ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY
SCHOOL OF EN GINEERING
DEPARTMENT OF COMPUTING

AUTOMATIC COMPLEX SENTENCE PARSING FOR AFAN OROMO TEXT: AN EXPERIMENT USING HIDDEN MARKOV MODEL (HMM).

BY

HACHALU ATOMSSA DABA

A Thesis Submitted to the School of Engineering Department of Computing In Partial Fulfilment Of The Requirements for the Degree of Masters of Science in Information Systems Engineering.

Adama, Ethiopia
September, 2016

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF COMPUTING

AUTOMATIC COMPLEX SENTENCE PARSING FOR AFAN OROMO TEXT: AN EXPERIMENT USING HIDDEN MARKOV MODEL (HMM).

BY

HACHALU ATOMSSA DABA

Signature of the Board of Examiners for Approval

Name	signature	Date
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

Declaration

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

_____

HACHALU ATOMSSA

September 2016

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

_____

Dr.Solomon Teferra (Ph.D.)
Addis Ababa University

Linguistic Advisor

Lemma Tesfaye (MA)
Arsi Univeristy

Acknowledgments

My heartfelt gratitude goes to my advisors, Dr. Solomon Teferra, who was my driving force throughout this study. He provided me a consistent support, and without whom such a study would have been incomplete. His constructive and invaluable comments have brought life to this study. Without the presence of his critical suggestions, this project would have been more demanding.

My deepest gratitude goes to my wife, Tsehay, my parents, W/ro Birhane Zikargachew and Ato Atomssa Daba, my sisters Hawii, and Meti who have all been my backbone in terms of providing me with moral and technical supports during my stay in the University.

I am also indebted to Ato Lemma Tesfay from Arsi University, who shared me his precious time voluntarily and allowed me to get his linguistic advice periodically.

My special thanks goes to my friends, Ato Alemayehu Taddese and Ato Mengistu Deyasa for their moral support.

My indebtedness goes to my beloved friend Ato Wondante Tolera, for his unreserved, all-rounded support and invaluable inspiration, and who technically assisted me during the coding of the algorithm.

Table of Contents

ACKNOWLEDGMENTS ..... **I**

LIST OF FIGURS ..... **V**

LIST OF TABLES ..... **VI**

ABBREVIATIONS AND SYMBOLS USED ..... **VII**

ABSTRACT ..... **VIII**

INTRODUCTION **1**

1.1. BACKGROUND 1
1.2. STATEMENT OF THE PROBLEM 4
1.3. OBJECTIVE OF THE STUDY 5
1.3.1. General Objective *5*
1.3.2. Specific Objectives *6*
1.4. METHODOLOGY 6
1.4.1. Literature Review *6*
1.4.2. Discussion with Linguists *7*
1.4.3. Data Collection *7*
1.4.4. Parsing Techniques and Prototype Development *7*
1.4.5. Testing Techniques *8*
1.5. APPLICATION OF RESULTS AND BENEFICIARIES 8
1.6. SCOPE OF THE STUDY 9
1.7. LIMITATION OF THE STUDY 9
1.8. ORGANIZATION OF THE THESIS 9

CHAPTER TWO **11**

REVIEW OF LITERATURE **11**

2.1 INTRODUCTION 11
2.2 AUTOMATIC SENTENCE PARSING (ASP) 11
2.3 APPROACHES TO AUTOMATIC SENTENCE PARSING 13
2.3.1 Rule-Based Approach to Automatic Sentence Parsing *14*
2.3.2 Stochastic Approach to Automatic Sentence Parsing *15*
2.3.3 Parsing Strategies *16*
2.3.3.1 Top-down Vs Bottom-up Parsing *16*
2.3.3.2 Left-to-right Vs Right-to-left *18*
2.3.3.3 Depth-first Vs Breadth-first *19*
2.3.3.4 Chart Parsing *19*
2.4 KNOWLEDGE REQUIRED BY THE PARSER 22
2.4.1 The Lexicon *22*
2.4.2 The Grammar Rule *23*
2.4.2.1 Context Free Grammars *24*
2.4.2.2 Transition Network Grammars *25*
2.4.2.3 Context Sensitive Grammars *25*
2.4.2.4 Unification-based Grammars *26*

2.4.2.5 Probabilistic Context Free Grammars (PCFG) *26*
2.5 RELATED NLP IN PARSING 28
2.6 RELATED NLP IN AFAN OROMO 28
2.7 RELATED NLP COMPONENT SYSTEMS 29
2.7.1 Morphological Analyzer *29*
2.7.2 Part-Of-Speech Tagger *30*
2.8 CONCLUSION 31

CHAPTER THREE **32**

THE STRUCTURE OF AFAN OROMO **32**

3.1 INTRODUCTION 32
3.2 AFAN OROMO ALPHABET AND WRITING SYSTEM 32
3.3 PUNCTUATION MARKS IN AFAN OROMO 33
3.4 WORD CATEGORIES IN AFAN OROMO 33
3.4.1 Categories of Nouns *34*
3.4.2 Categories of Verbs *36*
3.4.3 Categories of Adverbs *37*
3.4.4 Adpositions in Afan Oromo *37*
3.4.5 Conjunctions in Affan Oromoo *38*
3.4.6 Numerals *39*
3.4.7 Interjections *39*
3.5 PHRASAL CATEGORIES 40
3.5.1 A Phrase *40*
3.5.2 Noun phrases *41*
3.5.2.1 Accusative and Nominative Case *43*
3.5.3 Verb Phrases *45*
3.5.4 Adjective Phrase *46*
3.5.5 Adverb Phrase *46*
3.5.6 Adpositional Phrase *46*
3.6 SENTENCES 47
3.6.1 Afan Oromo Simple Sentences *47*
3.6.2 Afan Oromo Complex Sentences *50*

CHAPTER FOUR **53**

DATA PREPARATION AND PCFG EXTRACTION **53**

4.1 INTRODUCTION 53
4.2 THE DESIGN APPROACH OF THE PARSER 53
4.3 THE SAMPLE CORPUS 55
4.4 THE MORPHOLOGICAL PRE-PROCESSING 56
4.5 THE PART OF SPEECH TAGGER 56
4.6 EXTRACTION OF A PROBABILISTIC CONTEXT FREE GRAMMAR 59
4.7 CHOMSKY NORMAL FROM (CNF) REPRESENTATION 60
4.8 CONCLUSION 61

CHAPTER FIVE **62**

PARSING ALGORITHM AND EXPERIMENTATION **62**

5.1 INTRODUCTION 62
5.2 THE PARSING ALGORITHM 62
5.3 PCFG PARSING 64
5.3.1 Parse Tree *64*
5.3.2 Parse Chart *65*
5.4 THE DESIGN OF THE PARSER 67
5.4.1 Preprocessing the Parser's Input *67*
5.4.2 The Input & Output Interface *69*
5.5 THE EXPERIMENT 73
5.5.1 Experiment on the Training Set *75*
5.5.2 Experiment on the Test Set *75*
5.5.3 Results of the Experiment *75*
5.5.3.1 Result on the Training Set *76*
5.5.3.2 Result on The Test Set *76*
5.5.4 Solution to Identified Problems *77*

CHAPTER SIX **79**

CONCLUSION AND RECOMMENDATION **79**

6.1 CONCLUSION 79
6.2 RECOMMENDATIONS 81

REFERENCES ..... **XIII**

APPENDICES ..... **XIX**

APPENDIX A: THE SAMPLE TEXT ..... **XIX**

APPENDEX B ..... **XXII**

SAMPLE OUTPUT ..... **XXII**

List Of Figurs

:

Figure 2.1 : Example of parse tree ……13

Figure 2.2 : Example of parse tree 13

Figure 3.1: Sample Structure of simple Sentence. 49

Figure 3.2: The Structure of a Complex Noun Phrase 51

Figure 3.3: The Structure of a Complex Verb Phrase 52

Figure 5.1:- The words outside and inside Nk, lj..... 63

Figure 5.-2:-Tree structure diagram of the complex sentence 65

Figure 5.3:- An 8-level-chart..... 66

Figure 5-4:-The calculation of probabilities of non-terminal nodes 67

Figure 5.-5:-An algorithm for preprocessing an input sentence to the parser..... 69

Figure 5.6:- The Parse Chart Procedure to Implement the Inside-Outside Algorithm 72

