified value difference, Jeffrey divergence and Jensen-Shannon divergence [3, 53]. The feature values and classes in NLP tasks are often represented by symbolic labels. Weighted overlap metric is limited to either a match or a mismatch to calculate the similarity of feature values as described in equation 1. Modified value difference metric is a method used to determine the similarity of the values of a feature by looking at co-occurrence of values with target classes [53]. For the distance between two values v1, v2 of a feature, the difference of the conditional distribution of the classes C*i* for these values can be computed as:

$$\delta(v1, v2)=\sum_{i=0}^{n} |P(Ci|v1) - P(Ci|v2)| \quad (2)$$

There are different feature weighting metrics used within memory-based learning algorithms: gain ratio, information gain, and other statistical metrics [3]. Information gain feature

weighting looks at each feature in isolation, and estimates how much information it contributes to our knowledge of the correct class label. The information gain estimate of feature *i* is measured by computing the difference in entropy between the situations without and with knowledge of the value of that feature:

$$IG(wi) = H(C) - \sum_{v \in Vi} P(v) x\, H(C|v) \quad (3)$$

Where IG is information gain, wi is the weight of a feature, C is the set of class labels, $H(C) = -\sum_{c \in C} P(c) \log_2 P(c)$ is the entropy of the class labels and Vi is the set of values for feature i.

A well-known problem with IG is that it tends to overestimate the relevance of features with large number of values. To normalize information gain over features with high number of values, gain ratio was introduced [53]. Gain ratio can be computed by information gain divided by split info *(si(i))*, the entropy of the feature values:

$$wi = \frac{H(C) - \sum_{v \in Vi} P(v) x\, H(C|v)}{si(i)} \quad (4)$$

$$si(i) = H(V) = -\sum_{v \in Vi} P(v) \log_2 P(v) \quad (5)$$

The other parameter that determines the result of MBL systems is the number of nearest neighbors. Once a set of nearest neighbors is determined, there are different ways in which the output class can be decided [3, 53]. Majority voting method is among the ways which decides the output class, the distance-weighted class voting. It is the most straightforward method for letting the k-nearest neighbors vote on the class of a new case, in which the vote of each neighbor receives equal weight, and the class with the highest number of votes is chosen. Dudani [57] proposed inverse linear (IL), a voting rule in which the votes of different members of the nearest neighbor set are weighted by a function of their distance to the query. In inverse linear, a neighbor with smaller distance is weighted more heavily than one with a greater distance: the nearest neighbor gets a weight of 1, the furthest neighbor a weight of 0 and the other weights are scaled linearly to the interval in between. Inverse linear can be defined as:

$$wj = \begin{cases} \frac{dk - dj}{dk - d1} & if\ dk \neq d1 \\ 1 & if\ dk = d1 \end{cases} \quad (6)$$

where *dj* is the distance to the query of the j^{th} nearest neighbor, *d1* the distance of the nearest neighbor, and *dk* of the furthest (k^{th}) neighbor.

Dudani [57] also proposed the inverse distance (ID) weight which can be defined as:

$$wj = \frac{1}{dj+\epsilon} \quad (7)$$

Where ϵ is a small constant added to the denominator to avoid division by zero.

B. Inductive logic programming (ILP)

Several techniques aiming to learn word morphology have been emerged since the beginning of the study of computational morphology. Inductive logic programming is one of them in which annotated data is required to learn morphology. Inductive logic programming (ILP) is a supervised machine learning framework based on logic programming [15]. It aims at the inductive learning of concepts from examples and background knowledge in a first order logic framework. In ILP, the training examples, the background knowledge and the induced hypothesis are all expressed in a logic program form, with additional restrictions imposed on each of the languages [58]. It has been successfully applied to a variety of natural language processing applications such as morphological analysis, part of speech tagging and parsing.

C. Support Vector Machine

Support vector machines (SVM), also called support vector networks are one of supervised learning models. Initially, it was built as a binary classification method and later regression and multi-class classification were added to it. In support vector machine, morphological problem is considered as a classification problem as in other machine learning models. SVM is another approach to supervised pattern classification which has been successfully applied to a wide range of classification problems. It has been successfully applied to numerous natural language tasks such as, morphological analysis, handwritten digit recognition, object recognition, text classifications [59].

ii. Unsupervised Morphology Learning

Unsupervised morphology learning is an alternative approach of machine learning approach that has become popular recently. It has a very long history in natural language processing to learn morphology from a large list of unannotated words in the target language [60]. Unsupervised learning approaches are attractive in morphology learning due to the availability of large quantities of unlabeled text and the lack of supervised labels makes it more important to leverage rich features and global dependencies [61]. Hammarström [62] defines the

unsupervised morphology learning problem as input, a raw (unannotated, non-selective) natural language text data and output, a description of the morphological structure of the language of the input text with little supervision. Hence, from a practical point of view, solving the morphology problem in unsupervised manner has the advantages of elegance, economy of time and money (no annotated resources required), and the fact that the same technology may be used on new languages. The unsupervised methods are appealing as they can be applied to any language for which there exists a sufficiently large set of unannotated words in electronic form. There are a wide range of methods and algorithms that have been used to develop unsupervised morphology learning since its inception. Hammarström [62], and Burcu Can and Suresh Manandhar [22] have presented an immense of review and survey on the work of unsupervised learning morphology of more than 5 decades. The methods and algorithms are briefly described here based on their category.

1. Letter Successor Variety models

Letter successor model is one of the varied methods of unsupervised learning morphology enhanced from time to time to its current standing. It is a measure of the number of different letters encountered after or before a certain substring, given a set of other strings as context [63]. The work of Harris [30] is given as the starting point of unsupervised learning morphology and was the first to introduce the idea of modeling letter successor variety model. According to this model, the possible segmentation points within a word can be characterized by the sharp changes in the number of successors of a letter within a word. For instance, the corpus comprises the words *walnut, wall, walks, walked, walking, walk.* The number of letter successors of the prefix *wal* equals 3, namely, *n, l,* and *k* and the number of letter successors of *walk* is 4, namely, *s, e, i* and *$*. The number of letters that can follow each letter in a word is successor variety whereas, the letters that precede other letters are called predecessor variety. In order to find morpheme boundaries, the successor counts are applied to all words in the corpus. For instance, the procedure may choose *-ing* as a morpheme. Consequently, all words that precede *-ing* are considered as stems. This problem causes the model to segment words that do not contain *-ing* as a morpheme such as *sing, string, spring, cling,* etc. In order to solve the problem happened in the model, different works have been undertaken.

Hafer and Waiss [31] have taken the original idea of Harris's [30] model to improve LSV using the entropy of the successors and predecessors instead of using raw counts. Using entropy in the segmentation process allows the importance of each successor letter to be weighted by its probability of occurrence. Rare successor letters will have a smaller effect on the segmentation decisions than will highly probable successors. The letter successor entropy (LSE) of a prefix *w* is defined as follows:

$$\text{LSE}(w) = -\sum_{c \in \Sigma} \frac{f(wc)}{f(w)} log_2 \frac{f(wc)}{f(w)} \quad (8)$$

where Σ is the alphabet, $f(wc)$ is the number of word entries in the corpus that have prefix *w* followed by the letter *c*, and $f(w)$ is the total number of the word entries that begin with *w* and can be followed by any letter.

Déjean [64] developed another version of letter successor variety. The author was inspired by the distributional approach developed by Harris [30], and Hafer and Waiss [31] to develop the newest version of letter successor variety. The new version of letter successor variety method being developed is based on the initial method by dividing the process into three different phases. The first phase is, the construction of morpheme dictionary by using the letter successor variety technique and choosing only the high frequency morphemes based on Harris [30]. The second phase is, the segmentation of words in the corpus using the morpheme dictionary to generate more morphemes. The final phase is, analyzing the corpus by using the morpheme dictionary being constructed. For instance, given the words *lights, lighting, lighted, lightly, lightness, lightest, lighten*. In the first phase, the most frequent morphemes are selected such that */-s/, /-ing/, /-ed/, /-ly/* that have a higher LSV frequency than a given threshold value. In the second phase */-ness/, /-est/,* and */-en/* are captured by segmenting the words *lightness, lightest,* and *lighten*. Finally, the entire corpus is morphologically analyzed using the combined morpheme dictionary */-s/, /-ing/, /-ed/, /-ly/, /-ness/, /-est/, /-en/*.

Bordag [65] tries to enhance the work of Harris [30], and Hafer and Waiss [31]. The model developed involves two steps. In the first step, a local LSV is used to segment words that are contextually similar. The contextual similarity is intended to group words that are syntactically similar. Thus, the idea is to identify syntactically similar words such as subclasses of adjective, verbs etc. and choose a different local LSV cutoff value for each subclass. With this technique, orthographically similar words such as *early* and *clearly* are

analyzed independently since they tend to be contextually different. The final LSV score is computed for each position in the input word by multiplying the original LSV score, the weighted average of substring frequency and the inverse bigram weight. To obtain a combined score, the combination of local LSV weights, the inverse bigram weights are used in addition to the original LSV score. In the second step, a PATRICIA compact tree [63, 65] was used to train the classifier.

2. Minimum Description Length

Minimum description length is a form of analysis rigorously based on information theory [29, 66]. Golsmith [66] states the minimum description length, given a corpus, an MDL model defines a description length of the corpus, given a probabilistic model of the corpus: the description length is the sum of the most compact statement of the model expressible in some universal language of algorithms, plus the length of the optimal compression of the corpus, when we use the probabilistic model to compress the data. Grünwald [67] argues the concept of minimum description length as any regularity in the data can be used to compress the data, i.e. to describe it using fewer symbols than the number of symbols needed to describe the data literally. Accordingly, the more regularities there are, the more the data can be compressed. There are numerous works that are involved in utilizing the concept of minimum description length. Michael R. Bent *et al*. [68] encodes the stems and suffixes as binary codes and the encodings are kept in tables. The most frequent stems and suffixes are encoded with shorter encodings.

There is also another system called Linguistica [29, 66], based on minimum description length. It utilizes signatures to encode data in addition to using stem and affix codebooks. A signature represents the inner structure of a list of words that have similar inflective morphology.

iii. Semi-supervised learning

Semi-supervised morphology learning is the combination of supervised morphology learning, the labeled data and unsupervised learning, the unlabeled data. When dealing with a large volume of dataset, supervised machine learning is time consuming and very costly process because the dataset has to be hand labeled by either machine learning engineer or a data scientist. The unsupervised machine learning application spectrum is limited. To counter the

disadvantages of both supervised and unsupervised machine learning, the concept of semi supervised machine learning was introduced to natural language processing applications. This approach emerged to solve the problems that cannot be sufficiently solved by either supervised or unsupervised machine learning. It is widely applied on natural language processing problems such as morphology analysis [69], grammar induction [70]. In semi-supervised morphology learning, the learning system has access to both labeled and unlabeled data [69].

2.8 Evaluation Metrics to Morphological Analysis

One of the fundamental tasks in building a machine learning model is to evaluate its performance even if it is a complex task [71]. The choice of metrics or techniques used to evaluate the performance of machine learning is very important because the choice of metrics influences how the performance of machine learning algorithms is measured and compared. Cross validation, feature selection and confusion matrix are the most widely used evaluation techniques.

Cross validation is an effective evaluation procedure used for estimating the generalization accuracy of machine learning model. It is applicable and very useful technique for machine learning tasks, such as accuracy estimation, feature selection or parameter tuning and also extensively used within a wide range of machine learning approaches, such as instance-based learning, artificial neural networks, or decision tree induction [72]. Cross validation is a statistical method of evaluating the performance of a model and comparing learning algorithms by dividing data into two groups: one is used to learn or train a model and the other used to test the model. Presently, cross-validation is widely accepted in data mining and machine learning community, and serves as a standard procedure for performance estimation and model selection. The main idea behind cross-validation is that each sample in the dataset has the chance of being tested. The main problem with evaluating a machine learning model is that it may demonstrate adequate prediction capability on the training data, but might fail to predict unseen data as done for training data [73]. But, the fundamental principle of estimating the accuracy of trained model of machine learning is that the dataset used for training should not also be used for testing at the same time. Hence, the idea of cross validation

was raised to tackle this issue, starting from the remark that testing the output of the algorithm on new data would yield a good estimate of its performance.

K-fold and leave-one out cross validations are the most widely used among the different cross validation techniques that can be applied to machine learning model to estimate the performance of the learned model from available data and to compare the performance of two or more different algorithms [73, 74]. K-fold cross validation is one of the cross-validation techniques that splits the dataset, let's say, X dataset into k equally or nearly equally sized segments (groups), where k is the number of segments [73, 75, 76]. One part of the segment is used for test and the remaining k -1 parts are merged into a training subset for model evaluation. In k-fold cross validation, the training and test sets must cross-over in successive rounds such that each data point has a chance of being validated against. Leave-one out cross validation is a special case of k-cross validation where the number of segments equals the number of instances in the dataset. From the entire dataset, LOOCV needs n-1 instances of training dataset and a single instance of test dataset, where n is the total number of instances in the dataset. LOOCV is unsuitable for large dataset and computationally expensive than k-fold cross validation [73, 75].

Confusion matrix is another metric of performance measurement for machine learning classification algorithms model which is usually used to show errors of a classifier. A classifier induced by supervised learning algorithm such as memory learning algorithm performance can be evaluated based on the output class produced by the classifier and the rank it generates in making a decision [77, 78]. A classifier is a model which assigns an unclassified instance to a predefined set of classes. But relying only on standard performance indicators such as accuracy may not give much clue on the generalization or specific quality of the learned model or classifier [79, 80]. Sometimes the accuracy might be biased for a certain class and this may not provide a good indication of the overall performance for the predictive model. In this case, the accuracy is not necessarily the best measurement for predictive models, whereas the confusion matrix is still the most valuable source of performance indicators from classifiers to be analyzed.

2.9 Afaan Oromoo Language

2.9.1 Background

Afaan Oromoo, literally Oromo mouth, is one of the natural languages classified as one of the Cushitic languages spoken in Ethiopia, Somalia, Sudan, Tanzania, and Kenya [81]. Cushitic languages are a branch of the Afro-Asiatic or Harnito-Semitic language family which is reckoned to be divided into six major branches or families. The Cushitic branch is divided into a further four groups of North, Central, South and East. Of the Cushitic languages spoken in Ethiopia, Afaan Oromoo, Somali, Sidama, Haddiya, and Afar-Sabo are the languages with the largest number of speakers. Afaan Oromoo is one of the languages of lowland groups within the east Cushitic group. It is also one of the major African languages and probably the third-most widely spoken Afroasiatic language in the world, after Arabic and Hausa. [82, 81].

Afaan Oromoo is spoken as a native language (L1-first language) in one of the nine administrative regional states in Ethiopia called Oromia National Regional State. It is taught as a subject in the first cycle (0-4), second cycle (5-8) and third cycle (9-10) of education and also serves as a medium of instruction in the first and second cycles [83]. It is also taught as a subject and functions as a medium of instruction in different Ethiopian universities where Afaan Oromoo departments are established to train teachers, journalists and anyone who works in the fields of public relations and culture. It is also used in courts, media, religious organization and for administration. In Ethiopia, Afaan Oromoo is used as a lingua franca and as a day–to-day means of communication by a great many people of various ethnonational groups other than its native speakers [81].

2.9.2 Writing System of Afaan Oromoo

Tilahun Gemta [82] argues that linguistically, writing itself has passed through three stages of development before reaching the alphabet stage. The three stages are iconography, logography and syllabary. Iconography consist of drawings of animals or objects. The drawings were disconnected and fragmented, and they were intended to give merely a static impression. Later, standardized pictures were selected, arranged in a series, and were made to tell a story in the same way as today's action photographs do. Iconography was common among North American Indian tribes. Logography is the use of signs to represent word. For instance, In English, words such as one, two, three and dollar are respectively represented by the signs

1,2,3 and $. Syllabary is a set of characters which represent syllables. For example, The Amharic syllabary, used in Ethiopia today, is a very good example of a syllabic writing.

Afaan Oromoo had remained essentially a well-developed oral tradition which has been transmitted from mouth to mouth and preserved in the memories of the people until the early 1970's [82, 84]. An understanding of the phonemic inventory and phonology of a language is vital component in the discussion of a writing system. Phonemically, Afaan Oromoo has twenty-six consonants, five short and five long vowels which can occur in a word initially, medially or finally. Afaan Oromoo consonant and vowel phonemes are depicted in Table 2.2 and Table 2.3 respectively.

Table 2.2: Consonant phoneme chart

Stops	**Labial**	**Alveolar/dental**	**Palatal**	**Velar**	**Glottal**
Voiceless	(p)*	t	tʃ	k	ʔ
Voiced	b	d	(z) dʒ	g	
Ejectives	pʹ	tʹ	tʃʹ	kʹ	
Implosive		ɗ			
Spirants	f	s	ʃ		
Nasals	m	n	ɲ		h
Sonorants		r l			
Glides	w	j			

As indicated in Table 2.2, the two loan consonants (/ p / and / z /) are becoming more and more common in the Afaan Oromoo vocabulary especially since 1991 as a result of mainly of the process of modernization of the language [81].

Table 2.3: Vowel phoneme

Short			**Long**	
i	u	high	i:	u:
e	o	mid	e:	o:
a		low	a:	

All Afaan Oromoo vowels, both short and long, can occur in a word initially, medially or finally, though the rounded vowels (/ o / and / u /) only rarely occur in short form in word final position.

Qubee as a term referring to Afaan Oromoo alphabet has been in use clandestinely in Oromia and openly in the diaspora at least since the 1970s. The exact date and how it was coined, or the identity of the person or persons responsible for coining the term, however, are not clear since relevant information is not readily available [81]. *Qubee* is a Latin script that has been adopted and taken as official Afaan Oromoo alphabet for writing Afaan Oromoo on November 3, 1991 [82]. Since then, *Qubee* became an official Afaan Oromoo writing system adopted from the Roman alphabet using the principle of writing through a one-to-one correspondence between sound segments and graphemes or symbols of the alphabet. Twenty-five letters of the Roman alphabet and the apostrophe mark (′) were adopted without making changes to their shapes or sizes (for example, by way of diacritic marks) to represent Afaan Oromoo sounds. The letter "v" is the only symbol from the Roman alphabet that is not used in the *Qubee* system. Most of Afaan Oromoo phonemes, except the ejectives such as /tʃ′/, / t′/, and / k′/, have a corresponding symbol in the Roman alphabet and matching them has not proved to be difficult. However, the matching of the ejectives, and the representation of other aspects, for example, gemination and vowel length, have proved to be a challenging task especially in terms of consistency and simplicity of the writing system [81]. Table 2.4 shows a one-to-one correspondence between sound segments and graphemes or symbols of the alphabet.

Table 2.4: The Qubee, Roman characters adopted for Afaan Oromoo sound representation

No.	Phonemes	Qubee	Examples in Afaan Oromoo	Gloss
1	/a/	A	harka	hand
2	/b/	B	eeboo	spear
3	/ tʃ/	Ch	achi	there
4	/ tʃ′/	C	cabbii	snow
5	/d/	D	dargaggoo	youth
6	/ ɗ/	Dh	dhagaa	stone
7	/e/	E	ergaa	message

8	/f/	F	fayyaa	health
9	/g/	G	guyyaa	Day
10	/h/	H	haaraa	New
11	/i/	I	iddoo	place
12	/ dʒ /	J	jecha	saying
13	/k/	K	kennaa	Gift
14	/l/	L	lama	Two
15	/m/	M	mana	house
16	/n/	N	nama	Man
17	/ ɲ /	Ny	nyaata	food
18	/o/	O	oromoo	oromo
19	/p/	P	poolisii	police
20	/p'/	Ph	tapha	game
21	/k'/	Q	marqaa	porridge
22	/r/	R	raafuu	cabbage
23	/s/	S	soquu	to dig
24	/ ʃ /	Sh	ishee	she
25	/t/	T	hantuuta	mouse
26	/t'/	X	fixuu	to finish
27	/u/	U	urjii	star
28	/w/	W	waggaa	year
29	/j/	Y	wayyaa	cloth
30	/z/	Z	zeeroo	zero
31	/ ʔ /	′	yaa'uu	to flood

Tilahun Gemta [82] argue that, in addition to the 31 symbols of Afaan Oromoo, it is recommended to know the principles related to Afaan Oromoo Alphabet (Qubee):

1. Two vowels in succession indicate that the vowel is long, e.g. *bitaa* (left);
2. Gemination (a doubling of a consonant) is phonemic in Afaan Oromoo, e.g. *damee* (branch), *dammee* (sweety);

3. H and ′ are not geminated at all;
4. The same word can have two or more forms depending on its context, e.g. nama kadhu (ask person), namaa kadhu (ask for person);
5. When it occurs at the end of a word, the single "a" is pronounced schwa (inverted e) whereas it is pronounced (delta) elsewhere;
6. Understandably, instead of diacritic signs, the combined letters are used so as to align them with typewriter characters.

