



JIMMA UNIVERSITY
JIMMA INSTITUTE OF TECHNOLOGY
SCHOOL OF COMPUTING
MSc in Information Technology
Afaan Oromo Parser using Hybrid Approach

Tesfaye Gadisa Ayana

The Thesis Submitted to the School of Graduate Studies in Partial Fulfillment for the Degree of Master of Science in Information Technology.

Jimma, Ethiopia

16 November 2017


Jimma University
Jimma Institute of Technology
School of Computing

Tesfaye Gadisa Ayana
Advisor: Mr. Debela Tesfaye (PHD Candidate)

Co-Advisor: Mr. Kibret Zewdu (MSC)

This is to certify that the thesis prepared by Tesfaye Gadisa Ayana, titled: Afaan Oromo Parser using Hybrid Approach and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Information Technology 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:	<u>Mr. Debela Tesfaye (PHD Candidate)</u>		<u>11/16/17</u>
Co-Advisor:	<u>Mr. Kibret Zewdu (MSC)</u>		<u>11/16/17</u>
Internal Examiner:	_____		
External Examiner:	_____		

Declaration

I hereby certify that this material, which I now submit for assessment on the program of study leading to the award of Masters of Science (MSC) is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Tesfaye Gadisa Ayana

16 November 2017

The thesis has been submitted for examination with my approval as university advisors.

Mr. Debela Tesfaye (PhD Candidate)

Mr. Kibret Zewdu (MSC.)

16 November 2017

Dedicated to:

My father, **Gadisa Ayana** and my mother, **Gelane Gada**; who have brought me to be a man I am today without having academic educations themselves.

Acknowledgement

First and foremost, I thank **God** for bestowing his blessings on me which has enabled me to complete the thesis with astounding success. I would like to thank my parents for their support and belief which had led to the success of the thesis.

I would like to express my heart felt gratitude to **Mr. Debela Tesfaye**, my thesis advisor who was inspiring me and providing consistent support and timely help in every possible way by constantly monitoring the performance and making this thesis successful. Also I have special thanks to my co-advisor **Mr. Kibret Zewdu** for his support.

I would like to thank program coordinators and all other staff members of school of computing who have helped me. Finally, I would like to thank all the staff members and friends who have helped, guided and encouraged me directly or indirectly during the thesis work.

Table of Contents

Abstract	i
Chapter One	1
1. Introduction	1
1.1. Back ground of the study	1
1.2. Statement of the problem	3
1.3. Objective	4
1.3.1. General Objective	4
1.3.2. Specific Objective	4
1.4. Scope and Limitation of the study	5
1.4.1. Scope	5
1.4.2. Limitation	5
1.5. Research Methodology	5
1.5.1. Literature Review	5
1.5.2. Discussion with Expert	6
1.5.3. Technique and Tools	6
1.6. Evaluation Technique	7
1.7. Application of Result and Beneficiary	7
1.8. Organization of the Report	7
Chapter Two	9
2. Literature Review	9
2.1. Introduction	9
2.2. Basic Concept of Parsing	9
2.2.1. Component of Parser	11
2.3. Machine Learning Technique	13
2.4. Related work	15

2.4.1. Automatic sentence parser for Oromo language using supervised learning technique.....	15
2.4.2. Syntax Parsing	19
2.4.3. Rule based shallow parser for Arabic language.....	20
2.4.4. Bootstrapping statistical processing into a rule based parser.....	23
2.4.5. Chunking with Support Vector Machines.....	24
Chapter Three.....	25
3. Afaan Oromo Grammar.....	25
3.1. Introduction.....	25
3.2. Overview of Afaan Oromo.....	25
3.3. Word class in Afaan Oromo	26
3.3.1. The Noun Class	26
3.3.2. The verb class.....	28
3.3.3. Adjective Class.....	29
3.3.4. Adverb class	31
3.3.5. Pre- and Post- Position class	31
3.4. Afaan Oromo Phrase Structure	33
3.4.1. Noun Phrases.....	35
3.4.2. Verbal Phrase	35
3.4.3. Adjective phrase.....	36
3.4.4. Adverb phrases.....	36
3.4.5. Pre- and Post- positional phrase.....	36
3.5. Sentence in Afaan Oromo.....	37
3.5.1. Types of sentences	39
1. Structurally.....	39
I. Simple Sentence.....	39
II. Compound Sentence	40

III.	Complex Sentence	40
IV.	Compound Complex Sentence.....	40
2.	Functionally.....	40
A.	Declarative Sentence.....	41
B.	Imperative Sentence.....	41
C.	Interrogative Sentence	41
D.	Exclamatory Sentence.....	41
Chapter Four	42
4.	Methodology of the study	42
4.1.	Introduction	42
4.2.	Rule Based Approach.....	42
4.3.	System Architecture	44
4.4.	Parser Algorithm	46
4.5.	Grammar Rule.....	46
4.5.1.	Sentence Structure Rule.....	47
4.5.2.	Noun Phrase Structure Rule.....	48
4.5.3.	Verb Phrase Structure Rule.....	50
4.5.4.	Adjective Phrase Structure Rule	51
4.5.5.	Adverb Phrase structure Rule	52
4.5.6.	Pre- and Post- Positional Phrase Structure Rule	52
4.6.	Parse Tree.....	52
4.7.	Support Vector Machine	55
4.8.	Hybrid Approach.....	57
4.9.	Training Set.....	57
4.9.1.	Corpus Preprocesses Module.....	58
4.10.	Learn patterns encoding syntactic structure.....	60
4.10.1.	Part-Of-Speech (POS) Tagger Pattern	60
4.10.2.	Morphological Information.....	62

Chapter Five.....	66
5. Experimentation, Result and Evaluation	66
5.1. Introduction	66
5.2. Support vector machine feature selection	66
5.3. Experimentation using SMO Function.....	68
5.4. Test Set.....	70
5.5. Evaluation of the parser	73
5.6. Discussion	75
Chapter Six.....	77
6. Conclusion and Recommendation	77
6.1. Summary	77
6.2. Conclusion.....	78
6.3. Recommendation.....	79
Reference	80

Lists of Table

Table 2:1: Result of automatic sentence parser for Oromo language [2]	18
Table 2: 2: Rule based Shallow parser result.....	22
Table 3. 1 Personal pronoun in Afaan Oromo	28
Table 4. 1. Tag sets used.....	62
Table 5. 1: Value of paired phrase classes	71
Table 5. 2: Detailed Accuracy of test set	72
Table 5. 3: Confusion Matrix of test set	72
Table 5. 4: Accuracy comparison with previous work of Afaan Oromo parser	75

Lists of Figure

Figure 2. 1. How supervised machine learning is work.....	14
Figure 2. 2: Sentence extraction algorithm parsing system[2]	18
Figure 3. 1: Afaan Oromo sentence structure tree	38
Figure 4. 1: Architecture of the system.....	45

Lists of Abbreviations

AI	Artificial Intelligence
Adj	Adjective
AdjP	Adjective phrase
AV	Adverb
AVP	Adverb phrase
CFG	Context Free Grammar
CONJ	Conjunction
Det	Determiner
dP	dependent Postposition
iP	independent Preposition
N	Noun
NLP	Natural Language Processing
NP	Noun phrase
SVM	Support Vector Machine
SMO	Sequential Minimal Optimizing
V	Verb
VP	Verb Phrase

Abstract

Nowadays, Natural Language Processing (NLP) concerns with the interaction between computers and human natural languages. The most difficult task in NLP is to learn natural languages for the computer. Parsing is one of the very important tasks in natural language processing. It is the task of analyzing the structural relationship between the words in a sentence. For a free word order language like Afaan Oromo, parser suits the best to extract the relation between the words in the sentences. Development of hybrid sentence parser for Afaan Oromo will avoid the large amount of time wasted to manually process sentences in the language to show its syntactic structure. The parser is also useful for semantic parsing which extracting meaning from a sentence and checking the well-formed-ness of a sentence, which is useful in a number of applications such as language teaching. Corpus used in this study as training and test set are manually parsed by researchers with linguistic advisor. Manually parsed sentences are given to machine for machine learning.

In this thesis, Weka tool is used for machine learning technique. The algorithm used for machine learning is support vector machine (SVM). The SVM algorithm is implemented using sequential minimal optimizing function (SMO). The features for the parser to machine learning include parts of speech, word and Lexicalized features. The algorithm achieved precision and recall of 82% for complex sentence parser and 89.5% for simple sentence. Accuracy of the result is 73.11%.

The model created for the parser differs from the previous work since the model developed includes machine leaning technique and also the tag set used is different. At the end, the developed model gives satisfactory results

Chapter One

1. Introduction

1.1. Back ground of the study

Natural language is the primary means of communication between individuals. It is the tool everyone uses to express the greater part of ideas and emotions. It shapes thought, has a structure, and carries meaning. Natural language processing (NLP) is concerned with the progress of computational models of human language processing. NLP refers to artificial intelligence (AI) method of communicating with an intelligent system which is overlapping in information and interferes significantly with the progress of linguistics with regard to the linguistic profile required for computers. Through the science of the software industry, we are able to analyze and simulate the understanding of natural language. It is the automated approach to analyze text that is based on a set of theories and a set of technologies together. In fact, Natural language processing has recently received attention in terms of research and development. Syntax analysis is a fundamental area of research in computational linguistics. Syntax analysis is used in key areas of computational linguistics such as machine translation, storytelling, question-answering, information retrieval and information extraction[1]. Identifying the syntactic structure is useful in determining the meaning of the sentence. The identification is done using a procedure known as parsing.

Semantic processing must operate on sentence constituents. If there is no syntactic parser, then the semantic system must decide on its own constituents. On the other hand, if parsing is done, i.e. if a parser is used as a component, it constrains the number of constituents that a semantic parser can consider. Syntactic parsing is computationally less expensive than is semantic processing (which may require substantial inference). Thus it can play a significant role in reducing overall system complexity.

Rule based parsing deals with the syntactic structure of a sentence. Syntactic analysis identifies certain patterns of words in a sentence as forming phrases of different types, such as noun phrases, verb phrases, adverbial phrases, adjectival phrases and prepositional phrases. Parsing, as defined by many people, is a procedure that explores various ways of combining grammatical rules to find a combination that generates a tree that could represent the structure of the input sentence. In other words, it is the step in which a flat input sentence is

converted into a hierarchical structure that corresponds to the units of meaning in the sentence. In parsing, a grammar and search strategy is used to assign a complete analysis for each sentence. Researchers have proposed a number of parsing methods for natural language sentences. [5], [6], [11], [12]. The rule based parser is provided with the set of rules used to identify the syntactic structure of a sentence and parsing them accordingly whereas the machine learning method learns the patterns form the example sentence in the training set annotated with syntactic structures. . Hence, in contrast to the rule based approach, machine learning algorithm is learns the syntactic structure of sentences form training sets. The algorithm used in this study is Support Vector Machines (SVM). Support Vector Machines is a supervised machine learning algorithm, which has achieved state of the art performance on learning tasks. In particular, SVM is a popular learning algorithm for natural language processing (NLP) tasks. After machine learns form training sets it creates model. The model is tested using test set.

As natural language processing was fast developing in a field of computer science, design and developing NLP task is important for each and individual natural language in the world. One of NLP task is parsing. So, Afaan Oromo needs good parser system that can be used as component for natural language applications like machine translation. But still not reach on the point where we say best Afaan Oromo sentence parser is developed. That's why; we develop rule based Afaan Oromo parser to solve the problem. Developing Afaan Oromo sentence parser using hybrid approach is our main target in this study. The most traditional understanding of parsing would be such where parsing is a mapping which assigns all possible syntactic structures to the input. This is inherent mainly to hand crafted systems. As for the automatic procedures parsing is not just a mapping but a function, a single structure is assigned to a single input sentence. There are two approaches that used for developing parsers: statistical and rule based approach.

Statistical approach is the method that associate grammar rules with a probability. Grammar rule are traditionally viewed in computational linguistics as defined the valid sentence in a language. Within this mindset, the idea of associating each rule with a probability then provides the relative frequency of any given grammar rule and by deduction, the probability of a complete parse for sentence. There are number of methods that statistical algorithms frequently use. Most statistical algorithms are based on a modified form of chart paring. The modifications are necessary to support an extremely large number of grammatical rules and search space. Although statistical

parsing algorithms can be programmed to output not only the most probable output but also the less probable ones, thus giving more than a single output.

There are a number of works carried out on sentence parser area in different languages including English. The main purpose of all these works are to enable computers understand human languages. However, as far as the knowledge of the researches, there is some research work done so far on Ethiopian languages, including Afaan Oromo like an automatic sentence parser for Oromo language using supervised learning technique[2], the concern of the present work. Thus, this study has taken the opportunity of carrying out a research on syntactic processing.

1.2. Statement of the problem

The task of the parser is to obtain the syntax structure for a given input sentence. This structure gives us the information as how the words and phrases in the sentences are related and what kind of relationship exists between the words and phrases in the sentence. From the syntax structure, the head, the modifier of words and phrases in the sentence can be identified. Then machine learning algorithm is learning from pattern of the sentence structure. This study is identifying sentence structure using grammar rule and give structure of the sentence to machine learning. Amount of information on the internet could be used to enhance development by making it accessible to the public. To fully localize and utilize these resources which are available on the Internet, translation of documents from one language to another may be necessary. Machine translation, which uses natural language as an input, and sentence parsers as a component, plays a great role in solving the translation problem. And hence, the need to develop an Afaan Oromo sentence parser.

Today there are a lot of parsing systems developed for various languages of the world including English . From then on wards, there have been a lot of attempts to develop sentence parsing to the languages in the world. As Afaan Oromo is one of such languages that should have rule based sentence parser, as far as the knowledge of the current study is concerns, there is no such kind of system developed so far for the language. So, development of such hybrid parsing system has some importance. Hence, this study addresses the problem and tries to develop a hybrid sentence parser for the Afaan Oromo and shades lights for further works in this area. The absence of good parsing in Afaan Oromo limits; if not completely delayed, later pains

of making computer understand Afaan Oromo. Hence, this study will try to fill such gap in the language.

In addition, the development of hybrid sentence parser for Afaan Oromo will avoid the large amount of time wasted to manually process sentences in the language to show its syntactic structure. The parser is also useful for semantic parsing which extracting meaning from a sentence and checking the well-formed-ness of a sentence, which is useful in a number of applications such as language teaching. All these benefits of the system necessitate the development of rule based parser for Oromo to reap the fruit.

1.3. Objective

1.3.1. General Objective

The general objective of this study is to design Afaan Oromo syntactic parser and combine with machine learner.

1.3.2. Specific Objective

The specific objective of the study is

- ✓ To Review related work and different techniques of parsing adopted for other languages.
- ✓ To identify Afaan Oromo word classes
- ✓ To identify the entire Noun Phrases (NPs), Verb Phrases (VPs), Adjective Phrases(AdjP), Adverb Phrases(AVBP), and Pre- and Post-positional Phrases (PPs) boundaries.
- ✓ To identify type of lexicon required for Oromo Parser and designs the appropriate rule for the parser.
- ✓ To explore grammar of the language and represent it in a computer.
- ✓ To develop SVM model
- ✓ Combine the rule based approach with the SVM model
- ✓ To evaluate the performance of the rule based, SVM and hybrid.

This study will discuss various idiosyncrasies of Afaan Oromo sentences to derive more accurate rules to detect start and the end boundaries of each clause in an Afaan Oromo sentence.

1.4. Scope and Limitation of the study

1.4.1. Scope

The scope of this study is confined to dealing with simple sentences and complex sentences of Afaan Oromo, Because of resource limitation in terms of time, cost and labor. Thus the current parser parses sentences of simple and some of complex types. It does not deal with compound and compound-complex sentences which consist of clauses as phrases in the sentences. Moreover, the study does not deal with complete features of all type of sentences.

1.4.2. Limitation

Although rule based parser is widely used in real, working natural language processing systems, they have the disadvantage that extensive amounts of dictionary data and labor to write the rules by highly skilled linguists are required in order to create, enhance, and maintain them. This is especially true if the parser is required to have broad coverage, i.e., if it is to be able to parse natural language text from many different domains what one might call general text. Due to those problems, the study has the following limitations:

1. All kinds of Afaan Oromo sentences are not included in this study. The sentences that are included in the training and testing dataset do not contain interrogative and imperative sentences.
2. The size of the corpus is very small. The corpus is prepared manually for the purpose of the work.
3. Ambiguity like Structural ambiguity that occurs when the grammar can assign more than one parses to a sentence.

1.5. Research Methodology

1.5.1. Literature Review

To accomplish the objectives of this study, various appropriate and related literature resources, i.e. books, research reports, journal articles, manuals, and other published and unpublished documents including those from the internet have been reviewed for the purpose of this study. All these helped the researcher understand both the issues regarding NLP,

particularly sentence parsing (e.g., approaches, techniques and strategies), and issues of the language considered (i.e., the basic word categories, morphological property, phrase structure, and the various types and kinds of sentences of Afaan Oromo). This understanding, in turn, has enabled the researcher to find the features of the language that have been found appropriate to the study, and to adopt the parsing algorithm appropriately.

1.5.2. Discussion with Expert

Successive discussion with linguists and experts in the area of Afaan Oromo at Jimma University/Institute Language Studies have also been made for better understanding and analysis of Afaan Oromo sentences and its sub components, particularly complex phrase structures of the language.

1.5.3. Technique and Tools

A. Technique

Several researchers applied different techniques to deal with parsing in several languages. These techniques are rule based, Stochastic based and the hybrid approach for sentence parser. Stochastic based parsers use probability in analyzing the problem of parsing. The stochastic approach is based on the ideas of Bayes (Network) theorem, independent events and the Markov assumption in sentence parsing. Thus, the approach uses these ideas to determine the most likely lexical sequence of each word in a given sentence. After well study from the control of the NLP applications, we are interest in hybrid approaches to solve common problems such as syntactic parsing. The advantages of rule based parser are that rules can be hand written and easily comprehended. Rule based approach needs lots of linguistic knowledge like Part of Speech tagging, grammatical relations, prepositional phrase attachments and morphological analysis. Means it needs high linguistic skill. So, that we combine with machine learning. Machine learning algorithm we use is SVM.

B. Tool

Weka tool is data mining software that uses a collection of machine learning algorithms. Named after a flightless New Zealand bird, Weka is a set of machine learning algorithms that can be applied to a data set directly, or called from your own Java code[3]. Machine learning systems crawl through the data to find the patterns and, when these are found, adjust the program's

actions accordingly. We used Weka for running SVM algorithm. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

1.6. Evaluation Technique

The evaluation plan is plan how the final work is evaluated to either it achieve it is a goal or not. This will be carrying out on a set of sentences take as test set. Test sets sentences are 100 select from the corpus. Test set has 75 simple sentences and 25 complex sentences. Those sentences are different from training sets. Those sentences are first manually parsed according linguistic knowledge into phrases. Then we give machine and identify detailed accuracy means precision, recall and F-score of set. After detailed accuracy identified, overall accuracy the test set is calculated and compare with accuracy of experiments. This is how the accuracy of the work is evaluated.

1.7. Application of Result and Beneficiary

Sentence parser is useful in many natural language processing systems. Researchers in the area of NLP of Afaan Oromo could get some advantage out of the output of the study. Specifically, researchers interested in area of semantic parsing, machine translation, Information Extraction, Information Retrieval, Text Summarization, and a variety of text-mining operations and etc. are among the top beneficiaries.

Linguists and students in the area of Afaan Oromo could also apply the output of this research to parse Oromo sentences automatically. The output can also be used in language teaching for recognition of phrasal categories, and to see the relationship between words in a sentence. Moreover, those who are interested in generating the syntactic structure can use it.

1.8. Organization of the Report

The research report is organized into six Chapters. The first Chapter is made up of the background of the study, the statement of the problem, objective of the study, significance of the study, scope of the study, limitation of the study, beneficiary of the research in this study and organization of the study. Different types of approaches and techniques of parsing are discussed in chapter two. Different components of parsers are presented in this chapter.