Figure 5.7: The Overall Implementation of the Parsing Algorithm..... 73

Figure 5.8:-Diagrammatic Representation of the Parser 73

List of Tables

Table 4.1:- An example of the CNF rules with their respective probability values. 61

Table 5.1 :- Parsing result on Training Set before making no error correction 71

Table 5.2:- Parsing result on Training Set before making no error correction 76

Table 5.3:-Parsing result on Training Set after making some error correction..... 76

Table 5.4:-Parsing result on Test Set 77

Abbreviations and symbols used

Symbols

( )	What is inside is optional except those used to indicate cite (source) of documents used.
\	Used to separate words from their tags
‘ ’	What is inside is Afan Oromo word or sentence
“ ”	The Translation for Afaan Oromo into English

Abbreviations

CFG	Context Free Grammar
CNF	Chomsky Normal Form
CS	Complex Sentence
MC	Main Clause
NL	Natural Language
NLP	Natural Language Processing
NLU	Natural Language Understanding
PCFG	Probabilistic Context Free Grammar
SC	Subordinate Clause

Abstract

Natural language processing is a research area which is becoming increasingly popular each day for both academic and commercial reasons. Higher NLP systems (e.g., machine translation) are materialized only when the lower ones (e.g., part-of-speech tagger, syntactic parser) are successfully built. This functional dependency exists even among the lower NLP systems.

This thesis can be taken as an attempt to integrate ideas and outputs of previously attempted Afan Oromo part of speech tagger towards solving a bit further problem in parsing of complex sentence language.

Syntactic parsing underlies most of the applications in natural language processing. Parsers are already being used extensively in a number of disciplines such as in computer science (for compiler construction, database interfaces, artificial intelligence, etc), and in computational linguistics (for text analysis, corpora analysis, machine translation, etc.).

Although there have been some comprehensive studies of Afan Oromo syntax from a linguistic perspective, attempts for investigating it from a computational point of view is a very recent story. In this thesis, an attempt to extract such features as Afan Oromo word and phrase classes, sentence formalisms that enable implementation of automatic Afan Oromo complex sentence parser is presented.

The sample data used in this study has been taken from references that are widely used in the teaching-learning process of the language. This data has also been manually annotated and analyzed, tagged, parsed, and then used as a corpus to extract the grammar rules and to assign probabilities.

Experiments have been conducted in 300 complex sentences having 3029 words. In this study, 80% of the corpus (240) sentences have been used for training set and 20% (60) complex sentence have been used as test set.

The part of speech tagger integrated on this study using 3029 manually annotates words used 28 category of tag sets. Out of these the 27 of them are a tag category whereas the 28th one "X" is used for unknown words. The result of this experiment showed that the tagger attained 89.7% and 84% of accuracies on the training set and the test set, respectively.

The experiments on complex sentence parsing showed 80.0% accuracy result on the training set and 71.6% accuracy result on the test set prepared for this purpose.

CHAPTER ONE

INTRODUCTION

1.1.Background

A natural language or an ordinary language is a language that is spoken, written, or signed by humans for general-purpose communication, as distinguished from formal languages (such as computer-programming languages or the "languages" used in the study of formal logic). [1]

Language is one of the fundamental aspects of human behavior and it constitutes a crucial component of our lives. In its written form it serves as a means of recording information and knowledge on a long term-basis and transmitting what it records from one generation to the next. In its spoken form it serves as a means of coordinating our day-to-day life with others. [2]

Human language, whether in the written or spoken mode, is a fundamental part of human communication. Any hope of providing computer systems that claim intelligence approaching that of a human, therefore, rests on the hope of providing communication in Natural language. Machine translation of natural languages accurately and in real time is a typical example of such systems that activates researches in the area of computational linguistics. [3] The computational activities required for enabling a computer to carry out information processing using natural language is called natural language processing. [4]

Natural language processing (NLP) is a branch of artificial intelligence that deals with analyzing, understanding and generating the languages that humans use naturally in order to interface with computers in both written and spoken contexts using natural human languages instead of computer languages. It studies the problems of automated generation and understanding of natural human languages. [1]

NLP is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications. [5]

In order to fully realize impressive capabilities of NLP, intensive research works should be conducted at different levels of natural language processing (NLP): phonological (i.e. Sounds or combinations of sounds), morphological (processing of individual word forms, lexical (procedures operating on full words), syntactic (grouping the words of a sentence into structural units), semantic (contextual knowledge to the purely syntactic process in order to restructure the text into units that represent the actual meaning of a text), and pragmatic (using additional information about the social environment in which a given document exists). [6] As Warner [6] stated, the incorporation of information about the underlying organization of the data at various levels as described above helps in understanding the nature of the language under consideration and, therefore, to effect the desired systems.

Currently, many applications in NLP area require linguistic knowledge to be readily available to successfully implement many NLP systems at various levels of processing. There are, for instance, systems developed for processing natural language at phoneme, word, sentence, and pragmatic levels. These systems are developed in such a way that the output of a lower system can serve as an input for the next higher level. For instance, the output of a morphological synthesizer that works at word level could serve as an input for syntactic and semantic parsers that work at sentence level. [2]

There are a number of works carried out on this area in different languages including English. The main purpose of all these works are to enable computers understand human languages [6]. However, as far as the knowledge of this paper is concerned, there are few research works done so far on any one of the Ethiopian languages, including Afan Oromo, the concern of the present paper. Thus, this research will take the opportunity to shade light on syntactic processing. [7]

Afaan Oromoo is one of the major African languages that is widely spoken and used in most parts of Ethiopia and some parts of other neighboring countries like Kenya and Somalia [8] and [9] . It is used by Oromoo people, who are the largest ethnic group in Ethiopia, which amounts to 34.5% of the total population. Besides first language speakers, a number of members of other ethnicities who are in contact with Oromoo's speak it as a second language, for example, the Omotic-speaking Bambassi and the Nilo-Saharan-speaking Kwama in northwestern Oromia [10].

Currently, Afan Oromo is an official language of Oromia regional state (which is the largest Regional State among the current Federal States in Ethiopia). Being the official language of the regional state, it has been used as medium of instruction for primary and junior secondary schools of the region. [10]

Syntactic processing (Parsing, now on wards) is the step in which a flat input sentence is converted into a hierarchical structure that corresponds to the units of meaning in a sentence. Volk [11] explains syntactic parsing as "partial functions that map features to atomic features values or to syntactic categories". Broadly speaking, parsing is the de-linearization of linguistic input, i.e. the use of syntax to determine the functions of words in the input sentence in order to generate a data structure that can be used to get at the meaning of the sentence [12]

The syntactic structure of a sentence indicates the way that words in the sentence are related to each other. The major objective of this phase is to transform the potentially ambiguous input phrase into unambiguous forms.

Today there are a lot of parsing systems developed for various languages of the world starting from the early 1960s, including English [13]. The first of this kind is Earley's parser. This parser is considered as the first efficient chart parser for English language. From then on wards, there have been a lot of attempts to develop such Automatic Sentence Parsing (ASP) to the languages in the world. [14]

As Afan Oromo is one of such languages that should have an automatic sentence parser, as far as the knowledge of the current paper is concerned, there is only one such kind of system developed so far (Diriba , [7]) for the language. Thus the development of such an automatic system is of some importance. Hence, this paper addresses this problem and will try to fill the gap experienced by previously conducted research regarding the development of an automatic sentence parser for the language and shades lights for further works in this area.

1.2.Statement of the Problem

There have been comprehensive studies of Afan Oromo syntax from a linguistic perspective. However, attempts for investigating Afan Oromo extensively from a computational point of view were almost nonexistent for a long time. Very recently, NLP for local languages such as Afan Oromo has become an area of research interest and there are some studies that have been and are being conducted on NLP of Afan Oromo. 'An automatic sentence parser for Afan Oromo by Driba [7], a Part-of-Speech Tagger (POST) for Afan Oromo by Getachew [15], Automatic Part-of-speech Tagging for Oromoo Language Using Maximum Entropy Markov Model by Abraham Tesso Nedjo [16] Designing a Rule Based Stemmer for Afan Oromo Text by Debela Tesfaye [17], Analysis of Rule Based Approach for Afan Oromo Automatic Morphological Synthesizer by Abebe Abeshu , Afan Oromo News Text Summarizer by Girma Debele [10].