Tabor Wami [85] also argues that using Latin alphabet than other script to write Afaan Oromo has at least four main advantages:

1. It solves the problem of vowel length; i.e., the combination of consonant and vowel.
   Haraa- clean it Haaraa-new
   Bita –to the left Bitaa- buy
2. It solves the problem of germination
   Butuu- to abduct Buttuu- he who abducts
   Sodaa- fear Soddaa- marriage relatives
3. Since Latin alphabet is international alphabet, those who want to learn Afaan Oromoo do not have to learn a new alphabet.
4. It is relatively easy to use Latin alphabet in computer.

2.9.3 Afaan Oromoo Words Syllable Structure and Typological Classification

Afaan Oromoo has four different syllable structure such as: cv, cvv, cvc and cvvc, where c represents consonant and v represents vowel [86]. The syllables in Afaan Oromoo usually start with consonants. Phonemes like /i/, /e/, /a/, /o/, /u/ may appear at the beginning of any words, but there is always a glottal stop (ʔ) in front of them by default. For instance, in words like uumaa, ooluu, aannan, vowels are at the beginning. Table 2.5 shows Afaan Oromoo words syllable structure.

Table 2.5: Afaan Oromoo words syllable structure

Words	Syllable	No. of Syllable	Gloss
sun	cvc	1	that
dee-maa	cvv-cvv	2	go
aan-nan	cvvc-cvc	2	milk
qul-lub-bii	cvc-cvc-cvv	3	garlic
ob-bo-leet-tii	cvc-cv-cvvc-cvv	4	sister

Linguistic typologists often split languages into types according to the so-called basic word order, often understood as the order of subject (S), object (O) and verb (V) in a typical declarative sentence. The vast majority of the languages of the world fall into one of the three groups of SOV, SVO and VSO. Accordingly, the word order in Afaan Oromoo language falls under Subject-Object-Verb (*Matima-Aantima-Gochima*) word order pattern as numerous languages fall under SOV pattern [87]. The verb in Afaan Oromoo is strictly at the end. Example. *Inni foon nyaate, He ate meat.*

All typological classifications were almost exclusively morphological, since morphology was for a long time the most developed field of linguistics. Körtvélyessy [9] states the typological classification of languages as index of synthesis and index of fusion. The former one, index of synthesis deals with how much syntactic information is obtained in the average word. There are three language types that distinguished in index of synthesis: analytical, synthetic and polysynthetic. The index of fusion which deals with how do morphemes usually build words. There are four possibilities of index of fusion: isolating, agglutinative, fusional and symbolic languages. These indexes show that, a classification of language typology should be viewed as a continuum or a scale. Both indexes are overlapped with each other. Analytic languages overlap with isolating languages. In the same way, synthetic languages can be viewed from the point of view of index of fusion as either agglutinating or fusional. Depending on these classifications, Afaan Oromoo is categorized into synthetic/agglutinative language.

2.9.4 Word Formation of Afaan Oromoo

In this section, we present the different types of words, word categories and word formation of Afaan Oromoo that have great contributions to our study, particularly nouns, verbs, and adjectives, since the study focuses on the word formation of such word classes. Words are the

basic things for the grammar of a language. A word, in Afaan Oromoo “jecha”, is a part of language containing one or more sounds that can stand independently and make sense [88]. It can be defined as a sequence of characters delimited by spaces, punctuation marks, etc. in case of a written text. Addunyaa Barkeessaa [28] indicates that Afaan Oromoo word can have three major parts on its body. These main parts that appear on the body of words are root (hundee), base (bu’uura) and stem (bu’uur-hortee).

A word can be of two types: simple and compound. A simple word consists of a root or stem together with suffixes and prefixes. A compound word can be broken into two or more independent words. Each of the constituent words in a compound word is either a compound word or a simple word and may be used independently as a word [24]. Words are divided into two kinds of lexical classes: open and closed classes. In most languages, nouns, adjectives, and verbs form open classes. Function words such as determiners, conjunctions, pronouns, and ad positions (pre- and postpositions) form closed sets of words that cannot be extended by regular word-formation patterns [21]. Traditionally, linguists classify words into different categories based on their uses. Two related areas of evidence are used to divide words into categories. The first area concerns the word’s contribution to the meaning of the phrase that contains it, and the second area concerns the actual syntactic structures in which the word may play a role [1].

The first step while developing a morphological analyzer is to define the word classes and the grammatical information that will be required for words of these word classes. Every language around the world has its own word classes or parts of speech. For instance, the English language uses eight parts of speech, namely noun, pronoun, adjective, verb, adverbs, preposition, conjunction and interjection. Afaan Oromoo has five-word classes such as, verb, adjective, adverb, noun and conjunction [89]. Most linguists of Afaan Oromoo classify the word classes of Afaan Oromoo into ***gochibsa*** (Verb), ***maqaa*** (Noun), ***maqibsa*** (Adjective), ***durduubee*** (Affix), ***qabsiistuu*** (conjuction) and ***bamaqaa*** (Pronoun) [28]. Here, we prefer to discuss about the three major word classes: verb, noun and adjective because our study focuses on them. The words grouped into these word classes are formed through either derivation or inflection.

I. Derivation

Derivation is a process of word formation in which one or more affixes are attached to a root word (and stem) to produce a new word known as a derived word. The word formation discussed here is based on the analysis and discussion from [28, 90, 91].

A. Noun Derivation

Nouns are words that are used to name or identify any of categories of things, people, places or ideas or a particular of one of these entities. The process of driving a noun from the other word class is called normalization, and the types of affixes used for this purpose is called nominalizers. In Afaan Oromoo, there is a large stock of nominal derived from adjectival, verbal and nominal bases.

Suffixes involved in the derivation of nouns in Afaan Oromoo are classified into different groups based on the types of word class they change into nouns.

Group 1 */-eenya/, /-ina/*

These suffixes are used to derive nouns from adjectives as the following set of examples illustrate.

Adjectives	**Gloss**	**Suffix**	**Derived Noun**	**Gloss**
adii	white	-eenya	addeenya	whiteness
dhiyoo	near	-eenya	dhiyeenya	closeness
bareedaa	beautiful	-ina	bareedina	beauty

Group 2 */-a/, /-aa/, /-ee/, /-ii/, /-sa/, /-aatii/, /-cha/, /-choo/, /-maata/, /-nsa/, /-noo/*

These suffixes are used to derive nouns from verbs. The following example shows the derivation of nouns from verbs.

Verbs	**Gloss**	**Suffix**	**Derived Noun**	**Gloss**
ibse	make it clear	-aa	ibsaa	light
lole	he fought	-a	lola	war
dhuge	he drunk	-aatii	dhugaatii	drink

Group 3 */-ummaa/, /-ooma/*

These suffixes are used to derive nouns from other nouns. It changes the concrete noun (maqaa waan qabatamaa) to abstract noun (maqaa waan yaadaan jiruu). The following example shows the derivation of nouns from other nouns

Noun	Gloss	Affix	Derived Noun	Gloss
Bilisa	free	-ummaa	bilisummaa	freedom
Garba	slave	-ummaa	garbummaa	slavery
Fira	relative	-ummaa/-ooma	firooma	relationship

B. Verb Derivation

The verbs are words or compound of words that express action, a state of being and/or relationship between two things. Suffixes involved in the derivation of verbs (verbilizer) in Afaan Oromoo are classified into different groups based on the type of word class they change into verbs.

Group 1 */-oom-/, /-aa'-/, /-a'-/*

These verbilizers are used to derive verbs from nouns and adjectives. The following examples show the derivation of such verbs.

Word	Gloss	Verbilizer (Suffix)	Derived verb	Gloss
Arjaa	donator	-oom-	arjoome	donated
Gurraacha	black	-a'-	gurraacha'e	blacked
Qulqulluu	blessing	-aa'-	qulqullaa'e	blessed

Group 2 /-at-/, /-am-/, /-sis-/, /-siis-/, /-s-/

These verbilizers are used to derive verbs from adjectives and other verbs. The following examples show the derivation of such verbs.

Word	Gloss	Verbilizer (Affix)	Derived verb	Gloss
Mare	rolled	-at-	marate	have rolled
Diimaa	red	-at-	diimate	became red
Mare	rolled	-am-	marame	was rolled
Dhuge	he drunk	-siis-	dhugsiise	cause to drink

C. Adjectives Derivation

Adjectives are words that describe or modify nouns (and pronouns). Forming adjectives from another lexical category is termed as adjectivization. From stative verb like /**diim-at**/ 'become red' one can derive the adjective /**diim-at-aa (-tuu)/ '**reddened'. In Afaan Oromoo adjectives can be formed from verbs by taking adjectivizers like **/-aa/, /-tuu/, /-eessa/, and /-eettii/**. Afaan Oromoo adjectives are not derivatives as nouns and verbs, this is because there are a few numbers of adjectivizers in the language [28].

The following examples show the derivation of such adjectives.

Word	Gloss	Affix	Derived Adjectives	Gloss
Sodaate	feared	-aa/-tuu	sodaataa/sodaattuu	fearful
Iyye	shouted	-eessa/-etti	iyyeessa/iyyeettii	poor

II. Inflection

The word formation discussed here is based on the analysis and discussion from [28, 90, 91].

A. Noun (Maqaa)

Nouns in Afaan Oromoo end with a vowel except few which ends with consonants. Inflectional categories that are inherent to nouns indicate different grammatical functions like number, gender, definiteness, case and focus markers [91, 92]. Number and gender markers are considered as inherent categories, while case marker is considered as relational.

1. Number (Lakkofsa)

Afaan Oromoo distinguishes between singular and plural nouns as for other languages. In contrast to the English plural noun markers /-(e)s/, there are different types of suffixes that can be attached to nouns to indicate plurality. Addunyaa Barkeessaa [28] categorizes those suffixes into groups like, */-n/, /-oota/, /-lee, -lii/, /-yyii/*. Griefenow-Mewis [92] argues that the use of these suffixes is not obligatory but used if the plurality of the noun is not clear from the context. Most of the time, the suffix */-oota/* and its allomorphic variant */-ota/* is used to mark plurality. For example, *nam-a 'man'* becomes *nam-oota 'men'*.

2. Gender (Koorniyaa)

The gender class of every language can be marked in many ways. The determiner is used to mark the gender class of the noun in some languages (e.g., French), while in other language (e.g., Russia) inflectional affixes are used to mark gender class of the noun rather than determiner [28]. In Afaan Oromoo, there are two types of grammatical gender: masculine and feminine [87, 93]. These genders are identified through the semantics of the words, combining the word classes and gender marking suffixes. Semantically, distinct words are used for masculine and feminine. For instance, *abbaa* 'father' represents masculine gender, while *mother* 'haadha' represents feminine gender. In combining word classes, gender representing words can be used for animals and they are positioned immediately after the nouns they belong to. *Kormaa, '*male' and *dhaltuu* 'female' are the most common contrastive pair of words used

for animals in most cases. Gender suffix marking like */-ich-/* and */-ttii/* are used to show masculine and feminine gender class in some cases. These suffixes are rarely used. For example,

Word	**Gloss**	**Word**	**Gloss**
Abbaa	father	haadha	mother
Ilma	son	intala	daughter
Korma	male	naayee	female

3. Definiteness (Beekamtummaa)

In Afaan Oromoo, the grammatical properties of definiteness are marked using the suffix */-ich-/* for masculine gender and */-ittii/* for feminine gender respectively. The masculine marker is positioned between the stem and accusitave case marker, while the feminine marker is appended to the stem by removing the vowel endings. Afaan Oromoo doesn't have any overt marker of definiteness which means a specified noun that can be either singular or plural [91]. For example, *nama* 'man' becomes *nam-ich-a* 'a/the man' for masculine and becomes *nam-ittii* 'a/the man' for feminine. The definite marker requires definiteness being with singular nouns in Afaan Oromo.

4. Case (Maayii)

Afaan Oromoo marks nouns for case inflection, the relational category. Addunyaa Barkessa [28] states that, the suffixes used to mark the case are nominative, accusative (direct object), dative (indirect object) and instrumental. The nominative cases are used for nouns that are the subject of intransitive verbs and agent of the transitive verbs. In Afaan Oromoo, nominative case morphemes are */-ni/, /-i/, /-n/ and /Φ/*. For example, *luk-i 'leg',* the morpheme */-i/* shows nominative case. The */-n/* suffix is attached to the nouns that end with long vowels. Accusative case is a direct object marker marked by a morpheme */-a/*. For example, in a word *nam-a 'man'*, the last vowel or morpheme */-a/* shows the accusative marker. Dative marker is an indirect object case marker marked by a morpheme */-dhaa(f)/* and its allomorphic variants */-a(f)/, /-tii(f)/ and /-ii(f)/*. For example, *nam-a-af* 'for man', */-af/* shows a dative case marker. The instrumental case is marked by a morpheme */-n/*. For example, in *nam-a-a-n 'by man'*, the last marker */-n/* shows instrumental case.

5. Focus (Xiyyeeffannoo)

In Afaan Oromoo, focus suffixes are attached to the last constituent of a noun. The focus markers attached nouns are */-tu/, /-uma/ and /-dhuma/* [28]. When */-tu/* focus marker is attached to a noun, no case marker is allowed. For example, in *nam-a-tu 'it is man',* the last marker */-tu/* shows focus marker.

B. Adjective Inflection

In Afaan Oromoo, there are uninflected adjectives that are morphologically simple and do not take inflectional affixes for any of the grammatical categories as there are inflected adjectives [28]. The following examples show uninflected adjectives:

Uninflected word	**Glossary**
Haaraa	new
Doofaa	fool

Like nouns, the inflectional categories of adjectives are inflected for different grammatical categories such as number, gender, definiteness and case. But there is a different way of affixation that is used for inflectional purpose of adjectives only. For instance, unlike nouns, adjectives are inflected by reduplication of stem to mark the plurality. The */-(o)ota/* suffixes are also used to show plurality. Adjectives are inflected for gender class. The marker */-aa/* is used to show masculine, while the markers */-tuu/* and */-oo/* show feminine gender. Both */-tuu/ and /-oo/* morphemes are suffixed to most adjectives that end with long */-aa/* by removing them. The case and definiteness grammatical categories of adjectives are identical with noun.

C. Verb Inflection

Afaan Oromoo verb inflection occurs for inherent, the basic members of a word class triggering inflection on that word class such as aspect, mood, and voice and agreement properties, inflection of a word class for properties out of its members such as person, number and gender [91]. There are some studies that include tense to inflectional categories of a verb independently but from the three major tenses present, past and future, Afaan oromoo mainly identifies between past and non-past in its morphology because the morphological markers do not differentiate each tense categories [28, 94]. Because of this, Afaan Oromoo conflates tense with aspectual categories.

Aspect (Haala Raawwii)

Afaan Oromoo is one of those languages which expresses aspect grammatically. There are two major aspectual categories such as perfective and imperfective. Perfective aspect presents a situation as completed, whereas imperfective aspect presents the situation as ongoing or not completed. The two aspects are distinguished primarily by their suffix vowel attached to root or derived stem. Table 2.6 shows the summary of the suffix vowel of perfective and imperfective aspects.

Table 2.6: Afaan Oromoo aspects

Aspect	**Subject**	**Root**	**Agreement**		**Suffix**	**Inflected form**
			Per.	**Num.**		
Perfective	1SG	deem-	-ø-	-	-e	deem -ø-e
	2SG		-t-	-	-e	deem -t-e
	3SGM		-ø-	-	-e	deem -ø-e
	3SGF		-t-	-	-e	deem -t-e
	1PL		-n-	-	-e	deem -n-e
	2PL		-t-	-an-	-i	deem -t-an-i
	3PL		-ø-	-an-	-i	deem -an-i
Imperfective	1SG		-ø-	-	-a	deem -ø-a
	2SG		-t-	-	-a	deem -t-a
	3SGM		-ø-	-	-a	deem -ø-a
	3SGF		-t-	-	-i	deem -t-i
	1PL		-n-	-	-a	deem -n-a
	2PL		-t-	-	-u	deem -t-u
	3PL		-ø-	-	-u	deem -ø-u

The suffix vowels /-e/ and /-a/ primarily distinguish between perfective and imperfective aspects. However, the allomorph /-*i/* is used within both perfective aspect when the subject is 2PL and 3PL, and imperfective aspect when the subject is 3SGF respectively. On the other hand, the allomorph /-*u/* is an imperfective aspect marker occurring with 2PL and 3PL subjects. Therefore, the markers /-*i/* and /-*u/* are allomorphs of the aspect marker /-*a/* whereas the marker /-*i/* occurs as allomorphic variant of /-*e/* for perfective aspect. The form for a 1SG

subject is identical with a 3SGM subject, and a 2SG subject with a 3SGF subject. 1SG, 3SGM and 3PL subjects are marked by person marker /-ø-/ (zero morpheme) while 2SG, 3SGF, 2PL subjects are marked by person marker /-t-/.

Mood (Gochaalaa)

Afaan Oromoo shows mood as other languages show mood inflection. Afaan oromoo has two major moods such as imperative (ajaja) and jussive (eyyama) moods. The imperative mood is the form of a verb that indicates command given to the second person singular and plural [95]. It is formed by means of adding suffix /-*i/* and /-*aa/* to the root or derived stems [91, 92]. Table 2.7 shows the markers used in imperative mood without reflexive.

Table 1.7: Imperative mood markers

Person	**Root**	**Marker**	**Inflected form**	**Gloss**
2sg	utaal-	-i	utaal-i	jump
2pl		-aa	utaal-aa	jump

The jussive mood is marked by the preverbal particle */haa-/* and the dependent suffix /-u/ or /-i/ on the verb root [91]. Table 2.8 shows the markers used in jussive mood.

Table 2.8: Jussive mood markers

Person	**Jussive**	**Root**	**Agreement**	**Aspect**	**Inflected form**	**Gloss**
3sgm	haa	guut-	-ø-	-u-	haaguut-u	Let him fill
3sgf			-t-	-u-	haaguut-t-u	Let her fill
1pl			-n-	-u-	haaguut-n-u	Let us fill
3pl			-an-	-i-	haaguut-an-i	Let them fill

Voice

Afaan Oromoo has primarily two voices such as active voice and passive voice. Active voice is when the subject performs the action whereas passive voice is the form in which the subject receives the action [91]. The active voice is unmarked and the passive voice is marked by /-*am- /* morpheme. Table 2.9 shows active and passive verbs.

Table 2.9: Active and Passive voice

Voice	Root	Marker	Aspect	Inflected form	Gloss
Active	mar-	-	-e	mar-e	rolled
	gurgur-	-	-e	gurgur-e	sold
Passive	mar-	-am-	-e	mar-am-e	was rolled
	gurgur-	-am-	-e	gurgur-am-e	was sold

Addunyaa Barkessaa [96] argues that negation marker, */hin-/.../-n/* and affirmative marker, */ni/* can be attached to verbs. Table 2.10 shows negation and affirmative markers.

Table 2.10: Negation and Affirmative

Word	Morpheme	Negation/Affirmative	Gloss
nideeme	ni-deem-e	ni-	He went
nideemna	ni-deem-n-a	ni-	We will go
hindeemin	hin-deem-i-n	hin-, -n	Don't go

2.10 Summary

In this chapter, we presented the background information concerning the current research work such as natural language processing, morphology, morpheme, morphological analysis, approaches to morphological analysis, evaluation metrics and the linguistic properties of Afaan Oromoo. Natural language processing has a deep and diverse concept. It is a computer processing of natural language. It is successfully applied to solve different natural language problems like morphological analysis. Morphological analysis is one of the linguistic levels. It is an analysis of words that aimed at segmenting words into their component morphemes and usually the assignment of grammatical information to grammatical categories. The problem of morphological analysis is solved by either rule based or machine learning. Memory-based learning is one of the machine learning algorithms that solves the problem of morphological analysis described as classification problems. It is based on the assumption that in learning a cognitive task from experience, people do not extract rules or other abstract representations from their experience but reuse their memory of that experience directly. Commonly, there are two different morphological system performance evaluation methods. They are cross validation methods and the metrics computed from confusion matrix.