In chapter three also introduces Afaan Oromo language, word classes of Afaan Oromo, types of phrase in language and different types of sentences including simple, complex, compound, compound complex sentences is presented.

The core of this study is discussed in chapter four. In chapter four the approach, technique used for the parsing, algorithm, parse tree, machine learning techniques, sample rules of the phrase construction, training set, learn patterns encoding and some special keys in language are presented. Experimentation and evaluation techniques are discussed in chapter five. Finally, the conclusions and recommendations made based on the findings of the study are presented in chapter six.

Chapter Two

2. Literature Review

2.1. Introduction

This chapter is concerned with the review of literature. The chapter contains two sections. First section is explaining basic concept in parsing. General over view of parsing, different approaches and techniques to the task of sentence parsing are reviewed. The later sections of this chapter discuss related work to our study. For related works research papers and articles are selected based on the criteria related approach or the same approach with our study and the language we develop for. Goal of the chapter is to more explain area of the research and clarify the approaches in area by reviewing related works.

2.2. Basic Concept of Parsing

Parsing is an important process of natural language processing (NLP) and computational linguistics which is used to understand the syntax and semantics of a natural language sentences confined to the grammar. Parsing is actually related to the automatic analysis of texts according to a grammar. Technically, it is used to refer to practice of assigning syntactic structure to a text. It is a computational system which processes input sentence according to the productions of the grammar, and builds one or more constituent structures called parse trees which conform to the grammar.

Parsing or syntactic analysis is the process of analyzing a text, made of a sequence of tokens, to determine its grammatical structure with respect to a given (more or less) formal grammar. Parsing is a standard technique used in the field of natural language processing. Parsing means taking an input and producing some sort of structure for it. Before a syntactic parser can parse a sentence, it must be supplied with information about each word in the sentence. Parsing may be defined as the process of assigning structural descriptions to sequences of words in a natural language (or to sequences of symbols derived from word sequences). In another way, a parser accepts as input a sequence of words in some language and an abstract description of possible structural relations that may hold between words or sequences of words in the language, and produces as output zero or more structural descriptions of the input as permitted by the structural rule set. There will be zero descriptions if either the input sequence cannot be analyzed by the grammar, i.e. is ungrammatical, or if the parser is incomplete, i.e. fails to find all of the structure

the grammar permits. There will be more than one description if the input is ambiguous with respect to the grammar, i.e. if the grammar permits more than one analysis of the input.

The most traditional understanding of parsing would be such where parsing is a mapping which assigns all possible syntactic structures to the input. This is inherent mainly to hand crafted systems (e.g. manually designed formal grammar parsers). As for the automatic procedures parsing is not just a mapping but a function, a single structure is assigned to a single input sentence[4]. In the development of parsers for the purpose of examining how the syntactic structure of a sentence can be computed, it is a standard practice to consider two things: the grammar and the parsing technique. The grammar is a formal specification of the structures allowable in the language while the parsing technique is the method of analyzing a sentence to determine its structure by using the grammar as the source of syntactic knowledge. Parsing is the process of structuring a linear representation in accordance with a given grammar[5].

Parser is a process of assigning a labeled or unlabeled syntactic tree structure to a sentence at the input; the input originally does not exhibit any kind of syntactic information. The parser is contains the requirement its input and output. The input is a sentence or a word group acting as a sentence. The sentence can be associated with its morphological. The output is analytical syntactic structure of the input sentence[4]. In parsing two widely used approaches are Statistical and rule based approaches.

Statistical approach is the task of computing the most probable parse of a sentence given a probabilistic context free grammar (CFG). The weights of the probabilistic or weighted CFG are typically learned on a corpus of texts. In statistical parsing, grammar rules specify the structures allowable in the language, while probabilities specify the distributional regularities of sentence structures in the language. That is, probabilistic reasoning by way of statistical probabilities is introduced to assist reasoning. It means that linguistic specifications and statistical regularities of syntax are combined to be used for better syntax analysis [6]. But in rule based parsers, knowledge about the syntactic structure of a language is written in the form of linguistic rules, and these rules are applied by the parser to input text segments in order to produce the resulting parse trees[7]. In the Rule based approach there are two

ways in which parsing can be done. These are top down and bottom up parsing techniques. Both techniques are reviewed in related work under syntax parsing.

2.2.1. Component of Parser

Rule based system learns a set of rules automatically based on a given inputs and then parses sentences following these rules. Rule based parser consists of different components in the process of parsing a sentence. Those are:

A. Grammar Rule

The major component of any parser is the grammar rules. These are the rules that the parser consults every time it starts parsing a sentence. The grammar rules can be considered as a friendly guider to the system on what action should be and should not be taken. They are important for the parser to assign appropriate grammatical categories for each constituent in a sentence. Therefore, the inclusion of such rules in the knowledge base of the parser to be developed is inevitable.

A grammar can be defined as a description language plus a set of structural constraints according to which a parser attempts to analyze the symbol sequences presented to it. Put another way, a parser accepts as input a sequence of words in some language and an abstract description of possible structural relations that may hold between words or sequences of words in the language, and produces as output zero or more structural descriptions of the input as permitted by the structural rule set or the grammatical formalism. Grammar specifies two things:

- A grammar's weak generative capacity:- The set of grammatically correct sentences that are contained within the language
- A grammar's strong generative capacity:- The structure to be assigned to each grammatical sentence in the language.

Generally, the motivation for parsing is the belief that grammatical structure contributes to meaning and that discovering the grammatical structure of a natural language word sequence is a necessary step in determining the meaning of the sentence.

B. Lexicon

A lexicon or dictionary is another major component of rule based parser. Lexical analysis is used to convert sequence of characters into identified meaning full tokens (or sequence of tokens). A parser requires lexicon, which are lists of all the grammatical categories of words (categories of words means part-of-speech in linguistics) and phrases used in parsing process. It provides distinct coding for all classes of words having distinct grammatical behavior. The lexicon is important because as soon as the parser receives the input tokens (strings), it refers to this dictionary to parse the sentence into a syntactic tree structure. The lexicon contains a list of all possible lexical categories that the word can be assigned.

C. Morphological Rule

Morphological rules are also useful components in rule based approach. The morphological rules provide information useful to treat words that are not in the lexicon of the parser. In other words, such rules are useful to make reasonable guesses as to the grammatical categories of unknown words. For instance, assuming the English word worked is not in the lexicon, this word will morphologically be analyzed, i.e. using the lexical rules, as work + ed and it will be parsed as an arbitrary symbol for past tense of the lexical verb for this work. In line with these morphological rules, there is also morphological analyzer which is used to strip of some affixes from a word to get the word class. The base form with the syntactic feature is then passed to the syntactic parser to assign its grammatical categories. For example, assume Afaan Oromo sentence ‘Caaltuun gaba deemte’ [Chaltu went to market]. The word ‘Caaltuun’ is NP. If morphologically not analyzed, the phrase cannot be parsed into word class. When phrase can morphologically analyzed, it parsed into noun ‘Caaltuu’ and dependent position ‘-n’. so, morphological rule is more useful in Afaan Oromo rule based parser as words and phrases in language has full of pre- and post- positional that changes words into phrases or one word class to other in sentences.

Consequently, it is possible to guess the lexical categories of unknown words based on the morphological information stored in the Knowledge base. In treating words that are unknown during the parsing process, some systems include rules pertaining to capitalization and punctuation. These rules of capitalization and punctuation are additional rules. In corporate in such systems besides contextual and lexical rules. But, such information on capitalization and

punctuation may or may not be useful in the parsing process depending on the language being parsed. In Afaan Oromo, information on capitalization does not useful. In German, for instance, information about capitalization proves to be extremely useful in the parsing of unknown nouns. This indicates that if not all in all some of the morphological rules are language dependents.

Advantages of parsers developed using rule-based approaches require less storage than those developed using stochastic approach. That is, it is compact. In addition, some argue that parsers developed using rule based approaches are ten times faster than the fastest stochastic parsers. Also, enhance or modifying rule based parsers for the purpose of correcting errors is easy and straightforward. On the other hand, alteration to statistical parser is very difficult for it is hard to predict the effect of alteration the parameters of the system.

2.3. Machine Learning Technique

Machine learning is usually considered a broad subfield of artificial intelligence has grown in scope and popularity over recent years. Different researches took a different approach; they built a parser that employs a machine learning method to induce grammar rule. Improvements in computer speed and memory have allowed machine learning of larger patterns from a parsed corpus learns all possible substructures from a parsed corpus, and parses a new sentence by finding the optimal combination of sub trees to span the sentence[8].

Machine learning approach does not strictly follow explicit theory of linguistics. The approaches are completely based on training and testing corpora, which constitute the input data. Approaches in this category use some algorithms to learn, say about the phrase formation process of a language from a given corpus and perform the parsing based on this knowledge. There are two types of methods in the machine learning approach. They are: supervised and unsupervised methods[9].

Unsupervised approach use probability information generated from the test corpora. In unsupervised learning approach, the learners exclusively receive unlabeled data and infer unseen points. Since there is no labeled training data, the performance of the classifiers is generally difficult to quantitatively evaluate. Data will be given and the system will learn from the data itself and will create a model based on those data. This method of learning induces the model from the large corpora.

In Supervised approach, the learner receives a set of labeled data as training data. This strategy allows the learner to infer unseen points, classifying unknown data according to models built by learning the training data. The ultimate aim is to learn a mapping from the input to an output. It is fairly common in classification problems because the goal is often to get the computer to learn a classification system that we have created.

To learn a mapping from input x to output y , given a labeled set of input output pair $D = \{(x_i, y_i)\}_{N_i=1}$. Here D is called the training set, and N is the number of training example. In general x_i could be a complex structured object, such as a sentence in our concept etc.

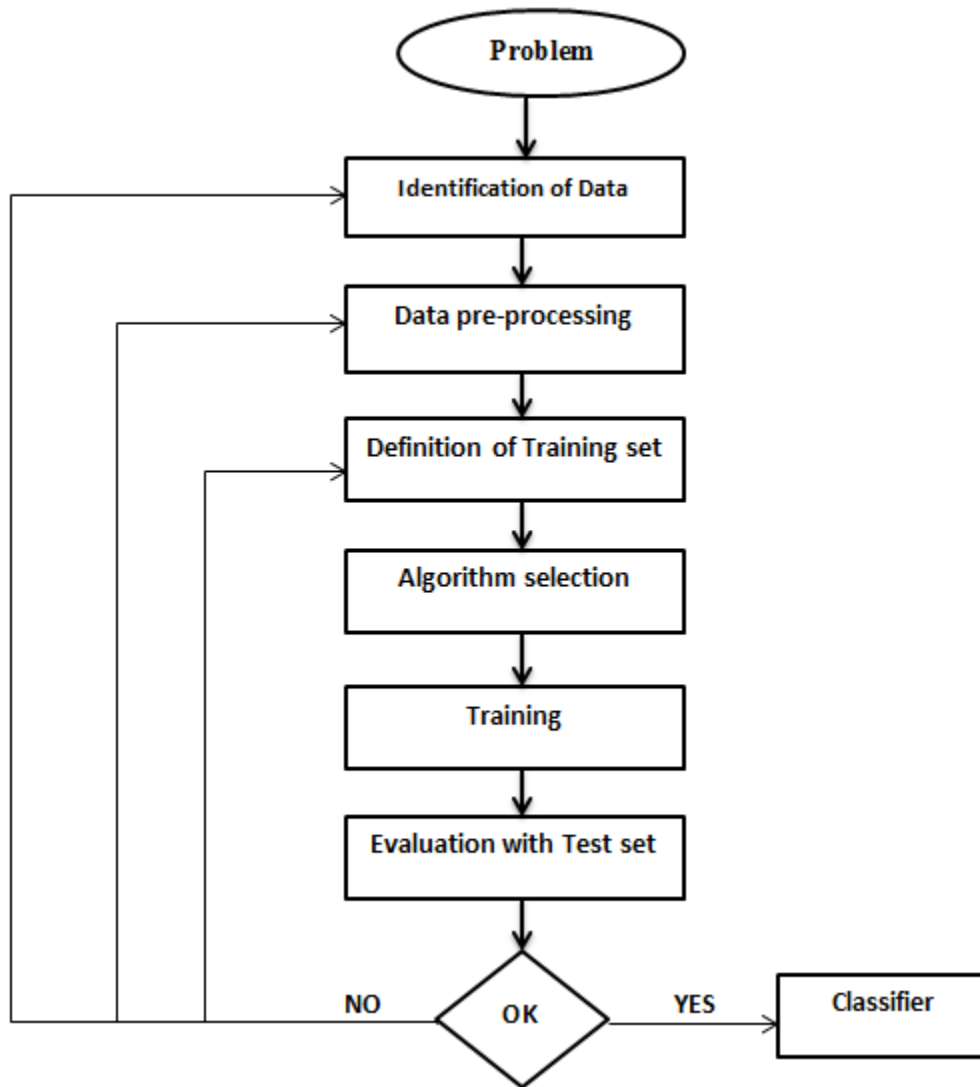


Figure 2. 1. How supervised machine learning is work

The mapping is realized by the classifier by learning the training data provided by the supervisor. It requires annotated text corpora. Annotated corpus will be given and the system will learn from the annotated corpus and will create a model based on the training set given to the system for learning. In this case, a teach input is provided which tells the system the output required for a given input.

When we discuss classification; Here the goal is to learn a mapping from inputs x to outputs y , where $y \in \{1, \dots, C\}$, with C being the number of classes. If $C = 2$, this is called binary classification (in which case we often assume $y \in \{0, 1\}$); if $C > 2$, this is called multiclass classification. If the class labels are not mutually exclusive we call it multi-label classification, but this is best viewed as predicting multiple related binary class labels (a so called multiple output model). When we use the term classification, we will mean multiclass classification with a single output, unless we state otherwise.

One way to formalize the problem is as function approximation. We assume $y = f(x)$ for some unknown function f , and the goal of learning is to estimate the function f given a labeled training set, and then to make predictions using $\hat{y} = \hat{f}(x)$ (We use hat symbol to denote an estimate). Our main goal is to make predictions on novel inputs, meaning ones that we have not seen before, since predicting the response on the training set is easy.

2.4. Related work

2.4.1. Automatic sentence parser for Oromo language using supervised learning technique

Automatic sentence parser for Oromo language is unpublished work of the author master's thesis. Parsing is a procedure that explores various ways of combining grammatical rules to find a combination that generates a tree that could represent the structure of the input sentence[2]. It is the step in which a flat input sentence is converted into a hierarchical structure that corresponds to the units of meaning in the sentence. The input string is passed to the parser token by token.

Automatic sentence parser uses hybrid approach means stochastic + Rule-based.

🚩 Stochastic Approach

Stochastic based parsers use probability in analyzing the problem of parsing. The stochastic approach is based on the ideas of Bayes Network theorem, independent events and the Markov

assumption in sentence parsing. The approach uses these ideas to determine the most likely lexical sequence of each word in a given sentence. Stochastic approach contains two sub approaches: supervised and unsupervised parsers based on the type of data they use.

The important information in a supervised stochastic parser is a lexicon (lists of each word with the entire possible lexical category) and list of contextual probabilities for each lexical category. Problem in developing supervised parsers is the lack automatically parsed corpora and it needs to manually parse each time the parser is applied to a new text. But if pre-tagged corpora are easily available, the Hidden Markov Model approach in particular and stochastic parsers in general can be adopted in new languages with little effort. But parsers developed using unsupervised stochastic technique does not require any pre-tagged corpora. For unsupervised learning, Pre-processed corpus is not necessary during the learning processes, training is more difficult for the set of state transitions used to generate the training corpus are not visible and they use different algorithms called Baum-Welch algorithm while supervised use the Viterbi algorithm.

The similarity that supervised and unsupervised approaches in stochastic parsers share are;

- Both of them assume the same underlying assumption called Hidden Markov Model.
- Both of them use a large dictionary and inflectional information to determine the possible lexical category for words in a corpus.
- Both of them involve calculation of parsers accuracy to improve performance and get better results.
- In both cases disambiguation can be achieved using statistical, hybrid or rule based approaches.

Rule based Approach

This approach is doesn't make any use statistics to parse a sentence. It is entirely based on the information from the knowledge base and some kind of learning technique, if any, to handle ambiguity and guess unknown words. It needs the knowledge of grammar rule, lexicon and morphological rules. Two ways in which parsing can do in rule based approach are: Top-down and Bottom-up parsing techniques.

The top down parsing begins with the start symbol (sentence) and apply the grammar rules forward until the symbols at the terminals of the tree correspond to the components of the

sentence being parsed. However, bottom up parsing begins with the sentence to be parsed and applies the grammar rules backward until a single tree whose terminals are the words of the sentence and whose top node is the start symbol (sentence) has been produced. That is, it starts from each word and assign its grammatical category until it reaches the start symbol.

Comparison of Stochastic vs. Rule-Based Approaches

These compare similarity and difference of two approaches.

✚ Similarity of two approaches are:

- Both of them may or may not need pre-processed corpora
- Both of them use contextual information from an already set of possible lexical category to disambiguate words.
- Until recently, none of them handle unknown words in a way that is completely portable.

✚ Difference between two approaches are:

- Rule based approach uses contextual and morphological information to deal with unknown words while stochastic approach uses probabilities to deal with such words.
- A stochastic approach has no rule-based mechanisms for disambiguation. Instead it use only probabilities for dealing with disambiguation.
- Rule-based parsers generally perform as good as or better than that of stochastic parsers.

The steps in automatic sentence parser are considering three things. Those are generally essential to automatically parse a sentence using an intelligent (hybrid of Rule-based and supervised learning) System Approach, an approach selected for the study. These are,

- Creation of a dictionary listing of the allowable grammatical categories for each word in the small tokens.
- Writing of grammar rule in the language
- Creation of a matrix of Mapped Rules

Parsing algorithm presents chart parser and other algorithms used to develop Oromo sentence Parser. It discusses part that actually performs the Parsing. The algorithm is sentence extraction algorithm. The sentence extraction algorithm is initiated when the button labeled Open File to parse from file is pressed. Then word extraction is called to split the complete word into sentence constituent parts. After the word is extracted from the input sentence, the chart parsing algorithm

is invoked which starts parsing the words in the sentence into its phrasal and lexical categories. Finally the tree construction is called and the tree is constructed together with the grid table which displays the information required after parsing. The block diagram below shows the parsing of the algorithm designed. Figure is original copy the author.

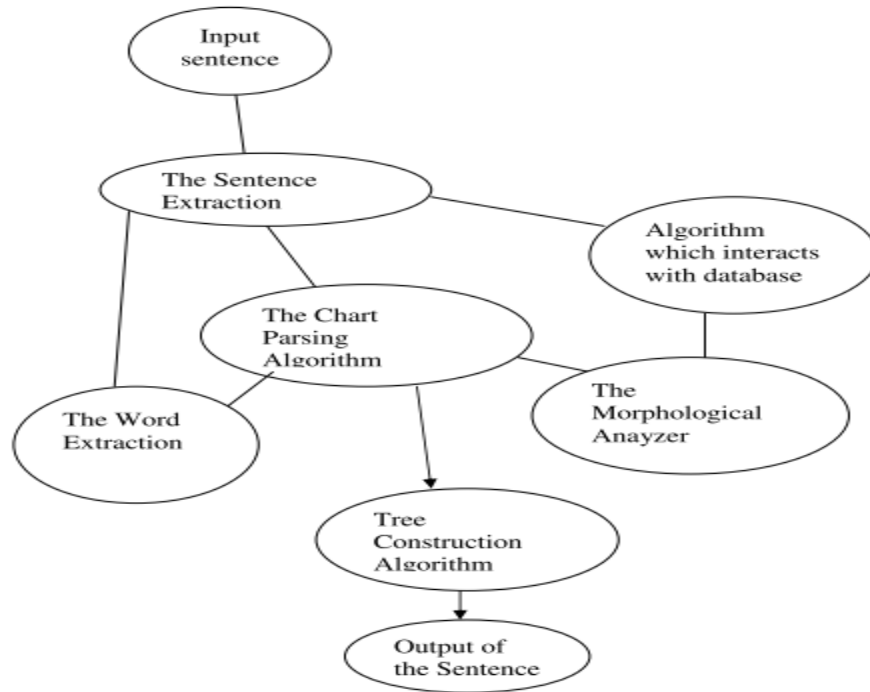


Figure 2. 2: Sentence extraction algorithm parsing system[2]

For the result of the parser, it uses two experiments, training set and test set. Training set use 300 and test set use 52 sentence. The following table indicates final result of the automatic sentence parser for Oromo language.