Even though there are some attempts towards an automatic sentence parsing for simple Afan Oromo sentence and phrases by Diriba [7], still having Parsing systems which could be applicable for all kinds of Afan Oromo sentence are useful in many areas of NLP for the language. Moreover, parsing is the step toward the process of semantic representation. Thus, the development of automatic parsing technique for both simple and complex Afan Oromo sentences would help the later development of such systems as semantic parsers, spell checker, machine translation, text extraction, and other important components in the language, Afan Oromo.

The absence of syntactic parser for both simple and complex sentences in Afan Oromo limits, if not completely hinders, later efforts of making computer understand Afan Oromo. Hence, this paper tried to fill such gap in the language.

In addition, the development of automatic sentence parser for both simple and complex Afan Oromo sentences will avoid the large amount of time wasted to manually pars sentences in the language to show its syntactic structure. The current Parsing system developed for this language will also be used as component in many other applications. These include phrase recognition, grammatical function assignment, recognition in message extraction systems, and generation of intonation in speech production systems and many others [18]. As a result, Afan Oromo language will benefit a lot from the development of automatic sentence parser.

In language teaching of the sentences and in identifying phrase, simple and complex sentence categories of the language, the member of each category and the relationships that hold between categories would be taught. In this regard, the linguistic structure of sentences in the language could be discovered easily if a parser is developed for the language. The lexical categories information is obtained using the automatic sentence parser as a component which facilitates higher levels of analysis such as noun phrase recognition, semantic and others.

1.3. Objective of the Study

1.3.1. General Objective

The general objective of this study is to explore the possibilities of integrating previously attempted studies for simple sentence parsing in order to be able to parse complex Afan Oromo sentences.

1.3.2. Specific Objectives

In line with achieving the general objective stated above, the research would attempt to address the following specific objectives.

- Review the basic word categories, morphological properties, phrase structures, and the various kinds of sentences of Afan Oromo with the aim of investigating patterns that allow computer representation;
- Collect sample simple and complex sentences to be used in the experiment;
- Build the database of the part-of-speech tagger using the stems of words taken from the sample corpus and calculate the lexical and transitional probabilities for them.
- Generate the grammar rules appropriate for the language;
- Use a combination of statistical lexical co-occurrence techniques in order to guess the POS for unknown words;
- Customize the parsing algorithm that was used for automatic simple sentence parser for Afan Oromo Diriba [7];
- Develop a prototype parser that will implement the findings of the study and test it to measure its performance;
- Finally, forward recommendations based on the findings of the study.

1.4. Methodology

Obviously, developing an NLP system such as sentence parsing, requires thorough investigation and identification of the properties of the target language. Hence, in order to achieve the objectives of this research, the following methods have been used.

1.4.1. Literature Review

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 of these helped the researcher

to understand both the issues regarding NLP, particularly automatic sentence parsing (e.g., approaches, techniques and strategies), and issues of the language under consideration (i.e., the basic word categories, morphological property, phrase structure, and the various types and kinds of sentences of Afan Oromo). Besides this, literature in the area of parsing in particular and computational linguistics in general (e.g. algorithms and data structures used) was reviewed to better understand how a language works. This understanding, in turn, has enabled the researcher to implement the features of the language that have been found appropriate to the study, and to adopt the parsing algorithm appropriately.

1.4.2. Discussion with Linguists

Successive discussion with linguists and experts in the area of Afan Oromo language at different universities and Institutes of Language Studies have also been made for better understanding and analysis of Afan Oromo complex sentences and its sub-components, particularly complex phrase structures of the language. These thorough discussions with linguists; both native and non-native Afan Oromo speakers, helped the researcher to understand the well-formed-ness of sentences in Afan Oromo and about the correctness of the parser.

1.4.3. Data Collection

Two Different types of sentence i.e 10 simple and 300 complex of Afan Oromo, have been collected from published books, magazines and newspapers. The sentences has been selected in such a way that the complex sentences would have simple noun phrase and complex verb phrase. Whereas simple sentences would have simple noun phrases and simple verb phrase. Hence, due attention to language structure and type so that all the sentence selected could fulfill the required language structure that will benefit the research process have been given.

1.4.4. Parsing Techniques and Prototype Development

In data preprocessing, 3029 words have been annotated with their respective category of tags. Then, the string that stems were tagged using HMM, have been used to determine the

necessary lexical dictionary, Probabilistic Context Free Grammar (PCFG) rules. Based on these a parsing algorithm have be adopted and modified in order to generate the sentence parser.

The development of the prototype using NLTK Tkinter and implementation of the parsing algorithm using python have been carried out.

1.4.5. Testing Techniques

Two types of data sets have been used during the experimentation: training set and test set. Out of the total 300 complex sentences 80% of the sample corpus were picked randomly as training set while the rest 20% of the corpus were used as a test set. More over 10 simple sentences were taken and tested to confirm that the algorithm also works for simple sentence to. The experiment have been conducted in two phases: first on the training set and next on the test set and the results have been evaluated.

The parse outputs were crosschecked with manually parsed sentences and the experiment iteratively continued until no more improvement is seen. In other words this have been done until the prototype reaches the maximum possible level which is the closest to the one that had been found in the manual.

Finally, the figures obtained from the observed results have been statistically summarized and analyzed in a way that is suitable to report the attained accuracy level.

1.5. Application of Results and Beneficiaries

Syntactic parsers are one of the core components of higher level NLP systems. That means, parsing systems would also play decisive roles in many areas of NLP for Afan Oromo language. Hence, conducting research on the area of NLP spatially sentence parsing has tremendous significance in such a way that researchers who involve themselves in increasing the capability of computers in processing Afan Oromo language will benefit from the results of this study. Specifically, researchers interested in the area of conceptual parsing, machine translation, phrase recognition, spell checker, text summarization, etc. will be among the top beneficiaries. Besides, linguists and students in the field of Afan Oromo language might

also apply the output of this research to parse sentences in the language automatically. The output could also be used in language teaching, for recognition of phrasal categories, and to see the relationship between words in a sentence.

1.6. Scope of the Study

The scope of this study is confined to dealing with Afan Oromo complex declarative sentences that are formed in one of the three possible ways of complex sentence formation, i.e. from a simple noun phrase and a complex verb phrase. This is because of such resource limitations as time, cost and labor. Therefore, this prototype parses complex Afan Oromo sentences that are composed of a simple noun phrase and a complex verb phrase (predicate) only.

1.7. Limitation of the Study

This study has the following limitations:

1. The study did not incorporate all kinds of Afan Oromo sentences with there attributes like case, numbers, genders, person and tenses.
2. The prototype developed used manually annotated morphological analysis prepared for the purpose of this study. This is due to lack of the source code of the morphological analyzer for Afan Oromo which was previously developed.
3. The complex sentences that are included in the sample do not exhibit complex noun phrases, interrogative.
4. Moreover, the researcher of this study believed that the size of the corpus used is still very small.

1.8. Organization of the Thesis

This study is organized in six chapters. Chapter one introduces what an NLP is and the role of a syntactic analysis for NLP. This chapter also presents the statement of the problem and the objective of the study. Different approaches and strategies for developing an automatic sentence parser and issues related to sentence parsing are discussed in chapter two. The third

chapter describes the Afan Oromo language, i.e. the various word classes, phrase categories, and complex sentence formalisms in the language. Also, pertaining to complex phrases and sentence structures in the language are discussed here. Chapter four describes the design of the lexicon for Afan Oromo complex sentence parser. The algorithm designed, the experiment conducted, and the results observed are reported in chapter five. Finally, the conclusions and recommendations made based on the findings of the study are presented in chapter six.

CHAPTER TWO

REVIEW OF LITERATURE

2.1 Introduction

As already been indicated in the first chapter, the main objective of the study is to design and implement an automatic Afan Oromo sentence parser for complex sentence. It is a standard practice to consider the grammar as well as the parsing technique during the development of parsers for the purpose of examining how the syntactic structure of sentence can be computed. 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.

In this chapter different techniques and approaches to sentence parsing are discussed. The first section gives an overview of automatic sentence parsing and its evaluation criteria. In the second section, the different approaches and techniques to the task of automatic sentence parsing are reviewed. The third section of this chapter discusses the grammar rules and the lexicon which is, the knowledge required by the parser. And, under the last section, some previous Afan Oromo NLP research endeavors that are related to this study are summarized subsequently.

2.2 Automatic Sentence Parsing (ASP)

A natural language system must use considerable knowledge about the structure of the language, including what the words are, how words are combined to form sentences, what the words mean, how word meanings contribute to sentence meanings and so on [19].