In order to solve the morphological problem of Afaan Oromoo morphological analysis, the linguistic properties of Afaan Oromoo were studied. Afaan Oromoo is one of the Cushitic languages spoken widely in Ethiopia. It has a writing system adopted from a Latin alphabet called *'Qubee'. Qubee* has 26 consonants and 10 vowels. Afaan Oromoo has four different syllable structure such as: cv, cvv, cvc and cvvc, where c represents consonant and v represents vowel. Afaan Oromoo follows Subject-Object-Verb (*Matima-Aantima-Gochima*) word order pattern. Typologically, Afaan Oromoo is categorized into synthetic/agglutinative language. The words of Afaan Oromoo can be formed through either derivational or derivational morphology.

Chapter Three: Related Work

3.1 Introduction

Morphological analysis is a significant component in any system of natural language processing. To build this important component of natural language processing, numerous researches have been conducted on different languages around the world with different approaches. This Chapter presents, the review of various studies that have been carried out on different languages with different approaches related to the current research work.

3.2 Morphological Analyzer Developed Using Machine Learning Approach

3.2.1 Morphological Analyzer for Dutch

Bosch and Daelemans [3] have built a model called memory-based morphological analysis (MBMA) for a Dutch language. It is a memory-based learning approach which models morphological analysis of complex wordforms. They employed CELEX lexical database as a corpus. CELEX contains a large lexical database of Dutch wordforms and features a full morphological analysis for 247,415 of them. They took each wordform and its associated analysis, and created instances using a windowing method. In this way, they generated an instance base of 2,727,462 instances. Within these instances, 2422 different class labels occur. Ambiguity in syntactic and inflectional tags occurs in 3.6% of all morphemes in their CELEX database. The memory-based learning algorithm used to develop memory-based morphological analysis for Dutch language is IB1-IG.

They performed an experiment to evaluate the performance of the model developed for Dutch language. In order to do their experiment, they used 10-fold cross validation in an experimental matrix in which MBMA is applied to the full instance base. They organized the presentation of the experiments into generalization accuracies on test instances. The generalization accuracies in terms of percentage of correctly classified test instances with three lower-granularity tasks derived from MBMA's full output is 95.88% for morphological analysis, 98.83% for derivation/inflection and 98.97% for segmentation respectively. The precision and recall of morphemes can be computed at different levels of granularity. Precision and recall of morphemes, derived from the classification output of MBMA applied to the full task and two lower-granularity variations of Dutch morphological analysis is 84.33% and 83.78%, derivation/inflection is 94.72% and 94.07%, and segmentation is 94.83%

and 94.18% respectively. However, the system has shown an encouraging value, it cannot return more than one possible segmentation for a wordform.

3.2.2 Morphological Analyzer for Macedonia

The Macedonian language belongs to the South-Slavic family of languages and with the other Slavic languages shares a rich system of inflections [97]. Aneta Ivanovska et al. [97] developed learning rules for morphological analysis and synthesis of Macedonian nouns, adjectives and verbs. First, they prepared morphosyntactic annotation of words according to the multext-east specification, where each wordform is associated with morphosyntactic description (MSD) presented as a packed string. Then, they converted lexicons into the format suitable for running Clog, the inductive logic programming system. To learn rules for morphological analysis and synthesis, they used word-forms from the Macedonian translation of Orwell's 1984 as a corpus for the three grammatical categories: adjectives, verbs and nouns. The morphological analysis and synthesis were carried out over 5,078 word-forms of adjectives, 5483 word-forms of verbs and unspecified for nouns. 10-fold cross validation evaluation method was used to evaluate the performance of the rules for morphological analysis and synthesis of adjectives, verbs and nouns.

The obtained average accuracies of the learned rules for analysis and synthesis of adjectives are 93.12% and 82.77%, of verbs are 91.65% and 95.71%, and of nouns are 97.01% and 94.81%. The system has shown a best result but morphosyntactic descriptions (MSDs) with less than 100 examples do not provide enough data to induce good rules.

3.2.3 Morphological Analyzer for English

Tang [98] developed English morphological analysis with machine-learned rules for English language. In order to develop the system, they have adopted the approach proposed by Keshava and Pitler [99] in learning affix rules from wordlist and tested the approach using wordlist of different scales. To learn the affix rules, one forward lexicographic tree and one backward lexicographic tree were built. For the forward and backward lexicographic trees, they used a corpus of 24,447,034 tokens. Like word segmentation in Chinese, there are two types of ambiguities in morphological analysis such as intersectional and combinatory ambiguities. The key to intersectional ambiguity is to decide where the morphological boundary is while the key to combinatory ambiguity is to decide whether there is a morphological boundary inside the wordform. Then, they applied disambiguation and affix

rule order since a wide-covering and correct set of affix rules alone does not guarantee a successful analysis. In order to solve the intersectional ambiguity, they employed transitional probability proposed by Keshava and Pitler [99], but it does not work for all wordform. Combinatory ambiguity is more difficult to solve than intersectional ambiguity. Even a simple finite automaton surely cannot solve the problem, as every rule may have an exception. To solve combinatory ambiguity, they have also chosen to rely on letter transitional probability. For the affix rule order, they used inflectional morphemes→derivational morphemes→lexical morphemes sequence.

Finally, they conducted an experiment to evaluate the performance of the analyzer. The evaluation of the analyzer shows a promising result with an 88.46% precision, 78.61% recall and 83.24% F-score.

3.2.4 Morphological Analyzer for Amharic

Wondwossen Mulugeta and Gasser [15] developed learning morphological rules for Amharic verbs using Inductive logic programming system, CLOG. Learning morphological rules with ILP requires preparation of the training data and background knowledge. To handle the complexity of Amharic language, they required background knowledge predicates that can handle stem extraction by identifying affixes, root and vowel identification and grammatical feature association with constituents of the word. They run three separate training experiments to learn the stem extraction, root patterns, and internal stem alternation rules. In order to learn stem extraction, the background predicate *'set_affix*' uses a combination of multiple '*split'* operations to identify the prefix and suffixes attached to the input word. This predicate is used to learn the affixes by taking only the *word* and the *stem.* The root extraction predicate, '*root_vocal'*, extracts *root* and the *vowel* with the right sequence from the *stem*. This predicate learns the root by taking only the *stem* and the *root.* Another challenge for Amharic verb morphology learning is handling stem internal alternations. For this purpose, they have used the background predicate *'set_internal_alter'.* This predicate works much like the '*set_affix*' predicate except that it replaces a substring which is found in the middle of *stem* by another substring from *valid_stem*. In order to learn stem alternations, they required a different set of training data showing examples of stem internal alternations. Along with the three experiments for learning various aspects of verb morphology, they have also used two utility predicates to support the integration between the learned rules and to include some language

specific features. These predicates are '*template'* and '*feature'*. '*Template'* is used to extract the valid template for *stem* while '*feature'* is used to associate the identified affixes and root CV pattern with the known grammatical features from the example. For the training purpose, they used 216 manually prepared Amharic verbs. After training the program using the example set, which took around 58 seconds, 108 rules for affix extraction, 18 rules for root template extraction and 3 rules for internal stem alternation have been learned. Finally, they have combined the background predicates used for the three learning tasks and the utility predicates. They have also integrated all the rules learned in each experiment with the background predicates.

After building the program, to test the performance of the system, they started with verbs in their third person singular masculine form, selected from the list of verbs. They then did inflection of verbs for the eight subjects and four tense-aspect-mood features of Amharic, resulting in 1,784 distinct verb forms. The system is able to correctly analyze 1,552 words, resulting in 86.99% accuracy. Even if the system achieves a promising result, it has limitations. Amharic verbs have more prefixes and suffixes for various morphological features. So, the system is limited to only subject markers.

Mesfin Abate and Yaregal Assabie [7] developed morphological analyzer for Amharic using memory-based learning. It contains two major components: training phase and analysis phase. Since there were no inputs, morphological database, for training datasets, they prepared annotated datasets manually. In order to prepare annotated datasets, they identified and performed different tasks such as identifying inflected words; segmenting the word into prefix, stem, suffix; putting boundary marker between each segment; and describing the representation of each marker. Then, the annotated words are stored in a database and instances are extracted automatically from the morphological database based on the concept of windowing method. The morphological analysis phase contains the feature extraction to de-construct a given text, morpheme identification to classify and extrapolate, stem and root extraction to label segmented inflected words with their morpheme functions. Given a new word to be analyzed, the feature extraction accepts and de-constructs as instances to make similar representation with the one stored in memory. When new or unknown inflected words are deconstructed as instances and given to the system to be analyzed, an extrapolation is performed to assign the most likely neighborhood class with its morphemes based on their

boundaries. After they identified morpheme in this manner, they performed stem extraction. In stem extraction, reconstruction of individual instances into meaningful morphemes (to their original word form) and insertions of identified morphemes in their segmentation point are performed. Then, they did root extraction. In order to extract the root from verbal stems, they removed the vowels from verbal stems.

The corpus they prepared contains 1,022 words, of which 841 are verbs and 181 are nouns and adjectives. The number of instances extracted from nouns and adjectives are 1,356 and from verbs are 6,719 which accounts a total of 8,075 instances. A total of 26 different class labels occur within these instances. For their task, they employed TiMBL tool. To get an optimal accuracy of the model they used the default settings, modified value difference metric and chi-square, information gain, inverse distance, and k from the nearest neighbor. The classifier engines they used were IB1 and IGTREE which construct databases of instances in memory during the learning process. In order to evaluate the performance of the model and the capability of learnability of the dataset, they conducted an experiment by combining nouns and verbs. They used two popular cross validation evaluation methods: leave out-one and 10-fold cross validation. The accuracy of the system is 96.40% and 93.59 % for IB1 algorithm using leave-out-one and 10-fold respectively. The accuracy of the system is also 82.26% for IGREE algorithm using 10-fold. The system doesn't extract the root.

3.2.5 Morphological Analyzer for Tamil

T. Mokanarangan et al. [100] developed Tamil morphological analyzer using support vector machine. In order to develop the system, they devised to get all possible lexical units and annotate each lexical unit with part of speech tags. The frequency of each word is also required. To get the total frequency of each word, they used two sources: a lexicon corpus along with PoS annotations, and a list of high frequency words along with the frequency score for each word. They obtained annotated lexicon corpus created online by University of Mandras. The corpus had 16 different types of lexical labels, but they reduced into 5 types of lexical labels which consist of verb, adjective, noun, adverb and others. They built high frequency of word lists by crawling Wikipedia and other news websites. Each entry in this list has the word and the word count. The word count was used to calculate the frequency score. They used morphological engine which is a vital component in the system. This engine generates all possible candidates along with their lexical labels by encompassing all grammar

rules regarding morphological construction. Support vector machine classifier is used to select the best suitable candidate. Finally, they performed an experiment to evaluate the classifier with 70,000 manually annotated words. The system achieved the accuracy of 89.73%. The main intention of the system is to tackle the ambiguity, but sometimes it fails when encountering name entities.

3.3 Morphological Analyzer Developed Using Rule Based Approach

3.3.1 Morphological Analyzer for Amharic, Afaan Oromoo and Tigrigna

Gasser [13] developed HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya, for the three major languages widely spoken in Ethiopia by employing finite state transducer (FST). The system performs both analysis and generation of words. The function of the analysis part is analyzing single words and all of the words in a file. For Amharic and Afaan Oromoo, there are two additional analysis functions, which segment input verbs into sequence of morphemes. For Amharic only, there are also two more functions, which convert the input orthographic form to a phonetic form. The function of the generation part is to take a stem or root and a set of grammatical features and produce the possible output.

In order to evaluate the performance of the system, 200 Amharic verbs, 200 Amharic nouns and adjectives and 200 Tigrinya verbs were collected. The results of the system were evaluated by a human reader familiar with the languages. The accuracy of the analyzer was for Amharic verbs 99%, nouns and adjectives 95.5% and for Tigrinya verbs 96%. The generator was expected to generate 10 to 25 verbs. Accordingly, the accuracy of the morphological generator for Amharic was 100% and for Tigrigna 93%. Afaan Oromoo words have not been evaluated because of its great variation in the use of double consonants and vowels.

3.3.2 Morphological Analyzer for Bengali

Priyanka Das and Arjun Das [101] developed Bengali noun morphological analyzer using rule-based approach. To develop Bengali noun morphological analyzer, the nouns in Bengali language were studied and analyzed in a well manner. All the nominal suffixes and the hierarchical structure of their occurrence in Bengali language were identified and studied. The identified nominal suffixes comprise 15 classifiers, 9 case markers and 2 emphatic markers. Based on these suffixes, the hierarchical structure of the concerned suffixes was identified and listed according to the different possibilities of their occurrence forming the nominal words.

After completing linguistic analysis (nominal suffix identification and hierarchical structure), they had done the computational analysis of the nominal inflections. In order to do the computational analysis, they employed the finite state transducer grammar for each and every individual nominal suffix with and without its hierarchical combinations. They developed linguistic resource. They also collected Bengali text file that mainly comprises different news articles and unknown text file from the web. The corpus has passed through the process of tokenization, word-splitting and sort using Perl program. Meanwhile, the list of suffixes is created manually. Based on the created suffixes list, suffix extraction Perl programs were executed. For suffix extraction, 174 Perl programs were written individually to extract 174 nominal suffixes. Finally, they evaluated the total system using the accuracy metrics with unknown Bengali corpus consisting of 6157 unique tokens. The accuracy of the system is 43.96%. The system is unable to solve ambiguities like nominal suffixes and quantifier.

3.3.3 Morphological Analyzer for Af-Somali

Mahdi Yonis [102] developed morphological analyzer for Af-Somali using rule-based approach. They employed finite state transducer technology. For the successful development of morphological analyzer, they created a lexicon of Af-Somali most important part of speech such as verbs, nouns and adjectives separately and alternation rules. Accordingly, separate FSTs were created for lexicon and rules, and then combined into one big FST by applying FST composition operation. The lexicon contains the list of root words and its category separated by a tab. For doing experimentation, they used XFST tool developed by xerox.

They evaluated the performance of the developed morphological analyzer using a dataset which contains 220 tokens, 90 nouns, 120 verbs and 8 adjectives. The results were evaluated by a human reader familiar with the language. Out of 220 words, the analyzer analyzed 77 nominal, 105 verbal and 6 adjectives correctly. Thus, the overall accuracy of the system is 84.1%. The system is only limited to inflectional category.

3.4 Summary

In this chapter, we presented the review of a number of morphological analyzers developed focusing on different techniques. The techniques presented and discussed are rule based and machine learning based approaches. Most of the morphological analyzers developed using machine learning approaches specifically, memory-based learning has shown an interesting

performance. Memory-based learning has the capability to capture the morphological problems of concatenative and non-concatenative languages. For instance, the works of Bosch and Daelemans [3], and Mesfin Abate and Yaregal Assabie [7] have shown the impact of memory-based learning. Our morphological analysis is based on memory-based learning algorithms.

As described in Chapter One, there were attempts made to developing morphological analyzer/synthesizer for Afaan Oromoo. They are based on the traditional and hand-engineered approach which is arduous, time-consuming task and difficult to debug, modify, or adapt to other similar languages. The work made to develop morphological analysis for Afaan Oromoo was limited to verb word class only. The work doesn't clearly describe some of affixes occurring, the different possibilities of their occurrence to form words and doesn't seriously consider the hierarchical structure of the affixes. For example, Afaan Oromoo verb has three major prefixes such as */haa-/*, which is used to indicate jussive mood, */ni-/*, affirmative, and */hin-/*, which is used to indicate negation. Finally, it doesn't test Afaan Oromoo words. The current work tried to model morphological analysis that handles the derivation and inflection of the three Afaan Oromoo word classes: noun, adjective and verb using memory-based learning.

Chapter Four: Design of Afaan Oromoo Morphological Analyzer

4.1 Introduction

This chapter presents the proposed solution of Afaan Oromoo morphological analyzer using inductive machine learning called memory-based learning. Section 4.2 discusses the general architecture of the proposed Afaan Oromoo morphological Analyzer. In this section, the components of the proposed architecture are discussed thoroughly. Section 4.3 discusses the creation of Afaan Oromoo morphological database called OROLEX.

4.2 General Architecture of Afaan Oromoo Morphological Analyzer

The proposed morphological analyzer has two major components: training phase and analysis phase. The training phase contains feature extraction and memory learning and trained model sub-components. The analysis phase contains morpheme identification and morpheme extraction components. The morpheme identification and morpheme extraction components contain other sub-components. Each components of the system with their functions are discussed thoroughly. The general architecture of the system is shown in the Figure 4.1.

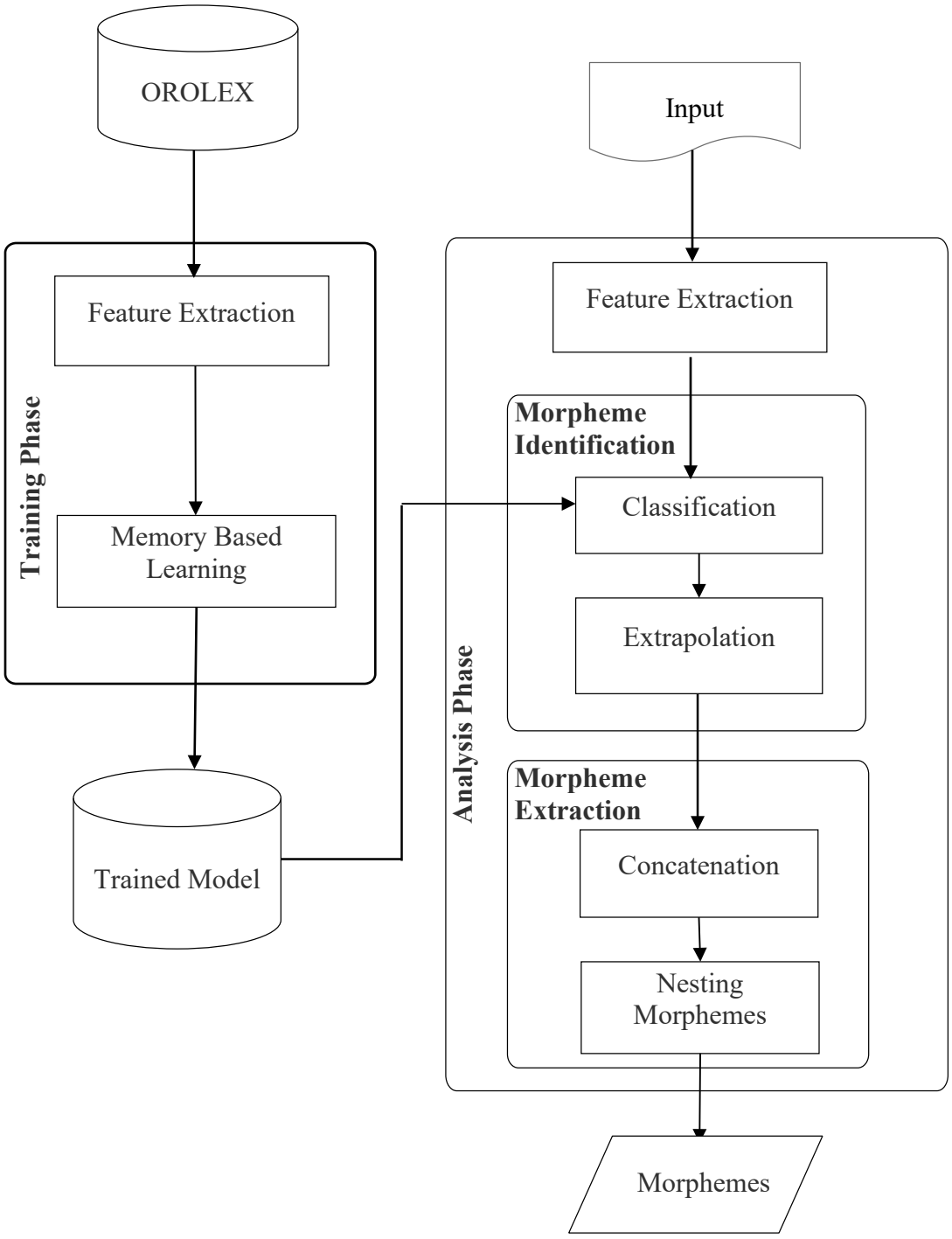

Figure 4.1: Architecture of the proposed Afaan Oromoo morphological analyzer

4.2.1 Training Phase

The training phase comprises necessary components that are used in the process of training the learning component of memory-based learning. The components within this phase operate on the data from the training corpus, OROLEX and produce the trained model.