Date set	No of sentences	No of erroneously parsed sentences	Accuracy
Training set	300	15	95%
Test set	52	6	88.5%

Table 2:1: Result of automatic sentence parser for Oromo language [2]

2.4.2. Syntax Parsing

Syntactic parsing deals with syntactic structure of a sentence. The parsing is implemented using English grammar rules. In[10], use of identifying syntactic structure is to determine meaning of sentence. The identification is done using a procedure known as parsing. In many languages, words are brought together to form larger groups termed constituents or phrases, which can be modeled using context free grammar.

As the authors describe, Context free grammar (CFG) is a set of rules that expresses which elements can occur in a phrase and in what order. A CFG defines the syntax of a language but does not specify how structures are assigned. The task that uses the rewrite rules of a grammar to either generate a particular sequence of words or reconstruct its derivation is termed parsing.

The two approaches used for syntax parsing are Top down parsing and Bottom up parsing.

Top down parsing

The main concept in top down parsing is that the input can be derived from the chosen start symbol of the grammar. Then find all sub-trees which can start with symbol. To generate the sub trees of the second level search; it expands and root node using all the grammar rules with symbol on their left hand side. Also, all non-terminal symbols in the resulting sub-trees is expanded next using the grammar rules having a matching non-terminal symbol on their left hand side. The right hand side of grammar rules provides the nodes to be generated, which are the expanded recursively.

As the expansion of symbol continues, the tree grows downward and eventually reaches a point where the bottom of the tree consists only of part-of-speech (POS) categories. At this point, all trees whose leaves do not match words in the input sentence are rejected, leaving only trees that represent successful parses.

Bottom up Parsing

A bottom-up parsing approach is starts with the words or POS in the input sentence. Main target of this approach is to construct a parse tree in an upward direction towards the root. At each step, the parser looks for rules in the grammar where the right hand side matches some of the rule in

the parse tree constructed so far, and reduces it using the left hand side of the rule. The parse is considered successful parser if the parser reduces the tree to the start symbol of the grammar.

When the authors compare two approaches; each of these has advantages and disadvantages. In the top-down search starts generating trees with the start symbol. The grammar, it never wastes time exploring a tree leading to a different root. However, it wastes considerable time exploring symbol trees that eventually result in words that are inconsistent with the input. This is because a top down parser generates trees before seeing the input. But, in bottom-up parser never explores a tree that does not match the input. Still, it wastes time generating trees that have no chance of leading to a symbol rooted tree.

The algorithm designed for parsing by syntactic parsing methods is:

1. Accept a sentence.
2. Categorize English sentence into simple, complex, facts, interrogative, active, passive or Subject-object-verb (SVO) with adjective.
3. Check the phrases of sentences using various tags returned by part-of-speech (POS) tagger.
4. Partition the sentence into noun phrase (NP) and verb phrase (VP).
5. Parse the NP and VP by matching it against Grammar rules.
6. If all parts of the sentences are parsed correctly then sentence is syntactically correct, else the sentence is syntactically incorrect.

To verifying accuracy of the algorithm, the experimentation sample sets had chosen for different categories of sentences such as simple, complex, active, passive voice, questions and each holds 50 random sentences. So, overall 400 samples have been verified. The algorithm has accomplished an accuracy of 81%. The sample sentences and their corresponding syntactic understanding whether they are syntactically correct or not. The authors suggest accuracy of the system can be further increased through corpus training.

2.4.3. Rule based shallow parser for Arabic language

Syntactic analysis identifies certain patterns of words in sentence as forming phrases of different types, such as noun phrases, verb phrases and adjectival phrases[11]. Syntactic analysis categorized into full parsing and shallow parsing. In shallow parser for natural languages is often

separated into two major parsing of sentence parts is analysis without building a complete typical parser tree. It represents the task of recovering only a partial amount of syntactic information to identify phrases from natural language sentences and process of grouping consecutive words together to form phrases by a chunkier. Chunking does not provide information on how the phrases attach to each other. The structures generally specified by shallow parsers include phrasal heads and their immediate and unambiguous dependents and these structures are usually non-recursive. But full parser tree defines completely by specifying the syntactic relationships between all constituents.

A shallow parsing method can be more robust than a full parsing method in cases of low quality input, because sometimes in the input there exists noise, mistakes and missing words. Full parsing is expensive, is not very robust, much slower and it gives more information than needed. That means, partial parsing can be much faster, more robust and be sufficient for many natural language processing applications.

The authors use rule based approach to develop rule based shallow parser for Arabic language. The reason they select rule based approach is to solve common problems like partial syntactic parsing. The advantages of rule-based approaches are rules can be hand-written and easily comprehended. On the other hand, the disadvantages are that the rules are language and corpus specific and it takes a large amount of work and needs lots of linguistic knowledge.

A rule based constituent for the shallow parser is used when the input is a sequence of lexical trees with no constituent structure. The input data is prepared in a specific format and each line contains only a POS tag matching with the word in the sentence. The rule formalism has been designed specifically for group sequences structures to facilitate the dependency analysis. These rules are structured in layers that are applied on to the input sequences of sequential categories and they deal with syntactic structure and typical linguistic grammars to recognize several major categories of words in a language.

Rules are incrementally built and applied using corpus. Total designed rule contains 150 rules. The shallow parser then checks whether the first word of the input can belong to the category. The first 110 rules are implemented in the first stage to generate the first level from a shallow parser. The second rule set which contains 40 rules is run as a post-processing element in the second phase to generate the second level from a shallow parser.

Firstly, the hand crafted rules for the first level are derived, based on the experience through manual tagging for NP, VP and PP chunking. The rules will describe sequences of tagged words to identify the three types of chunks which are covered by rules. Those rules have two phases.

- ✚ **First Phase:** - The first aspect to consider is the rules that aim at identifying NP, VP and PP parse boundaries. The essential point is identifying phrase parsing.
- ✚ **Second Phase:-** Having developed first phase based on NP, VP, PP rules and some other linguistic grammatical requisites, the next step is to build grammatically correct phrases boundaries.

The result of shallow parser is evaluated by conducting a series of experiments which depend on the length of the full sentence as well as each phrase separately. The generic rule set was applied to all experiments to identify and then compare the phrase output of the system with a human chunking standard set of the input text. However, the careful choosing of the Part-of-Speech (POS) tag set has a directly impact on higher level syntactic processing. The accuracy of the chunkier heavily depends in turn on the accuracies of the POS tagging. For evaluation of the result, three experiments were performed on the same sentence. These experiments are divided into two phases. Firstly, the experiments are conducted for the first level of shallow parser in the first phase and then the experiments proceeded with the second level of shallow parser, in the second phase. The results obtained after executing all the experiments.

Finally, the result is presented by F-score. Over all F –scores of three chunkier (NP, VP and PP) and it is averages are listed as following table.

Over all F-score	
NP chunks	95.94 %
VP chunks	97.82%
PP chunks	97.5%
Average	97.08%

Table 2: 2: Rule based Shallow parser result

The average of chunks indicates accuracy of the parser. That means, the study confirmed that, the system performs with F-scores that are 97.08%.

2.4.4. Bootstrapping statistical processing into a rule based parser

The paper is describe bootstrapping method which uses a broad coverage, rule based parser to compute probabilities while parsing an untagged corpus of natural language text, and which then incorporates those probabilities into the processing of the same parser as it analyzes new text. In [12], both rule based and statistical approach are explained.

In rule based, knowledge about the syntactic structure of a language is written in the form of linguistic rules, and these rules are applied by the parser to input text segments in order to produce the resulting parse trees. While rule based parsers are widely used in real, working NLP systems, they have the disadvantage that extensive amounts of data and labor to write the rules by highly skilled linguists are required in order to create, improve, and modify them. But in statistical methods, parser acquires information from large corpora of natural language text, and on using that information in statistical natural language parsers. Instead of being stored in the form of dictionary data and grammatical rules, linguistic knowledge in these parsers is represented as statistical parameters, or probabilities. These probabilities are commonly used together with simpler, less specified dictionary data and rules. Advantages of the statistical method are decreases of amount of rule coding required to create a parser that performs adequately, and the ability to tune a parser to a particular type of text simply by extracting statistical information from the same type of text. Disadvantage of the approach is requirement of large amounts of training data.

The approach used in paper is "bootstrapping". This method uses a rule based parser to compute part-of-speech and rule probabilities while processing a large, non- annotated corpus. It begins by using the rule based parser to parse a large corpus of untagged natural language text. During parsing, frequencies that will be used to compute rule and part-of-speech probabilities are obtained. For These probabilities are then incorporated into the very same parser, thereby providing guidance to the parser as it assigns parts of speech to words and applies rules during the processing of new text. The approach is relies on the existence of a broad coverage, rule based parser. Benefits of this approach are that relevant statistical information can be obtained automatically from large untagged corpora, and that this information can be used to improve significantly the speed and accuracy of the parser.

Main disadvantage of the bootstrapping method is that the parser can reinforce its own bad behavior. However, this may be controlled by parsing a large amount of data and then by using only the probabilities computed for shorter sentences (in a paper, those less than 35 words) for which a single, well-formed parse is obtained. Other factor in avoiding the reinforcement of bad behavior is making linguist's skill that sure to the most common structures parse accurately.

2.4.5. Chunking with Support Vector Machines

Support vector machines (SVM) are the most complex machine learning technique explored in this paper, however they are also the most accurate and computationally inexpensive. SVM's are binary classifiers. This means that they are best used to separate one class of items from another. In the NP extraction case, they can be used to separate noun phrases from non-noun phrases. Kudo and Matsumoto[13] implemented SVM learning technique to identify English base phrases. They also applied weighted voting of 8 SVM based system to achieve higher accuracy. They derived weighting strategy from theoretical basis if the SVM for the weighted voting systems. They have used three annotated corpora for their experiments. The base NP standard dataset and base NP large data set which consists sections (15-18) and sections (02-21) of WSJ part of the Penn Treebank for the training data and section 20 and section 00 for the testing data, which are used for the noun phrase identification, respectively. The chunking data set that is used for all types of phrase identification also consists of sections (15-18) of the WSJ part of Penn Treebank for the training data and section 20 for the test data. As they reported their approach achieves 94.15% precision and 94.29% recall for baseNP-S data set, 95.62% precision and 95.93% recall for baseNP-L data set and 93.89% precision and 93.92% recall for the chunking dataset.

Chapter Three

3. Afaan Oromo Grammar

3.1. Introduction

A natural language is used as a tool for communication and people use it for communication by combining phonologies to form words, by combining words to form phrases and by combining phrases to form sentences. Afaan Oromo is one of most widely spoken in Ethiopia next to Amharic language, official language of the country. It contains own grammar rule like other natural languages. Most of them this grammar rules define characters of the languages. This chapter discusses the structure of Afaan Oromo word classes, phrases types and sentences formation with their types.

3.2. Overview of Afaan Oromo

Afaan Oromo is among the major languages that are widely spoken and used in Ethiopia. It is considered to be one of the five most widely spoken languages from among the roughly one thousand languages of Africa. Afaan Oromo, although relatively widely distributed within Ethiopia and some neighboring countries like Kenya and Somalia, is one of the most resource scarce languages.

Afaan Oromo is part of the lowland east Cushitic group within the Cushitic family of the Afro-Asiatic phylum. Although it is difficult to identify the actual number of Afaan Oromo speaking society (as a mother tongue) , due to lack of appropriate and current information sources, according to census taken in 2007 it was estimated that 34.5 percent of Ethiopians are ethnic Oromo.

Afaan Oromo is the fourth most widely spoken language of Africa (after Arabic, Hausa, and Swahili), It is one of the major language of Africa[14]. It is widely used as both written and spoken language in Ethiopia and neighboring countries like Kenya and Somalia . Currently, Afaan Oromo is an official language of Oromia Regional State (which is the largest region in Ethiopia) and used as an instructional media for primary and junior secondary schools of the region. It is also given as a subject starting from grade one throughout the schools of the region.

Furthermore, few literature work, newspaper, magazine, educational resource, official document and religious writings are written and published in this language.

3.3. Word class in Afaan Oromo

Word class in Afaan Oromo is some different from English word class. According to the [15], [16], Based on the context and form, In Afaan Oromo there are five types of main word classes. Those are noun, adjective, adverb, verb and pre- and post- position. The classifications of word types are almost common in most of the linguistic books[15]–[17]. According to [2] and linguistic experts define, there is five word class that are head to phrases.

3.3.1. The Noun Class

Like English and in other languages, Afaan Oromo nouns are words used to name or identify any of a class of things, people, places, ideas or a particular one of these. For this study, the Afaan Oromo noun class is considered to consist of nouns and pronouns.

Example: Caalaan sa'a bite. (Chala bought cow)

In above sentence the underline word is noun. The reason word 'Caalaa' to be noun is position Occupied. This gives a conclusion to some of subject word in Afaan Oromo sentence is noun. Moreover, two numbers are recognized in Oromo nouns: singular and plural. Except a noun ended by {-n} in Afaan Oromo; A singular noun is marked by zero morpheme whereas a plural noun is marked by various forms. But those ended by {-n} is the same for singular and plural.

The plural forms are not used as often in Oromo as they are in English. Typically, the plural form is used to specify that one is talking about more than one object where no other indicators are given. For example, in conversation the plural is rarely used when the noun is modified by a number. One would say “**muka lama**” for “two trees”, keeping **muka** [tree] in the singular, instead of “**mukkeen lama**”, where **mukkeen** is the plural of **muka**. When a plural noun in modified by an adjective, only the adjective shows plurality (discussed in next). In written text Oromo, plural forms tend to be more common, and may occur with numbers, adjectives, and other indicators.

When the plural form is used, there are several forms it may take. Typically, the final vowel is dropped and the correct suffix attached: *-oota*, *-toota*, *-lee*, *-een*, *-yyii*, *-wwan*, *-ootii*, or *-olii*. Unfortunately, the correct suffix cannot be predicted from the noun, meaning plural forms must be learned individually. Plural forms also vary across dialects, and multiple forms may be correct for some words. The most common suffix is *-oota*.

Examples:

<u>Singular</u>	<u>Plural</u>	<u>English</u>
Ilka	Ilkaan	Tooth
Wanta	Wantoota	Thing
Guyyaa	Guyyoota	Day
Gaara	Gaarreen	Mountain
Laga	Laggeen	River
Muka	Mukkeen	Tree
Waggaa	Waggoota	Year
Kitaaba	Kitaaboolii	Book

For nouns that may take either a masculine or feminine form, the feminine form is used as the stem to which the plural suffix is attached. For example, the plural of ‘barataa/ barattuu [student] is **barattoota**. Many nouns have irregular plural forms (e.g. biraa [another] while [biroo] others).

Derived Noun in Afaan Oromo

As expert define in [15], [17] noun can be derived from non-noun or other noun words by affixing nominal affixes. Those derived nouns are created in three different ways.

A. Noun derived from adjective.

Let we see the following example how nouns are derived from adjectives.

	Base word	Nominal Affixes	Drive Noun
Examples,	Bal’aa	(-ina)	Bal’ina
	Gabaabaa	(-ina)	Gabaabina
	Adii	(-eenya)	Addenya

Jabaa

(-ina/-eenya)

Jabina/jabeenya

B. Noun derived from base noun

	Base word	Nominal Affixes	Drive Noun
Examples,	Guyyaa	(-saa)	Guyyaasaa/Guyyee
	Galgala	(-oo)	Galgaloo
	Waaree	(-iyoo/-tee)	Waariyoo/Waaritee

According linguistic experts define in [15], [17], [18], There are also personal pronouns which are included under noun class. See the following table.

Person	Plural /singular	Case		Saalaa[Sex]
		Mathima (Subject)	Antima (Object)	
First	Singular	Ani [I]	Ana [Me]	General
	Plural	Nuti [We]	Nuti [Us]	
Second	Singular	Sii [You]	Sii [You]	General
	Plural	Sii [You]	Sii [You]	
Third	Singular	Inni [He]	Isa [Him]	Male
	Singular	Ishee [She]	Ishii [Her]	Female
	Plural	Isaan [They]	Isaan [They]	General

Table 3. 1 Personal pronoun in Afaan Oromo

3.3.2. The verb class

An important property of the Afaan Oromo verbs is that any word that comes at the end of a complete grammatical Afaan Oromo sentence is a verb[15]. It is independent words. As a consequence of this property a word at the end of such a sentence is expected to be tagged as a verb by an Afaan Oromo tagger. Example in following sentences;

- Daraartuun kaleessa kitaaba ishee **gatte**
- Finfinneen handhuura oromiyaatti **argamti**
- Tolaan sanga diimaa guddaa **bite**
- Aadde sooromeen gara gabaa **deemte**

In all above sentences the underline words ‘gatte’, ‘argamti’, ‘bite’ and ‘deemte’ in respectively to sentences are verbs. Characters of these all words are to terminate the sentences within.

3.3.3. Adjective Class

In Afaan Oromo words that come after noun or pronoun to modify noun or pronoun is adjectives.

Example: Inni buna **Jimmaa** bite. [He bought Jimma coffee.]

In this sentence the word ‘Jimmaa’ declare type of ‘buna’ [coffee] he bought. Also it has some position difference from English adjectives. It is not come before noun or pronoun.

Afaan Oromo adjective can be male, female, or neutral. Masculine adjective are used with masculine noun, feminine adjective modify feminine nouns, and neutral adjectives can be used with any noun. All non-neutral adjective can be made masculine or feminine by attaching the appropriate suffix. Masculine suffixes for adjectives are: *-aa*, *-aawaa*, *-acha*, and *-eessa*. Feminine suffixes are: *-oo*, *-tuu*, *-ooftuu*, and *-eettii*. Standard morphology rules apply when attaching suffixes.

Examples:

<u>Masculine</u>	<u>Feminine</u>	<u>English meaning</u>
Bareedaa	Bareedduu	Beautiful
mi'aawaa	mi'ooftuu	Sweet
xinnaa, xiqqaa	xinnoo, xiqqoo	Small
Boosacha	Booseettii	Messy
Gurraacha	Gurraattii	Black

Hiyyeessa	Hiyyeettii	Poor
Godeessa	Godeettii	Skinny

Neutral adjectives (e.g. **adii** – “white”) use the same form for both masculine and feminine nouns. When adjectives are used to modify a noun, typically the noun remains in the singular and number is shown by the adjective only. Plural adjective are formed by repeating the first syllable.

Examples:

<u>Singular</u>	<u>Plural</u>	<u>English</u>
Adii	Adaadii	White
bareedduu	Babbareedduu	beautiful
Gogaa	Goggogaa	Dry

Some masculine adjectives will change their ending to -oo when pluralized. Some of these do not repeat the first syllable as a plural marker.

Examples:

<u>Singular</u>	<u>Plural</u>	<u>English</u>
Beekaa	Beekoo	knowledgeable
Cimaa	Ciccimmoo	Strong
Guddaa	Guguddoo	Large
olaanaa	Olaanoo	High

In written Oromo, the noun may be pluralized as well as the adjective, so that ‘**nama sosoressa lama**’ and ‘**namoota sossooreyyii lama**’ are correct ways to say [two rich people]. In conversational Oromo, the first method, keeping the noun in the singular, is more common.