The term parsing is derived from the Latin phrase *pars orationis* (part of speech) and it refers to the process of assigning a part of speech (Noun, Adjective, etc.) to each word in a sentence, and grouping the words into phrases Alen [19]. However, in natural language processing the task of parsing may be carried out at a word or sentence level. Word parsing is a

process of tokenizing a word into its components or individual morphemes. The word should be tokenized into morphologically valid components. These tokenized components will further be analyzed for their contribution to the meaning and categorization of the whole word [2] [20].

Sentence Parsing (syntactic 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. In other words, it is a task in NLP in which a flat input sentence is converted into a hierarchical structure that corresponds to the units of meaning in the sentence, Allen [19]. The input string (or tokens) is passed to the parser token by token. For each token the parser calls the morphological analyzer, which segments words into roots and affixes according to the morphological rules of the language (Afan Oromo). Roots and affixes are stored in a lexicon, which consists of a set of records relating various types of linguistic information.

The parse tree is nothing but a diagrammatic representation of the input sentence and it records the rules of the language and how they are matched. Every node of a parse tree corresponds either to an input word or to a non-terminal in the grammar, Allen [19]. Each level in the parse tree corresponds to the application of one grammatical rule. However, the final terminal symbols are connected to the input word related to its lexical category.

'Gaangeen, Tolaan kalessa bitee hara ganama duute'

"The mule that Tola bought ide yesterday"

```
(S
  (NP
    (NP (N Gaangeen))
    (ADVP (NP (N Tolaa)) (ADVP (ADV kaleesa) (V bitee))))
  (VP (ADVP (ADV hara) (ADP ganama)) (VP (V duute))))
```

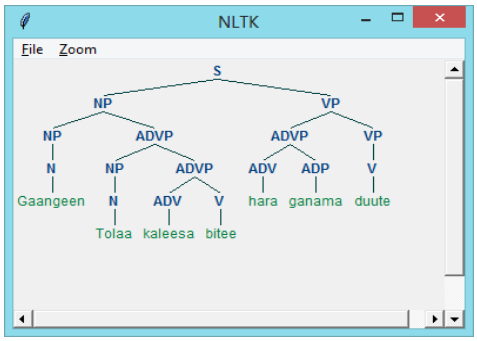


Figure 2.2 : Example of parse tree

The symbols like S, NP, VP, ADVP, N, V, ADP, ADV and JJ are used here to represent sentence, noun phrase, verb phrase, adverb phrase noun, V, adverb and adjective respectively. The process of parsing can be carried out either manually or automatically. The Manual process is tiresome, prone to error and expensive as the volume of text to be parsed gets bigger and bigger. On the other hand, automatic sentence parsing avoids such complexities and plays important roles in natural language understanding systems. Currently there exist several different approaches to the task of sentence parsing, which are broadly grouped into rule-based approaches and statistical approaches. The following section discuss those approaches.

2.3 Approaches to Automatic sentence parsing

Parsing is a complex process because it deals with the investigation of computational and Psychological issues at the same time. Viewing parsing as putting a mental grammar to use has often imposed stringent requirements on the forms of the grammars used, mostly requiring that the grammar proposed by the linguists be used directly. Were the study of grammars not viewed as the study of a mental system, there would be no need to have a precise and motivated mapping between the grammar and the parser. [21] [22]

As long as the parser recovers the same structural descriptions that are assigned by the grammar to the strings in the language, no other stricter relation would be needed. The parser is subject to time/space constraints, which are irrelevant to the grammar, and these are the only pressuring constraints to shape the architecture of the parser and the techniques. [13] [21] [23]

Sentence parsing requires exploring a way in which that sentence could have been generated from the start symbol (usually sentence written as an **S**). [24] [19] [21] A lot of investigations

have been made so far and still under way to explore such techniques that efficiently parse a sentence. But the attempts made so far and the techniques discovered so far to respond to this problem can be categorized, in general, in two ways: Rule-based versus Stochastic. [7]

2.3.1 Rule-Based Approach to Automatic Sentence Parsing

As Mao [23] reports, Rule-Based system learns a set of rules automatically based on a given tokens (strings) and then parses sentences following these rules. But, this approach doesn't make any use of probabilities (or 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.

Rule-Based parser consists of different components in the process of parsing a sentence. [19] The major component of any parser is the grammar rules (sometimes called Rewrite rules, production rules). These are the rules that the parser consults every time it starts parsing a sentence.

As Diriba [7] reports lexicon (or dictionary) is another major component of rule-based parser. A parser requires a lexicon, which are lists of all the grammatical categories of words 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.

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 [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. [7]

2.3.2 Stochastic Approach to Automatic Sentence Parsing

Stochastic-based parsers use probability (i.e. statistics) in analyzing the problem of parsing. The stochastic approach, also called corpus-based approach, is based on the ideas of Bayes (Network) theorem, independent events and the Markov assumption in sentence parsing. The approach uses these concepts to determine the most likely lexical sequence of each word in a given sentence [7]. Based on the type of text corpora used, the corpus-based approach can be further categorized into supervised and unsupervised approaches [19].

Supervised approaches use annotated text corpora while unsupervised approaches use natural corpus as those found in newspaper and books.

Automatic syntactic analysis systems developed following the supervised approach are called supervised parsers, and they use probability (i.e. statistics) in analyzing the problem of parsing. There are two important information sources in a supervised parser: the lexicon, which lists each word with the entire possible lexical category for each word with its respective estimate of lexical probabilities,[1] and the list of contextual probabilities for each lexical category. The list of contextual probabilities indicates the particular lexical category that is appropriate for a particular context. [19]

The major problems in developing supervised parsers include lack of manually or automatically parsed text (corpora) and the fact that it needs to manual parsing each time the parser is applied to a new text [19]. Manual parsing is really expensive and time consuming, but if pre-tagged corpora are easily available, the Hidden Markov Model (HMM) approach in particular and stochastic parsers in general can be adopted for new languages with little effort.

Parsers developed using unsupervised stochastic techniques do not require any pre-tagged text in the training process. Some heuristics or probabilistic information generated from the corpus is used to develop the syntactic analysis system [7] [3] . These parsers are similar to their supervised counterparts in that both assume the HMM assumption. HMM is a set of states

[1] *Lexical probability is the probability of a word to have a particular label and is usually denoted by p([word]lexical category).*

(lexical categories in this case) with directed edges labelled with transition probabilities that indicate the probability of moving to the state at the end of the directed edge, given that one is now in the state at the start of the edge. The states are also labelled with a function which indicates the probabilities of outputting different symbols if in that state (while in a state, one outputs a single symbol before moving to the next state). In this case, the symbol output from a state/lexical category is a word belonging to that lexical category [3]. Besides, both types of parsers use a large dictionary to list all possible lexical categories for each entry, and also use statistical, rule-based or hybrid techniques to achieve disambiguation, etc. However, the unsupervised stochastic parser has such unique features as training takes place on an unparsed or fresh text, uses the Baum-Welch algorithm (which is different from the Viterbi algorithm), and so on.

2.3.3 Parsing Strategies

Various strategies have been proposed to address such questions concerning to parsing: Where do we start from?, In what order is the string or the Right Hand Side (RHS) of a rule looked at? , and how are alternatives explored? Researchers in the area of NLP provided strategies like top-down, bottom-up, left-to-right, right-to-left, depth-first, breadth-first, and chart parsing as solutions. The following consecutive subsections discuss some of the major solutions provided at various times.

2.3.3.1 Top-down Vs Bottom-up Parsing

Top-down and bottom-up are rival solutions that have been proposed as alternative solutions for the strategy question regarding the direction of the parsing process. Top-down parsing begins with the start symbol (usually a sentence S) and applies the grammar rules forward until the symbols at the terminals of the tree correspond to the components of the sentence being parsed. For example, the parser starts in state (S), after applying the rule S → NP VP, the symbol list will be (NP VP). It then applies the rule NP → ART N and the symbol list will be (ART N VP), and so on. The parser could recursively continue in this fashion until it arrives entirely at terminal symbol states, and then it could check the input sentence to see if the classes of words in it matched with the rewritten

sequence of terminal symbols [19]. Parsers developed using this approach are called top-down parsers. This parser makes a hypothesis about what it is looking for and to decide its next move. Hence, top-down parser is characterized by a sequence of goals to determine the remaining words.