4.2.1.1 Feature Extraction

Features are properties of a text that are used to provide necessary information associated to a given word annotation and increase the confidence level of predicting instance as a morpheme. Feature extraction is one of the essential components that supply necessary information (features) during the development of the model. In memory-based morphology learning, instance making is an important component. The feature extraction component performs a task of generating required information that is used for the actual training procedure. To do this, we first annotated Afaan Oromoo words from three-word classes namely verbs, nouns, and adjectives and stored in Afaan Oromoo morphological database called OROLEX that will be discussed in Section 4.3. Once the annotated words are stored in the database, instances are extracted automatically from the morphological database based on the concept of windowing method in a fixed length of left and right context.

In contrast to the decomposition of morphological problems into three components by rule-based approach, we reformulate the task of morphological analysis as a one-pass segmentation task, in which a sequence of letters with a focus position is to be classified as indicating a morpheme boundary at that position. This classification approach demands that the number of input features be fixed, hence we cannot use whole words as input. Instead, we converted a word into fixed-sized instances of which the letter at focus position is mapped to a class denoting a morpheme boundary. To generate fixed-sized instances from OROLEX, we adopt the windowing method developed by Sejnowski and Rosenberg [103] which generates fixed-sized snapshots of words. So, in order to extract the instances from the OROLEX, we employed Algorithm 4.1.

Input: Annotated words

1. Define the length of window size (7-1-7).
2. Mark the middle positions of arrays as a focus letter (the focus letter represents where the first letter of a word starts at).
3. Read from the database and push one step forward each character until the right context reached(filled).
4. Put 0(zero) at the class if there are no any special character like @, &, digits and capital letters, next to the characters placed in the focus letter; if any one of those symbols exist put the value as a class (in the last index)
5. Push the previous focus letter to the left and start putting each letter (as in step 3)
6. Go until it finishes that line
7. Go to the next line and repeat the steps 3,4,5,6.

Output: - Instances

Algorithm 4.1: Instance Extraction

Windowing schema converts each word form into as many instances as it has letters. Each instance focuses on one letter, and includes a fixed number of left and right neighbor letters, chosen here to be seven because the average word length in OROLEX is eight. In this way each instance spans fifteen letters, which also happens to be the longest word length in the OROLEX database. For example, applying the windowing method to the example word *'**haamararsiisani**' 'let them treat'* leads to the instances displayed in Table 4.1, listing instances with the appropriate classifications. The morphological analysis of the full word is simply the concatenation of the instance classifications, in which all classifications other than '0' mark morpheme boundaries.

*Table 4.1: Instances with morphological analysis derived from the word '**haamararsiisani**' analyzed as [haa]J[marar]V[siis]6[an]S[i]W*

Instance number	Left Context							Focus Letter	Right Context							Class
1	-	-	-	-	-	-	-	**h**	a	a	m	a	r	a	r	0
2	-	-	-	-	-	-	h	**a**	a	m	a	r	a	r	s	0
3	-	-	-	-	-	h	a	**a**	m	a	r	a	r	s	i	J
4	-	-	-	-	h	a	a	**m**	a	r	a	r	s	i	i	0
5	-	-	-	h	a	a	m	**a**	r	a	r	s	i	i	s	0
6	-	-	h	a	a	m	a	**r**	a	r	s	i	i	s	a	0
7	-	h	a	a	m	a	r	**a**	r	s	i	i	s	a	n	0
8	h	a	a	m	a	r	a	**r**	s	i	i	s	a	n	i	V
9	a	a	m	a	r	a	r	**s**	i	i	s	a	n	i	-	0
10	a	m	a	r	a	r	s	**i**	i	s	a	n	i	-	-	0
11	m	a	r	a	r	s	i	**i**	s	a	n	i	-	-	-	0
12	a	r	a	r	s	i	i	**s**	a	n	i	-	-	-	-	6
13	r	a	r	s	i	i	s	**a**	n	i	-	-	-	-	-	0
14	a	r	s	i	i	s	a	**n**	i	-	-	-	-	-	-	S
15	r	s	i	i	s	a	n	**i**	-	-	-	-	-	-	-	W

The hyphen mark (-) is used as a filler symbol which shows there is no character at that position. To illustrate the construction of instances, Table 4.1 displays the 15 instances derived from the Afaan Oromoo word '***haamararsiisani***' and their associated classes. In this example, the classification of the third instance is 'J' which means that the morpheme ending in ***a*** is a prefix which represents jussive. The second morpheme, ***marar*** has a class 'V'. This tag indicates that the morpheme is the root of a verb. The class of the twelves instance is '6', which indicates that ***siis*** morpheme is a derived stem from root verb. The class of the fourteenth instance is S, which signifies number agreement and the class of the fifteenth instance is W, which implies allomorphic perfective aspect. For the classes bearing no morphological identification, we assigned a zero (0) label.

4.2.1.2 Memory Based Learning

Memory-based learning is one of the symbolic supervised machine learning methods that learn from instances. Our main concern with the learning component of memory-based learning system is to build a trained model. The learning component of memory-based learning is the component designed to learn an instance and then build a trained model that will be discussed in Section 4.2.1.3. The main goal of the learning component of memory-based learning is to add training examples to a memory. The examples are represented as a fixed-length vector of n feature-value pairs, and the information field containing the classification of that particular feature-value vector as shown in Table 4.1. During training, a set of examples, the training instances are presented to the memory-based learning component and those instances are added to the memory (the instance base or case base) without abstraction, selection, or restructuring. The procedure of training the model is done by the learning component of memory-based learning. The memory-based learning algorithms employed in our work is IB1 and IGTREE algorithms described in Chapter 2 in order to perform learning instances. Therefore, the product of the learning component is used as a basis for mapping input to output in the morphological analysis phase.

4.2.1.3 Trained Model

The trained model is what we look for as part of the training phase and it is the final output of the learning process. Since we used memory learning algorithms in the training process and the model is trained with the instances created from Afaan Oromoo words, we can call the trained model as *Afaan Oromoo MBL Model*. The model contains a set of instances called instance base or case base which is extracted from OROLEX using windowing scheme and fed to memory-based learning component. The model is the main component that plays a crucial role in suppling trained instances information during analysis.

4.2.2 Morphological Analysis

In the morphological analysis phase, the performance component of a memory-based learning system, the trained model is used as a basis for mapping input to output. This phase comprises feature extraction, morpheme identification and morpheme extraction. The description of each components and their sub-components are provided thoroughly.

4.2.2.1 Feature Extraction

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. It is the transformation of original data to a data set which contains the most discriminatory information. The learning component of memory-based learning learns instances by storing training data into the memory, trained model. So, the instances stored in trained model is later used in analysis phase in order to map the input to the output. The input here is the features or instances extracted from a word in analysis phase. When new words to be classified are given to the system, the feature extraction component will deconstruct the word in a fixed-length of instance. The features are extracted as the process of feature extraction described in Section 4.2.1.1.

4.2.2.2 Morpheme Identification

Morpheme identification is used to classify and extrapolate the class of new instances. The morpheme identification component also contains two sub-components: classification and extrapolation.

Classification

The classification component accepts previously unseen test instance from feature extraction component to perform classification problem. After accepting those unseen test instances, it computes similarity between the new instance, unseen test instance and all examples in the memory, trained model, using distance metric based on memory-based learning algorithms. The result of the similarity computation performed passes to the extrapolation component to further do more morpheme identification.

Extrapolation

The extrapolation is another sub-component of morpheme identification. The fundamental task of extrapolation is that, it assigns the most frequent category within the found set of most similar example(s) (the k-nearest neighbors) as the category of the new test example. For the new test instances presented to the system, if there is an exact match on the trained model, it extrapolates the class of that instances to the new instances. In order to assign the class to the new test instance, the extrapolation tries to find the instances which have similar structure and pattern, and assigns the class of that instance to it. For example, if previously unseen instance

as shown in Figure 4.2 is offered to the extrapolation, it tries to find the instances which most closely resemble the unseen instance from the trained instances.

7	6	5	4	3	2	1	Focus	1	2	3	4	5	6	7	Class
h,	a,	a,	s,	a,	r,	a,	r,	s,	i,	i,	s,	a,	n,	i	?

Figure 4.2: Assigning a class to new instance

Most of the structure and patterns of the instance presented in Figure 4.2 are similar to the instances extracted from the '**haamararsiisani**' shown in Table 4.1. So, the extrapolation might assign the class 'V' to the instance in Figure 4.2 because they share almost all features of the left context and right context except the left context 4. 'V' is the eighth class of the instances extracted from the verb '**haamararsiisani**'. If there are more related classes found in a set of similar examples (the k-nearest neighbors), for a given new instance, a tie breaking resolution method is used.

4.2.2.3 Morpheme Extraction

This process is done after the appropriate morphemes are identified during morpheme identification. In morpheme extraction, reconstruction of individual instances into a meaningful (to their original word form) and insertions of identified morphemes in their segmentation point are performed.

Concatenation

The main goal of concatenation is to concatenate morphemic segments and insert the morphemes identified during morpheme identification. Five non-null classes (class bearing morphological identification) and ten null classes (classes bearing no morphological identification) morphemic segments are identified in morpheme identification from the word '**haamararsiisani**'. Only instances of the non-null classes morpheme segments are concatenated to show morphemes in a word as shown in Figure 4.3.

Figure 4.3: Concatenating morphemes of the word 'haamararsiisani'.

As we observe from Figure 4.3, the morpheme segments of five morphemes are concatenated together to show morphemes in a verb '*haamararsiisani*'. The first morpheme is, *'haa-'*, which shows jussive, the second morpheme is, *'marar-'*, which shows the root of a verb, the third morpheme is, *'-siis-'*, which shows causative derived stem morpheme which is derived from root verb, the fourth morpheme is, *'-an-'*, which shows the plurality of number agreement, and the last morpheme is, *'-i'*, which shows perfective aspect variant.

Nesting Morphemes

The nesting morphemes is a subcomponent of morpheme identification. Nesting morpheme subcomponent will reconstruct the whole morphemic segmentation tied together in concatenation into a meaningful word properly. For example, the morpheme identification, concatenation of morphemic segments and nesting of the whole morpheme of the word *'haamararsiisani'* is illustrated in Figure 4.4.

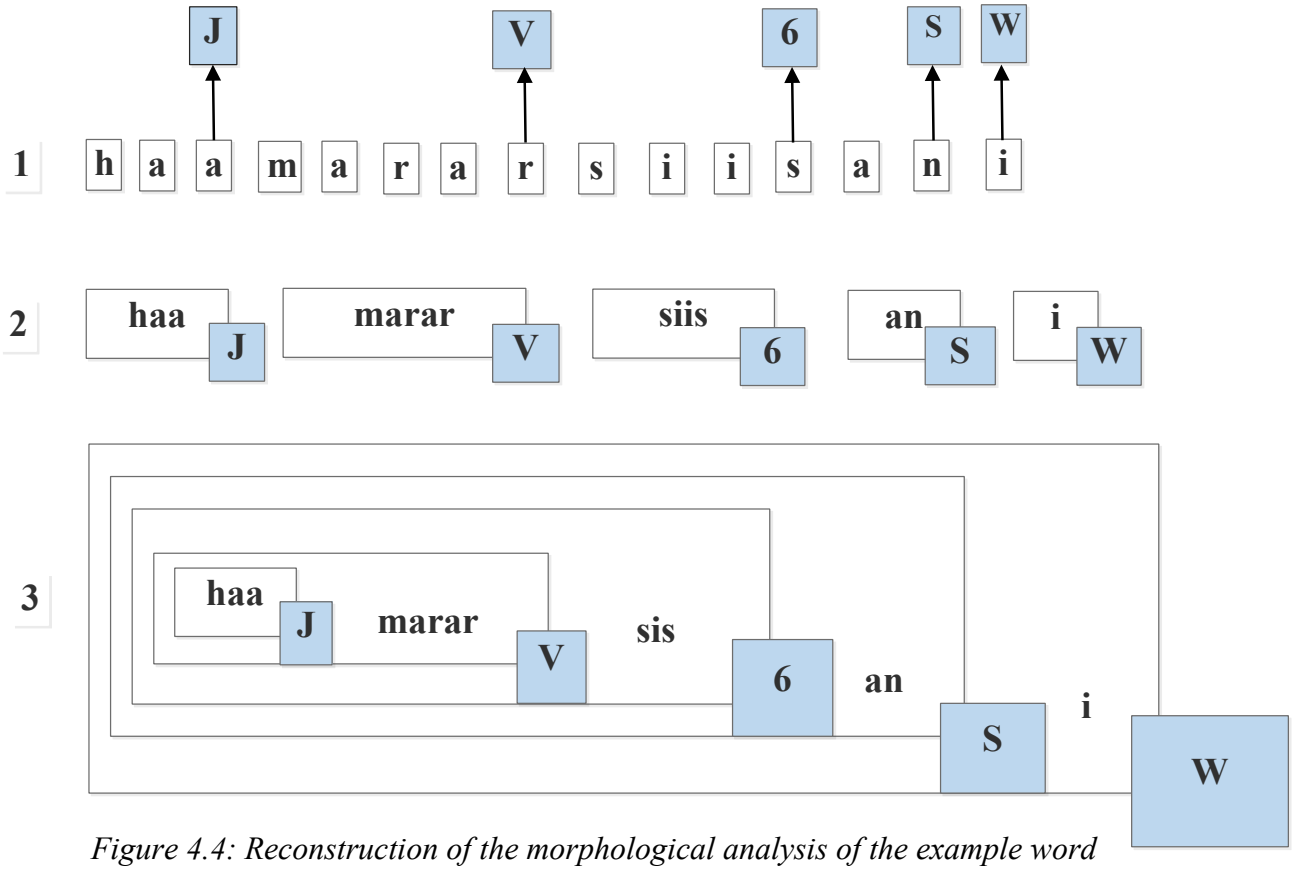

Figure 4.4: Reconstruction of the morphological analysis of the example word haamararsiisani

As it can be seen, the reconstruction of the morphological analysis of the example word ***'haamararsiisani'*** in Figure 4.4, five non-null classes (the classes bearing morphological

identification) are predicted in classification step. In the second step, the letters of the two identified morphemic segments are concatenated, and finally the morphemes are nested.

4.3 Afaan Oromoo Morphological Database (OROLEX)

All supervised machine learning models need annotated or labeled data. In less resourced languages such as Afaan Oromoo, there is no formally well-developed morphological database or annotated words in order to facilitate the study carried out at the word level. Morphological database is a must to develop a powerful memory based morphological analysis. In order to carry out our study, we are faced with the problem of having a morphological database of Afaan Oromoo. To tackle the problem, there is a need of preparing morphologically annotated word lists. Therefore, the following tasks were identified and performed to prepare annotated datasets used for the training and test purpose:

- Identifying inflected Afaan Oromoo words.
- Segmenting the words into prefix, stem, suffix based on the affixes of a word class.
- Putting boundary marker between each segment.
- Describing the representation of each marker with grammatical description.

By following the steps of the listed tasks, we collected different Afaan Oromoo words from three-word classes of the language, namely: nouns, adjectives and verbs. We annotated those words and stored them in Afaan Oromoo morphological database called OROLEX. OROLEX is a lexical database of Afaan Oromoo word forms and contains 33 grammatical descriptions of morphological analysis. The annotation was cross checked by a language expert at Oromo research center, department of language studies. We used different Afaan Oromoo data sources such as textbooks, dictionaries, research papers and websites for the annotation.

A. Annotation for Nouns

Afaan Oromoo nouns have zero prefix and eight suffix slots. The suffixes are attached in a particular order to the stem. Not all suffixes are attached to all stems. The suffixes that are attached to the stems are derivation, direct object (accusative case), plurality, definiteness, nominative case, dative case, instrumental case and focus. Table 4.2 shows the order in which suffixes are attached to noun stem.

Table 4.2: The suffixes attached to nouns in their slot order

Stem	Der	Plu	Def	Nom	Acc	Dat	Inst	Foc
Nam- Durb-	-ummaa	-oota =	-ich-, -ittii =	-ni, -i =	-a =	-a(f) =	-n =	-tu =

Where Stem, Der, Plu, Def, Nom, Acc, Dat, Inst and Foc indicate the root of noun, derivation, plurality, definiteness, nominative, accusative, dative, instrumental and focus marker respectively. The '=' sign shows that null value can be possible in that position. In order to create OROLEX, a collection of noun words should be annotated manually based on its grammatical function. The annotation is done according to the stem-suffix ([S]-[S]) structure as illustrated in Table 4.3.

Table 4.3: Nouns annotation

Word form	Stem	Suffix
namoota	nam- [N]	-oota [M]
fardi	fard- [N]	-i [E]
fardaaf	fard- [N]	-a-[L], -af-[O]
garbicha	garb- [N]	-ich-[B], -a[L]
garbittii	garb- [N]	-ittii[X]

Where M, E, L, O, B, and X indicate end of plural, nominative case, accusative case, dative case, masculine definiteness, and feminine definiteness markers respectively and N indicates the stem of nouns.

B. Annotation for Adjectives

Adjectives have one prefix and nine suffixes in their proper slots. All of the suffixes that are attached to the stem of adjectives are similar, except gender marker not found in noun. The way suffixes are attached to adjective stems are similar with noun except few affixes. The affixes attached to adjective stems are reduplication of stem as a prefix and derivation, gender,

plurality, definiteness, nominative case, accusative case, dative case, instrumental case and focus marker as a suffix. Table 4.4 shows the order in which affixes are attached to adjective stem.

Table 4.4: The affixes attached to adjectives in their slot order

Redup	Stem	Der	Gen	Plu	Def	Nom	Acc	Dat	Inst	Foc
qa- =	qall-	-ina =	-aa, -tuu, -oo =	-oota, -ota =	-ich-, -ittii =	-ni, -i =	-a	-af =	-n =	-tu =

Where Redup, Stem, Der, Gen, Plu, Def, Nom, Acc, Dat, Inst and Foc indicate stem reduplication, stem, derivation, gender, plurality, definiteness, nominative, accusative, dative, instrumental and focus respectively. The '=' sign shows that null value can be possible in that position. In order to create OROLEX, a collection of adjective words should be annotated manually based on its grammatical function. The annotation is done according to the prefix-stem-suffix ([P]-[S]-[S]) structure as illustrated in Table 4.5.

Table 4.5: Adjectives annotation

Word form	Prefix	Stem	Suffix
qaqallaa	qa-[R]	-qall-[A]	-aa[&]
qalloo	-	qall-[A]	-oo[@]
ballaa	-	ball-[A]	-aa[&]

Where R, A, & and @ indicate the end of reduplication of stem, the stem of adjectives, masculine gender, and feminine gender marker respectively.

C. Annotation for Verbs

As discussed in Chapter 2, Afaan Oromoo verbs carry different kind of information on their affixes like person, gender, number and others. Basically, Afaan Oromoo verbs have base stems and derived stems. The base stems of verbs are the infinitive (verbal noun) forms ending with morpheme *-uu* as in *mur-uu*-VN 'to cut/cutting'. The latter one, derived stems are formed

by several markers such as passive, causative, the autobenefactive or the middle voice, and intensive affixed to the verb roots or other derived stems. The affixes are attached to both base stem and derived stem. The affixes that are attached to verb stems are negative/jussive and affirmative as a prefix and derivation, subject, aspect-mood, negative marker as a suffix. Table 4.6 shows the order in which affixes are attached to verb base stems.