3.3.4. Adverb class

In Afaan Oromo, adverbs are used to modify the coming verbs. Adverbs always come before the modified verb but it should be noted that any words come before verbs cannot be always considered as an adverb.

Example: Caaltuun kaleessa dhufte. [Chaltu was come yesterday]

In this example, the adverb ‘kaleessa’ [yesterday] precedes the verb ‘dhufte’ [came] that it modifies. However, it should be noted that any word that comes before a verb is not necessarily an adverbs.

In their nature, adverbs can be found either in their primitive form or compound form as grouping of preposition and other word categories. Adverbs indicate manner, time, place, cause, or degree and answers questions such as ‘akkamitti’ [how], ‘yoom’ [when], ‘essaa’ [where]. The primitive adverbs are very few in number and these are: ‘haamayyu’ [yet], ‘daddafii’ [quickly] and etc.

3.3.5. Pre- and Post- Position class

Prepositions give meanings only if they combine with other words such as noun, adjective, verb, etc, unless they have no meaning. Pre- and post- positions link with nouns, pronouns and with other words in a sentence. The main properties of pre- and post- positions are: they never use affixes and they don’t assist to form other words. According to [19], A preposition links a noun to an action (e.g., “achirraa deemi”) or to another noun (“biriin minjaalarraa jia”). For the purpose of clarity, this section will divide Oromo prepositions into two categories: prepositions and postpositions, with prepositions coming before the noun and postpositions coming after the noun they relate to.

Some Common Prepositions and Postpositions:

Postpositions

‘ala’ — [out, outside]
‘bira’ — [beside, with, around]
‘booda’ — [after]
‘cinaa’ — [beside, near, next to]

Prepositions

‘gara’ — [towards]
‘eega’, ‘erga’ — [since, from, after]
‘haga’, ‘hanga’ — [until]
‘hamma’ — [up to, as much as]

‘dur, dura’ — [before]

‘akka’ — [like, as]

‘duuba’ — [behind, back of]

‘waa'ee’ — [about, in regard to]

‘irra’ — [on]

‘irraa’ — [from]

‘itti’ — [to, at, in]

Examples:

boqonnaarra (boqonnaa irra) – [on vacation]

mana keessa – [in the house]

waaree booda – [afternoon]

irra deebi’i – repeat [lit. return on it]

mana nyaataa kanatti – [at this restaurant]

waa'ee fiilmii sun natti himi – [tell me about that film]

Chaaltuun akka Hawwiituu baratuu dha. – [Chaltu is a student like Hawitu.]

hanga torban dhufu – [until next week]

gammachuu wajjin – [with pleasure]

shaayee annan malee – [tea without milk]

Ani meetirii lama gadi. – [I am below (shorter than) 2 meters.]

Keeniyaan Itoophiyaarraa (gara) kibbatti argamti – [Kenya is located (to the) south of Ethiopia]

From the examples above, you may notice that the postpositions itti, irra, and irraa most often occur as suffixes, *-tti*, *-rra*, and *-rraa*, on the nouns they relate to. Often with place names, no preposition or postposition is used to be mean “in”. Therefore, one can say ‘Jimmaa jiratta’ for [you live in Jimma], or ‘hospitaalan ture’ for [I was in the hospital], using no preposition. Personal pronouns are not used with prepositions. Instead, possessive pronouns are used as personal pronouns.

Examples:

gara koo(tti) (it not similar ‘gara na’) --- [toward me]

akka keenya — [like us]

akka isaatti — [according to him]

waa'ee kee — [about you]

Postpositions, on the other hand, take the accusative form of personal pronouns.

Examples:

sitti --- [at you]

narraa ---[from me]

isa jala ---- under him

When an adjective modifies a noun, the postposition follows the adjective, as in ‘nama guddarraa’ [from the big man].

3.4. Afaan Oromo Phrase Structure

A phrase is a structure in a language which is constructed from one or more words in the language. Phrases are composed of either only head word or other words or phrases with the head combination. The other words or phrases that are combined with the head in phrase construction can be specifies, modifiers and complements. As Afaan Oromo linguistic book [15] report, the phrase is part of sentence.

In Afaan Oromo; Single part of sentence or group words to be a phrase there is some criteria that identified by linguistic expert. A phrase is to be phrase one of the following three criteria must fulfilled [16].

✓ Movement

Order of word in Afaan Oromo sentences is allowed by Afaan Oromo grammar. When move/interchange place those phrases are move with each. i.e. when it move in sentence moves as one part. See the example;

Examples:

- a. Bishaan [fayyaa namaatiif] gaariidha. [water is good for health of man]
- b. [Fayyaa namaatiif] bishaan gaariidha.

In both sentence the part ‘[Fayyaa namaatiif]’ [for health of man] is phrase. We cannot interchange the place without each to give full meaning.

✓ **Replacement**

To be phrase, those words that move as one word must be replaced by other word. One of this known replacement word is pronoun. Let we see following example;

- a. Boontuun [meeshaa taphaa mucaa isheeti bitte.]
- b. Kumeeni[s.]
- c. Kumeeni[s **akkasuma.**]
- d. Kumeeni[s **sanuma** mucaa ishiitii] bitte.

✓ **Interconnection**

According to this criterion, any word can’t enter between parts of sentence that construct single sentences.

Examples:

- a. Sangaa diimaa guddaa kaleesssa argine sana
- b. Sangaa **qalle** diimaa guddaa kaleesssa argine sana
- c. Sangaa diimaa **qalle** guddaa kaleesssa argine sana
- d. Sangaa diimaa guddaa **qalle** kaleesssa argine sana
- e. Sangaa diimaa guddaa kaleesssa **qalle** argine sana
- f. Sangaa diimaa guddaa kaleesssa argine **qalle** sana
- g. Sangaa diimaa guddaa kaleesssa argine sana **qalle.**

In Afaan Oromo grammar only (a & g) is correct. ‘a’ is phrase and ‘g’ is sentence. Else others are grammatically incorrect.

With above three criteria, In Afaan Oromo there are five phrase classes, which are noun phrases, verb phrases, adjectival phrases, adverbial phrases and prepositional phrases [15]. Those phrases are constructed from word classes. i.e.

Examples:

- a) [sangaa]

- b) [sangaa diimaa]
- c) [sangaa diimaa guddaa]

3.4.1. Noun Phrases

Noun phrases are consists of a noun or pronoun and other related words that modify the noun or pronoun. It consists of noun or pronoun as head word and other words which come after or before the noun. The simplest NP consists of a single noun (e.g. Gammachuu) or pronoun such as ‘Inni’ [he], ‘Isheen’ [she], ‘Isaan’ [they], etc. A complex NP can consists of a noun (called head) and other constituents (like complements, specifiers, adverbial and adjectival modifiers) that modify the head from different aspects.

Eg. Gammachuun mana citaa guddaa kallessa ijaare.

In this example the head word is noun ‘mana’ [house] that indicates phrase is noun phrase. But other words are use as relate words to more modify head word. The parsed structure of this sentence is:

S

(NP (N Gammachuu) (dP -n))

(VP (NP (NP (N mana) (N citaa)) (Adj guddaa))

VP (AVP (AVB (kallessa)) (VB ijaare))

3.4.2. Verbal Phrase

When we say verbal phrase we must see main components of sentences [Hima] in Afaan Oromo. In Afaan Oromo sentence structure verbs are found at the end of sentences. Example, In sentence ‘Leensaan xalayaa barreessite.’ [Lensa wrote letter], ‘barreessite’[wrote] is head word and ‘xalayaa’[letter] is predicate. When categories sentence ‘Leensaan xalayaa barreessite.’ in to noun phrase and verb phrase, ‘Leensaan’ is noun phrase and ‘xalayaa barreessite’ is verb phrase. The head word to phrase ‘xalayaa barreessite’ is verb ‘barreessite’ that decided phrase is verb phrase.

3.4.3. Adjective phrase

To take part of sentence is as adjective phrase or word or group of words are adjective phrase; head word of that phrase must be adjective. Example, ‘Caalaan akkuma abbaa cimaadha.’ [Chala is strong as his father]. ‘akkuma abbaa cimaa’ is adjective phrase in verb phrase ‘akkuma abbaa cimaadha’. For this adjective phrase head word is a word ‘cimaa’. Because; concentration of adjective phrase ‘akkuma abbaa cimaa’ is based on strengthens of chala not about chala like his father. That is why ‘cimaa’ is head word to the phrase.

3.4.4. Adverb phrases

An adverb phrase is a phrase that is adverb is head word to phrases. In Afaan Oromo, adverbs are used to modify the coming verbs. Adverbs always come before the modified verb but it should be noted that any words come before verbs cannot be always considered as an adverb. In Afaan Oromo the main difference of adverb phrases from other phrase is adverb phrase is never come with other words [15]. Adverbs indicate manner, time, place, cause, or degree. Sometimes it is single word in sentences. In following example all the underline one is adverb phrases.

- Eg. 1) Guutaan har’a dhufe. [Guta is come today.]
2) leencichi kallessa ajjefame. [The lion is killed yesterday.]

In above sentence the underline words ‘har’a’ and ‘kallessa’ are adverbs.

3.4.5. Pre- and Post- positional phrase

Afaan Oromo pre- and post- positional phrase (PPhr) is constructed from a pre- or post-position (PP) head and other constituents such as nouns, noun phrases, verbs, verb phrases, etc. that means there is no phrase that construct from only pre- or post- position. In Afaan Oromo prepositional phrases are not similar to other language like English. As in [20], there are not many possible forms for prepositional phrases in English, though adverbs can act as modifiers to prepositional phrases.

Example: post positional phrase with dependent post positional.

- a) Meeshaadhan
- b) Mukatti
- c) Mukarra
- d) Fardaan

In example (a-d) all are postpositional phrases. For these phrases head words are the bold once. Those are dependent one. But they have a right to guide structure that creates post positional phrases. In next example let we see prepositional and post positional phrase that created from independent prepositional.

Example: prepositional and post positional phrase with independent pre- and post- positional.

- a. **Gara** manaa
- b. Mana **jala**
- c. **Waa'ee** mataaaa
- d. Garaa **keessa**

The bold words are head word to those phrases. Positions are either left or right of the structure. The difference of prepositional phrases from other phrases are head word can be either left or right of the structure. Also, dependent and independent prepositional can be taken as head in structure at the same time. However, it guides the structure by order or can't guide the structure at the same time. Let we see with example.

Example: prepositional phrase with independent and dependent prepositional in the same structure.

- a. **Gara** manaatti
- b. **Waa'ee** kootiif

In above phrases a and b there is independent and dependent prepositional. But both can't guide the phrase. In both phrases the word '**gara**' and '**waa'ee**' is guide pre- and post- positional phrases '**manaatti**' and '**kootiif**'.

3.5. Sentence in Afaan Oromo

A sentence is group of words that are correctly structured to give one meaning. This definition is common in any language. But may be the structure is different based on the language. In English sentence must have subject, object, verb (S+O+V). But in Afaan Oromo, there is some difference. According to [15][16], sentence is a complete thought or idea-subject + predicate.. So, to be a sentence in Afaan Oromo it must have a subject(S) and a Verb (V). Structure of Afaan Oromo sentence look like this:-[16]

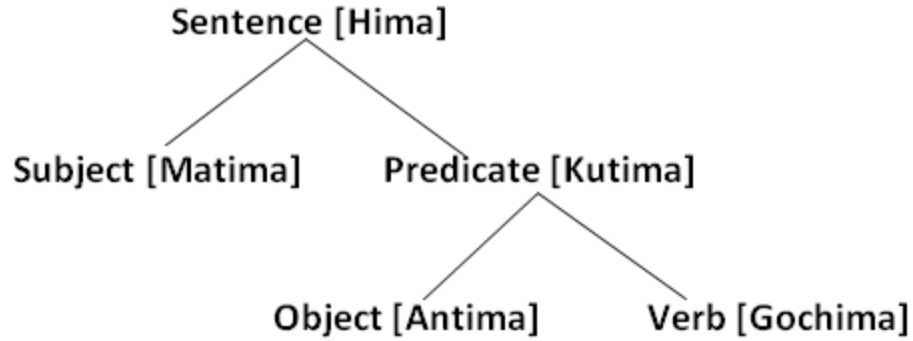


Figure 3. 1: Afaan Oromo sentence structure tree

Some sentences may have adjectives, adverbs and conjunctions. But that is not the case to define main structure (component) of the sentences. Let us see some of them in the following:

Example 3.1:

Waraabessi yuuse.

Margaan barsiisaadha.

Lagni Hawwaas guute.

A. Subject (Noun Phrase)

Subject (Noun Phrase) one of two main parts of a sentence containing the subject noun or a pronoun person, place or thing often accompanied by modifiers. Therefore, the noun or pronoun is who or what the sentence is about.

If noun take as subject, it takes prefix like ‘-ni’, ‘-n’, ‘-ti’, ‘-i’ and may be zero morpheme (%). Those morphemes are used in different ways.

- ✓ ‘ni’, ‘ti’, and ‘i’:- these are affixed on noun that end with single vowel letters. Among these three morphemes ‘ni’ is mostly used.

Example: Mukni/Mukti/muki jige.

- ✓ ‘n’ is affixed with noun end with double vowel letters to be noun as subject of the sentence.

Example: Galataan arba ajjesse.

- ✓ Zero morphemes (Φ) are noun end with 'n' letters is no change when we take noun as subject of the sentence.

Example: Bishaan dhuge.

B. Predicate

Predicate is one of two main parts of a sentence containing the verb, objects, or phrases governed by the verb. Basically sentence is made up of a noun and a verb. Verb is words that more declare a subject in sentence. In addition to subject and verb the sentence may have other words. Those words are used declare more subject and verb clearly. Verb is always at the end of the sentence in Afaan Oromo. In example 3.1; 'yuuse', 'barsiisadha' and 'guute' are verbs. The object of a sentence is the noun or pronoun directly related to and affected by the subject's action (verb). The object is not who or what a sentence is mainly about; it's not the focus of the sentence.

3.5.1. Types of sentences

Basically Afaan Oromo sentences can be categorized in to different types based on structurally and functionally[16].

1. Structurally

Structurally, Afaan Oromo sentences can be categorized in to four types. Namely, Leexima[simple], Dachima[Compound], Xaxima[Complex] and Dachima xaxima[Compound Complex] sentences.

I. Simple Sentence

Simple sentence in Afaan Oromo contains subject (noun/noun phrase) and a predicate (Verb/verb phrase). It communicates one complete idea as an independent clause. It can be one independent clause. It's a complete sentence. To more explain let we see the following example.

- i. Tolaan mana ijaare.
- ii. Biliseen, intalli obbo Margaa, barsiistuu taate.

In example above both sentences are simple sentences because it has single verb. But subject of the sentence can be contains other words to more modify.

II. Compound Sentence

Compound sentence is constructed from the combination of two and more than two simple sentence or two and more than two independent clauses. In Afaan Oromo two or more simple sentences are combined to construct compound sentence, there are different techniques is there. Among those techniques some of them a using conjunction ‘fi’ [and], using semicolon (;) and make affix {-e} double on verbs are techniques we can call.

Example:

- I. Namni gaariidha.
- II. Namni mana ijaare.
- III. Namni horii horsiise.

The above three examples are simple sentences we can construct compound sentences from this three sentences, it create compound sentences ‘**Namni mana ijaarefi namni horii horsiise gaariidha**’.

III. Complex Sentence

A complex sentence includes a dependent clause linked to an independent clause by a subordinating conjunction of some kind to form a complete sentence. It contains a dependent clause and one or more independent clause. In complex sentence subordinating conjunction is affixed on dependent clause. Let we see in example.

- I. Yoo dhufuu baattellee, xalayaa naaf barreessi.
- II. Yoo finfinnee deemteef, meeshaa naa bitta.

IV. Compound Complex Sentence

Compound complex sentence is a sentence that contains one or more dependent clause and one or more independent clause. Example;

- I. Yommuu deemtuu fi yommuu deebitu naa dubbisii darbi.
- II. “sagal elmamus sagaltamni elmamus kan koo qiraaciidhuumaatti” jette adurreen.

2. Functionally

Whereas sentence structure refers to the form of sentences in a language, sentence purpose refers to the function of sentences. Four types of sentence purposes exist in Afaan Oromo are similar to

the other like English language: declarative sentences, interrogative sentences, imperative sentences, and exclamatory sentences.

A. Declarative Sentence

The first type of sentence in the Afaan Oromo is the declarative sentence. Declarative sentences, or declarations, convey information or make statements. They usually provide information, and are used to make statements. For example:

- Barattootni daree jiruu [Students are in class]

B. Imperative Sentence

The third type of sentence in the Afaan Oromo is the imperative sentence. Imperative sentences, or imperatives, make commands or requests. For example:

- Hojii manaa hojjedhu [Do homework]

C. Interrogative Sentence

The second type of sentence in the Afaan Oromo is the interrogative sentence. Interrogative sentences, or questions, request information or ask questions. It is a type of sentence that always has a question mark at the end. For example;

- Yoom biyya deemta? [When you go to your homeland]

D. Exclamatory Sentence

The fourth type of sentence in the Afaan Oromo is the exclamatory sentence. Exclamatory sentences, or exclamations, show emphasis. Unlike the other three sentences purpose, exclamatory sentences are not a distinct sentence type. Instead, declarative, interrogative, and imperative sentence become exclamatory through added emphasis. For example;

- Na gargaarii?

Chapter Four

4. Methodology of the study

4.1. Introduction

This chapter may be considered as the core of this study. Based on the assumptions and approaches discussed the syntactic property of Afaan Oromo reviewed in chapter two and chapter three. The chapter discusses over all how rule based parser approach is designed and how it has been employed to parser Afaan Oromo text. The chapter begins by discussing the approaches taken to develop the parser, and then moves to the second section which presents the system architecture using block diagram to show how the system works. Section three explains the algorithm followed to develop the parser. Then fourth section discusses the model of rules and the fifth section is presents the grammatical parse tree of the sentences.

Section six is describes the machine learning algorithm which learns the parsing patterns and section seven describes the details of training set used in the study. Section eight is discuss pattern learned in studying; part of speech pattern in Afaan Oromo sentence and, following that, the part of speech tagger developed by [21] and how it was embedded in the development of the parser is described. Also this section is deal with a simple morphological analyzer and the section of this chapter section nine is describe special keywords identified in this study.

4.2. Rule Based Approach

The Rule based approach does not make any use of probability to parse a sentence. It is entirely based on the information from the knowledge base and some kind of learning technique. It learn a set of rules from grammar rule of language and then parses sentences based on syntax rule of a language. Rule developed for one language is not compatible for other language. The reason is most of the natural language structure and grammar rule are different from each other. Example in English and Afaan Oromo at least minimum complete sentence contains the structure (subject, verb and object) and order of the structure is different. In the following complete simple sentence in both Afaan Oromo and English are:

✚ Inni dhufe [He was come].

✚ Tolaan barata cimaadha [Tola is clever student].

The first sentence contains two words means only subject and verb. The first word in first Afaan Oromo sentence is pronoun ‘Inni’[He] that used as subject of the sentence and the second word is verb ‘dhufe’[came] that is also used as verb in sentence. That means the structure of this sentence is (subject + verb) that as clearly explained in chapter 3 under section 3.5 of this study. But the translated sentence to English is different. The sentence contains structure (subject + verb + object). From this, what we understand is at least Afaan Oromo complete sentence contains subject and verb which is different from English complete sentence which contains subject, verb and object. Also, if Afaan Oromo sentence contains the structure subject and predicate (verb and object), meaning it contains the structure (subject, verb and object). Order of the structure in sentence is (subject + object +verb). This is different from English structure (subject + verb +object). As grammar rule and structure in each language is different, the rules developed in one language might not be directly applied to other language.

The rules developed in rule based approach for this study are developed by the researchers. During developing the rules, researchers use linguistic books, learn rules from training set and take advice from linguistic adviser. Developed rule are take grammar rules as central concept. Because before developing rule, grammar rule of the language must be clearly understand. Rules are grammar rule written by researchers and the system is learns from rule using supervised learning technique.