On the other hand, 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 (usually S, for sentence) has been produced [3]. That is, it starts from each word and assign its grammatical category until it reaches the start symbol. This operation is repeated, at each state, using the sequence of highest-level labels as the new string to operate on. The task of the parser would now appear to be that of attempting to group words into their respective categories together (e.g. take a sequence ART ADJ N and identify it as an NP) in a manner permitted by the grammar.

Top-down methods have the advantage of being highly predictive. This means that a word might be ambiguous when considered in isolation, but if some of those grammatical categories cannot be used in a legal sentence, then these categories may never even be considered [19].

However, this method has a reduplication of effort that becomes a serious problem, and large constituents may be rebuilt again and again as they are used in different rules. In contrast, the bottom-up parser only checks the input sentence once, and only builds each constituent exactly once. The basic observation to make about the bottom-up parser is that it works from left to right: it does everything it can with the first item before exploring what it can do with the next two items, and so on. That is, the parser is bottom up, driven entirely by the data presented to it and building successive layers of syntactic abstraction on the basis of data provided.

Unfortunately, Allen [19] states whether one chooses top-down or bottom-up to implement, the payback is prohibitively expensive, as the parser would tend to try the same matches again and again, duplicating much of its work unnecessarily. Hence, to

avoid such reduplication problem there should be a mechanism that allows parser to store results of the matching it has done so far. Such technique is called chart-based parsing.

Therefore, the combination both strategies can result in a better parser. A small variation in the bottom-up chart algorithm yields a technique that is predictive like the top-down approaches, yet avoids any reduplication of work as in the bottom-up approaches.

2.3.3.2 Left-to-right Vs Right-to-left

These are the competing solutions that have been forwarded for the question regarding the order of looking at substrings or an RHS. Left-to-right parsing processes the words of the sentence from left to right (i.e. from beginning to end), as opposed to right-to-left (i.e. end-to-beginning) parsing. That is, it uses the leftmost symbol first, continuing with the next to its right. Logically, it may not matter which direction parsing proceeds in, and the parser will work, eventually, in either direction [25]. However, right-to-left parsing is likely to be less intuitive than left-to-right. But there are situations in which using both strategies becomes useful. If the sentence is damaged, for example, by the presence of a misspelled word, it may help to use a parsing algorithm that incorporates both left-to-right and right-to-left strategies, to allow one to parse material to the right of the error.

The top-down strategy encounters trouble with rules that exhibit left recursion when it is performed from left to right [25]. Left recursion arises when the first category on the RHS of a rule is more general than the one on the LHS (Left Hand Side). In this case, it is possible to transform a left-recursive grammar into an equivalent one with no left-recursive rules, and one that generates exactly the same set of strings (although it will not assign the same structures). However, the bottom-up never makes such thing and it only checks rules to see if there is a legitimate way of putting together the words already in hand [19]; [26]; [27] . Such a parser encounters problem with empty grammar rules.

2.3.3.3 Depth-first Vs Breadth-first

As far as the question of exploring parse space (i.e., alternative parses) of a given input sentence is concerned, there are two contending solutions, depth-first and breadth-first. In the former case a single alternative parse is completely pursued at every choice point before trying another alternative. State of affairs at the choice points needs to be remembered and choices can be discarded after unsuccessful exploration. Breadth-first, on the other hand, pursues every alternative parse at every choice point for one step at a time. It requires massive bookkeeping since each alternative computation needs to be remembered at the same time. While depth-first search is generally incomplete, breadth-first search is guaranteed to be complete.

2.3.3.4 Chart Parsing

The chart is a data structure for representing partial results of parsing process in such a way that they can be reused later on. The chart for an n-word sentence consists of n+1 vertices and a number of edges that connects vertices. A chart parser is a variety of parsing algorithm that maintains a table of well-formed substrings found so far in the sentence being parsed. While the chart techniques can be incorporated into a range of parsing algorithms, mostly they have been used in the context of a particular bottom-up parsing algorithm. [12]

The main idea behind this technique is that the essence of improving parsing efficiency. As indicated by Russell and Norvig [28], there are three points to keep in mind for efficiency consideration of chart parsing: it is advisable not to do twice what can be done once, not to do once what can be avoided altogether and not to represent distinctions if that is not the concern of the study.

Chart parsers maintain the records of all the constituents derived from the sentence so far in the parse. That is, it stores the intermediate results and maintains the record of rules that have matched but are not completed. To be more specific, for example, once it is discovered that '***reenfi tiskee hoolota loolaan ajjesee,' "***'The body of the shepherd that

was killed by flood" is a complex NP as it is used in the sentence '***reenfi tiskee hoolota loolaan ajjesee gara hospitalaatti ergame'***, "The body of the shepherd that was killed by flood was sent to hospital", it is recommended to record that results in a data structure known as a chart. Recording of intermediate results is a form of dynamic programming that avoids duplicate work [19] [28].

It can be observed from the ongoing scenarios that, chart-based techniques use a combination of top-down and bottom-up processing in a way that means it never has to consider certain constituents that couldn't lead to a complete parse. (This also means it can handle grammars with both left-recursive rules and rules with empty Right Hand Sides without going into an infinite loop). The result of the algorithm is a **packed forest** of parse tree with its constituents labeled with their respective grammatical categories in the trees.

The basic operation of a chart-based parser involves combining an active arc (also called edge) with a completed constituent. The result is either a new completed constituent or a new arc that is an extension of the original active arc. New complete constituents are maintained on a list called agenda until they themselves are added to the chart. For example, [0, 5, S NP VP•] means that an NP followed by a VP combine to make an S that spans the string from 0 to 5. The numbers here indicate the indexing of the grammar rules. The symbol • in an edge separates what has been found so far from what remains to be found. The numbers before the symbol S indicates the indexing of the grammatical categories. Edges with • at the end are called complete edges. The edge [0, 2, S NP • VP] says that an NP spans from 0 to 2, and if the parser could find a VP to follow it, then the parser would have an S. Edges like this with the dot before the end are called active arcs.

Chart-based parsers are more efficient than those that rely on a search since the same constituent is never constructed more than once. A pure top-down or bottom-up search strategy could require up to C^n operations to parse a sentence of length n, where C is a constant that depends on a specific algorithm used. Even if C is very small, this exponential complexity rapidly makes the algorithm unusable [19] [16] [29].

However, it is reported that chart-based parser would require K * N^2 time and space complexity to build each constituent as lexical categories between every positions, where N is the length of the sentence and K is a constant depending on the algorithm. Thus, it reduces parsing operations substantially. In chart parsing, the process of parsing an n-word sentence consists of forming a chart with n+1 vertices and adding edges to the chart one at a time, trying to produce a complete edge that spans from vertex 0 to n and is of category S. There is no backtracking; everything that is put in the chart stays there.

In general, there are two different issues that are of concern. The first involves improving the efficiency of parsing techniques by reducing the search but not changing the final outcome, and the second involves techniques for choosing between different interpretations that parser might be able to find. To accomplish this, the following technique is usually employed.

It was noted earlier that the bottom up failed to store any intermediate results. That is the key reason for its obsessively time-wasting activity in rechecking things that it has already checked and which cannot be changed. This can be considered as amnesiac! It checks each new category to see if it exhausts an RHS and check each new adjacent pair of categories to see if they exhaust an RHS. This makes it slightly harder to use, since the implementation now has to make it impossible to keep a record of which categories a parser is in [19] [28] [29].

This issue of storage of intermediate results is independent of the distinctions already discussed, even though any parser needs to store some states, simply to remember what it is currently doing; in particular, chart parser has to remember the multiple hypothesis states that are currently being entertained. Storage of intermediate results also turns out to be the key to efficient parsing. Chart-based parsers use a data structure called a chart to encode all intermediate results obtained in the course of a parse [19] [28] [27].

One way to improve the efficiency of parsers is to use techniques that encode uncertainty, so that the parser need not make an arbitrary choice and later backtrack. Rather, the uncertainty is passed forward through the parse to the point where the input eliminates all but one of the

possibilities. The efficiency of the technique described here arises from the fact that all the possibilities are considered in advance, and the information is stored in a table that controls the parser, resulting in parsing techniques that can be much faster.

It can be observed that chart parsers are more efficient than any other parsers. They encode intermediate results so that reduplication of effort is avoided. Moreover, chart parser encodes uncertainty to avoid ambiguity and predicts the word category of unknown words (those that are not in the knowledge base). Therefore, the present study will implement chart parser suggested by Allen [19] [30] [31] to avoid uncertainty and predict unknown words' category.