Table 4.6: The affixes attached to verbs in their slot order

Neg/ Juss	Aff	Stem	Der _Stem	Der	sbj	am	Neg
hin- haa- =	ni- =	deem-	-sis- =	-noo =	-t- =	-e	-n =

Where Neg/Juss, Aff, Stem, Der_Stem, Der, sbj, am and Neg indicate negation/jussive, affirmative, base stem, derived stem, derivation, subject, aspect-mood and negation respectively. The '=' sign shows that null value can be possible in that position. This affix order is used to annotate verbs correctly. The annotation of verb is done according to the prefix-stem-suffix ([P]-[S]-[S]) structure as illustrated in Table 4.7.

Table 4.7: Verbs annotation

Wordform	prefix	stems	suffix
deeme	-	deem- [V]	-e [P]
deemsise	-	deem- [V]	-sis- [5], -e [P]
deemtani	-	deem- [V]	-t- [2], -an- [S], -i [W]
nideema	ni- [T]	deem- [V]	-a [I]

Where V, P, S, T, W, I, 2, and 5 indicate base stem, normal perfective aspect marker, number marker, affirmative marker, allomorphic variation of perfective aspect marker occurring with 2PL and 3PL, imperfective aspect marker, person marker and causative derived stem marker respectively. Table 4.7 shows the annotation of agreement between the subject and verbs by the markers affixed on verb stem. The perfective aspect doesn't show the person agreement on first person singular, third person singular masculine, first person plural, and third person plural and number agreement on first person singular, second person singular, third person singular feminine and masculine. The imperfective aspect doesn't show the person agreement

on first person singular, third person masculine, and third person plural and all number agreement. For the plural person types (subjects), the verb is marked for agreement separating person and number unlike the singular subjects whose agreement morpheme represents both person and number features of agreement.

D. Annotation for Derivations

Derivation is the combination of a word stem with a grammatical morpheme which may usually change a word class. This kind of morphology is also available in Afaan Oromoo as in other languages. Table 4.8 shows the annotation of Afaan Oromoo words derived from other word classes by attaching suffixes.

Table 4.8: Annotation of words derived from noun, adjective and verbs

Word form	**Stems**	**Suffix**
namummaa	nam-[N]	-ummaa [D]
filmaata	fil-[V]	-maata [D]
filatnoo	fil-[V]	-at- [4], -noo[D]

Where N, V, 4 and D indicate the noun stem, verb base stem, derived verb stem and derivation marker respectively.

4.4 Summary

In this chapter, basically we have discussed the two main components of the proposed architecture of Afaan Oromoo morphological analyzer. The two main components are: training phase and analysis phase. The training phase takes an input from the morphological database and produces the trained model. The trained model offers basic information for the analysis phase in order to carry out the analysis of morphology. The analysis phase produces a collection of morphemes of words. We also discussed the development of Afaan Oromoo morphological database called OROLEX. It was made to fill the gap of morphology of Afaan Oromoo we have faced with during developing Afaan Oromoo morphological analyzer. OROLEX consists of grammatical description of manually annotated Afaan Oromoo verb, noun and adjective words. During annotation, we set out the order in which affixes are attached to a root/stem of any words based on the rule of the language. We put a marker which marks the boundary between the morphemes in a word.

Chapter Five: Experimentation

5.1 Introduction

This chapter is fully devoted to the discussion of the experimentation aspects of our system, Afaan Oromoo Morphological Analysis. In Section 5.2, development environment is discussed. Section 5.3 deals with the data collection. Section 5.4 presents the experimentation process: training and testing. Section 5.5 presents the performance evaluation and the results obtained from the experiments in detail. The last section, Section 5.6, presents the discussion.

5.2 Development Environment

The prototype was developed using C++ and Java programming language. TiMBL, a memory-based learning tool developed using C++ programming language is used to test the feasibility of the model. We used ucto 0.3.5 for text tokenization, geany2.1 a C++ editor for code writing, and GCC 4.6.4 compiler for compiling the source. We also used NetBeans IDE 8.0 for automatic feature extraction component.

The morphological analyzer is developed and tested on a computer with the following specifications:

- Ubuntu 18.10 LTS operating system
- Intel® Core™ i5 2.60Ghz
- Hard disk size 750GB and
- RAM size 8GB

5.3 The Corpus

Afaan Oromoo has a shortage of existing annotated data for any work of its natural language processing applications. This shortage holds true for morphology too. As far as we know, there is no official and freely available Afaan Oromoo words morphological database. As described in Chapter 4, we have developed OROLEX in order to fix the problem of lack of morphological database. We split OROLEX corpus into training set and test set. The performance of the model is evaluated by the dataset taken from the OROLEX. The corpus contains 2,270 tagged words. Out of 2,270 tagged words, nouns are 438, adjectives are 17 and verbs are 1,815 respectively. The number of verbs in the OROLEX is more than the number of nouns and adjectives because verbs have more inflections and derivations than nouns and

adjectives. Since Afaan Oromoo verbs are the most inflected words, our focus is more on a verb. When we look at Afaan Oromoo nouns and adjectives, comparatively, they have more or less the same number of suffixes except that reduplication of root prefix and gender suffix appears in adjective. So, again our focus is more of on nouns than adjectives. The total number of instances or features extracted from those words is 17,386: 3,072 instances from nouns, 114 instances from adjectives and 14,200 instances from verbs. Within these instances, 33 different class labels occur.

5.4 Training and Test Experiment

TiMBL allows us to use different formats for training and test dataset because it has the capability to guess the type of format in most cases. For the experiment, the dataset is organized with the well-known format called C4.5. The experiment started by executing TiMBL with the training dataset, *AfaanOromoo.train* and test dataset, *AfaanOromoo.test* files as an argument. Upon the completion of the execution, a new file similar to the input test file is created except that an extra comma-separated column is added with the class predicted. The name of the file provides information about the MBL algorithms and metrics used in the experiment. During the execution of the experiment, we performed training and test together at one time even if it is possible to execute it separately with TiMBL. Right after the execution, the output which consists of three phases is displayed. The first phase holds information about the training data analysis phase. The second phase contains the information of building multi index on training data. The last phase contains the information of how all training items are stored in a memory and the trained classifier which is applied to the test set. The summary information of the three phases is illustrated in Figure 5.1.

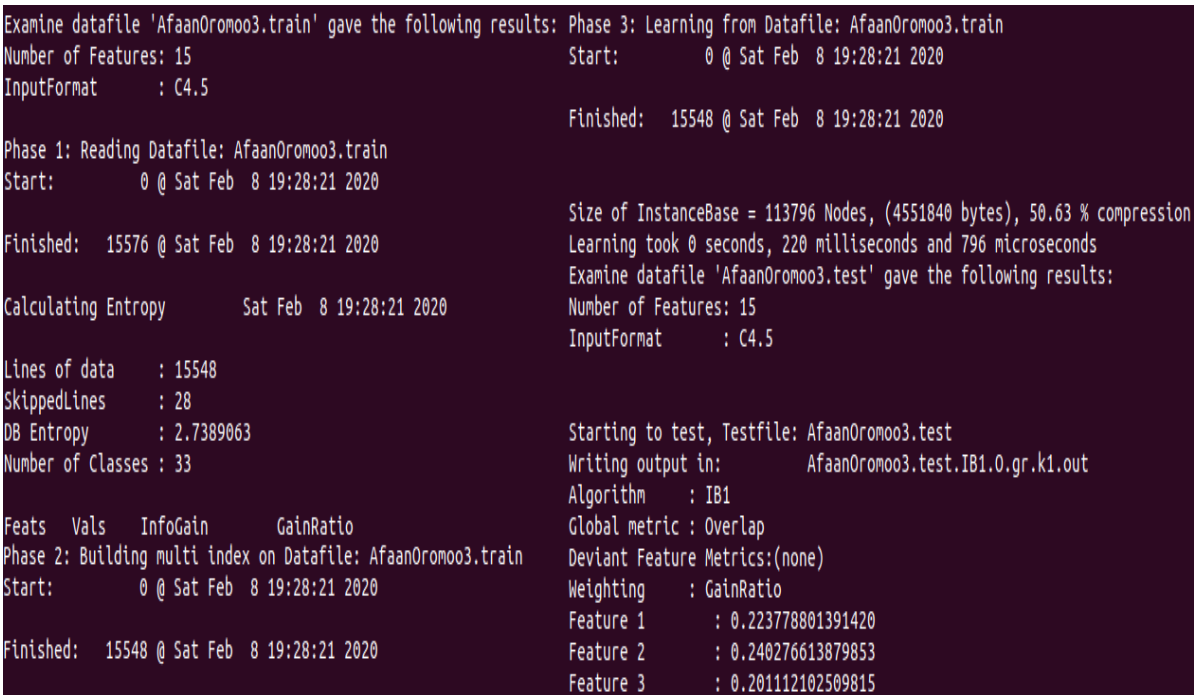

Figure 5.1 Training and testing experiment

5.5 Performance Evaluation

The efficiency of machine learning is determined by the proper evaluation methods used. In order to evaluate the performance of our morphology learning model, we employed different model performance evaluation techniques, such as k-fold cross validation evaluation techniques and other evaluation metrics that can be derived from confusion matrix in order to evaluate the predictive performance of our model.

5.5.1 K-fold Cross Validation

For our experiment, we have used k-fold cross validation because of its successful achievement in classification assessment. We selected the value of k, 10 (10-fold) cross validation based on the available data. In this case, the dataset is arbitrarily divided into 10 groups or folds. Subsequently 10 iterations of training and testing are performed such that within each iteration a different fold of the data is held-out for testing while the remaining folds are merged for learning. The results of 10 tests are averaged together to show the generalization performance of the model. In this way, we trained the memory-based model with 90% (k-1) of the dataset and used the remaining 10% (k-9) of the dataset for testing iteratively until the process is over.

Practically, we performed 10 different experiments on our dataset (2,270) by dividing them into 10 equally sized segments. All the instances in the dataset are eventually used for both training and testing consecutively. In each experiment, the number of instances extracted from both training and testing words vary because there exist unequal number of word length. For instance, in experiment 1, the number of instances extracted from both learning dataset and testing dataset is 15,664 instances and 1,722 instances respectively. None of the test datasets are present in the training datasets during the process of our experiment except the instances that are extracted from the words in both datasets are identical. The parameter settings of both algorithms and feature selection methods are considered during the experimentation. The experiments are performed with the default parameter settings and optimized parameters of the learning algorithms (IB1 and IGTREE). The default parameter settings are k-nearest neighbor (k=1), gain ratio, overlap distance metric, normal majority voting because both algorithms have the same default parameter settings. The parameter settings, default parameter settings and their corresponding descriptions are depicted in Table 5.1.

Table 5.1: The default parameter settings

Parameter settings	Value	Description
Nearest neighbor	k=1	The number of nearest neighbors
Feature weighting	-w1	Gain ratio
Distance metrics	-mO	Overlap
Class voting	-dZ	Normal majority voting

Accordingly, 10 separate experiments are performed with our dataset. The randomly partitioned dataset is trained and tested on IB1 and IGTREE algorithms. The accuracy of the trained model is predicted using test dataset. Accuracy in classification problem is calculated as the total number of correct predictions of new instances made by the model divided by the total number of test instances in the dataset.

$$\text{Accuracy} = \frac{Total\ number\ of\ correctly\ predicted\ test\ instances}{Total\ number\ of\ test\ instances\ in\ the\ dataset} \quad (9)$$

The accuracy of both IB1 and IGTREE with default parameter settings using 10-FCV are depicted in Table 5.2.

Table 5.2: The results of IB1 and IGTREE algorithm with default parameter settings

Experiments	Training Instances	Test Instances	Accuracy (%)	
			IB1	IGTREE
1	15664	1722	98.61	95.01
2	15714	1672	98.92	92.28
3	15548	1838	98.26	90.64
4	15646	1740	99.37	95.98
5	15694	1692	99.41	96.10
6	15601	1785	97.87	92.44
7	15590	1796	99.61	94.15
8	15683	1703	98.65	94.95
9	15653	1733	98.15	91.46
10	15681	1705	99.18	91.38
Average	15647	1739	98.80	93.44

The random partitioning of 10-FCV procedure creates the imbalance of training set and test set problem. When we see Table 5.2, we understand that there exists the disparity of word length in both training set and test set during corpus splitting. The result obtained using 10-FCV shows that IB1 version is found to be quite better than IGTREE in terms of generalization accuracy. The obtained accuracy of IGTREE is not as good as IB1 because it invests more time in organizing the instance base to obtain simplified and faster processing during classification.

In machine learning algorithms, often all features of a particular task don't have equal importance to the model performance. Thus, a good feature selection determines the performance of one model built for a particular task. In order to influence the performance of our model, we selected the features of instances that have high accuracy using forward selection method. We used forward selection method for computational reason. We started the experiment of feature selection by computing the accuracy of each feature independently as only feature while ignoring the other features as being unwanted features. The features with highest accuracy were selected. Then, the accuracy of sets of features with highest accuracy were computed until no more accuracy increase is reported. Accordingly, from 15 features of

our morphological analysis, 13 features with highest accuracy were selected and run on IB1 algorithm with default parameter settings. The features that are ignored from the features of instances during learning instances and classifying the belongingness of instances to classes by the model were feature 2 and 15 because they are with low accuracy. Again, the same process is applied on IGTREE algorithm with default parameter settings. With respect to IGTREE, from 15 features, 6 features were selected as well performed features. Feature 1, 2, 3, 4, 6, 12, 13, 14 and 15 were ignored. The combination of the accuracy of features that achieved highest accuracy were feature 5, 7, 8, 9, 10 and 11. The accuracy of those combination of selected features is shown in Table 5.3.

Table 5.3: The results of IB1 and IGTREE with default parameter settings and feature selection

Experiments	Training Instances	Test Instances	Accuracy (%)	
			IB1	IGTREE
1	15664	1722	98.61	96.23
2	15714	1672	99.16	93.60
3	15548	1838	98.15	90.10
4	15646	1740	99.43	96.44
5	15694	1692	99.41	96.63
6	15601	1785	97.87	94.62
7	15590	1796	99.55	94.27
8	15683	1703	98.83	95.36
9	15653	1733	97.98	90.08
10	15681	1705	99.18	94.02
Average	15647	1739	98.82	94.14

As we observe, Table 5.3, the results achieved in combination with both feature selection and default parameter settings is slightly better than the results obtained with default settings on both IB1 and IGTREE algorithms.

In addition to the experiments done on both algorithms with default parameter settings and the combination of default parameter setting with feature selection, we also performed other experiments using 10-FCV on IB1 and IGTREE by tuning their parameter settings. It is

important to adjust the parameters of algorithms to maximize their performance because a good choice of the parameter settings can have a large effect on the accuracy of IB1 and IGREE. The default parameter settings of both IB1 and IGTREE are by no means certain that they will be ideal parameter settings for our morphological analysis. Hence, after a number of parameter adjustment, we identified the parameters that have great effect on the performance of the model. In our experiment, the optimized parameters of IB1 are modified value difference (MVD), No weighting, i.e., all features have the same importance (weight=1), inverse distance (ID) and k-nearest neighbor value (k=1) which outperforms the default parameter settings. As the success of IGTREE algorithm is determined by only a good judgement of feature relevance ordering, we have tried to optimize the feature weighting parameter. Optimizing the other parameters of IGTREE doesn't have any effect on the performance. Finally, we found that the accuracy obtained with information gain feature weighting outperforms the accuracy obtained with default parameter of feature weighting. The generalization accuracy obtained with optimized parameters of IB1 and IGTREE are depicted in Table 5.4.

Table 5.4: The results of IB1 and IGTREE parameter optimization

Experiments	Training Instances	Test Instances	Accuracy (%)	
			IB1	IGTREE
1	15664	1722	98.49	95.06
2	15714	1672	99.46	93.00
3	15548	1838	98.91	90.86
4	15646	1740	98.85	95.86
5	15694	1692	99.35	96.04
6	15601	1785	98.66	93.50
7	15590	1796	99.50	93.88
8	15683	1703	98.36	95.30
9	15653	1733	98.21	91.81
10	15681	1705	98.42	91.32
Average	15647	1739	98.82	93.66

As we observe, Table 5.4, the overall accuracy shows that tuning the IB1 parameter settings has shown a 0.02% accuracy improvement over its default parameter setting, while tuning the IGTREE parameter settings has shown 0.22% accuracy improvement over its default parameter settings.

In order to enhance further the accuracy of IB1, in addition to examining feature selection with default parameter settings of the algorithms and tuning their parameters, we did another new experiment. In the experiment, we interleaved combined feature selection and parameter optimization to see the effects of their interaction together in influencing the accuracy. Table 5.5 shows the overall accuracy of IB1 and IGTREE obtained by interleaving combined features and parameter tuning.

Table 5.5: The results of interleaved of combined features and parameter optimization

Experiments	Training Instances	Test Instances	Accuracy (%)	
			IB1	IGTREE
1	15664	1722	98.55	96.23
2	15714	1672	99.40	93.60
3	15548	1838	98.91	90.10
4	15646	1740	98.91	96.44
5	15694	1692	99.05	96.63
6	15601	1785	98.77	94.62
7	15590	1796	99.61	94.27
8	15683	1703	98.59	95.36
9	15653	1733	98.33	90.08
10	15681	1705	98.48	94.02
Average	15647	1739	98.86	94.14

When we observe Table 5.5, the overall accuracy obtained with combined feature selection and parameter optimization lead to a slight accuracy improvement. The reason that improved the generalization accuracy of IB1 with interleaved combined feature selection and parameter optimization is the best selection of features and optimal parameters.

There is a general question that needs to be answered concerning the classifier model: which classification algorithm achieves the highest classification accuracy? To answer this question,

we compared the generalization accuracy achieved with both IB1 and IGTREE after doing different experiments on them using the same dataset.

Finally, there is an interesting trade-off between generalization accuracy and efficiency that IB1 usually leads to more accuracy at the cost of memory and slower computation than IGTREE. However, Table 5.6 shows the results of 10-FCV experiments comparing generalization accuracy, storage requirements and speed of the two algorithms on the average of 15,647 instances of training set and 1,739 instances of test set.

Table 5.6: Comparison of IB1 and IGTREE

Algorithms	Experiments	Instance Base (Bytes)	Learning time (sec)	Test time (sec)	Compression (%)	Accuracy (%)
IB1	Default settings	4566432	0.1493	0.1743	50.61	98.80
	Feature selection	3688000	0.1445	0.1349	51.31	98.82
	Parameter tuning	4566432	0.1495	0.2330	50.61	98.82
	Interleaved	3688000	0.2205	0.2761	51.31	98.86
IGTREE	Default settings	40620	0.2711	0.0165	99.56	93.44
	Feature selection	32668	0.1995	0.0156	97.83	94.14
	Parameter tuning	35680	0.2768	0.0170	99.61	93.66
	Interleaved	32668	0.1986	0.0158	97.83	94.14

When we observe Table 5.6, the generalization accuracy obtained with four scenarios from both algorithms are slightly close to each other. IB1 achieved a maximum generalization accuracy of 98.86% and minimum accuracy of 98.80%, while IGTREE achieved a maximum accuracy of 94.14% and minimum accuracy of 93.44%.

5.5.2 Confusion Matrix

A confusion matrix is a table that can be produced from a classifier and which associates the predicted class with the real class of the test items given. In our experiment, there exists a confusion matrix of size 33 x 33 associated with a classifier that shows the predicted classification and actual classification, where 33 is the number of different classes. Table 5.7 shows the confusion matrix of IB1 with default parameter settings using 10-FCV.