Rule based approach has two components that can usually distinguish in an analyzer; a declarative component corresponding to linguistic knowledge and a procedural component which represent the analysis/generation strategy. Linguistic knowledge includes the grammar and the lexicon of the language while analysis strategy is an algorithm which specifies in detail each of the operations involved in the process of analysis.

The real power of the rule based approach is ability to learn the rules from a syntactically preprocessed input. The supervised learning is performed on a rather small set of manually syntactically processed sentences. Manually in this sense means correctly syntactically analyzed sentences. The set of manually analyzed sentences used for the process of learning is usually referred as the accurate. A generalization of this type of algorithm is familiar from computer science. Under this method, a parse tree is represented as a sequence of decisions made by the

parser, so that generating a parse tree for a sentence is a series of classification problems in that the parser needs to decide the next move given previously built structure.

✓ **Problem of rule based approach**

Challenges of pure rule based approach are:

- Writing all rules is need good linguistic skill. So including all rules for parser is difficult due to natural language has complex structure. Also getting phrase type in a sentence is challenging.
- Updating rule is need perfection.

4.3. System Architecture

The implementation of parsing involves a sequence of steps. The input sentences are sent to the POS tagger to get the POS tagged data. And input sentence is tokenized and sent to morphological analysis to get the morphological data. Now the words contain the POS tags, the morphological tags and word sequence. Now construct sentence structural pattern (model). Then machine learning algorithm learns from sentence structural pattern. After taking structural pattern it train with training set. After train with training set; Evaluated with test set. The architecture of the system is look like this:

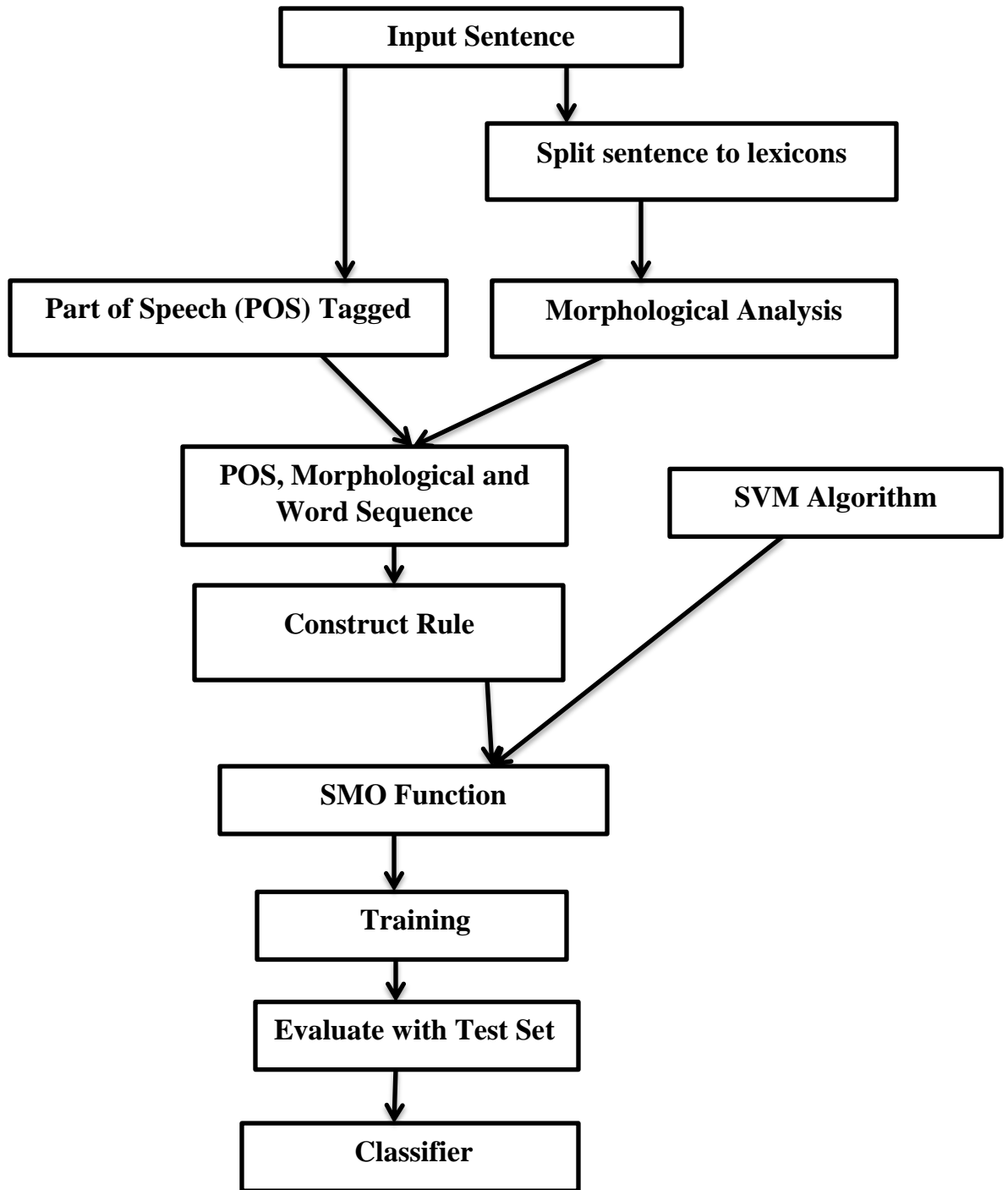


Figure 4. 1: Architecture of the system

4.4. Parser Algorithm

Parser algorithm is usually developed for classes of grammar rather than tailored towards individual grammar. There are several important properties that a parsing algorithm should have if it is to be practically useful. It should be sound with respect to a given grammar and lexicon; that is, it should not assign to an input sentence analyses which cannot arise from the grammar. For this study the algorithm we used is based on modified chart parser that operates on CFG using key, chart, edge and agenda. Key is holds the current constituent we are attempting to match. Chart is a form of well-formed substring table that plays the role of the memo-table and keeps track of partial derivations so nothing has to be rederived. Edge stores information about a particular step in the parsing process. Agenda is set of edges waiting to be added to the chart. It is used to prioritize constituents to be processed and organizes the order in which tasks are executed using philosophy of queue data structure. The input is processed left to right, one word at time. The algorithm works as follows;

Parsing input $X = X_1 \dots X_n$

1. Begin
2. $i=0$
3. If Agenda is empty and $i < n$ then set $i=i+1$, find all POS of X_i and add them as constituent $C(p_i, p_{i+1})$ to the Agenda, where n is number of word in string.
4. Pick a Key constituent $C(p_i, p_{i+1})$ from the Agenda
5. For each rule in the grammar of form $X \rightarrow CX_1 \dots X_n$, add an edge of form $X \rightarrow C \cdot X_1 \dots X_n$ from p_1 to p_2 .
6. Use Key to extend all relevant edge
7. Insert the Key in to the Chart
8. If Key is $S(1,n)$ then Accept the input Else go to (2)
9. End

4.5. Grammar Rule

A grammar is a set of rules for putting strings together and so corresponds to a language. There are hundreds of grammar rules but the basic ones refer to sentence structure and parts of speech,

which are noun, pronoun, verb, adjective, adverb, preposition and conjunction. Let's look at the way sentences are put together and the words that form them.

4.5.1. Sentence Structure Rule

The sample rules used to parse the sentence into noun phrase, verb phrase and to noun phrase and verb phrase are as the following;

Rule 1: $S \longrightarrow [<VP>]$

The Sentence may contains only verb phrase. But Noun phrase is hidden or not there. Example:

- I. Deemte.
- II. Mana barumsa deeme

In first sentence the subject 'Isheen'[She] is hidden and in a second sentence subject 'Inni'[He] is hidden. Both sentences have hidden pronoun. But the phrases may contain other type of phrases. In second sentence 'mana barumsa'[school] is noun phrase. That means verb phrase is contains noun phrase and verb.

Rule 2: $S \longrightarrow [<NP> + <VP>]$

The sentence is categorized into two main phrases. Noun phrase and Verb phrase. Those phrases may be contains other type of phrases. In the following sentences there are some different in a details. Example:

- I. Ibsaan ife
- II. Keennataan mana guddaa ijaare.
- III. Barsiisan gara mana barumsaa deeme

In above sentences the subject 'Ibsaan', 'Keennataan' and 'Barsiisan' are noun phrases and the predicate 'ife'[shine], 'mana guddaa ijaare'[build big house] and 'gara mana barumsaa deeme' are verb phrases respectively. The difference between three sentences is number of words and type of phrases each verb phrase of the sentence contain.

Rule 3: $S \longrightarrow [<VP> + [S [<N> + <VP>]]]$

Such like structure is when the sentence has a dependent and one or more independent clause. It is a complex sentence structure rule Example:

I. Yoo dhufuu baattellee, xalayaa naaf barreessi.

The first part before comma is dependent clause and the part after comma is independent clause that is also taken as simple sentence.

4.5.2. Noun Phrase Structure Rule

Due to natural language is very board and it is full of disambiguate, we cannot define all rules that explain noun phrase construction in Afaan Oromo. Now we see sample rule used to create noun phrase in language based on our training set.

Rule 1: NP= {<N>}

In Afaan Oromo, noun phrase can be constructed from single noun or pronoun. For example; in sentence ‘Harbuu jigse’, the word ‘Harbuu’[fig tree] is noun phrase and ‘jigse’[cut down] is verb phrase.

Rule 2: NP= {<N> + <N>}

Noun phrase can be constructed from two different nouns with special order to be meaningful. For example, ‘Sa’a aananii’ this is noun phrase constructed from two word ‘sa’a [cow] and ‘aananii’ [milk]. The word ‘aananii’ [milk] modify the use of cow. When we interchange two words we get ‘aanan sa’aa’ with correct grammar. But the meaning is different from the first one. Second phrase identify type of milk.

Rule 3: NP= {<N> + <Adj>}

We can construct noun phrase from noun and adjectives with respected order. Example, ‘mana guddaa’[big house] is noun phrase constructed from the noun ‘mana’ [house] and adjective ‘guddaa’ [big]. Order of word is very important to give meaning full phrase because in example ‘mana guddaa’ when inter change place ‘guddaa mana’ is not meaning full in Afaan Oromo phrase.

Rule 4: NP= {<N> + <PrN>}

As we define in rule 1 and rule 2 it can be constructed from noun or pronoun. Additionally, combination of noun and pronoun also create noun phrases. Example, ‘Abbaan koo qotee bulaa cimaadha.’ [My father is a good farmer.]. In this sentence the underline part ‘Abbaan koo’ [My father] is noun phrase. A noun phrase ‘Abbaan koo’ [My father] is constructed from noun ‘Abbaa’ [father] and pronoun ‘koo’ [my].

Rule 5: NP= {<N> + <Num>}

In this noun phrase can constructed from noun and number. Example, ‘Qeerransi re’oota shan nyaate.’ [a Tiger ate five goats]. The underline part ‘re’oota shan’ [Five goats] is noun phrase in sentence. It constructed from noun ‘re’oota’ [goats] a plural form of ‘re’ee’ [goat] and number ‘shan’ [five].

Rule 6: NP= {<N> + <Det>}

For example, ‘Intalli kun barsiistuu Afaan Oromooti.’ [This girl is Afaan Oromo teacher]. ‘Intalli kun’ [This girl] is noun phrase constructed from noun ‘Intalli’ [girl] and determiner ‘kun’ [this].

Rule 7: NP= {<NP> + <Det>}

Also noun phrase can be constructed from noun phrase and determiner. Example, [[Mannen guguddaa]kanneen]. This noun phrase constructed from noun phrase ‘Mannen guguddaa’ and determiner ‘kanneen’. As we define in rule 4, noun phrase ‘Mannen guguddaa’ is constructed from noun ‘Mannen’ plural form of ‘mana’ and adjective ‘guguddaa’.

Rule 8: NP= {<N> + <NP>}

Example, [dhagaa [konkolataa lama]]

This noun phrases is constructed from one noun and one noun phrase. Noun ‘dhagaa’ [stone] is head word to the phrase and noun phrase ‘konkolataa lama’ [two car] is modify noun structured with.

Rule 9: NP= {<NP> + <NP>}

Example, [[biiraa Beddellee][sanduuqaa shan]]

As in example noun phrase can be constructed from other combination two or more noun phrases. The ‘biiraa Beddellee sanduqaa shan’ [Five box of Bedele beer] is constructed from noun phrases ‘biiraa beddellee’ and ‘sanduqaa shan’.

Rule 10: NP= {<NP> + <Det>}

Noun phrase can be constructed from noun phrase and determiner. For example, the phrase ‘intaala diimtuu kana’ is constructed from noun phrase ‘intala diimtuu’ and determiner ‘kana’.

Rule 11: NP= {<NP> + <NP> + <AdjP>}

In other case noun phrase can be constructed from two or more phrases that are the same or different. Example, The phrase ‘mana barumsaa Gaara mul’ataa guddaa sun’ is from two noun phrase ‘mana barumsaa’[school] and ‘Gaara mul’ataa’[mulata mountain] and adjective phrase ‘guddaa sun’[this big]. This phrase contains two types of phrases.

4.5.3. Verb Phrase Structure Rule

To construct verb phrases, we take some sample rules that are analyzed from training set. Due to natural language is flexible we cannot include all rules of verb phrases. Now we see sample rule used to create verb phrases in language as the following.

Rule 1: VP= {<AV> + <V>}

We can construct verb phrases from adverb and verb. For example, In sentence ‘Margaan dafee dhufe.’ [Merga come quickly.]. The underline part ‘dafee dhufe’[came quickly] is verb phrase created from adverb ‘dafee’[quicky] and verb ‘dhufe’[came].

Rule 2: VP= {<V> + <V> *}

Verb phrase can be created from combination of two or more different phrases. In sentence ‘Boonsaan Jimmaa bulee dhufe’ [Bonsa spent night at Jimma and came], Verb phrase is ‘bulee dhufe’ which contains two verbs ‘bulee’ [spent night] and ‘dhufe’ [came].

Rule 3: VP= {<N> + <VP>}

We can construct verb phrase from noun and other verb phrase. For example, ‘Boonsaan Jimmaa bulee dhufe’. ‘Jimmaa bulee dhufe’ is verb phrase constructed from noun ‘Jimmaa’ and verb

phrase 'bulee dhufe'. Both word 'bulee' and 'dhufe' are verbs. In rule 2 we say verb phrase can be constructed from two different verbs with correct Afaan Oromo verb order in phrase.

Rule 4: VP = {<AVP> + <V>}

Combination of verb and verb phrase also constructs other phrases. For example, in 'Caaltuun sa'a daddaafte elmite'. The phrase 'sa'a daddaafte elmite' is a verb phrase built from noun 'sa'a' [cow] and verb phrase 'daddaafte elmite'. For verb phrase 'daddaafte elmite', the word 'daddaafte' [quickly] is an adverb and 'elmite' is a verb. Combination of adverb and verb is built verb phrase is already defined in rule 1.

4.5.4. Adjective Phrase Structure Rule

Similar to other phrases, we cannot define all rules that explain adjective phrase construction in Afaan Oromo. Now we see sample rules used to create adjective phrases in language based on our training set.

Rule 1: AdjP = {<Adj> + <Adj>}

Adjective phrases can be constructed from combinations of two or more phrases. Example, in sentence 'Caalaan sanga diimaa guddaa bite' [Chala bought red big ox]. The underline is an adjective phrase (JJP). The phrase 'diimaa guddaa' [red big] is an adjective phrase constructed from adjective 'diimaa' [red] and 'guddaa' [big].

Rule 2: AdjP = {<Adj> + <Det>}

Adjective phrases can be constructed from determiners and adjectives. Example, 'Baay'ee dheeraa' [very long] is constructed from determiner 'baay'ee' [very] and adjective 'dheeraa' [long].

Rule 3: AdjP = {<PP> + <Adj>}

Adjective phrases can be constructed from pre- and post-positional phrases and adjectives. In sentence 'Bishaan fayyaaf gaariidha' [water is good for health], the phrase 'fayyaaf gaarii' [good for health] is an adjective phrase constructed from postpositional phrase 'fayyaaf' [for health] and adjective 'gaarii' [good].

4.5.5. Adverb Phrase structure Rule

Rule 1: AVP= {<AV> *}

Adverb phrases can be constructed from one and more than one adverbs. Example, in sentence ‘Guutaan har’a dhufe.’ [Guta is came today.], ‘har’a’ [today] is adverb phrase and also adverb. This is based on what Afaan Oromo linguistic experts is define [15][18] and the same to others example what we analysis from our training sets.

4.5.6. Pre- and Post- Positional Phrase Structure Rule

As we say in other phrases, we cannot define all rules that explain Pre- and Post- Positional phrases construction in Afaan Oromo. Now we see sample rules used to create Pre- and Post- Positional phrases in language from our training set.

Rule 1: PP= {<N> + <P>}

Pre- and Post- positional phrases can be constructed from noun and post-position. Example, in sentence ‘Konkolaataan mana irraa fagoo dhaabata’, the underline phrase ‘mana irraa’ [from house] is post- positional phrase. The phrase is constructed from noun ‘mana’ [house] and ‘irraa’ [from].

Rule 2: PP= {<NP> + <P>}

We can construct Pre- and Post- positional phrases from noun phrases and Pre- or Post- positional. For example, in a phrase ‘mana dhugaatiitti’[at bar house], the head word is post position ‘-tti’ and noun phrase is ‘mana dhugaatii’[bar house].

Rule 3: PP= {<P> + <PP>}

Also, we can construct Pre- and Post- positional phrases from prepositional and Pre- and Post- positional phrase. Example, the phrase ‘gara barnootaatti’[towards to education] is constructed from preposition ‘gara’ and post position phrase ‘barnootaatti’.

4.6. Parse Tree

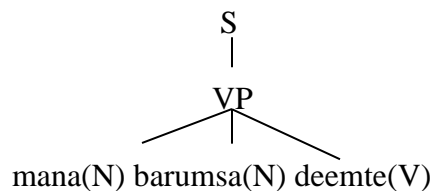
A parse tree is an ordered, rooted tree that represents the syntactic structure of a sentence according to some context free grammar. The expected output of syntactic parsing is parse tree of a sentence constructed from grammar rules. This tree can be described with the same analogy of real world tree but inverted structure; the root of parse tree is always the sentence(S symbol),

the leaves of the trees are terminal symbols (actual words), and the branches of the tree are phrasal, and lexical constituents (POS) tags. The branches of the tree are actually the nodes in between the root and the leaves representing different grammar rules which are applied to obtain that particular tree.

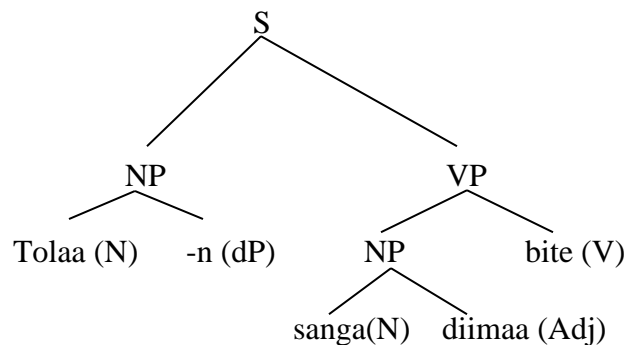
A parse tree is based on top down parser that searches for a parse tree by trying to build from the root node *S* down to the leaves. It is rooted with *S*, denoting Sentence; the sentence is composed of a noun phrase (NP) followed by a verb phrase (VP) and period. The leaves of the tree are the words in the sentence, and the pre-terminals are part of speech tags.