2.4 Knowledge Required by the Parser

In order to write efficient parsing algorithms, one needs to have a secure definition of the language to which he or she designs the parser. Much of the knowledge for NLP purpose must come from linguistic studies. This helps us understand better the organization and operation of the language. Knowledge of the grammar of the language also helps to have an idea on how the system should behave under a wide range of conditions. For this reason NLP systems will have to be based on what linguists can determine about the structure the language [32] [33].

The syntactic analysis system to be developed involves a knowledge base to guide the parser. The main components of the system to be built include a lexicon, a list of grammatical rules, a morphological analyzer, a POS tagger. While the first two of these knowledge base components of the parser are discussed here, the last three are discussed in another section of this chapter.

2.4.1 The Lexicon

The term lexicon, which is a linguistic term for dictionary, is a collection of information about the words of a language and the lexical categories to which they belong. It usually serves as the starting point of any NLP system and must contain information about every potential word from which the system might come across, in order to guide processing.

A lexicon is usually structured as a collection of lexical entries, like ("pig" N V ADJ). "Pig" is familiar as a noun, but also occurs as a verb ("Jane pigged herself on pizza") and as an adjective as in "pig iron". In practice, a lexical entry will include additional information about the roles the word plays, such as feature information - for example, whether a verb is transitive, intransitive, bi-transitive, etc., what form the verb takes such as present participle, or past tense, etc. [12].

According to Allen [19], a grammar need not contain any lexical rules of the form, for example, N - flower, so long as a lexicon is specified. The following simple context free grammar lexicon for Afan Oromoo is shown by Abebe [2]

N - Nama
V - Deeme
Adj - guddaa
Adv - ariifatee

In this example the symbols on the left hand sides are POS categories for the words on the right hand sides.

2.4.2 The Grammar Rule

Grammar is a formal system for describing the rules and syntax allowable in the language. The most common way to represent grammars is as a set of grammar rules that capture generalizations to classify words into what are often called 'parts of speech' or grammatical categories. Grammar rules underlie many linguistic theories, which in turn provide the bases for many natural language understanding systems [7]. The grammar of Oromoo is compiled into a LR (Left to Right) table. The following is an example of simple grammar for the sentence '***Tolaan gara manabarumsaa demee,'*** "Tola went to school".

S => NP VP
VP => PP VP
NP => NAME
PP =>P N

VP => V

NAME => **Tolaan(n)**

P => gara

N => ***manabarumssa***

V => ***demee***

There are different ways of specifying grammars often called grammatical formalisms. The most commonly and widely used formalisms include: Context Free Grammar [34], Transitional Grammar (Chomsky [34]), Transition Network Grammars (Wood [35]), Unification Based Grammar (Kay, [36]) and Probabilistic Context Free Grammars [19]. Hence, the grammar rules will take on different forms depending on the theoretical framework of the specific grammar.

2.4.2.1 Context Free Grammars

Grammars consisting entirely of rules with a single symbol on the left-hand side are called *context-free grammar* (CFG). A CFG is a formal system that describes a language by specifying how any legal text can be derived from a distinguished symbol called *axiom*, or *sentence symbol*. CFG is required to be monotonic and the only way a CFG rule can be non-monotonic is by having an empty right-hand side. CFGs are very important for two main reasons: the formalism is powerful enough to describe most of the structure in natural languages, yet it is restricted enough so that efficient parsers can be built to analyze sentences [19].

This formalism consists of a set of *productions*, each of which states that a given symbol can be replaced by a given sequence of symbols. S ➔NP VP, for instance, is one of such productions stating that S can be replaced by the sequence of NP and VP. NP and VP, in turn, have sequences of symbols that replace each (for example, NP➔Adj N and VP➔V NP). Symbols that are to be replaced are called *non-terminals*, and are always represented by *identifiers*, which are sequences of letters and digits. Every non-terminal must appear before a colon in at least one production. The axiom is a non-terminal that appears before the colon in exactly one production, and does not appear between the colon and the period in any production. There must be exactly one non-terminal satisfying the conditions for the

axiom. Symbols that cannot be replaced are called *terminals*, and may be represented by either identifiers or *literals* (which are a sequence of characters bounded by apostrophes).

2.4.2.2 Transition Network Grammars

The Transition Network Grammars is another formalism that is useful in a wide range of applications. It is based on the notion of a transition network consisting of nodes and labeled arcs. Simple transition networks are often called finite state machines (FSMs) and are equivalent in expressive power to regular grammars, and thus are not powerful enough to describe all languages that can be described by a CFG [19]. The notion of recursion in the network grammar is needed in order to get the descriptive power of CFGs. Thus, the grammatical formalism that is based on such a notion is called Recursive Transition Network Grammars (RTNG). The RTNGs consist of nodes and labeled arcs where one of the nodes is specified as the initial state, or start state.

Woods [35] also introduced another widely used grammatical formalism called Augmented Transition Network (ATN). This formalism has become one of the most popular forms for writing natural language grammars [37]. A transition network is a representation of regular (or finite-state) grammar. The network is a directed graph whose arcs are labeled by terminal symbols (words or word categories). One node of the graph is designated as the start state; one or more nodes are marked as final states.

The assumption is that if there is a path from the start state to some final state such that the labels on the arcs of the path match the words of the sentence, a sentence is in the language defined by the network.

2.4.2.3 Context Sensitive Grammars

Still another commonly used grammatical formalism is the Context Sensitive Grammar, which is a phrase structure grammar with context-sensitive rules. There are two equivalent formulations of the definition of a context-sensitive grammar rule:

- Rules of the form x → y where x and y are strings of alphabet symbols, with the restriction that length (x) <= length of (y).
- Rules of the form A →y | x z where A is a non-terminal symbol, y is a sequence of one or more terminal and non-terminal symbols, and x and z are sequences of zero or more terminal and non-terminal symbols.

The meaning of the latter rule (or production) is that A can be rewritten as y if it appears in the context 'x z', i.e. immediately preceded by the symbols x and immediately followed by the symbols z [37]. Context-sensitive grammars are more powerful than CFGs though the former kinds of grammars are much harder to work with than the latter [12].

2.4.2.4 Unification-based Grammars

The grammar formalism that makes extensive use of feature structures (such as case, gender, tense) including their values represented in the lexical entries of words is called *unification-based grammars*. These feature structures are manipulated by the operation of unification (i.e. the entire grammar can be specified as a set of constraints between feature structures). CFGs or any of the grammar formalisms discussed above can serve as the backbones for the unification-based grammars, CFGs being the most common. Joshi (NY), as cited in Daniel [3] indicated, the main reason for the excessive power of the unification-based grammars is that recursion can be encoded in the feature structures.

2.4.2.5 Probabilistic Context Free Grammars (PCFG)

Just as Finite State Machines could be generalized to the probabilistic case, CFGs can also be generalized. This can be achieved through having some statistics on rule use. That is, simply by counting the number of times each rule is used in a corpus containing parsed sentences and by using this statistical information to estimate the probability of each rule being used. Given a category C and m grammar rules R_1,..., R_m with left-hand side C, the probability of using rule R_j to derive C can be estimated by

$Pr(R_j \mid C)$ = count(#times R_j used) / Sum i=1,m (#times R_i used)

Such a context free grammars with associated probabilities is called Probabilistic Context Free Grammars (PCFG) formalism. A typical PCFG grammar that is based on a parsed version of a given corpus, therefore, comprises a count for LHS, a count for rule, and probability figures for each of the rules generated. A probabilistic context-free grammar (PCFG) is, therefore, commonly defined as a four-tuple (W, N, S, R), where

$W = \{w^1, w^2, w^u\}$ is a set of terminal symbols like words in a sentence,

$N = \{N^1, N^2 \ldots N^v\}$ is a set of non-terminal symbols,

$S = \{N^1\}$ is a set that only has one starting symbol, and

$R = \{R^1, R^2 \ldots R^w\}$ is a set of grammar rules with probabilities. For a rule $R^m \in R$, it is a context-free grammar (CFG) rule with the form $R^m : N^i \rightarrow \xi^j$, and its probability $P(R^m) = P(N^i \rightarrow \xi^j)$. [38]

PCFG implementation techniques involve making certain independence assumption about rule use. In particular, it is assumed that the probability of a constituent being derived by a rule **Rj** is independent of how the constituent is used as a sub constituent. With this assumption, a formalism can be developed based on the probability that a constituent C generates a sequence of words $W_i, W_i+1, \ldots, W_j$, written as W_{ij}, where i and j refer to the position of the word in the sentence.