Table 5.7: Confusion matrix of IGTREE with default parameter setting

Class	0	V	I	P	Q	U	R	1	2	Y	S	W	T	3	4	J
0	**9521**	123	0	6	24	61	0	1	1	0	9	0	4	7	0	29
V	225	**1501**	0	0	0	0	0	0	0	0	0	0	0	4	0	0
I	0	0	**311**	0	0	0	0	0	0	0	0	0	0	0	0	0
P	1	0	0	**407**	0	4	0	0	0	0	0	0	0	0	0	0
Q	2	0	0	0	**173**	0	0	0	0	0	0	0	0	0	0	0
U	2	0	0	0	0	**520**	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	**18**	0	0	0	0	0	0	0	0	0
1	6	0	0	0	0	0	0	**79**	0	0	0	0	0	0	0	0
2	5	0	0	0	0	0	0	0	**417**	0	0	0	0	0	0	0
Y	0	0	2	0	0	0	0	0	0	**70**	0	0	0	0	0	0
S	0	2	0	0	0	0	0	0	0	0	**269**	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	**271**	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	**532**	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	**288**	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**197**	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**222**
K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	114	81	0	0	0	0	0	6	2	0	0	0	0	6	1	0
L	25	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0
E	3	0	0	0	7	0	0	0	0	2	0	2	0	0	0	0
O	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
@	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
Z	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
&	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Class	K	6	N	L	E	O	B	M	X	D	G	A	@	Z	5	&	C
0	3	3	72	46	1	48	0	0	0	0	0	3	0	0	0	0	0
V	0	0	66	0	0	22	0	0	0	0	0	2	0	0	0	0	0
I	0	0	0	19	0	0	0	0	0	1	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
1	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	**267**	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	**296**	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
N	4	6	**195**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	**216**	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	**41**	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	**148**	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	**39**	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	**60**	0	0	0	0	0	0	0	0	0
X	0	0	0	0	0	0	0	0	**11**	1	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	1	**27**	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	**59**	0	0	0	0	0	0
A	0	0	0	0	0	0	0	0	0	0	0	**12**	0	0	0	0	0
@	0	0	0	0	0	0	0	0	0	0	0	0	**6**	0	0	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	**2**	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**61**	0	0
&	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**6**	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

For every cell in the matrix, as it can be seen from Table 5.7, the row represents the predicted class as the classification model produced and the column represents the original class. The diagonal in the matrix which is the darkest and bold, represents the ideal case in which the instance was correctly classified. For example, 9521, 1501, 311, 407, 173, 520, 18, 79, 417, 70, 269, 271, 532, 288, 197, 222, 267, 296, 195, 216, 41, 148, 39, 60, 11, 27, 59, 12, 6, 2, 61, 6, and 0 are the number of instances correctly classified diagonally. All cells outside the diagonal illustrate errors of one class being mistaken for another because according to confusion matrix procedure, all the off-diagonal cells represent miss classified instances. For instance, the I class (-a which represents imperfective aspect) is mis-predicted thirty-three times as class L (-a which represents accusative case). Additionally, there is also relatively high degree of mutual misprediction errors between some classes such as root verb (V), null class (0), and noun stem (N) accounting 225, 123, and 114 respectively.

In confusion matrix, the correctness of a classification can be evaluated by computing the number of correctly recognized class examples (true positives), the number of correctly recognized examples that do not belong to the class (true negatives), and examples that either were incorrectly assigned to the class (false positives) or that were not recognized as class examples (false negatives). Based on this information, it is possible to compute the different score values for each class in the datasets. The different metrics used to compute the score values per class are true positive (TP), false positive (FP), true negative (TN), false negative (FN), precision, recall (TPR), false positive rate (FPR), F-score and area under curve (AUC). Table 5.8 shows the detail results.

Table 5.8: The results of TP, FP, TN, FN, precision, recall FPR, F-score and AUC on IB1 with default parameter settings

Scores per Value Class:

class	TP	FP	TN	FN	precision	recall(TPR)	FPR	F-score	AUC
0	1113	0	710	15	1.00000	0.98670	0.00000	0.99331	0.99335
V	175	18	1638	7	0.90674	0.96154	0.01087	0.93333	0.97533
I	34	2	1801	1	0.94444	0.97143	0.00111	0.95775	0.98516
P	43	0	1794	1	1.00000	0.97727	0.00000	0.98851	0.98864
Q	21	1	1815	1	0.95455	0.95455	0.00055	0.95455	0.97700
U	51	1	1786	0	0.98077	1.00000	0.00056	0.99029	0.99972
R	1	0	1837	0	1.00000	1.00000	0.00000	1.00000	1.00000
1	13	1	1824	0	0.92857	1.00000	0.00055	0.96296	0.99973
2	14	3	1821	0	0.82353	1.00000	0.00164	0.90323	0.99918
Y	2	0	1836	0	1.00000	1.00000	0.00000	1.00000	1.00000
S	24	0	1814	0	1.00000	1.00000	0.00000	1.00000	1.00000
W	24	0	1814	0	1.00000	1.00000	0.00000	1.00000	1.00000
T	50	0	1788	0	1.00000	1.00000	0.00000	1.00000	1.00000
3	28	0	1810	0	1.00000	1.00000	0.00000	1.00000	1.00000
4	4	0	1834	0	1.00000	1.00000	0.00000	1.00000	1.00000
J	18	0	1820	0	1.00000	1.00000	0.00000	1.00000	1.00000
K	35	0	1803	0	1.00000	1.00000	0.00000	1.00000	1.00000
6	34	0	1803	1	1.00000	0.97143	0.00000	0.98551	0.98571
N	35	3	1797	3	0.92105	0.92105	0.00167	0.92105	0.95969
L	23	1	1812	2	0.95833	0.92000	0.00055	0.93878	0.95972
E	4	1	1832	1	0.80000	0.80000	0.00055	0.80000	0.89973
O	12	0	1826	0	1.00000	1.00000	0.00000	1.00000	1.00000
B	4	0	1834	0	1.00000	1.00000	0.00000	1.00000	1.00000
M	2	0	1836	0	1.00000	1.00000	0.00000	1.00000	1.00000
X	2	0	1836	0	1.00000	1.00000	0.00000	1.00000	1.00000
D	4	0	1834	0	1.00000	1.00000	0.00000	1.00000	1.00000
G	4	0	1834	0	1.00000	1.00000	0.00000	1.00000	1.00000
A	4	0	1834	0	1.00000	1.00000	0.00000	1.00000	1.00000
@	2	0	1836	0	1.00000	1.00000	0.00000	1.00000	1.00000
Z	2	0	1836	0	1.00000	1.00000	0.00000	1.00000	1.00000
5	20	1	1817	0	0.95238	1.00000	0.00055	0.97561	0.99972
&	2	0	1836	0	1.00000	1.00000	0.00000	1.00000	1.00000
C	2	0	1836	0	1.00000	1.00000	0.00000	1.00000	1.00000

From Table 5.8, we can understand that the F-score of most classes in confusion matrix obtained the accuracy of 100%. This accuracy indicates that those classes are fully predicted without any limitation. The accuracy with F-score of some classes are accurately predicted above 90% while the E class, which represents nominative case marker is predicted with the

lowest F-score of 80%. This may happen because of the availability of insufficient training dataset during the dataset preparation for such suffixes.

From such a matrix, not only accuracy can be derived, but also a number of additional metrics that have become popular in machine learning, information retrieval, and subsequently also in computational linguistics: recall, precision, and their harmonic mean F-score. Precision can be defined as the proportional number of times the classifier has correctly made the decision that some instance has class C whereas recall can be defined as the proportional number of times an instance with class C in the test data has indeed been classified as class C by the classifier. F-score also can be defined as the harmonic mean of precision and recall. The obtained average result of precision, recall and F-score on unseen words are depicted in Table 5.9.

Table 5.9: The average precision, recall and F-score on IB1 and IGTREE with all options

Algorithms	Experiments	Precision	Recall	F-score
IB1	Default parameters	97.54	97.76	97.65
	Feature selection	97.40	97.62	97.51
	Optimal Parameters	96.95	97.06	97.01
	Interleaved	97.32	97.54	97.43
IGTREE	Default parameters	91.80	92.97	92.43
	Feature selection	89.55	91.19	90.37
	Optimal Parameters	92.47	93.63	93.05
	Interleaved	89.55	91.19	90.37

Table 5.9 shows the average results of precision, recall and F-score on IB1 and IGTREE with default parameter settings, feature selection, optimal parameters and interleaving the combination of feature sub selection with optimal parameters. The result is achieved with a dataset of 2270 collected words. We can conclude that, the classifier classified a number of classes in a matrix into true positive.

5.6 Discussion

The proposed Afaan Oromoo morphological analyzer model is evaluated to classify the new instances that were not found in the training dataset based on the accumulated knowledge on the model. Basically, in the experiments performed to classify new instances, the generalization accuracy of 10-FCV, precision, recall and F-score were used as evaluation method.

The generalization accuracy of the model is evaluated using four different alternatives such as default parameter setting, feature sub selection on default parameter setting, parameter optimization and interleaving the combination of feature sub selection and parameter optimization. The generalization accuracy obtained with default parameter settings are 98.80% and 93.44% for IB1 and IGTREE respectively. Feature selection is performed. Features with highest accuracy are selected and the features with low accuracy are ignored. Two features are selected for IB1 as well performing features, while the others 13 are ignores as unnecessary features. Six features are selected for IGTREE as well performing features while the others 9 features are ignored as unnecessary features. Accordingly, the generalization accuracy obtained with combination of feature sub selections and default parameter settings are 98.82% and 94.14% for IB1 and IGTREE respectively. After a number of parameter adjustment, modified value difference (MVD), no-weighting (Weighting=1), inverse distance (ID) and k-nearest neighbor value (k=1) found to be the optimal parameters for IB1 which out performs the generalization accuracy of default parameter settings. Information gain (IG) is the only parameter optimized for IGTREE. Therefore, the generalization accuracy of IB1 with optimal parameters is 98.82%, while the generalization accuracy of IGTREE with optimal parameter is 93.66%. The combination of feature sub selection and optimal parameters are interleaved simultaneously to get high accuracy. The obtained generalization accuracy is 98.86% and 94.14% for IB1 and IGTREE respectively. The accuracy of IB1 outperforms all scenarios. The accuracy of IGTREE shows that it is the same achievement with the combination of feature selection. The accuracy of both algorithms leads to high accuracy as expected.

The precision, recall and F-score (the harmonic mean of precision and recall) were computed by taking the average of the 10-FCV. The precision of IB1 and IGTREE with default

parameter settings, feature selection, optimal parameters, and interleaved feature selection and optimal parameters are 97.54% and 91.80%, 97.40 % and 89.55%, 96.95% and 92.47%, and 97.32% and 89.55% respectively. The recall of IB1 and IGTREE with default parameter settings, feature selection, optimal parameters, and interleaved feature selection and optimal parameters are 97.76% and 92.97%, 97.62% and 91.19%, 97.06% and 93.63%, and 97.54% and 91.19% respectively. The F-score of IB1 and IGTREE with default parameter settings, feature selection, optimal parameters, and interleaved feature selection and optimal parameters are 97.65% and 92.43%, 97.51% and 90.37%, 97.01% and 93.05%, and 97.43% and 90.37% respectively. Finally, we conclude that the selection of features with highest accuracy plays a vital role in getting the best accuracy.

Chapter Six: Conclusion and Future Works

6.1 Conclusion

This study clearly addressed the morphological properties of Afaan Oromoo word classes such as nouns, verbs and adjectives. The morphological properties of these word classes are reviewed and investigated from most of linguistic works. It includes both inflectional and derivational words. The computational approaches to morphological analysis are studied thoroughly in order to tackle the morphological problem of Afaan Oromoo nouns, verbs and adjectives word class. We proposed and designed Afaan Oromoo morphological analyzer using machine learning approach to solve the morphology learning of Afaan Oromoo. From machine learning approach, we selected memory-based learning algorithm because of its appropriateness for Afaan Oromoo morphological analysis. The proposed system has two main components: training and analysis phase. The training phase contains feature extraction, memory learning and trained model sub-components. The analysis phase contains morpheme identification and morpheme extraction components.

For training and testing the system, we developed OROLEX which is a morphological database consisting of the grammatical description of Afaan Oromoo noun, adjective and verb words. It contains 2,270 annotated nouns, verbs and adjectives. Moreover, we automatically extracted 17,386 instances from OROLEX. We partitioned the dataset into training and test dataset using 10-fold evaluation method. The training dataset shares 90%, while test dataset shares 10% of the total dataset.

We used IB1 and IGTREE memory-based learning algorithms implemented in TiMBL in order to train and test our dataset. The model is being evaluated in four scenarios by default parameter settings, feature selection, parameter optimization, and interleaving feature selection and parameter optimization. We organized the presentation accuracy of each scenarios into generalization accuracies. Among all the scenarios, interleaving the combination of selected features and optimal parameters of IB1 obtains the generalization accuracy of 98.86% with compression ratio of 51.31%, while IGTREE obtains the generalization accuracy of 94.14% with compression ratio of 97.83%. From this result, we conclude that the feature selection plays a vital role in getting the best accuracy. Finally, there

is a trade-off between IB1 and IGTREE algorithms. IB1 usually leads to more accuracy at the cost of memory and slower computation than IGTREE.

6.2 Contribution of the Work

Among the major contributions of this study are:

- Proposing a new architecture for Afaan Oromoo morphological analysis with the state-of-the art approach.
- Developing the first Afaan Oromoo morphological database corpus which consists grammatical descriptions of the words.
- Confirming that morphological analyzers are dependent on the combination of features.
- Preparing an encouraging environment for the development of other Afaan Oromoo NLP studies that need morphological analysis as a component in their work.
- Selecting the possible affixes of the words, identifying the hierarchical structure of the affixes and listing according to the different possibilities of their occurrence forming the words.

6.3 Feature Works

The developed Afaan Oromoo morphological analyzer has portions that require further improvements that we want to recommend them as future works.

- Although we proposed a morphological analysis for verb, noun and adjective words, we implemented only simple words. Hence, Compound words can be performed.
- The morphological features of Afaan Oromoo words on MBL can be extracted in different granularities. We performed morphemes boundary on individual characters of a word. Hence, segmentation on the full words and insertions of grammatical descriptions in each morpheme boundary can be performed.
- The morphological analyzer we developed works only analysis by classification. It is also possible to develop morphological generation as other classification.
- The complex nature of Afaan Oromoo morphology cannot let the system to segment some words easily. Therefore, managing spelling changes can be the future work.

- Afaan Oromoo nouns, verbs and adjectives have a lot of affixes. We implemented most of them not all. Therefore, developing the system that includes all affixes can be the future work.
- The system being developed in this study is just a prototype. A full-fledged Afaan Oromoo morphological analyzer that can be easily integrated into different NLP applications can be developed.

References

[1] J. Allen, Natural Language Understanding, Redwood, California, USA: The Benjamin/Cummings Publishing Company, Inc, 2005.

[2] D. Jurafsky and J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Englewood Cliffs, New Jersey, United States: Prentice Hall, 2007.

[3] W. Daelemans and A. van den Bosch, Memory-Based Language Processing, New York, USA: Cambridge University Press, 2005.

[4] K. Anand, V. Dhanalakshimi, K. Soman and S. Rajendran, "A Sequence Labeling Approach to Morphological Analyzer for Tamil Language," *International Journal on Computer Science and Engineering,* vol. 2, no. 6, pp. 1944-1955, 2010.

[5] V. Abeera, S. Aparna, R. Rekha, M. A. Kumar, V. Dhanalakshmi, K. Soman and S. Rajendran, "Morphological Analyzer for Malayalam using Maching Learning," in *Proceedings of the 2nd International Conference on Data Engineering and Management*, Verlag Berlin Heidelberg, German, 2012.

[6] K. Debbarma, D. Das, S. Bandyopadhyay and B. Gopal, "Morphological Analyzer for Kokborok," in *Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing (SANLP)*, Mumbai, India, 2012.

[7] M. Abate and Y. Assabie, "Development of Amharic Morphological using Memory-Based Learning," in *Proceedings of the 9th International Conference on Natural Language Processing (PolTAL2014)*, Warsaw, Poland, 2014.

[8] A. Shaji and Sindhu L, "Morphological Analyzer for Malayalam: A Literature Survey," *International Journal of Computer Applications,* vol. 107, no. 14, pp. 24-27, 2014.

[9] L. Körtvélyessy, Essentials of Language Typology, Košice, Slovakia: Department of British and american Studies, 2017.

[10] K. Shaalan, "Rule-Based Approach in Arabic Natural Language Processing," *International Journal on Information and Communication Technologies,* vol. 3, no. 3, pp. 11-19, 2010.

[11] G. Olani and D. Midekso, "Design and Implementation of Morphology Based Spell Checker," *International Journal of Scientific and Technology Research,* vol. 3, no. 12, pp. 118-125, 2014.

[12] N. Habash, R. Eskander and A. Hawwari, "A Morphological Analyzer for Egyptian Arabic," in *Proceedings of the 12th Meeting of the Special Interest Group on Computational Morphology and Phonology (SIGMORPHON2012)*, Montre´al, Canada, 2012.

[13] M. Gasser, "HornMorpho: A System for Morphological Processing of Amharic, Oromo, and Tigrinya," in *Proceedings of Conference on Human Language Technology for Development*, Alexandria, Egypt, 2011.

[14] A. Abeshu, "Analysis of Rule Based Approach for Afan Oromo Automatic Morphological Synthesizer," *Science, Technology and Arts Research Journal,* vol. 2, no. 4, pp. 94-97, 2013.

[15] W. Mulugeta and M. Gasser, "Learning Morphological Rules for Amharic Verbs Using Inductive Logic Programming," in *Workshop on Language Technology for Normalisation of Less-Resourced Languages (SALTMIL8/AfLaT2012)*, Istanbul, Turkey, 2012.

[16] G. Chowdhury, "Natural Language Processing," *Annual Review of Information Science and Technology,* vol. 37, no. 1, pp. 51-89, 2003.

[17] A. Reshamwala, D. Mishra and P. Pawar, "Review on Natural Language Processing," *Engineering Science and Technology: An International Journal (ESTIJ),* vol. 3, no. 1, pp. 113-116, 2013.

[18] A. Joshi, "Natural Language Processing," *American Association for the Advancement of Science,* vol. 253, no. 5025, pp. 1242-1249, 1991.

[19] S. Vikram, "Morphology: Indian Languages and European Languages," *International Journal of Scientific and Research Publications,* vol. 3, no. 6, pp. 1-5, 2013.

[20] M. Haspelmath and A. Sims, Understanding Morphology, London, UK: Hodder Education, 2010.

[21] G. Booij, The Grammar of Words: An Introduction to Linguistic Morphology, New York, USA: Oxford University Press Inc, 2007.

[22] B. Can and S. Manandhar, "Methods and Algorithms for Unsupervised Learning of Morphology," in *Proceedings of the 15th International Conference on Computational Linguistics and Intelligent Text Processing*, Kathmandu, Nepal, 2014.

[23] J. Rizvi and M. Hussain, "Analysis, Desing And Implementation Of Urdu Morphological Analyzer," in *Student Conference on Engineering Sciences and Technology*, Karachi, Pakistan, 2005.

[24] V. Goyal and G. Singh, "Hindi Morphological Analyzer and Generator," in *First International Conference on Emerging Trends in Engineering and Technology*, Nagpur, Maharashtra, India, 2008.

[25] M. Sawalha and E. Atwell, "SALMA: Standard Arabic Language Morphological Analysis," in *1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, Sharjah, United Arab Emirates, 2013.

[26] J. Goldsmith, "Segmentation and Morphology," in *The Handbook of Computational Linguistics and Natural Language Processing*, Chichester, England, JohnWiley and Sons Ltd, 2010, pp. 364-393.

[27] R. Sproat, Morphology and Computation, London, England: MIT Press, 1992.

[28] A. Barkeessaa, Sanyii Jechaa fi Caasaa Isaa (Word and its Structure), Finfinnee, Ethiopia: Alem Printing PLC, 2011.

[29] J. Goldsmith, "Unsupervised Learning of the Morphology of a Natural Language," *Computational Linguistics,* vol. 27, no. 2, pp. 153-198, 2001.

[30] H. Zellig, "From Phoneme to Morpheme," *Linguistic Society of America,* vol. 31, no. 2, pp. 190-222, 1955.

[31] M. Hafer and S. Weiss, "Word Segmentation by Letter Successor Varieties," *Information Storage and Retrieval,* vol. 10, no. 11-12, pp. 371-385, 1974.

[32] C. Borg and A. Gatt, "Morphological Analysis for the Maltese Language: The challenges of a Hybrid System," in *Proceedings of the 3rd Arabic Natural Language Processing Workshop*, Valencia, Spain, 2017.

[33] A. van den Bosch, W. Daelemans and T. Weijters, "Morphological Analysis as Classification: An Inductive-Learning Approach," in *arXiv:cmp-lg/9607021v1*, 1996.

[34] G. Durrett and J. DeNero, "Supervised Learning of Complete Morphological Paradigms," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, USA, 2013.