Tree encodes these relations by the very nature of a tree as a directed acyclic graph. It is a set of connected nodes, each of which is labeled with a category. It common to use a family metaphor to talk about the relationships of nodes in a tree; for example, *S* is the parent of *VP*; conversely *VP* is a daughter (or child) of *S*. Also, since *NP* and *VP* are both daughters of *S*. We construct parse tree based on the sentence structural rule. Here is an example of a tree:

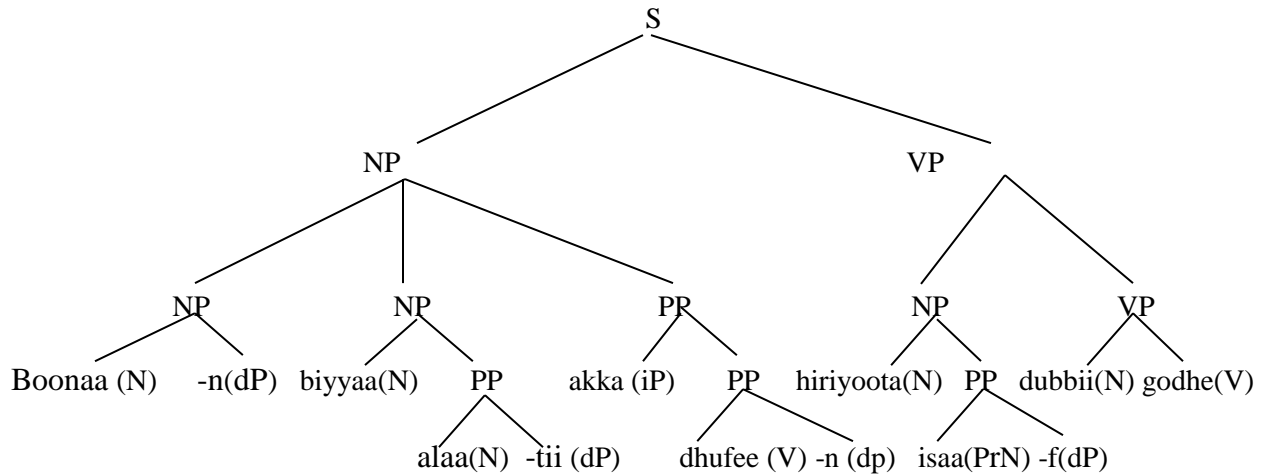
When the sentence contains only verb phrase (VP), that means noun phrase is hidden or not there.



When the sentence contain noun phrase (NP) and verb phrase (VP); Example for sentence ‘Tolaan sanga diimaa bite’ [Tola was bought red bull].



Example for complex sentence means when the sentence contains independent clause as phrase;
 Boonaa biyyaa alaatii akka dhufeen, hiriyoota isaaf dubbii godhe



Here, the node value is a constituent type (NP or VP), and the children encode the hierarchical contents of the tree. Although it is helpful to represent trees in a graphical format, for computational purposes we usually need a more text oriented representation. We will use the same format as the Penn Treebank, a combination of brackets and labels.

When the sentence contain only verb phrase (VP), that means noun phrase is hidden or not there for first parse tree.

```
(S
  (VP (N Mana) (N barumsa) (V deemte))
)
```

When the sentence contain noun phrase (NP) and verb phrase (VP);

```
(S
  (NP (N Tolaa ) (P n))
  (VP (NP (N sanga) (Adj diimaa)) (V qale))
)
```

If we apply complex structures in syntactic parsing, this model will expectedly contains more items (rules) than a context free which grammar produced with the same corpus. The tree pattern description compensates this growing with containing similar structures in one pattern. The flexibility of description allows building in elements of other formalisms. So, various phenomenon's of natural languages can be handled. We apply a modified version of chart parser to syntactic parsing of text with tree patterns.

4.7. Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm, which has achieved state of the art performance on many learning tasks. In particular, SVM is a popular learning algorithm for natural language processing.

SVM is an optimal classifier in the sense that, given training data, it learns a classification hyper plane in the feature space which has the maximal distance to all the training examples. Consequently, on classification tasks the SVM tends to have better generalization capability on unseen data than other distance or similarity based learning algorithms such as k-nearest neighbor (KNN) or decision tree. Another feature of the SVM is that, by using different types of kernel function, the SVM can explore different kinds of combinations of the given features without increasing computational complexity. In contrast, it would be difficult for many other learning algorithms to deal with a huge number of feature combinations efficiently.

On the other hand, NLP tasks typically represent instances by very high dimensional but very sparse feature vectors, which lead to positive and negative examples being distributed into two distinctly different areas of the feature space. This is particularly helpful for the SVM to search a classification hyper plane in feature space and for the generalization capability of the classifier as well. That is a main reason why the SVM can achieve very good results in a variety of NLP tasks. Such very high dimensional representation is achieved by forming the feature vector explicitly from text using a huge amount of linguistic features and in many cases by exploring the so called kernel function to map the feature vector into higher dimensional space.

Furthermore, as the SVM is optimal margin classifier, the distance of an example to the SVM classification hyper plane indicates how important of the example to the SVM learning. The examples being close to the SVM hyper plane are crucial for the learning. The SVM active

learning is based on the distance of unlabeled example to the SVM hyper plane. In the applications, the SVM can select the useful features effectively from a large number of features for a particular classification problem. This is because the SVM learns a classifier by combining all the features (or feature combinations) with different weights. If a feature occurs almost equally in both positive and negative training examples and hence is irrelevant to the classification, its learned weight would have little contribution to the classifier.

In contrast, many other algorithms require careful manual feature selection. This is advantageous when applying the SVM to NLP problem. As there exists many types of NLP features from morphology, syntax, semantics as well as from different knowledge sources like thesaurus and gazetteers, in the application of SVM to NLP, those different kinds of features are just put together to form one feature vector for one example as input to SVM and the learning would automatically determine which features and/or combinations of features are useful for the task. When compared to other classification problems, NLP classification tasks have several unique characteristics, which were rarely considered in applications. Perhaps the most important one is that NLP tasks tend to have imbalanced training data, in which positive examples are vastly outnumbered by negative ones. This is particularly true for smaller data sets where often there are thousands of negative training examples and only few positive ones. Another unique characteristic is that annotating text for training the algorithm is a time consuming process, while at the same time unlabeled data is abundant.

Moreover, since the SVM is usually used as binary classifier for solving an NLP problem and an NLP problem in most case is equivalent to a multiclass classification problem, we need to transform the multi class problem into binary classification problems. This is true in syntax parsing (rule based parsing).

The SVM model needs to be solved using an optimization procedure. There are specialized optimization procedures that reformulate the optimization problem to be a quadratic programming problem. The most popular method for fitting SVM is the Sequential Minimal Optimization (SMO) method that is very efficient. It breaks the problem down into sub problems that can be solved analytically (by calculating) rather than numerically (by searching or optimizing). Support Vector Machine is binary classifiers in their basic form. Their theoretical advantages and their practical success motivated researchers to investigate extensions to

multiclass problems. It should be noted here that with the term multiclass we refer to problems in which any instance is assigned exactly one class label. To deal with multiclass problems, it consider the whole training set with all the classes at once and solve the multiclass problem directly and those that decompose the problem into constructing several binary classifiers and combining their output.

SVM has a technique called the kernel trick. This is function which take low dimensional input space and transform it to a higher dimensional space i.e. it converts not separable problem to separable problem, these functions are called kernels. The linear kernel function model from experiment is: Linear kernel $k(x,y) = \langle x,y \rangle$, where x and y are classes.

- **Problem of machine learning**

Challenges of using pure machine learning are:

- It needs huge tagged dataset or Treebank

4.8. Hybrid Approach

As pure rule based approach has the above two challenging problem and pure machine learning need Treebank to train. So, we used hybrid approach. Hybrid approach is use rule based approach + machine learning.

The learning process in this study is based on template rules. The first step is derivation of rules, second is scoring of rules, and third is selection of one rule with maximal positive effect. The process is iterative. This technique achieved precision and recall of 82% for complex sentence parser and 89.5% for simple sentence.

4.9. Training Set

For the purpose of this study, the parser is trained using the sentences of different patterns collected from various Afaan Oromo grammar books. In a typical supervised learning scenario, a training set is given and the goal is to form a description that can be used to predict previously unseen patterns. The parser model is created with the data set containing simple sentences and some complex sentences. Types of sentences are taken from the Afaan Oromo grammar books such as Semmoo: Basic language education and poets of Oromo by Barkessa Adunga [22], Afaan Oromo grammar by Aberra Nafa [18] and Oromo grammar by Getachew Rabirra [15]. Since the training sentences for the parser are taken from the Afaan Oromo grammar books, the

sentences covers different patterns for the sentence structure, the training data covers almost all the patterns available for simple sentences. These books not only cover the simple sentence and complex sentences patterns, but they also cover the sentence patterns for all sentence type. Also, some of the corpus has been collected from the Afaan Oromo newspapers. Some of the sentences are included from the essays and short stories from magazines also. With this corpus, around 250 sentences were collected and given to the machine for training. Training set is used for machine learning algorithms to build models, which will be used to identify boundaries of components in the parsing stage.

Due to lack of annotated text for the purpose of grammar induction and training, it was really much time that was spent during manual morphological analysis of each word, hand tagging and parsing each sentence in the sample corpus. The corpus is enough for identifying the rules. Although as corpus is collected from different source as raw text then corpus preprocess techniques is important.

4.9.1. Corpus Preprocesses Module

The system includes preprocessing segments that can be used before the automatically parsing corpus. As the corpus collected from different materials, corpus preprocesses is very important. There are some unnecessary punctuation marks, normalize each contiguous occurrence of white space between words, and convert all case into sentence case and removing numbers. It is the step to remove non informative words and characters from the text.

Preprocess components are language dependent. The utilization of these modules depends on the nature of the input. Actually these modules are used in the case when the input is raw text. However, these modules will not be used when the input is an annotated corpus. It is used to make the entire sentence in training set to be valid simple sentences. These modules are done by normalization, tokenize and stop word removal.

A. Normalization

In this study text normalization is process that transform raw text (corpus) into single recognized form. Normalizing corpus before processing it allows for separation of concerns, since input is guaranteed to be consistent before operations are performed on it. Normalization is a preliminary

step to Afaan Oromo tokenization to ensure that the text is steady and predictable. It is frequently used until numbers, dates, acronyms and abbreviation is existed in corpus. Also, changing sentence case means from capitalization into sentence case or from lower case to sentence case and removing all Afaan Oromo delimiters are included under normalization module. It is a basic task that researchers in Afaan Oromo NLP always apply with a common goal in mind: reducing noise and sparsely in the data.

B. Tokenization

Tokenization is the process of breaking up the sequence of characters in a text by locating the word boundaries, the points where one word ends and another begins. This was not seen to be a serious problem for researchers working on English and similar languages, where word boundaries are generally coincident with space characters. It is similar in Afaan Oromo with English since Afaan Oromo letter is Latin letters.

Afaan Oromo is a rich language with reference to inflection and derivation. The lexical items are created by attaching affixes to roots which require a very large corpus for good coverage of Afaan Oromo. Tokenization is a task of splitting a text into pieces called tokens, when are disjoint and meaning full texts. It manipulates the text on the level of individual words. Correspondingly, tokenization is often considered to be part of the text preprocessing, which includes removal of markup tags and excessive whitespace characters. A token is the minimal syntactic unit. The tokenization module is responsible for identifying a word, a part of a word, a multiword expression, or a punctuation mark (including all delimiters). Actually an Afaan Oromo token may consist of several lexical items which have their own meaning and part of speech (POS).

C. Elimination of Stop words

In collected raw text there are several types of stop words lists. The stop word list has conjunction, articles and particles. Those words are collected from data sets and eliminated based on grammatical information of the language described in [15], [16], [18] grammar books.

4.10. Learn patterns encoding syntactic structure

4.10.1. Part-Of-Speech (POS) Tagger Pattern

Part of speech tagger (POS tagger)[21] is the act of assigning each word in sentences tag that describes how that word is used in the sentences. POS is process of labeling automatic annotation of syntactic categories for each word in a corpus. It is similar to the process of tokenization for computer languages. It assigns a part of speech like noun, verb, pronoun, preposition, adverb, and adjective or other lexical class marker to each word in a sentence.

In our work the Part of speech (POS) pattern is determine how POS is interrelated in sentence. POS pattern is important in rule based parser for understanding interrelation in grammar rule of the language. Let to explain POS pattern in all phrase type with sample example.

Noun phrase is one type of the phrase we define in chapter 3 of this work. In noun phrase always adjective comes after noun. For example; in noun phrase (NP) ‘sanga diimaa guddaa’[red big ox], both ‘diimaa’[red] and ‘guddaa’[big] are adjective those come after noun ‘sanga’[ox]. This is true when noun phrase constructed from [NP + Adj] or [N + Adj] in Afaan Oromo. When noun phrase constructed from combination of adverb with noun or with other noun phrase, adverb is come after noun or noun phrase. The noun phrase ‘mana guddaa lama’ [two big house] or a phrase ‘mana lama’ [two house] is valid phrase in Afaan Oromo. An adverb ‘lama’ [two] is come after noun ‘mana’ [house] in noun phrase ‘mana lama’ [two house] or in a phrase ‘mana guddaa lama’ [two big house] it come after noun phrase ‘mana guddaa’ [big house]. This always true when noun phrase constructed from combination of noun or noun phrase with adverb in Afaan Oromo phrases.

Adverb phrase is similarly type of phrase we define in chapter 3 of our work. Adverbs always come before the modified verb but it should be noted that any words come before verbs cannot be always considered as an adverb. Adverb phrase is never come with other words means it contain only adverb.

Among all type of Afaan Oromo phrases noun phrase and verb phrase is very important part. That means, we cannot say other phrase is not important. At least in Afaan Oromo sentences, sentence is categorized into noun phrase and verb phrase or one of them. Even if we focus on our scope, the statement is true in all type of sentences. Except there is some postpositional those terminate sentence described as special key under this chapter 4, verbs are at end in verb phrase

or in sentence (especially in simple sentence), as we explain in chapter 3. This is common in Afaan Oromo. Example; Let we see the following sentences:

1. Laliseen qabxii gaarii **argatte**.
2. Daraartuun kaleessa kitaaba ishee **gatte**.
3. Finfinneen handhuura oromiyaatti **argamti**.
4. Roobni cimaan kallessa **roobe**.
5. Obbo Hordofaan bulchaa aanaa **ta'ani**.

In above all sentences words at the end of sentences ('argatte', 'gatte', 'argamti', 'roobe', 'ta'ani') in respectively are verbs. But word come before verb cannot common may be adjective, adverb, noun, pronoun or other verb. In first sentence 'gaarii'[good] is adjective , in second sentence 'ishee'[she] is pronoun, in third sentence 'oromiyaatti'[of oromia] is postpositional phrase, in a fourth 'kallessa'[yesterday] is adverb and in fifth sentence 'aanaa' [sub city] is noun.

POS tagging is a well understood problem in NLP, to which machine learning approaches are applied. The interest in annotated corpora is spreading, as there is increasing concern with using existing machine learning approaches for corpus processing. The input to a tagging algorithm is a string of words of a natural language sentence and a specified tag set (a finite list of Part-of-speech tags). The output is a single best POS tag for each word. The importance of the problem focuses from the fact that the POS is one of the first stages in the process performed by various natural language related process.

Moreover, since the selected corpus is small in number, the researcher assumed to use six tag sets, listed in Table 4.1. These tag sets are used in constructing lexical rules to hold lexicons of each sentence as in figure 4.1.

NO	Tag-Sets	Assigned to
1	N	Noun
2	V	Verb
3	P	Preposition
4	AV	Adverb
5	Adj	Adjective
6	CD	Cardinal

Table 4. 1. Tag sets used

4.10.2. Morphological Information

In simple examples and small systems one can list all the words allowed by the system, representing the lexicon for sentence parsers that entertain large vocabulary would face a serious problem. Not only there are a large number of words, but also each word may combine with affixes to produce additional related words. One way to address this problem is to preprocess the input sentence into a sequence of morphemes.

To break a word into morphemes, try starting at the beginning of the word and seeing how far into the word we need to go to find a sub part of the word that has some meaning. But in this study we start from phrase, because sentence is break into phrases and again phrases is break into sub phrase or into part of speech. For morphological analysis of words in sentences, there is some developed system like Gasser's HornMorpho. According to [23], HornMorpho performs morphological analysis of the word. For ambiguous words returns the first nbest analyses. The interface of Gasser's HornMorpho morphological analysis for sentence 'Tolaan mana guddaa lama bite' is:

['Tolaan', 'mana', 'guddaa', 'lama', 'bite']

word: tolaan

POS: verb, root: <tol>, citation: tol

participle, 1s subj

POS: verb, root: <tol>, citation: toluu

subject: 1, sing

TAM: present, 3pers/instrumental

POS: deverbial noun, root: <tol>, citation: tol
agent, case: sb

POS: deverbial noun, root: <tol>, citation: tol
agent, case: ins

POS: deverbial noun, root: <tol>, citation: tol
agent, case: gen, 1s subj

POS: deverbial noun, root: <tol>, citation: tol
agent, case: bs, 1s subj

POS: verb, root: <tol>, citation: toluu
subject: 3, sing, masc

TAM: present, 3pers/instrumental

POS: verb, root: <tol>, citation: toluu
subject: 2, plur

TAM: imperative, 3pers/instrumental

POS: noun, stem: tolaa, citation: tolaa
case: ins

masculine, singular

POS: noun, stem: tolaa, citation: tolaa
case: sb

masculine, singular

POS: noun, stem: tolaa, citation: tolaa
case: dat

masculine, singular, 1s subj

POS: noun, stem: tola, citation: tola
case: dat

singular, 1s subj

POS: noun, stem: tola, citation: tola
case: ins

singular

POS: noun, stem: tola, citation: tola

case: bs+gen

singular, 1s subj

POS: noun, stem: tolaa, citation: tolaa

case: bs+gen

masculine, singular, 1s subj

POS: noun, stem: tolaa, citation: tolaa

case: bs

masculine, singular, 1s subj

POS: noun, stem: tola, citation: tola

case: abl

singular, 1s subj

POS: noun, stem: tolaa, citation: tolaa

case: abl

masculine, singular, 1s subj

Tolaan

word: mana

POS: noun, stem: mana, citation: mana

case: bs

singular

mana

word: guddaa

POS: noun, stem: guddaa, citation: guddaa

case: abl

singular

POS: noun, stem: guddaa, citation: guddaa

case: bs+gen

singular

POS: noun, stem: guddaa, citation: guddaa

case: dat

singular

POS: noun, stem: guddaa, citation: guddaa

case: bs

singular

guddaa

?word: lama

lama

word: bite

POS: verb, root: <bit>, citation: bituu

subject: 3, sing, masc

TAM: past

POS: verb, root: <bit>, citation: bituu

subject: 1, sing

TAM: past

bite

In Afaan Oromo, some words in sentences are counted as phrase. This is especially true in noun phrases. For example, in sentence ‘Daraartuun kaleessa kitaaba ishee gatte’ [Derartu’s lost her book yesterday] the word ‘Daraartuun’ is noun phrase that constructed from a noun ‘Daraartuu’ [Derartu] and postposition ‘-n’. This means that ‘-n’ is a morpheme. Again, Daraartuu- is independently meaningful. So Daraartuun has two morphemes: Daraartuu-n. Some words just have one morpheme; of course we can’t break down the word ‘Bishaan’ [water] into any meaningful sub parts, for example, ‘Bishaan bollaa bishaan qulqulluudha.’ [Ground water is pure water], in this sentence ‘Bishaan bollaa’ [Ground water] is noun phrase that parsed into two noun ‘Bishaan’ [water] and ‘bollaa’ [ground]. But verb phrase ‘bishaan qulqulluudha’ [is pure water] is parsed into prepositional phrase ‘qulqulluudha’ [is pure] and noun ‘bishaan’ [water]. For parsing post positional phrase ‘qulqulluudha’ [is pure] using morphological information, it parsed into adjective ‘qulqulluu’ [pure] and post position ‘dha’ that taken as [is] when translated.