Although they do not take context or lexical co-occurrence into consideration, PCFGs are especially useful for sentence parsing as it allows handling such cases as structural ambiguity, ungrammatical sentence analysis, and grammar learning, even compared with CFGs. It gives a better probabilistic model for syntax analysis for statistical properties of natural languages are introduced [38]. In this study, the PCFGs grammars formalism is used in order to describe and represent Afan Oromo words into their allowable grammatical and syntactic categories. The remaining section of this chapter discusses related NLP systems that can function as essential component systems to develop automatic sentence parsing.

2.5 Related NLP in parsing

Win Win Thant, Tin Myat Htwe and Ni Lar Thein [14] described the use of Naive Bayes to address the task of assigning function tags and context free grammar (CFG) to parse Myanmar sentences. Part of the challenge of statistical function tagging for Myanmar sentences comes from the fact that Myanmar has free-phrase-order and a complex morphological system. They used function tagging to be a pre-processing step for parsing.

Mihai Lintean and Vasile Rus [29] described the use of two machine learning techniques, naïve Bayes and decision trees, to address the task of assigning function tags to nodes in a syntactic parse tree. They used a set of features inspired from Blaheta and Johnson [39]. The set of classes they used in their model corresponds to the set of functional tags in Penn Treebank.

Yong-uk Park and Hyuk-chul Kwon [4] tried to disambiguate for syntactic analysis system by many dependency rules and segmentation. Segmentation is made during parsing. If two adjacent morphemes have no syntactic relations, their syntactic analyzer makes new segment between these two morphemes, and find out all possible partial parse trees of that segmentation and combine them into complete parse trees.

2.6 Related NLP in Afan Oromo

As it has been discussed in chapter one, little was attempted in the area of NLP in Afan Oromo spatially in the case of Automatic sentence parsing. The only attempt to develop a simple automatic sentence parser for Affan Oromo language was conducted by Diriba [7]. On his study, Diriba [7], used the chart algorithm with some modification. A module for morphological analyzer, which splits words into root form and their corresponding morpheme, was also developed in order to facilitate the preparation of texts in a file to be parsed with appropriate lexical categories. In addition, the unsupervised learning algorithm was designed to guide the parser in predicting unknown and ambiguous words in a sentence. Grammar rules, lexicon, morphological rules and contextual information were also designed

on the basis of the review made on the linguistic properties of Afaan Oromo grammatical categories. This system, in fact, was the first in its kind for this language.

The study adopted an intelligent (Rule-Based+ learning module) approach to develop a prototype, which is a simple Oromo parser for the language. In short, it describes processes of automated sentence parsing of Free Texts. That is, it was aimed at developing a prototype and conducting an experiment with it. The result obtained was 95% on the training test and 88.5% on the test set.

2.7 Related NLP Component Systems

2.7.1 Morphological Analyzer

The first step in any NLP task involves recognizing and identifying individual word forms from the input text [31]. In some languages such as English, such information may be supplied by a lexicon that simply lists all word forms with their part-of-speech and inflectional information such as number and tense. Since such languages have a relatively simple inflectional system, the number of forms that must be listed in such a lexicon is manageable. But an exhaustive lexical listing is simply not feasible for many other highly inflectional languages such as Afan Oromo that have a number of inflected forms for each noun or verb. This is because each lexical word may have literally thousands of distinct surface forms differing in inflectional properties but otherwise the same vocabulary elements [40]. Therefore, NLP for these languages could be practical only if there is a morphological analyzer that will use the morphological information of the language to compute the parts-of-speech (POS) and inflectional categories of words [41] [39].

Hence, a morphological analyzer is a central element needed to parse words into their morphemic components, and also to give word classes (such as noun, verb, etc.) to which a particular word may fall before the task of parsing is carried out. It involves rules that are needed to 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. On top of this, a morphological analyzer would help in

assisting a parts-of-speech tagger (POST), which is a major component of a syntactic parsing system. A justification for the incorporation of a morphological analyzer into this study is provided in the next section where this second essential component for a sentence parser is discussed.

In this regard, a prototype morphological analyzer for Afan Oromo was developed by Abebe [2]. He used a Rule Based Approach for Afan Oromo Automatic Morphological Synthesizer to create morphological dictionary (known as signature) by extracting prefixes, stems and suffixes from a given corpus. Even though the attempt is remarkable it type for this study. The assumption in this study is that the results obtained from the prototype morphological analyzer for Afan Oromo developed by Abebe [2] would not play a significant role in preprocessing the input text to the parser together with the other NLP component system. Hence the manually processed words would be used in this study.

2.7.2 Part-Of-Speech Tagger

The other major and essential component of a sentence analyzing system is a POS tagger, an NLP system that automatically assigns the possible parts-of-speech categories to a given word in a sentence. Elimination of unlikely parses (wrong analyses of a sentence) is one of the principal motivations of incorporating POST into a given automatic sentence parser since a POST involves identifying the syntactic categories of words in a text. That is, a given sentence, for example, '**Gaangeen tolaan kalessa bitee hara ganama du'e'** 'The mule that Tolla bought yesterday died this morning' would become unambiguous if we can get the correct POS tag assignment as follows:

'Gaangeen\N Tolaan\N kalessa\Adj bitee\ V hara\Adv ganama\Adv dutee\V'

Also, a single morphological form may very likely occur with various syntactic categories and some mechanism is necessary to identify the correct syntactic category of the word in the given position in a sentence.

2.8 Conclusion

In summary, this chapter defined automatic sentence parsing as 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. The major approaches and strategies that were proposed at various times were also reviewed. The knowledge base of a sentence parser and the necessary components such as the lexicon, the grammar rules and the grammatical formalisms that are needed to store information that helps during the parsing process were also described in this chapter. Finally, NLP systems that could be essential components of a sentence parser were also discussed.

CHAPTER THREE

THE STRUCTURE OF AFAN OROMO

3.1 Introduction

This chapter discusses the structure of Afan Oromo word classes, phrases and sentences since each of these units has its own impact on the present study. But before exploring into the grammatical categories of the language, the paper begins with the general and brief discussion of the lexical categories of the language. The lexical categories are what the traditional grammar calls Parts of Speech. But since grammatical categories are more inclusive the paper inclines to using the word. And to differentiate between words and phrases, we use lexical categories to refer to individual words and non-lexical or phrasal categories to refer to types of phrases.

The lexical categories discussed in this chapter are nouns, verbs, adjectives, adverbs, adpositions and conjunctions. Pronouns are also considered separately but they fall under noun category. Other Afan Oromo words such as interjections and numerals are also topics discussed in this chapter. For better understanding of this section, the chapter begins with a brief review of the writing system and punctuation marks in Afan Oromo.

The analyses and discussions made in this chapter are based on the data extracted from Diriba [7], Abebe [42], Baye [43] [44] [45], Askale [46] and Tilahun [47], and Girma [10]. Further details on the subject can be obtained from these sources.

3.2 Afan Oromo Alphabet and Writing System

Afan Oromo writing system is a modification to Latin writing system. Thus, the language shares a lot of features with English writing except some modification. Fortunately the study gets advantage of the Afan Oromo writing alphabets called commonly “qubee Afan Oromo” that has been designed and used so far by the language experts in the area. The writing system of the language is straightforward which is designed based on the Latin script. Thus letters in

English language are also in Afan Oromo except the way it is written. A detailed description of Afan Oromo writing system can be found in any text related to the language but readers can be referred to Diriba [7] and Girma [10] for the detailed discussion of the language's writing system.

3.3 Punctuation Marks in Afan Oromo

Analysis of Afan Oromo texts reveals that different Afan Oromo punctuation marks follow the same punctuation pattern used in English and other languages that follow Latin writing system. The full stop (.) in statement, the question mark (?) in interrogative and the exclamation mark (!) in command and exclamatory sentences marks the end of a sentence and the comma (,) which separates listing in a sentence and the semi colon (;). Listing of ideas, concepts, names, items, etc are separated by the commas.