[35] I. Muhammad and Z. Yan, "Supervised Machine Learning Approaches: A Survey," *ICTACT Journal on Soft Computing,* vol. 5, no. 3, pp. 946-952, 2015.

[36] H. G. Kumar and R. Choudhary, "Comprehensive Review on Supervised Machine Learning Algorithms," in *International Conference on Machine learning and Data Science*, Greater Noida, Uttar Pradesh, India, 2017.

[37] S. Ray, "A Quick Review of Machine Learning Algorithms," in *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con)*, Faridabad, India, 2019.

[38] M. Xue and C. Zhu, "A Study and Application on Machine Learning of Artificial Intellligence," in *International Joint Conference on Artificial Intelligence*, Hainan Island, China, 2009.

[39] D. Kazakov, "Achievements and Prospects of Learning Word Morphology with Inductive Logic Programming," in *Learning Language in Logic*, Heidelberg, Berlin, Germany, Springer, 2000, pp. 89-109.

[40] S. Manandhar, S. Džeroski and T. Erjavec, "Learning Multilingual Morphology with CLOG," in *Inductive Logic Programming*, Heidelberg, Berlin, German, Springer, 1998, pp. 135-144.

[41] H. Hammarström and L. Borin, "Unsupervised Learning of Morphology," *Computational Linguistics,* vol. 37, no. 2, pp. 309-350, 2011.

[42] M. Ahlberg, M. Forsberg and M. Hulden, "Semi-Supervised Learning of Morphological Paradigms and Lexicons," in *14th Conference of the European Chapter of the Association for Computational Linguistics*, Gothenburg, Sweden, 2014.

[43] P. C. Sen, M. Hajra and M. Ghosh, "Supervised Classification Algorithms in Machine Learning: A Survey and Review," in *Emerging Technology in Modelling and Graphics*, Downtown Core, Singapore, Springer, 2019, pp. 99-111.

[44] A. v. den Bosch and W. Daelemans, "Memory-Based Morphological Analysis," in *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, 1999.

[45] S. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," in *Emerging Artificial Intelligence Applications in Computer Engineering*, Amsterdam, Netherlands, IOS Press, 2007, pp. 3-24.

[46] R. Rahmath and R. Raj, "A Memory Based Approach to Malayalam Noun Generation," in *International Conference on Control, Communication and Computing India (ICCC)*, Trivandrum, India, 2015.

[47] N. M, R. Rahmath, R. Raj and R. Raj, "Malayalam Morphological Analysis Using MBLP Approach," in *International Conference on Soft-Computing and Network Security (ICSNS)*, Coimbatore, India, 2015.

[48] Okfalisa, Mustakim, I. Gazalba and N. G. I. Reza, "Comparative Analysis of K-Nearest Neighbor and Modified K-Nearest Neighbor Algorithm for Data Classification," in *2nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia, 2017.

[49] R. Agrawal, "K-Nearest Neighbor for Uncertain Data," *International Journal of Computer Applications,* vol. 105, no. 11, pp. 13-16, 2014.

[50] L. Le, Y. Xie and V. Raghavan, "Deep Similarity-Enhanced K-Nearest Neighbors," in *IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, 2018.

[51] V. D. B. Antal, S. Canisius, I. Hendrickx, W. Daelemans and E. T. Kim Sang, "Memory-Based Semantic Role Labeling: Optimizing Features, Algorithm, and

Output," in *Proceedings of the 8th Conference on Computational Natural Language Learning (CONLL-2004)*, Boston, USA, 2004.

[52] B. antal van Den and W. Daelemans, "Do Not Forget:Full Memory in Memory-Based Learning of Word Pronunciation," in *New Methods in Language Processing and Computational Natural Language Learning*, Sydney, Australia, 1998.

[53] W. Daelemans, J. Zavrel and K. van der Sloot, "TiMBL: Tilburg Memory-Based Learner," Induction of Linguistic Knowledge, Tilburg University and CLiPS, University of Antwerp, Antwerp, Belgium, 2018.

[54] D. Aha, D. Kibler and M. Albert, "Instance-Based Learning Algorithms," *machine Learning,* vol. 6, no. 1, pp. 37-66, 1991.

[55] W. Daelemans, A. V. Den Bosch and T. Weijters, "IGTree: Using Trees for Compression and Classification in Lazy Learning Algorithms," *Artificial Intelligence Review,* vol. 11, no. 1, pp. 407-423, 1997.

[56] W. Daelemans and A. van den Bosch, "Language-Independent Data-Oriented Grapheme-to-Phoneme Conversion," in *Progress in Speech Synthesis*, New York, USA, Springer, 1997, pp. 77-89.

[57] S. Dudani, "The Distance-Weighted k-Nearest Neighbor Rule," *IEEE Transactions on Systems, Man, and Cybernetics,* Vols. SMC-6, no. 4, pp. 325-327, 1976.

[58] N. Lavrač and S. Džeroski, Inductive Logic Programming: Techniques and Applications, New York, USA: Ellis Horwood, 1994.

[59] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," *Journal of Machine Learning Research,* vol. 2, no. 1, pp. 45-66, 2001.

[60] O. Kohonen, "Advances in Weakly Supervised Learning of Morphology," School of Science, Helsinki, Finland, 2015.

[61] H. Poon, C. Cherry and K. Toutanova, "Unsupervised Morphological Segmentation with Log-Linear Models," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, USA, 2009.

[62] H. Hammarström, "A Survey and Classification of Methods for (Mostly) Unsupervised Learning of Morphology," in *Proceedings of the 16th Nordic Conference of Computational Linguistics*, Tartu, Estonia, 2007.

[63] S. Bordag, "Unsupervised and Knowledge-Free Morpheme Segmentation and Analysis," in *8thWorkshop of the Cross-Language Evaluation Forum, CLEF 2007*, budapest, Hungary, 2007.

[64] H. Déjean, "Morphemes as Necessary Concept for Structures Discovery from Untagged Corpora," in *98 Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning, NeMLaP3/CoNLL*, Sydney, Australia, 1998.

[65] S. Bordag, "Two-step Approach to Unsupervised Morpheme Segmentation," in *Proceedings of 2nd Pascal Challenges Workshop on Unsupervised Segmentation of Words into Morphemes*, Venice, Italy, 2006.

[66] J. Goldsmith, "An Algorithm for the Unsupervised Learning of Morphology," *Natural Language Engineering,* vol. 12, no. 04, pp. 353-371, 2006.

[67] P. Grünwald , "Introducing the Minimum Description Length Principle," in *Advances in Minimum Description Length: Theory and Applications*, Cambridge, Massachusetts, USA, The MIT Press, 2005, pp. 3-21.

[68] M. R. Bent , S. K. Murthy and A. Lundberg, "Discovering Morphemic Suffixes: A Case Study in MDL Induction," in *The 5th International Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, Florida, 1995.

[69] O. Kohonen, S. Virpioja and K. Lagus, "Semi-Supervised Learning of Concatenative Morphology," in *Proceedings of the 11th Meeting of the ACL-SIGMORPHON*, Uppsala, Sweden, 2010.

[70] J. Santamar´ıa and L. Araujo, "Semi-supervised Constituent Grammar Induction Based on Text Chunking Information," in *Conference on Intelligent Text Processing*, Samos,Greece, 2013.

[71] A. Zheng, Evaluating Machine Learning Models, Sebastopol, California, USA: O'Reilly Media Inc, 2015.

[72] H. Blockeel and J. Struyf, "Efficient Algorithms for Decision Tree Cross-Validation," *Journal of Machine Learning Research,* vol. 3, no. 4, pp. 621-650, 2002.

[73] P. Refaeilzadeh, L. Tang and H. Liu, "Cross-Validation," in *Encyclopedia of Database Systems*, Boston, USA, Springer, 2009, pp. 532-537.

[74] S. Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning," *arXiv preprint arXiv:1811.12808,* 2018.

[75] Y. Bengio and Y. Grandvalet, "No Unbiased Estimator of the Variance of K-Fold Cross-Validation," *Journal of Machine Learning Research,* vol. 5, no. 5, pp. 1089-1105, 2004.

[76] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning, New York, USA: Springer, 2017.

[77] N. David Marom, L. Rokach and A. Shmilovici, "Using the Confusion Matrix for Improving Ensemble Classifiers," in *26th Convention of Electrical and Electronics Engineers in Israel*, Eilat, Israel, 2010.

[78] M. Navin and P. R, "Performance Analysis of Text Classification Algorithms using Confusion Matrix," *International Journal of Engineering and Technical Research (IJETR),* vol. 6, no. 4, pp. 75-78, 2016.

[79] M. Makhtar, D. C. Neagu and M. . J. Ridley, "Comparing Multi-class Classifiers: On the Similarity of Confusion Matrices for Predictive Toxicology Applications," in *Proceedings of the 12th International Conference on Intelligent Data Engineering and Automated Learning*, Berlin, German, 2011.

[80] M. Sokolova and G. Lapalme, "A Systematic Analysis of Performance Measures for Classification Tasks," *Information Processing and Management,* vol. 45, no. 4, pp. 427-437, 2009.

[81] D. B. Teferi, "The Development of Oromo Writing System," Doctor of Philosophy (PhD) Thesis, School of European Culture and Languages, University of Kent, Canterbury, England, 2015.

[82] T. Gamta, "Qube Afaan Oromoo: Reasons for Choosing the Latin Script for Developing an Oromo Alphabet," *Journal of Oromo Studies,* vol. 1, no. 1, pp. 36-40, 1993.

[83] T. Fufa, "A Typology of Verbal Derivation in Ethiopian Afro-Asiatic Languages," LOT, Utrecht, Netherland, 2009.

[84] F. Demie, "Historical Challenges in the Development of Oromo Language and Some Agenda for Future Research," *The Journal of Oromo Studies,* vol. 3, no. 1 and 2, pp. 18-27, 1996.

[85] T. Wami, Partisan Discourse and Authentic History, Addis Ababa, Ethiopia: Artsistic Printing Press, 2015.

[86] A. Nafaa, L. Kabbabaa, W. Dachaasaa and T. Nagaasaa, Caasluga Afaan Oromoo, Finfinnee: Branna P.E, 1998.

[87] M. Ali and A. Zaborski, Handbook of the Oromo Language, Warsaw, Poland: Zaklad Narodowy im, 1990.

[88] T. Gamta, "Structural and Word Stress Patterns in Afaan Oromo," *Journal of Oromo Studies,* vol. 6, no. 1 and 2, pp. 173-194, 1999.

[89] G. Mamo, "Automotic Part of Speech Tagging for Afaan Oromoo Language," Thesis, School of Graduate Studies, Addis Ababa University, 2009.

[90] G. Olani, "Design and Implementation of Afaan Oromo Spell Checker," Thesis, School of Graduate Studies, Addis Ababa University, 2013.

[91] W. Gobena, "Inflectional Morphology in Mecha Oromo," *Journal of Languages and Culture,* vol. 8, no. 8, pp. 110-140, 2017.

[92] C. G. Mewis, A Grammatical Sketch of Written Oromo, Cologne, German: Rudiger Koppe Verlag, 2001.

[93] B. Yimam, "Oromo Substantives: Some Aspects their Morphology and Syntax," Thesis, Addis Ababa University, 1981.

[94] S. Mazengia, "Aspect and Tense in Oromo," in *Time in the Languages of the Horn of Africa*, Wiesbaden, German, Wiesbaden: Harrassowitz Verlag, 2016, pp. 117-137.

[95] A. Nefa, "Inflection in Oromoo," Thesis, Addis Ababa University, 1982.

[96] A. Barkeessaa, Natoo: Yaadrimee Caasluga Afaan Oromoo (Concept of Afaan Oromoo Grammar), Finfinnee, Oromiyaa: Subi Printing Press, 2012.

[97] A. Ivanovska, K. Zdravkova, T. Erjavec and S. Džeroski, "Learning Rules for Morphological Analysis and Synthesis of Macedonian Nouns, Adjectives and Verbs," in *Proceending of 9th International Multi-Conference Information Society: Language Technologies*, Ljubljana, Slovenia, 2005.

[98] X. Tang, "English Morphological Analysis with Machine-Learned Rules," in *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, Wuhan, China, 2006.

[99] S. Keshava and E. Pitler, "A Simpler, Intuitive Approach to Morpheme Induction," in *Proceedings of 2nd Pascal Challenges Workshop*, Venice, Italy, 2006.

[100] M. Thayaparan, T. Pranavan, U. M. N. Nadarasamoorthy, G. D. S. J. and S. Ranatunga, "Tamil Morphological Analyzer Using Support Vector Machines," in *21th International Conference on Applications of Natural Language to Information Systems*, Salford,UK, 2016.

[101] P. Das and A. Das, "Bengali Noun Morphological Analyzer," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Mysore, India, 2013.

[102] M. Yonis, "Development of Morphological Analyzer for Af-Somali," Thesis, Addis Ababa University, 2017.

[103] T. Sejnowski and C. Rosenberg, "Parallel Networks that Learn to Pronounce English Text," *Complex Systems,* vol. 1, no. 1, pp. 145-168, 1987.

Appendixes

Appendix A: Sample Afaan Oromoo Verbs

bade	basaase	nihursiise
badi	basaasu	nihursiisu
bana	caraani	nigodaane
banu	ciniine	nigodaanu
bara	ciniini	nihiratama
bare	nidaaru	nihiratame
malu	nideege	hinhorsiisin
mara	nideegu	waraansisu
mare	nideemu	waraabsisu
mari	nidhaba	haahorsiisani
maru	nidhabe	haawaraabnu
mugi	nijoore	haawaraannu
mugu	nijooru	haatarsaasani
mura	nihirsiise	haabansiisani
mure	niqaratte	haabarsiisani
muri	niqaratti	haacirsiisani
qaba	nisoofame	haacufsiisani
nibara	niqabsiisa	haahimsiisani
nibare	nihursiisa	haasafaratani

Appendix B: Sample Afaan Oromoo Nouns

jila	nafoota	dulaaba
nafa	bofaan	hirriba
jilaaf	mukaan	sabbata
bofa	beelaan	bunittii
muka	gaaraan	adalicha
bofni	bosonaaf	gurricha
mukni	bosonaan	adaloota
beela	bunaaf	hirribni
dirra	naacha	adalittii
gaara	naachi	bunaan
warra	looniif	adalaan
gurri	namoota	faaraan
warri	baalaaf	gurraan
bofaaf	gaafaaf	deemsa
mukaaf	harkaaf	dammicha
bosona	bunicha	warrummaa
boficha	harkicha	adalootaaf
mukicha	adalaaf	rifeensaan
nafaaf	biyyaaf	miseensaan
wadala	gurraaf	hirmaata
naficha	warraaf	dhumaatii

Appendix C: Sample Afaan Oromoo Adjectives

daamaa	baballoo	booraa
diimaa	qallaa	salphaa
ballaa	qalloo	salphoo
balloo	qaqallaa	dhiphaa
baballaa	qaqalloo	dhiphoo

Appendix D: Sample Manually Annotated Afaan Oromoo Verbs

fufVeP
fufViQ
marVeP
marViQ
sobVeP
sobViQ
yaabViQ
kuusVeP
kuusViQ
soofVeP
soofVuU
qorVaaR
qorVt2eP
qorVt2iY
soorVaaR
soorVt2aI
soorVt2eP
niTcabVaI
niTcabVeP
niTcabVuU
niTkufVaI
niTkufVeP
niTkufVuU
barVam3eP
barVam3uU
furVam3aI
furVam3eP
niTmarVeP
haguugVeP
haguugViQ
niThirVt2aI
niThirVt2eP
niThirVt2uU
soofVam3aI
soofVam3eP
qabVsiis6aI
qabVsiis6eP
qabVsiis6iQ
qabVsiis6uU
niTcabVn1aI
niTcabVn1eP
niTkufVn1aI
niTkufVn1eP
niTkufVt2aI
niTkufVt2eP
niTkufVt2iY
godaanVeP
godaanVuU
soorVt2anSiW
godaanVaaR
godaanVn1aI
godaanVn1eP
niThirVat4aI
niThirVat4eP
niThirVat4uU
niTcirVat4aI
niTcirVat4eP
niTcirVat4uU
niThubVat4uU
niTguutVam3aI
niTguutVam3eP
niTguutVam3uU
niTsoofVam3aI
niTsoofVam3eP
niTsoofVam3uU
haaJcirVanSiW
hinKguutViQnK
waraabVsis5iQ
waraabVsis5uU
niThirVsiis6aI
niThirVsiis6uU
niTqabVsiis6aI
niTqabVsiis6eP
niTqabVsiis6uU
hinKjoorViQnK
haaJjoorVanSiW
haaJhirVsiis6uU
haaJqabVsiis6uU
hinKcabVsiis6iQnK
haaJcabVsiis6anSiW
haaJwaraanVsis5uU
hinKwaraanVsis5Uu
haaJqotVam3anSiW
niTwaraanVsis5t2iY
hinKsafarVsiis6iQnK
haaJwaraanVsis5n1uU
haaJsafarVsiis6anSiW
haaJwaraabVsis5anSiW
haaJwaraanVam3anSiW
haaJwaraanVsis5anSiW

Appendix E: Sample Manually Annotated Afaan Oromoo Nouns

nafNaL
bofNaL
mukNaL
bunNaL
loonN
bonNaL
foonN
sabNaL
diinNaL
gaalNaL
bofNniE
mukNniE
harmNaL
harmNiE
nafNniE
amalNaL
dammNaL
farrNaL
harbNaL
farrNiE
harbNiE
jilNaLafO
nafNaLafO
araamNaL
foonNiifO
wadalNaL
baaburNaL
laadanNaL
mandarNaL
nafNichBaL
nafNootaM
dammNaLafO
harbNaLafO
anfaarNaL
kibaabNaL
laddanNaL
nafNittiiX
amalNichBaL
dammNichBaL
farrNichBaL
harbNichBaL
amalNootaM
farrNootaM
bonNichBaL
harbNootaM
kibaabNniE
cineensNaL
cineensNiE
amalNittiiX
bunNichBaL
adalNaLafO
biyyNaLafO
faarNaLafO
gurrNaLafO
warrNaLafO
farrNootaMafO
harbNootaMafO
cineensNaLafO
dammNaLaOnG
cineensNaLaOnG
bofNittiiX
bofNichBaL
bofNaLafO
mukNaLafO
mukNichBaL
mukNootaM
beelNaLafO
gaarNaLafO
kiyyNaLafO
gaarNotaM
gaarNichBaL
kanniisNaL
kanniisNiE
namNootaMafO
harkNootaMafO
finniisNaLafO
namNaLaOnG
baalNaLaOnG
diidNaLaOnG
gaafNaLaOnG
harkNaLaOnG
mukNootaMafO
baaburNootaM
mandarNootaM
kanniisNaLafO
bofNaLaOnG
mukNaLaOnG
beelNaLaOnG
bosonNaLafO
bosonNaLaOnG

Appendix F: Sample Manually Annotated Afaan Oromoo Adjectives

daamAaa@	diimAaa@	boorAaa@
ballAaa@	qallAaa@	dhiphAaa@
ballAoo&	qallAoo&	dhiphAoo&
baCballAaa@	qaCqallAaa@	salphAaa@
baCballAoo&	qaCqallAoo&	salphAoo&