In Afaan Oromo parser, some phrases are constructed from root word and pre- or post- position. So, to classify sentence into subject and complement morphological analysis must be important. Because when words in sentence is morphological analyzed, it analysis words is it subject or complement like verbs.

Chapter Five

5. Experimentation, Result and Evaluation

5.1. Introduction

This chapter is we present result from learning training set described in Chapter 4. All the training data was used as seed data in order to initiate the learning experiments. We use two data; training and test data. Training set is used to train the classifier, unless otherwise stated. The test set used to evaluate the performance of the classifiers is the same across all the experiments for each of the tasks considered. Both data's are taken from corpus prepared by researchers. The SVM algorithm used for machine learning is implemented using sequential minimal optimization function.

5.2. Support vector machine feature selection

Feature selection is one of the most important tasks in machine learning. It is especially useful when you are dealing with high dimensional data or when your dataset contains a large number of features and a limited number of observations. Reducing features also saves storage and computation time and makes your results easier to understand.

Instrumental to our system's performance is the choice of a set of salient characteristics (features) to be used as input to the SVM algorithm for training and classification. Once the features are determined, classification instances can be formally represented as a vector of values.

In the parlance of SVM literature, a predictor variable is called an attribute, and a transformed attribute that is used to define the hyper plane is called a feature. The task of choosing the most suitable representation is known as feature selection. A set of features that describes one case is called a vector. So the goal of SVM modeling is to find the optimal hyper plane that separates clusters of vector in such a way that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other size of the plane. The vectors near the hyper plane are the support vectors. Three different kinds of feature types can be collected are:

1. Word Features

Word form n -grams we used are unigrams. We are not using bigrams and trigrams suffice. Also, the sentence last word, which corresponds to a punctuation mark, is important. In our study sentences are break into unigram means it is break into individual words in sentences. Because it is used to assign words into it is part of speech. Part of speech is important feature in parsing. Words are the simplest, most intuitive textual features. Although not a grammatical feature such as rewrite rules or function words, we include them because of their prevalence and accuracy. Besides, we expect words will be a very useful feature to combine with more advanced methods. Stemming and stop word is optimizations often applied to word features. Both these methods greatly reduce the feature space, omitting irrelevant words (stop word removal) and grouping words with the same meaning together in corpus.

2. Part of Speech (POS) Features

Part of speech tags are a common grammatical feature. Their small, finite number lends them to be simple features for a classifier. When expanding to n -grams of part of speech tags, their small number also ensures that there are still a relatively low number of features generated (opposed, mainly, to words). POS tags are perhaps the syntactic analog of basic words, in that they are simple and robust. They capture grammar usage at its most basic level.

Annotated parts of speech and ambiguity classes n -grams, and may be plural form like “-oota”, ”-wwaan”, ”-een” and also when prefix letter like “-n” is affixed with word in sentences, it is counted as phrase in some sentences but in other it counted as word. As for words, considering unigrams is enough. The ambiguity class for a certain word determines which POS are possible. May be states, for a certain word, that certain POS may be possible, i.e. it belongs to the word ambiguity class.

3. Lexicalized Features

Including prefixes and suffixes, capitalization, hyphenation, and similar information related to a word form. In our study, for Afaan Oromo all the other features are accepted except capitalization will not use.

5.3. Experimentation using SMO Function

Different experiments have been conducted on the SVM classifier. The developed model showed better accuracy for the sentences of smaller length, especially for simple sentences. This doesn't mean that we are not getting the correct parsed data for the complex sentences. The parser output for the complex sentences is comparatively less accurate. We conduct three different experimentations for SVM algorithm implementation. The SVM algorithm is implemented using sequential minimal optimizing (SMO) function. The first one is using simple, this experiment has 200 simple sentences. Accuracy of an experiment is 89.5%. The second one is using complex sentences. The experiment has 50 complex sentences. Accuracy of an experiment is 82%. The last experiment we done is using combination of simple and complex sentences. This contains mixture of two types of sentences. The experiment has 250 sentences. Accuracy of an experiment is 84.5%.

Sample output for training set is:

Test mode: evaluate on training data
=== Classifier model (full training set) ===

SMO

Kernel used:

Linear Kernel: $K(x,y) = \langle x,y \rangle$

Classifier for classes: NP, VP

BinarySMO

Machine linear: showing attribute weights, not support vectors.

-0.646 * (normalized) W1=PrN
+ -0.6451 * (normalized) W1=NP
+ -0.6454 * (normalized) W1=N
+ -0.6456 * (normalized) W1=iP
+ 1.3543 * (normalized) W1=V
+ -0.6449 * (normalized) W1=Det
+ 1 * (normalized) W1=AV
+ 0.8727 * (normalized) W1=PP
+ -0.5561 * (normalized) W2=iP
+ 1.4443 * (normalized) W2=VP
+ -0.5556 * (normalized) W2=PrN
+ 1.4441 * (normalized) W2=V
+ -0.5558 * (normalized) W2=0
+ -0.4424 * (normalized) W2=Adj
+ -0.5557 * (normalized) W2=dP
+ -0.555 * (normalized) W2=CD
+ 0 * (normalized) W2=PP
+ -0.5555 * (normalized) W2=N

+ 1.4444 * (normalized) W2=Aux
 + -0.5567 * (normalized) W2=NP
 + -1 * (normalized) W3=P
 + -0.6077 * (normalized) W3=0
 + 1.3919 * (normalized) W3=V
 + -0.6058 * (normalized) W3=VP
 + -0.1273 * (normalized) W3=N
 + -0.6086 * (normalized) W3=PrN
 + 1.2793 * (normalized) W3=Aux
 + -1 * (normalized) W3=iP
 + 1.2783 * (normalized) W3=AdjP
 + -0.0007 * (normalized) W4=0
 + 0.0007 * (normalized) W4=V
 + 0.8097

Number of kernel evaluations: 22950 (86.831% cached)

Classifier for classes: NP, PP

BinarySMO

Machine linear: showing attribute weights, not support vectors.

0.1623 * (normalized) W1=PrN
 + -0.2703 * (normalized) W1=NP
 + -1.1892 * (normalized) W1=N
 + 0.811 * (normalized) W1=Adj
 + 0.0353 * (normalized) W1=iP
 + 0.811 * (normalized) W1=V
 + -0.1804 * (normalized) W1=Det
 + -0.1797 * (normalized) W1=AVP
 + 0.649 * (normalized) W2=iP
 + -0.5762 * (normalized) W2=VP
 + -0.5748 * (normalized) W2=V
 + -0.7031 * (normalized) W2=0
 + 0.6492 * (normalized) W2=dP
 + -0.2703 * (normalized) W2=AdjP
 + 0.1863 * (normalized) W2=N
 + 1 * (normalized) W2=Aux
 + -0.3601 * (normalized) W2=NP
 + 0.9483 * (normalized) W3=0
 + -1 * (normalized) W3=V
 + 0 * (normalized) W3=PP
 + -0.1346 * (normalized) W3=N
 + 0.1863 * (normalized) W3=PrN
 + 0.1346 * (normalized) W4=0
 + -0.1346 * (normalized) W4=V
 - 1.5426

Number of kernel evaluations: 4710 (70.843% cached)

Classifier for classes: NP, AdjP

BinarySMO

Machine linear: showing attribute weights, not support vectors.

```
-0.1878 * (normalized) W1=PrN
+ -0.1902 * (normalized) W1=NP
+ -0.5857 * (normalized) W1=N
+ 1.4153 * (normalized) W1=Adj
+ -0.191 * (normalized) W1=iP
+ -0.0546 * (normalized) W1=V
+ -0.1027 * (normalized) W1=Det
+ -0.1033 * (normalized) W1=AVP
+ -0.1211 * (normalized) W2=iP
+ -0.1186 * (normalized) W2=VP
+ -0.119 * (normalized) W2=V
+ -0.1214 * (normalized) W2=0
+ 0.2771 * (normalized) W2=Adj
+ 0.2762 * (normalized) W2=dP
+ -0.12 * (normalized) W2=AdjP
+ -0.0236 * (normalized) W2=N
+ 0.2764 * (normalized) W2=Aux
+ -0.206 * (normalized) W2=NP
+ 0.0783 * (normalized) W3=0
+ -0.0546 * (normalized) W3=V
+ -0.018 * (normalized) W3=PrN
+ -0.0057 * (normalized) W3=iP
+ 0.0057 * (normalized) W4=0
+ -0.0057 * (normalized) W4=V
- 0.7748
```

Number of kernel evaluations: 3241 (83.143% cached)

5.4. Test Set

The training set selected, which was discussed in chapter four, was used for experimentation. Each sentence in the training and test set had been tagged and hand parsed by the researcher, with comments and suggestions from linguist's advisor and other experts of the language at Jimma University. From sample selected text, 75 sample simple sentences and 25 complex sentences were randomly picked from the sample corpus for test set. The algorithm trained is tested using sets of test sentences.

5.4.1. Test Set Result

The test set is used for evaluating the system. For test set we use 100 sentences. 75 sentences are simple sentence and 25 are complex sentences. All test set sentences are randomly selected form corpus. Output result is in Appendix 2

All classes are bind in two classes means binary combination. Each paired classes has own individual Number of kernel evaluations and accuracy. Those paired classes using binary SMO are <NP, VP>, <NP, PP>, <NP, AdjP>, <NP, AVP>, <VP, PP>, <VP, AdjP>, <VP, AVP>, <PP, AdjP>, <PP, AVP> and <AdjP, AVP>. The following table indicates individual value of paired phrase classes.

No.	Classifier for classes	Number of kernel evaluations	Accuracy (%)
1	<NP, VP>	22950	86.83
2	<NP, PP>	4710	70.84
3	<NP, AdjP>	3441	43.14
4	<NP, AVP>	3005	68.34
5	<VP, PP>	5099	83.39
6	<VP, AdjP>	3423	74.79
7	<VP, AVP>	2308	66.64
8	<PP, AdjP>	212	89.81
9	<PP, AVP>	124	83.221
10	<AdjP, AVP>	36	76.159

Table 5. 1: Value of paired phrase classes

The classification summary of the SVM classifier is stated below along with a detailed accuracy by class and a confusion matrix shown on table 5. 1 and table 5.2 respectively.

Time taken to build model: 0.27 seconds

Detailed Accuracy by Class of Test Set

	TP Rate	FP Rate	Precision	Recall	Recall F-Measure	ROC Area	Class
	0.848	0.139	0.853	0.848	0.85	0.866	NP
	0.942	0.127	0.825	0.942	0.879	0.921	VP
	0.318	0.003	0.875	0.318	0.467	0.886	PP
	0.5	0.003	0.75	0.5	0.6	0.994	AdjP
	1	0	1	1	1	1	AVP
Weighted Avg.	0.845	0.118	0.847	0.845	0.834	0.895	

Table 5. 2: Detailed Accuracy of test set

=== Confusion Matrix of paired classes ===

	NP	VP	PP	AdjP	AVP
NP	128	22	1	0	0
VP	7	113	0	0	0
PP	13	1	7	1	0
AdjP	2	1	0	3	0
AVP	0	0	0	0	10

Table 5. 3: Confusion Matrix of test set

5.5. Evaluation of the parser

In order to evaluate SVMs learning for classification tasks, we used the task of discriminating named entities from non-named entities in text. As it was mentioned earlier, the distribution of the class labels in the dataset of the task is much skewed. The tokens that are entities are the minority class (15% of the dataset). In order to evaluate the performance of the classifier trained in each learning step we used precision and recall evaluation metrics, also F-score based both recall and precision from test set.

Precision is the fraction of the correctly predicted class units (True Positive) over the total number of class units predicted by the classifier (True Positive plus False Positive). Means in equation:

$$P = \frac{TP}{TP+FP}$$

Recall is the fraction of the correctly predicted class units (True Positive) over the total number of the class units existing in the testing set (True Positive plus False Negative).

$$R = \frac{TP}{TP+FN} = \frac{0.845}{0.845+0} * 100 \% = 87.7 \%$$

Where, TP = count of phrases correctly classified

FN = count of sensitive phrases missed

FP = count of phrases incorrectly classified

When we calculate overall precision, recall and F-score for test set, first we must calculate precision, recall and F-score for each individual phrase type we have. Then, when we calculate the precision, recall and F-score for each class from confusion matrix in table 5.2,

Class a→NP

$$P = \frac{TP}{TP + FP} = \frac{128}{128 + 22} * 100 \% = 85.33 \%$$

$$R = \frac{TP}{TP + FN} = \frac{128}{128 + 23} * 100 \% = 84.77 \%$$

Class b → VP

$$P = \frac{TP}{TP + FP} = \frac{113}{113 + 24} * 100 \% = 82.48 \%$$

$$R = \frac{TP}{TP + FN} = \frac{113}{113 + 7} * 100 \% = 94.17 \%$$

Class c → PP

$$P = \frac{TP}{TP + FP} = \frac{7}{7 + 1} * 100 \% = 87.5 \%$$

$$R = \frac{TP}{TP + FN} = \frac{7}{7 + 15} * 100 \% = 31.82 \%$$

Class d → AdjP

$$P = \frac{TP}{TP + FP} = \frac{3}{3 + 1} * 100 \% = 75 \%$$

$$R = \frac{TP}{TP + FN} = \frac{3}{3 + 3} * 100 \% = 50 \%$$

Class e → AVP

$$P = \frac{TP}{TP + FP} = \frac{10}{10 + 0} * 100 \% = 100 \%$$

$$R = \frac{TP}{TP + FN} = \frac{10}{10 + 0} * 100 \% = 100 \%$$

So, overall precision and recall of the test set are summation of precision and recall of all classes.

$$P = \frac{\sum_a^e P}{5} = \frac{85.33 \% + 82.48 \% + 87.5 \% + 75 \% + 100 \%}{5} = 86.06 \%$$

$$R = \frac{\sum_a^e R}{5} = \frac{84.77 \% + 94.17 \% + 31.82 \% + 50 \% + 100 \%}{5} = 72.15 \%$$

F-score based on both recall and precision:

$$F = \frac{2PR}{P + R} = \frac{2 * 86.06 \% * 72.15 \%}{86.06 \% + 72.15 \%} = 78.49 \%$$

The overall accuracy is $= \frac{TP+TN}{TP+TN+FP+FN}$, TP and TN here are the same = 261 because both are the sum of all true classified examples, regardless their classes false positives, which are items incorrectly labeled as belonging to the class=48 and false negatives, which are items which were not labeled as belonging to the positive class but should have been=48.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{261}{261+48+48} * 100 \% = 73.11 \%$$

5.6. Discussion

When we compare hybrid parser with previous work of Afaan Oromo parser called Afaan Oromo automatic sentence using supervised learning techniques. In [2], the study is cover only simple sentence but in our study we try to cover simple and complex using machine learning techniques. When we compare accuracy of our study with previous work, we are compare accuracy of training set on simple sentence. Because since we have three experiments on training set and previous work training set had only simple sentences, means with an experiment that was contain only simple sentences. The following table is summarizing difference accuracy between two studies.

Data Set	Hybrid Approach	Afaan Oromo automatic sentence paring
Training Set	89.5%	95%
Test Set	73.11%	88.5%

Table 5. 4: Accuracy comparison with previous work of Afaan Oromo parser

Accuracy of our study is less than the previous work in both training and test set. We calculate the accuracy from the result executed by supervised machine learning techniques using SVM algorithm implemented by SMO function. But Afaan Oromo automatic sentence paring[2] use Hidden Markov Model. This means we are not use similar algorithm. We are use 200 simple

sentences but [2] use 50 simple sentences for simple sentence evaluation. Also number of data set we used and used in [2] is different; this can be affect accuracy of parser.

Chapter Six

6. Conclusion and Recommendation

6.1. Summary

The purpose of this study was to develop parser for Afaan Oromo simple and complex sentence by exploring different experiments. Parsing of natural language sentence have been investigated using either rule based or statistical approaches. Investigating each approach is more or less dependent on the availability of language's resources. Afaan Oromo hasn't had as such well resources that could have helped for exploring statistical approaches.

Consequently, the hybrid approach was applied in this study because it is better for dealing with under-resourced languages and small dataset with better accuracy. Upon investigating context-free grammar parser on complex sentence, computational level of Afaan Oromo hopefully went one step forward. Thus, this model parses Afaan Oromo simple and complex sentence and could in turn be used for incorporating it to other NLP application.

The input sentence for parser was manually preprocessed i.e., striped punctuation marks, putting sentence per new line, and POS tagging. Every sentence was tokenized so tokens could match lexicons defined in lexical rules. Then, the grammar, and lexical rules for each sentence is defined so that the parser construct parse tree based on set of constituents defined in grammar production for developing model. The CFG rules, algorithm and productions rules were unit tested to find out errors, and each of them succeeded it.

Three different experiments were made on 250 Afaan Oromo simple and complex sentences by categorizing them into different ratios for defining grammar rules of each sentence. Upon incrementally adding raw data to previously parsed data and then correcting parse errors, the parser scored an average accuracy of 73.11%, and each parse tree was sound, terminate and complete. This result is promising for exploring further experiments for correcting ambiguities resulted due to rule contradictions so that the accuracy could be increased. Since length of sentence affects the parse tree, the prototype was developed for Afaan Oromo simple and complex sentences.

6.2. Conclusion

This thesis described the design and development Afaan Oromo text parser according to the language grammar construction rules. In this study the ways how to develop a text parser for Afaan Oromo using SVM algorithm with SMO classification function. All simple and complex Afaan Oromo declarative sentences are included in this study.

The thesis began with a brief discussion on the concept and applications of NLP at different levels. In this discussion, it is indicated that NLP and natural language understanding (NLU) require the general language structure of the data at different level of the applications to increase its capability. To achieve these text parser, which is a series of processes first identifying chunks from a sequence of tokens or words and second classify these chunks to some syntactically related classes, is used to achieve the above objective of NLP.

Parser is an important task in semantic parsing, question answering system, machine translation; information extraction, information retrieval, text summarization, and a variety of text-mining operations are among the top beneficiaries. The parser system developed is also useful for identifying the boundary of the parses, which will be very helpful in semantic parsing. When the text is converted to speech, identifying the clauses plays a major role.

Evaluation of the parser performance was made based on the evaluation procedures outlined in the thesis. The results achieved based on the small sample were high, 73.5%.

6.3. Recommendation

The computational infancy of Afaan Oromo needs the development of parser from the very beginning which in turn requires the computerization of other natural language features as prerequisites. There are some Challenges that the researcher could not be able to handle them easily. Those challenges are better research directions that need to be explored, and listed below as immediate recommendations and future works:-

1. Morphological Analysis

Morphological analysis is required for investigating parser in Afaan Oromo to remedy repeatedly defining lexicons of same root with different affixes so future researchers are recommended to develop this feature for Afaan Oromo.

2. Part-of-Speech tagger

Part-of-Speech tagger is also precondition for developing parser in Afaan Oromo; in hybrid approach it is useful for determining lexicons of given sentence and in statistical approach it is directly incorporated as one of the modules. This work is also recommended and targeted to coming researchers.

3. Parsing by chunks

Noun phrase chunking is shallower, easier and more efficient than CFG parsing (Abney, 1996). So for this short timed study development of chunker is recommended as future work; and again it will be used as a component for investigating full parsing.

Note: effective POS tagger is also mandatory for developing both full and shallow parser.

4. Treebank development

The manual development of annotated corpus (Treebank) minimizes the effort invested on parsing. Therefore, coming researchers are recommended to develop Afaan Oromo Treebank. The sequence of these recommendations minimizes the general challenges encountering in parsing. For example if there is no Treebank for the language, probabilistic parsing is really difficult.