Appendix G: Sample Extracted Features of Afaan Oromoo Verbs

-,-,-,-,-,-,-,m,a,r,e,-,-,-,-,0

-,-,-,-,-,-,-,m,a,r,e,-,-,-,-,-,0

-,-,-,-,-,m,a,r,e,-,-,-,-,-,-,V

-,-,-,-,m,a,r,e,-,-,-,-,-,-,-,P

-,-,-,-,-,-,-,d,a,a,r,e,-,-,-,0

-,-,-,-,-,-,-,d,a,a,r,e,-,-,-,-,0

-,-,-,-,-,d,a,a,r,e,-,-,-,-,-,0

-,-,-,-,d,a,a,r,e,-,-,-,-,-,-,V

-,-,-,d,a,a,r,e,-,-,-,-,-,-,-,P

-,-,-,-,-,-,-,s,a,f,a,r,e,-,-,0

-,-,-,-,-,-,s,a,f,a,r,e,-,-,-,0

-,-,-,-,-,s,a,f,a,r,e,-,-,-,-,0

-,-,-,-,s,a,f,a,r,e,-,-,-,-,-,0

-,-,-,s,a,f,a,r,e,-,-,-,-,-,-,V

-,-,s,a,f,a,r,e,-,-,-,-,-,-,-,P

-,-,-,-,-,-,-,n,i,m,a,r,t,i,-,0

-,-,-,-,-,-,n,i,m,a,r,t,i,-,-,T

-,-,-,-,-,n,i,m,a,r,t,i,-,-,-,0

-,-,-,-,n,i,m,a,r,t,i,-,-,-,-,0

-,-,-,n,i,m,a,r,t,i,-,-,-,-,-,V

-,-,n,i,m,a,r,t,i,-,-,-,-,-,-,2

-,n,i,m,a,r,t,i,-,-,-,-,-,-,-,Y

-,-,-,-,-,-,-,b,a,r,a,a,r,e,-,0

-,-,-,-,-,-,b,a,r,a,a,r,e,-,-,0

-,-,-,-,-,b,a,r,a,a,r,e,-,-,-,0

-,-,-,-,b,a,r,a,a,r,e,-,-,-,-,0

-,-,-,b,a,r,a,a,r,e,-,-,-,-,-,0

-,-,b,a,r,a,a,r,e,-,-,-,-,-,-,V

-,b,a,r,a,a,r,e,-,-,-,-,-,-,-,P

-,-,-,-,-,-,-,m,a,r,a,r,a,m,e,0

-,-,-,-,-,-,-,m,a,r,a,r,a,m,e,-,0

-,-,-,-,-,-,m,a,r,a,r,a,m,e,-,-,0

-,-,-,-,-,m,a,r,a,r,a,m,e,-,-,-,0

-,-,-,-,m,a,r,a,r,a,m,e,-,-,-,-,0

-,-,-,m,a,r,a,r,a,m,e,-,-,-,-,-,V

-,-,m,a,r,a,r,a,m,e,-,-,-,-,-,-,0

-,m,a,r,a,r,a,m,e,-,-,-,-,-,-,3

m,a,r,a,r,a,m,e,-,-,-,-,-,-,-,P

-,-,-,-,-,-,-,h,i,n,y,a,a,b,i,0

-,-,-,-,-,-,h,i,n,y,a,a,b,i,n,0

-,-,-,-,-,h,i,n,y,a,a,b,i,n,-,K

-,-,-,-,h,i,n,y,a,a,b,i,n,-,-,0

-,-,-,h,i,n,y,a,a,b,i,n,-,-,-,0

-,-,h,i,n,y,a,a,b,i,n,-,-,-,-,0

-,h,i,n,y,a,a,b,i,n,-,-,-,-,-,V

h,i,n,y,a,a,b,i,n,-,-,-,-,-,-,Q

i,n,y,a,a,b,i,n,-,-,-,-,-,-,-,K

-,-,-,-,-,-,-,n,i,w,a,r,a,a,b,0

-,-,-,-,-,-,-,n,i,w,a,r,a,a,b,s,T

-,-,-,-,-,n,i,w,a,r,a,a,b,s,i,0

-,-,-,-,n,i,w,a,r,a,a,b,s,i,s,0

-,-,-,n,i,w,a,r,a,a,b,s,i,s,e,0

-,-,n,i,w,a,r,a,a,b,s,i,s,e,-,0

-,n,i,w,a,r,a,a,b,s,i,s,e,-,-,0

n,i,w,a,r,a,a,b,s,i,s,e,-,-,-,V

i,w,a,r,a,a,b,s,i,s,e,-,-,-,-,0

w,a,r,a,a,b,s,i,s,e,-,-,-,-,-,0

a,r,a,a,b,s,i,s,e,-,-,-,-,-,-,5

r,a,a,b,s,i,s,e,-,-,-,-,-,-,-,P

-,-,-,-,-,-,-,h,i,n,s,a,f,a,r,0

-,-,-,-,-,-,-,h,i,n,s,a,f,a,r,s,0

-,-,-,-,-,-,h,i,n,s,a,f,a,r,s,i,K

-,-,-,-,-,h,i,n,s,a,f,a,r,s,i,i,0

-,-,-,-,h,i,n,s,a,f,a,r,s,i,i,s,0

-,-,-,h,i,n,s,a,f,a,r,s,i,i,s,i,0

-,-,h,i,n,s,a,f,a,r,s,i,i,s,i,n,0

-,h,i,n,s,a,f,a,r,s,i,i,s,i,n,-,V

i,n,s,a,f,a,r,s,i,i,s,i,n,-,-,0

n,s,a,f,a,r,s,i,i,s,i,n,-,-,-,0

s,a,f,a,r,s,i,i,s,i,n,-,-,-,-,0

a,f,a,r,s,i,i,s,i,n,-,-,-,-,-,6

f,a,r,s,i,i,s,i,n,-,-,-,-,-,-,Q

a,r,s,i,i,s,i,n,-,-,-,-,-,-,-,K

-,-,-,-,-,-,-,h,a,a,c,a,r,a,a,0

-,-,-,-,-,-,h,a,a,c,a,r,a,a,n,0

-,-,-,-,-,h,a,a,c,a,r,a,a,n,s,J

-,-,-,-,h,a,a,c,a,r,a,a,n,s,i,0

-,-,-,h,a,a,c,a,r,a,a,n,s,i,s,0

-,-,h,a,a,c,a,r,a,a,n,s,i,s,a,0

-,h,a,a,c,a,r,a,a,n,s,i,s,a,n,0

h,a,a,c,a,r,a,a,n,s,i,s,a,n,i,0

a,a,c,a,r,a,a,n,s,i,s,a,n,i,-,V

a,c,a,r,a,a,n,s,i,s,a,n,i,-,-,0

c,a,r,a,a,n,s,i,s,a,n,i,-,-,-,0

a,r,a,a,n,s,i,s,a,n,i,-,-,-,-,6

r,a,a,n,s,i,s,a,n,i,-,-,-,-,-,0

a,a,n,s,i,s,a,n,i,-,-,-,-,-,-,S

a,n,s,i,s,a,n,i,-,-,-,-,-,-,-,W

Appendix H: Sample Extracted Features of Afaan Oromoo Nouns

-,-,-,-,-,-,b,o,f,a,-,-,-,-,-,0

-,-,-,-,-,b,o,f,a,-,-,-,-,-,-,N

-,-,-,-,b,o,f,a,-,-,-,-,-,-,-,L

-,-,-,-,-,-,-,b,o,f,n,i,-,-,-,0

-,-,-,-,-,-,b,o,f,n,i,-,-,-,-,0

-,-,-,-,-,b,o,f,n,i,-,-,-,-,-,N

-,-,-,-,b,o,f,n,i,-,-,-,-,-,-,0

-,-,-,b,o,f,n,i,-,-,-,-,-,-,-,E

-,-,-,-,-,-,-,d,a,m,m,a,a,f,-,0

-,-,-,-,-,-,d,a,m,m,a,a,f,-,-,0

-,-,-,-,-,d,a,m,m,a,a,f,-,-,-,0

-,-,-,-,d,a,m,m,a,a,f,-,-,-,-,N

-,-,-,d,a,m,m,a,a,f,-,-,-,-,-,L

-,-,d,a,m,m,a,a,f,-,-,-,-,-,-,0

-,d,a,m,m,a,a,f,-,-,-,-,-,-,-,O

-,-,-,-,-,-,-,n,a,m,i,c,h,a,-,0

-,-,-,-,-,-,n,a,m,i,c,h,a,-,-,0

-,-,-,-,-,n,a,m,i,c,h,a,-,-,-,N

-,-,-,-,n,a,m,i,c,h,a,-,-,-,-,0

-,-,-,n,a,m,i,c,h,a,-,-,-,-,-,0

-,-,n,a,m,i,c,h,a,-,-,-,-,-,-,B

-,n,a,m,i,c,h,a,-,-,-,-,-,-,-,L

-,-,-,-,-,-,-,n,a,m,o,o,t,a,-,0

-,-,-,-,-,-,n,a,m,o,o,t,a,-,-,0

-,-,-,-,-,n,a,m,o,o,t,a,-,-,-,N

-,-,-,-,n,a,m,o,o,t,a,-,-,-,-,0

-,-,-,n,a,m,o,o,t,a,-,-,-,-,-,0

-,-,n,a,m,o,o,t,a,-,-,-,-,-,-,0

-,n,a,m,o,o,t,a,-,-,-,-,-,-,-,M

-,-,-,-,-,-,-,g,a,a,r,i,c,h,a,0

-,-,-,-,-,-,g,a,a,r,i,c,h,a,-,0

-,-,-,-,-,g,a,a,r,i,c,h,a,-,-,0

-,-,-,-,g,a,a,r,i,c,h,a,-,-,-,N

-,-,-,g,a,a,r,i,c,h,a,-,-,-,-,0

-,-,g,a,a,r,i,c,h,a,-,-,-,-,-,0

-,g,a,a,r,i,c,h,a,-,-,-,-,-,-,B

g,a,a,r,i,c,h,a,-,-,-,-,-,-,-,L

-,-,-,-,-,-,-,d,h,u,g,a,a,t,i,0

-,-,-,-,-,-,d,h,u,g,a,a,t,i,i,0

-,-,-,-,-,d,h,u,g,a,a,t,i,i,-,0

-,-,-,-,d,h,u,g,a,a,t,i,i,-,-,V

-,-,-,d,h,u,g,a,a,t,i,i,-,-,-,0

-,-,d,h,u,g,a,a,t,i,i,-,-,-,-,0

-,d,h,u,g,a,a,t,i,i,-,-,-,-,-,0

d,h,u,g,a,a,t,i,i,-,-,-,-,-,-,0

h,u,g,a,a,t,i,i,-,-,-,-,-,-,-,D

-,-,-,-,-,-,-,f,o,o,n,i,c,h,a,0

-,-,-,-,-,-,f,o,o,n,i,c,h,a,-,0

-,-,-,-,-,f,o,o,n,i,c,h,a,-,-,0

-,-,-,-,f,o,o,n,i,c,h,a,-,-,-,N

-,-,-,f,o,o,n,i,c,h,a,-,-,-,-,0

-,-,f,o,o,n,i,c,h,a,-,-,-,-,-,0

-,f,o,o,n,i,c,h,a,-,-,-,-,-,-,B

f,o,o,n,i,c,h,a,-,-,-,-,-,-,-,L

-,-,-,-,-,-,-,g,e,e,b,i,c,h,a,0

-,-,-,-,-,-,g,e,e,b,i,c,h,a,-,0

-,-,-,-,-,g,e,e,b,i,c,h,a,-,-,0

-,-,-,-,g,e,e,b,i,c,h,a,-,-,-,N

-,-,-,g,e,e,b,i,c,h,a,-,-,-,-,0

-,-,g,e,e,b,i,c,h,a,-,-,-,-,-,0

-,g,e,e,b,i,c,h,a,-,-,-,-,-,-,B

g,e,e,b,i,c,h,a,-,-,-,-,-,-,-,L

-,-,-,-,-,-,-,h,o,r,m,a,a,t,a,0

-,-,-,-,-,-,-,h,o,r,m,a,a,t,a,-,0

-,-,-,-,-,-,h,o,r,m,a,a,t,a,-,-,V

-,-,-,-,h,o,r,m,a,a,t,a,-,-,-,0

-,-,-,h,o,r,m,a,a,t,a,-,-,-,-,0

-,-,h,o,r,m,a,a,t,a,-,-,-,-,-,0

-,h,o,r,m,a,a,t,a,-,-,-,-,-,-,0

h,o,r,m,a,a,t,a,-,-,-,-,-,-,-,D

-,-,-,-,-,-,-,n,a,m,i,t,t,i,i,0

-,-,-,-,-,-,-,n,a,m,i,t,t,i,i,-,0

-,-,-,-,-,-,n,a,m,i,t,t,i,i,-,N

-,-,-,-,n,a,m,i,t,t,i,i,-,-,-,0

-,-,-,n,a,m,i,t,t,i,i,-,-,-,-,0

-,-,n,a,m,i,t,t,i,i,-,-,-,-,-,0

-,n,a,m,i,t,t,i,i,-,-,-,-,-,-,0

n,a,m,i,t,t,i,i,-,-,-,-,-,-,-,X

-,-,-,-,-,-,-,f,i,n,n,i,i,s,a,0

-,-,-,-,-,-,f,i,n,n,i,i,s,a,-,0

-,-,-,-,-,f,i,n,n,i,i,s,a,-,-,0

-,-,-,-,f,i,n,n,i,i,s,a,-,-,-,0

-,-,-,f,i,n,n,i,i,s,a,-,-,-,-,0

-,-,f,i,n,n,i,i,s,a,-,-,-,-,-,0

-,f,i,n,n,i,i,s,a,-,-,-,-,-,-,N

f,i,n,n,i,i,s,a,-,-,-,-,-,-,-,L

Appendix I: Sample Extracted Features of Afaan Oromoo Adjectives

-,-,-,-,-,-,-,d,a,a,m,a,a,-,-,0

-,-,-,-,-,-,-,d,a,a,m,a,a,-,-,-,0

-,-,-,-,-,d,a,a,m,a,a,-,-,-,-,0

-,-,-,-,d,a,a,m,a,a,-,-,-,-,-,A

-,-,-,d,a,a,m,a,a,-,-,-,-,-,-,0

-,-,d,a,a,m,a,a,-,-,-,-,-,-,-,@

-,-,-,-,-,-,-,b,a,l,l,a,a,-,-,0

-,-,-,-,-,-,b,a,l,l,a,a,-,-,-,0

-,-,-,-,-,-,b,a,l,l,a,a,-,-,-,-,0

-,-,-,-,-,b,a,l,l,a,a,-,-,-,-,A

-,-,-,b,a,l,l,a,a,-,-,-,-,-,-,0

-,-,b,a,l,l,a,a,-,-,-,-,-,-,-,@

-,-,-,-,-,-,-,b,a,l,l,o,o,-,-,0

-,-,-,-,-,-,b,a,l,l,o,o,-,-,-,0

-,-,-,-,-,b,a,l,l,o,o,-,-,-,-,0

-,-,-,-,b,a,l,l,o,o,-,-,-,-,-,A

-,-,-,b,a,l,l,o,o,-,-,-,-,-,-,0

-,-,b,a,l,l,o,o,-,-,-,-,-,-,-,&

-,-,-,-,-,-,-,b,a,b,a,l,l,a,a,0

-,-,-,-,-,-,b,a,b,a,l,l,a,a,-,C

-,-,-,-,-,-,b,a,b,a,l,l,a,a,-,-,0

-,-,-,-,-,b,a,b,a,l,l,a,a,-,-,-,0

-,-,-,b,a,b,a,l,l,a,a,-,-,-,-,0

-,-,b,a,b,a,l,l,a,a,-,-,-,-,-,A

-,b,a,b,a,l,l,a,a,-,-,-,-,-,-,0

b,a,b,a,l,l,a,a,-,-,-,-,-,-,-,@

-,-,-,-,-,-,-,b,a,b,a,l,l,o,o,0

-,-,-,-,-,-,b,a,b,a,l,l,o,o,-,C

-,-,-,-,-,-,b,a,b,a,l,l,o,o,-,-,0

-,-,-,-,-,b,a,b,a,l,l,o,o,-,-,-,0

-,-,-,-,b,a,b,a,l,l,o,o,-,-,-,-,0

-,-,-,b,a,b,a,l,l,o,o,-,-,-,-,-,A

-,-,b,a,b,a,l,l,o,o,-,-,-,-,-,-,0

-,b,a,b,a,l,l,o,o,-,-,-,-,-,-,-,@

b,a,b,a,l,l,o,o,-,-,-,-,-,-,-,&

-,-,-,-,-,-,-,b,o,o,q,a,a,-,-,0

-,-,-,-,-,-,b,o,o,q,a,a,-,-,-,0

-,-,-,-,-,b,o,o,q,a,a,-,-,-,-,0

-,-,-,b,o,o,q,a,a,-,-,-,-,-,-,A

-,-,b,o,o,q,a,a,-,-,-,-,-,-,-,0

-,b,o,o,q,a,a,-,-,-,-,-,-,-,-,@

-,-,-,-,-,-,-,q,a,l,l,a,a,-,-,0

-,-,-,-,-,-,q,a,l,l,a,a,-,-,-,0

-,-,-,-,-,q,a,l,l,a,a,-,-,-,-,0

-,-,-,-,q,a,l,l,a,a,-,-,-,-,-,A

-,-,-,q,a,l,l,a,a,-,-,-,-,-,-,0

-,-,q,a,l,l,a,a,-,-,-,-,-,-,-,@

-,-,-,-,-,-,-,q,a,l,l,o,o,-,-,0

-,-,-,-,-,-,q,a,l,l,o,o,-,-,-,0

-,-,-,-,-,q,a,l,l,o,o,-,-,-,-,0

-,-,-,-,q,a,l,l,o,o,-,-,-,-,-,A

-,-,-,q,a,l,l,o,o,-,-,-,-,-,-,0

-,-,q,a,l,l,o,o,-,-,-,-,-,-,-,&

-,-,-,-,-,-,-,q,a,q,a,l,l,a,a,0

-,-,-,-,-,-,q,a,q,a,l,l,a,a,-,C

-,-,-,-,-,q,a,q,a,l,l,a,a,-,-,0

-,-,-,-,q,a,q,a,l,l,a,a,-,-,-,0

q,a,q,a,l,l,a,a,-,-,-,-,-,-,-,@

-,-,-,-,-,-,-,q,a,q,a,l,l,o,o,0

-,-,-,-,-,-,-,q,a,q,a,l,l,o,o,-,C

-,-,-,-,-,-,q,a,q,a,l,l,o,o,-,-,0

-,-,-,-,-,q,a,q,a,l,l,o,o,-,-,-,0

-,-,-,-,q,a,q,a,l,l,o,o,-,-,-,-,0

-,-,q,a,q,a,l,l,o,o,-,-,-,-,-,A

-,q,a,q,a,l,l,o,o,-,-,-,-,-,-,0

q,a,q,a,l,l,o,o,-,-,-,-,-,-,-,&

-,-,-,-,-,-,-,s,a,l,p,h,a,a,-,0

-,-,-,-,-,-,-,s,a,l,p,h,a,a,-,-,0

-,-,-,-,-,-,s,a,l,p,h,a,a,-,-,0

-,-,-,-,-,s,a,l,p,h,a,a,-,-,-,0

-,-,-,-,s,a,l,p,h,a,a,-,-,-,-,A

-,-,-,s,a,l,p,h,a,a,-,-,-,-,-,0

-,s,a,l,p,h,a,a,-,-,-,-,-,-,-,@

-,-,-,-,-,-,-,s,a,l,p,h,o,o,-,0

-,-,-,-,-,-,-,s,a,l,p,h,o,o,-,-,0

-,-,-,-,-,s,a,l,p,h,o,o,-,-,-,0

-,-,-,s,a,l,p,h,o,o,-,-,-,-,-,0

-,-,-,s,a,l,p,h,o,o,-,-,-,-,-,A

-,-,s,a,l,p,h,o,o,-,-,-,-,-,-,0

-,s,a,l,p,h,o,o,-,-,-,-,-,-,-,&

Appendix J: Morpheme Boundary Markers

0	No boundary
V	Verb base stem
I	Imperfective aspect
P	Perfective aspect
Q	Singular imperative mood
R	Plural imperative mood
U	Allomorph of imperfective aspect occurring with 2PL and 3PL
1	1PL person marker
2	2SG, 3SGF and 2PL person marker
Y	Allomorph of imperfective aspect occurring with 3SGF
S	2PL and 3PL perfective aspect number marker
W	Allomorph of perfective aspect occurring with 2PL and 3PL
T	Affirmative marker
3	Passive stem
4	Autobenefactive or middle voice stem
J	Jussive mood
K	Negation
5	Causative stem 1
6	Causative stem 2
N	Noun stem
L	Accusative case
E	Nominative case
O	Dative case
B	Masculine definiteness
M	Plurality
X	feminine definiteness
G	Instrumental case
Z	Focus marker
D	Derivation
A	Adjective stem

@	Adjective masculine gender
&	Adjective feminine gender
C	Stem reduplication

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 dully acknowledged.

Declared by:

Name: _____

Signature: _____

Date: _____

Confirmed Advisor:

Name: _____

Signature: _____

Date: _____