Reference

- [1] M. A. Tayal *et al.*, “Syntax Parsing Implementation using Grammar-Rules for English Language (PDF Download Available).” .
- [2] M. Diriba, “AN AUTOMATIC SENTENCE PARSER FOR OROMO LANGUAGE USING SUPERVISED LEARNING TECHNIQUE,” ADDIS ABABA, 2013.
- [3] I. Russell, “An Introduction to the WEKA Data Mining System.”
- [4] K. Ribarov, “On the Rule-Based Parsing of Czech,” vol. 1993, pp. 77–100, 2002.
- [5] D. Grune, *PARSING TECHNIQUES*. 1990.
- [6] A. ALEMU, “AUTOMATIC SENTENCE PARSING FOR AMHARIC TEXT AN EXPERIMENT USING PROBABILISTIC CONTEXT FREE GRAMMARS,” ADDIS ABABA UNIVERSITY, 2002.
- [7] C. Paper and M. Tayal, “Syntax Parsing : Implementation using Grammar-Rules for English Language,” no. January 2014, 2015.
- [8] G. I. Webb, M. J. Pazzani, and D. Billsus, “Machine Learning for User Modeling,” no. 1978, pp. 19–29, 2001.
- [9] N. Khoufi, S. Louati, C. Aloulou, and L. H. Belguith, “Supervised learning model for parsing Arabic language 1,” 2003.
- [10] M. a. Tayal, M. M. Raghuwanshi, and L. Malik, “Syntax parsing: Implementation using grammar-rules for english language,” *Proc. - Int. Conf. Electron. Syst. Signal Process. Comput. Technol. ICESC 2014*, no. January 2014, pp. 376–381, 2014.
- [11] O. Al and A. Bayda, “Rule Based Shallow Parser for Arabic Language Mona Ali Mohammed and 2 Nazlia Omar Department of Computer Science , Faculty of Information Science and Technology , Department of Computer Science , Faculty of Science ,” vol. 7, no. 10, pp. 1505–1514, 2011.
- [12] S. D. Richardson and W. 98052 Redmond, “Bootstrapping Statistical Processing into a Rule-based Natural Language Parser.” .
- [13] T. Kudo and Y. Matsumoto, “Chunking with Support Vector Machines.”

- [14] “No Title,” 2014. [Online]. Available: <http://www.lmp.ucla.edu/Profile.aspx>.
- [15] R. Geetaachoo, *FURTUU: Seerluga Afaan Oromoo(Oromo Grammar)*. 2009.
- [16] B. Addunyaa, *SEMMOO: Bu’uura Barnoota Afaaniifi Afoola Oromoo*. 2014.
- [17] B. Addunyaa, *NATOO: Yaadrimee caasluga Afaan Oromoo*. Addis Abebe, 2012.
- [18] N. Abarraa, *Caasluga Afaan Oromoo*, Jildii 1. Finfinnee, 1995.
- [19] “Wikibooks,” *Wikibooks*, . .
- [20] C. Mellish and G. Ritchie, “The Grammatical Analysis of Sentences,” pp. 1–16.
- [21] G. M. Wegari, “Parts of Speech Tagging for Afaan Oromo,” pp. 1–5.
- [22] T. Mitsumori, M. Murata, and Y. Fukuda, “Semantic role labeling using support vector machines,” *Proc. Ninth ...*, pp. 5–8, 2005.
- [23] M. Gasser, “2.5 Quick Reference,” pp. 1–5, 2012.

Appendix 1: Training set

[[Inni]PrN [garuu]iP]NP [[[nama]N [magaala]N]NP [fakkaata]V]VP
[[Inni]PrN]NP [[[mana]N [gaarii]Adj]AdjP [ijaaree]V]VP
[[Tulluu]N[-n]dP]NP [[[mana]N [bareeda]P]AdjP [keessa]P [jiraata]V]VP
[[Caalaa]N[-n]dP]NP [[saree]N [baayee]Adj [jalaata]V]VP
[[Saree]N[-n]dP]NP [[isaaf]P [[hiriyyaa]Adj[-dha]dP]AdjP]VP
[[Buna]N [dhuguun]V]NP [[rakkoo]V [hin]iP [qabu]VB]VP
[[Mucaa]N [godhaachu]V [dura]P]NP [[karooraa]N [baafadhu]V]VP
[[Finfinnee]N[-n]dP]NP [[haandhura]N [[Oromiyaa]N[-tti]dP]PP [argamti]V]VP
[[Bunni]N [Jimmaa]N]NP [[bareedaa]Adj[-dha]Aux]VP
[[Caaltuu]N[-n]dP]NP [[[fardaa]N[-n]P]NP [galtee]V]VP
[[Tolaa]N[-n]dP]NP [[sanga]N [diimaa]Adj]NP [qale]V]VP
[[Inni]PrN]NP [[saree]N [ajjeesse]V]VP
[[Obbo]Det [[Caalaa]N[-n]dP]NP]NP [[bulchaa]N [aanaa]N [ta'an]V]VP
[[Addee]Det [[Roobee]N[-n]dP]NP]NP [[dubartii]N [harkaa]N [toletti]N[-dha]Aux]VP
[[Caaltuu]N[-n]dP]NP [[[suuta]AV]AVP [deemti]V]VP
[[Caalaa]N[-n]dP]NP [[[atattama]N[an]dP]PP [dhufe]V]VP
[[Har'a]AV [[aduu]N[n]dP]NP]NP [[baay'ee]Adj [ho'a]V]VP
[[Abdii]N[-n]dP]NP [[hojii]N [mana]N [sirrii]Adj[-tti]dP [hojjette]V]VP
[[Maraartuu]N[-n]dP]NP [[[kallessa]AV]AVP [deemte]V]VP
[[Inni]PrN]NP [[[gidduu]iP [kanaa]iP]PP [[biyyaa]N [alaa]N]NP [deema]V]VP
[[Margaa]N[-n]dP]NP [[dabboo]N [nyaate]V]VP
[[Margaa]N[-n]dP]NP [[[atattama]N[-an]dP]PP [gara]iP [Finfinnee]N [deeme]V]VP
[[Guutaa]N[-n]dP]NP [[har'a]AV [dhufe]V]VP
[[Leenci]N[-chi]dP]NP [[kallessa]AV [ajjefame]V]VP
[[Lalisee]N[-n]dP]NP [[[haalaan]AV]AVP [fiigdi]V]VP
[[Biqilaa]N[n]dP]NP [[gabaa]N [yemmuu]iP [deemu]V [kufe]V]VP
[[Ibsa]N[an]dP]NP [[ife]V]VP
[[Caalaa]N[n]dP]NP [[barata]N [[cimaa]Adj[dha]dP]AdjP]VP
[[Inni]PrN]NP [[[figiicha]N[an]dP]PP [dhufe]V]VP
[[Isheen]PrN]NP [[figiicha]N[an]dP]PP [dhufe]V]VP

[[Tolaa]N[n]dP]NP [[muka]N [kore]V]VP
 [[Lagni]N [Hawwaas]N]NP [[gute]V]VP
 [[Margaa]N[-n]dP]NP [[barsiisaa]N[-dha]iP]VP
 [[Finfinnee]N[-n]dP]NP [[magaala]N [guddoo]Adj [Oromiyaa]N[-ti]dP]VP
 [[Manni]N [Gammaada]N]NP [[gubate]V]VP
 [[Funyaan]N [isa]iP]NP [[[haanxaa]N [qoraanii]N]NP [fakkaata]V]VP
 [[Laataa]N[-n]dP]NP [[kubbaa]N [miilaa]N [taphaata]V]VP
 [[Gadaa]N[-n]dP]NP [[har'a]AV [[Finfinnee]N [deeme]V]NP]VP
 [[Tulluu]N[-n]dP]NP [[qoosaa]N [baayee]Adj [jalaata]V]VP
 [[Gammachuu]N[-n]dP]NP [[[mana]N [citaa]N]NP [guddaa]Adj]NP [ijaare]V]VP
 [[Gammachuu]N[-n]dP]NP [[[mana]N [citaa]N]NP [guddaa]Adj]NP [[Kaleessa]AV]AVP]NP
 [ijaare]V]VP
 [[Boonaa]N[-n]dP]NP [[[sangaa]N [foonii]N]NP [bite]V]VP
 [[Waraabessi]N]NP [[yuuse]V]VP
 [[Inni]PrN]NP [[muge]V]VP
 [[Mana]N [barumsa]N [deemte]V]VP
 [[Bishaan]N]NP [[dhuge]V]VP
 [[Raadiyoo]N[-n]dP]NP [[dubbate]V]VP
 [[Galataa]N[-n]dP]NP [[[farda]N[-af]P]PP [ukaa]N [haame]V]VP
 [[Boonaa]N[-n]dP]NP [[[Bilisee]N[-f]P]PP [qalama]N [kenne]V]VP
 [[Gadaa]N[-n]dP]NP [[[har'a]AV]AVP [[walgahii]N[-f]dP]PP [[Finfinnee]N [deeme]V]VP]VP
 [[Kitaabni]N [kun]iP]NP [[kan]iP [[Gammachuu]N[-ti]Aux]VP]VP
 [[Qalamni]N [kun]iP]NP [[Kan]iP [[koo]PrN[-ti]Aux]
 [[Inni]PrN]NP [[kophee]N [haaraa]Adj [bite]V]VP
 [[Abbaa]N[-n]dP [koo]PrN]NP [[qalama]N [naaf]PrN [kenne]V]VP
 [[Nuti]PrN]NP [rafnee]AV [jirra]V]VP
 [[Gammachuu]N[-n]dP]NP [[ni]iP [dhugaa]V]VP
 [[Ishiin]PrN]NP [[biyyaa]N [deemte]V [jirti]V]VP
 [[Yeroo]N]NP [[barnoota]N [kabaji]V]VP
 [[Tolaa]N[-n] [[gara]iP [mana]N [kitaabaa]N [deeme]V]VP
 [[[Abbaa]N[-n]dP]NP [mana]N [ishii]PrN]NP [[barsiisaa]N[-dha]Aux]VP

[[Guutaa][-n]dP]NP [[lallaafaa]N [dhuge]V]VP
 [[Rifeensi]N [[mata]N [koo]PrN]NP]NP [[luuccaa]N[-dha]Aux]VP
 [[Ani]PrN]NP [[[[kitaaba]N [dubbisuu]N[-n]dP]PP]NP [[jallaa]Adj[-dha]Aux]VP]VP
 [[Caaltuu]N[-n]dP]NP [[barattuu]N[-dha]Aux]VP
 [[Birbirsi]N]NP [[muka]N [[guddaa]Adj[-dha]Aux]VP]VP
 [[Intala]N gaarii]Adj]NP [[qabda]V]VP
 [[Muka]N [jiidhaa]N]NP [[falaxe]V]VP
 [[Inni]PrN]NP [[[[buna]N [Wallaggaa]N]NP [bite]V]VP
 [[Urgee]N[-n]dP]NP [[qarshii]N [shan]CD]NP [liqeeffette]V]VP
 [[Isaan]PrN [[mana]NP [tokkoo]CD [jiraatu]V]VP
 [[Obbo]Det [[Galataa]N[-n]dP]NP]NP [[obboleessa]N [kiyyaa]PrN]VP
 [[Galaanee]N[-n]dP]NP [[amala]N [gaarii]Adj [qabdi]V]VP
 [[Abdiisaa]N[-n]dP]NP [[gara]iP [shaambuu]N [deeme]V]VP
 [[Ibsaa]N[-n]dP]NP [[akka]iP [fardaa]N [fiiga]V]VP
 [[Ibsaa]N[-n]dP]NP [[Toleeraa]N[-rra]dP [kitaaba]N [fudhate]V]VP
 [[Namni]N [[mana]N [ijaare]V]NP [fi]iP]DC [[namni]N [horii]N [horsii]V [[gaarii]Adj[-
 dha]dP]PP]VP
 [[Yoo]iP [[dhufuu]V [[baatte]V[-llee]dP]PP]VP]DC [[xalayaa]N [naa]PrN[-f]dP]PP
 [barreessi]V]VP
 [[[Boonaa]N[-n]NP [biyyaa]N [alaa]N[-tii]dP]PP [akka]iP [[dhufee]V[-n]dP]PP]NP
 [[hiriyoota]N [[isaa]PrN[-f]dP]PP [dubbii]N [godhe]V]VP
 [[Yoo]iP [Finfinnee]N [[deemtee]V[-f]iP]PP]DC [[meeshaa]N [naa]PrN [bitta]V]VP
 [[Bokkaa]N [cimaa]Adj [waan]iP [[roobee]V[-f]dP]PP]DC [[lagni]N [guutee]V [riqicha]N
 [cabse]V]VP
 [[Ati]PrN]NP [[[[yoo]iP [deemsi]N [kee]PrN]NP [ture]V]DC [[biyya]N [gali]V]VP]VP
 [[Caalaa]N[-n]dP]NP [[[[osoo]iP [namni]N [isa]PrN]NP [ilaaluu]V]DC [[hin]iP
 [hamummatu]V]VP]VP
 [[Osoo]iP [qo'anna]N [hin]iP [jalqabiin]V]DC [[[Waaqayoo]N[-n]dP]NP [kadhahu]V]VP
 [[[Osoo]iP [qo'anna]N [hin]iP [jalqabiin]V]DC [[hin]iP [jalqabiin]V]VP]VP
 [[[yeroo]iP [qo'attu]V]DC [[hin]iP [jeeqiin]V]VP]VP

[[Boontuu]N[-n]dP]NP [[[yeroo]iP [baruulle]N [ishee]PrN]NP [qayyabattu]V]DC [[ni]iP
 [xiyyeefftti]V]VP]VP
 [[Buna]N]NP [[[yoo]iP [barbaadde]V]NP [[abidda]N [irra]V]NP [jira]V]VP
 [[Osoo]iP [nuyi]PrN [rafnu]V]DC [[[Caalaa]N[-n]dP]NP [[sawwaan]N [bobbaase]V]VP]
 [[Yeroo]iP [[Jigjigaa]N [deemne]V]VP]DC [[[Caalaa]N[-n]dP]NP [[nu]PrN [simaate]V]VP]
 [[Yeroo]iP [nama]N [loolu]V]DC [[ni]iP [dalaana]V]VP
 [[Inni]PrN]NP [[erga]iP [mataa]N [isa]PrN [dhukkubee]V]VP]NP [[ogeessa]N [gaafate]V]VP
 [[Erga]iP [[Qananiisaa]N[-n]dP]NP [[atilektiksii]N [dhiise]V]VP]NP [[[Moofara]N[-an]dP]NP
 [warqee]N [kaase]V]VP
 [[Erga]iP [[barnoota]N [xummure]V]VP [[daldalaa]N [ta'e]V]VP]VP
 [[yoo]iP [[seena]N [qoratte]V]VP]NP [[[eenyummaa]N [kee]PrN]NP [barta]V]VP
 [[Ati]PrN [yoommuu]iP [gabaa]N [dhaqxu]V]NP [[naa]PrN [waami]V]VP
 [[yoo]iP [[Gammada]N[-an]dP]NP [dhufe]V]NP [[Boolee]N [deemna]V]VP
 [[yoo]iP [hin]iP [dhufu]V [ta'e]V]VP [[xalayaa]N [barreessi]V]VP
 [[[Gaaddisaa]N[-n]dP]NP [yoo]iP [[dirqama]N [isaa]PrN [xumure]V]VP]NP [[Finfinne]N [gala]V]VP
 [[Yoo]iP [barnootni]N [cime]V]VP [[daldalaa]N [ta'i]V]VP
 [[[sangaa]N [abbaan]N]NP [ganfaa]N [cabse]V]NP [[Ollaan]N [ija]N [jaamsa]V]VP

Appendix 2

Scheme:weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K
 "weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"

Relation: simple

Instances: 309

Attributes: 5

W1

W2

W3

W4

Phrase_Type

Test mode:evaluate on training data

=== Classifier model (full training set) ===

SMO

Kernel used:

Linear Kernel: $K(x,y) = \langle x,y \rangle$

Classifier for classes: NP, VP

BinarySMO

Machine linear: showing attribute weights, not support vectors.

-0.646 * (normalized) W1=PrN
+ -0.6451 * (normalized) W1=NP
+ -0.6454 * (normalized) W1=N
+ -0.6456 * (normalized) W1=iP
+ 1.3543 * (normalized) W1=V
+ -0.6449 * (normalized) W1=Det
+ 1 * (normalized) W1=AV
+ 0.8727 * (normalized) W1=PP
+ -0.5561 * (normalized) W2=iP
+ 1.4443 * (normalized) W2=VP
+ -0.5556 * (normalized) W2=PrN
+ 1.4441 * (normalized) W2=V
+ -0.5558 * (normalized) W2=0
+ -0.4424 * (normalized) W2=Adj
+ -0.5557 * (normalized) W2=dP
+ -0.555 * (normalized) W2=CD
+ 0 * (normalized) W2=PP
+ -0.5555 * (normalized) W2=N
+ 1.4444 * (normalized) W2=Aux
+ -0.5567 * (normalized) W2=NP
+ -1 * (normalized) W3=P
+ -0.6077 * (normalized) W3=0
+ 1.3919 * (normalized) W3=V
+ -0.6058 * (normalized) W3=VP
+ -0.1273 * (normalized) W3=N
+ -0.6086 * (normalized) W3=PrN
+ 1.2793 * (normalized) W3=Aux
+ -1 * (normalized) W3=iP
+ 1.2783 * (normalized) W3=AdjP
+ -0.0007 * (normalized) W4=0
+ 0.0007 * (normalized) W4=V
+ 0.8097

Number of kernel evaluations: 22950 (86.831% cached)

Classifier for classes: NP, PP

BinarySMO

Machine linear: showing attribute weights, not support vectors.

0.1623 * (normalized) W1=PrN
+ -0.2703 * (normalized) W1=NP
+ -1.1892 * (normalized) W1=N
+ 0.811 * (normalized) W1=Adj
+ 0.0353 * (normalized) W1=iP
+ 0.811 * (normalized) W1=V
+ -0.1804 * (normalized) W1=Det
+ -0.1797 * (normalized) W1=AVP
+ 0.649 * (normalized) W2=iP
+ -0.5762 * (normalized) W2=VP
+ -0.5748 * (normalized) W2=V
+ -0.7031 * (normalized) W2=0
+ 0.6492 * (normalized) W2=dP
+ -0.2703 * (normalized) W2=AdjP
+ 0.1863 * (normalized) W2=N
+ 1 * (normalized) W2=Aux
+ -0.3601 * (normalized) W2=NP
+ 0.9483 * (normalized) W3=0
+ -1 * (normalized) W3=V
+ 0 * (normalized) W3=PP
+ -0.1346 * (normalized) W3=N
+ 0.1863 * (normalized) W3=PrN
+ 0.1346 * (normalized) W4=0
+ -0.1346 * (normalized) W4=V
- 1.5426

Number of kernel evaluations: 4710 (70.843% cached)

Classifier for classes: NP, AdjP

BinarySMO

Machine linear: showing attribute weights, not support vectors.

-0.1878 * (normalized) W1=PrN
+ -0.1902 * (normalized) W1=NP
+ -0.5857 * (normalized) W1=N
+ 1.4153 * (normalized) W1=Adj
+ -0.191 * (normalized) W1=iP
+ -0.0546 * (normalized) W1=V

+ -0.1027 * (normalized) W1=Det
+ -0.1033 * (normalized) W1=AVP
+ -0.1211 * (normalized) W2=iP
+ -0.1186 * (normalized) W2=VP
+ -0.119 * (normalized) W2=V
+ -0.1214 * (normalized) W2=0
+ 0.2771 * (normalized) W2=Adj
+ 0.2762 * (normalized) W2=dP
+ -0.12 * (normalized) W2=AdjP
+ -0.0236 * (normalized) W2=N
+ 0.2764 * (normalized) W2=Aux
+ -0.206 * (normalized) W2=NP
+ 0.0783 * (normalized) W3=0
+ -0.0546 * (normalized) W3=V
+ -0.018 * (normalized) W3=PrN
+ -0.0057 * (normalized) W3=iP
+ 0.0057 * (normalized) W4=0
+ -0.0057 * (normalized) W4=V
- 0.7748

Number of kernel evaluations: 3241 (83.143% cached)