3.4 Word Categories in Afan Oromo

As in the case with the grammatical categories of other languages, like English for example, the grammatical categories of Afan Oromo have undergone a series of improvement in terms of its word categories and other syntactic features. Thus, the eight traditional grammars is now summarized and divided into five categories in the language. In traditional word categories, Afan Oromo words are categorized into the following eight categories (or Grammatical Categories). These are the Noun, Verb, Adjective, Adverb, Adposition, Pronoun, Conjunction and Interjection.

Modern syntacticians such as Baye [43] [44] [45] divide Afan Oromo words into five putting pronouns and adjectives under the noun, conjunctions under adposition (pre-and postposition) categories, respectively plus adverbs, adjectives and verbs. Others like Askale [46] put adjectives and adverbs under the same lexical categories. Whatever the case, there are five syntactic categories serving as heads in phrase construction. Thus based on the above classification, the language has five grammatical categories headed by five word categories. In this categorization interjections, which are "words" without syntactic functions, are not considered as Grammatical Categories.

The current study adopts the classification scheme developed by Baye [43] [44] [45]. This is because traditional classification scheme is repetitive and adds redundancy to the parser to be developed by this paper. Rather a sub categorization scheme is used so that the grammar rule is compact and more expressive. The following sections of this chapter explores into the discussion of the grammatical categories of Afan Oromo as a background to the tasks to be carried out in chapter four and five, which are the central and major contribution of this thesis in the area.

3.4.1 Categories of Nouns

The definition of nouns in Afan Oromo is similar to the definition in other language like Amharic. Afan Oromo nouns are words used to name or identify any of categories of things, people, places or ideas or a particular one of these entities. For this study, the Afan Oromo noun categories consist of nouns, adjectives and pronouns. Positions occupied by words like '***Fardaa'*** "Horse" are considered as the nouns positions in the following sentence. '***Fardi marga dheeda'*** (the Horse grazed grass). Moreover, two numbers are recognized in Afan Oromo nouns: singular and plural. A singular noun is marked by zero morpheme whereas a plural noun is marked by various forms. The following are illustrative examples.

	Singular		**Plural**		**plural marker**
1.	nama	man	namoota	men	{-(o)ota}
2.	sangaa	ox	saangoota	oxen (pl)	{-(o)ota}
3.	godina	zone	godinalee	zones	{-lee}
4.	bineensa	hyena	bineeyyii	hyenas	{-yyii}
5.	jabbii	caf	jabbiilee	cafs	{-lee}

Nouns sometimes used as adjectives, like in the following sentences:

Tulluun mana *citaa* ijaare. [Tullu built a thatched house]

Adjectives are categorized under different categories. Like nouns Afan Oromo adjectives can be either primitive or derived. Adjectives come after the nouns they describe. For example,

the adjective '***guraacha'*** "black" in a phrase '***holaa guraacha'*** comes after the noun it modified. It is also true in the adjective case that not all words after nouns could be adjectives. For example, in ***'mana barumssa'*** "school", the word *barumssa* "education" is not adjective but noun as adjective. Moreover, nouns but not adjectives occur in subject and/or object positions. On the other hand, like nouns, Afan Oromo adjectives can be either primitive or derived.

As can be observed from the data above, Afan Oromo nouns are pluralized by suffixing various forms of suffixes, such as, **{-een}, {-wan}, {-(o)ota}, {-yyii} and {-lee}**. It is possible to use more than one type of different plural markers in some nouns. For instance, '***jabbii'*** "caf" can be pluralized both as ***jabbilee*** or ***jabbiloota***. Most nouns, however, prefer one plural marker to the other Abebe [42]. The word '***sagalee'*** "sound", for example, can be pluralized by suffixing {-(o)ota} or {-lee}, but it prefers the former form to the latter. When a noun stem and a plural marking suffix combine, various morphological processes take place among which the last vowel of the stem drops and be compensated by germination of the preceding consonant as in (1- 5) above.

Adjectives are similar to nouns in various forms. For example, Baye [43] concluded that both nouns and adjectives are inflected for number. He further discussed that both may be pluralized by affixing the above indicated plural maker for nouns, especially ***{-(o)ota}*** excluding those derived adjectives which are pluralized by reduplication. For example, the noun ***'nama'*** "man" in Afan Oromo is pluralized by adding the suffix {-ota} and becomes ***'namoota'*** "man" but by deleting the final vowel. The same is true for adjectives. For example, the adjective ***'guraacha'*** "black" can be pluralized in a similar fashion like nouns and becomes '***guraachota'*** "the black".

The two sub categories also share similar characteristics for the inflection of gender. However, it should be noted that adjectives cannot substitute nouns in a sentence construction. Only pronouns seem to substitute for each other since they can occur in the same position in a sentence of Afan Oromo. For example, consider the following sentences in which one is correct and the other is not.

Tollaan barsiisaadha. "Tolla is a teacher"

Inni barsiisaadha. "He is a teacher".

Guraacha barsiisaadha. "Black is teacher", which is ungrammatical.

There are also personal pronouns which are included under this category. See the following table.

Person	***Accusative***	***Nominative***
1sg	(a)na (me)	an-i (I)
1pl	Nu (Us)	nu-hi/nu-ti (we)
2sg	Si'I (you)	Ati (you)
2pl	Isin (you)	isin-φ (you)
3sgm	Isa (him)	in-ni (he)
3sgf	Ishii (her)	ishii-n (she)
3pl	Isaan (they)	isaa-φ (they)

Table 3.1: personal pronouns

3.4.2 Categories of Verbs

The discussion of this section is based on the information collected from Baye [43] [44] [45] and Askale [46] and Abebe [42]. These works consist of all the information required by the current study. Verbs are forms which occur in clause final positions and belong to a distinct category from that of nouns. For example in the following sentence,

Caalaan farda *bite*. "Chala bought a horse"

Leensaan *dhufte*. "Lensa has come"

Tulluun *dheeraadha*. "Tullu is tall"

The italicized part are all verbs. Baye [45] divides verbs into a number of sub categories based on the type of constituents they are associated with. These are intransitive, transitive, modals and auxiliaries verbs. The intransitive verbs are those verbs which do not take any phrase as their complement. For example in the sentence '***Abbabaan furdate'*** (Abebe got fat), '***furdate'***

"got fat" is an intransitive verb which has no complement. There are also what Sag and Wasow [48] call strictly transitive verb. These types of verbs are those which take one complement in Afan Oromo. For example,

inni [teechuma] $_{NP}$ ***cabse*** "he broke the chair"
Caalaan [mana]$_{NP}$ ***bite*** "Chala bought a house"

The NP in these two examples are complement to the verbs ***'cabse'*** broke and '***bite'*** bought. For the detailed treatment of these subcategorizations see Baye [45].

3.4.3 Categories of Adverbs

Afan Oromo adverbs are words which are used to modify verbs. Adverbs usually precede the verbs they modify or describe. Example;

Tolaan *kaleessa* dhufe. "Tola came yesterday"

In this example, the adverb '***kaleessa'*** "yesterday" precedes the verb '**dufe'** "came" that it modifies. However, it should be noted that any word that comes before a verb is not necessarily an adverbs. For instance, in ***'muka cabse'*** "broke wood", the word '**muka'** "wood" precedes the verb '**cabse'** "broke". In this case the word '**muka'** is a noun and in turn is modified by the verb '**cabse'**. Hence, the verb functionally shares the feature of an adjective (modifier). There are different types of adverbs. These are adverbs of time, place, manner, frequency, degree, etc. in general, adverbs are treated as the subclass of verbs. Days of the week in Afan Oromo language may be used also either as a noun or as an adverb. Readers again referred to Baye [45] for further discussion and examples.

3.4.4 Adpositions in Afan Oromo

The term Adpositions refers to words, which will have meaning only when they are attached or used together with other words such as nouns, verbs, pronouns and adjectives. Adpositions are characterized by having no inflectional or derivational morphology and belong to the closed system.

Adpositions can appear as:

A simple adpositions that stand alone as separate words

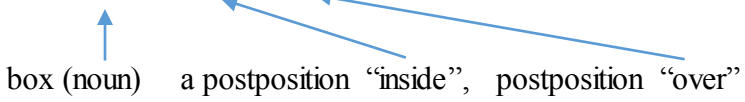
Examples **Toleraa *walin*** "with Tolera "
***Gara* mana** "to house"

A simple adposition prefixed or attached with other words (e.g. nouns and verbs).

Example **harka-*an*** "by hand"
Ummata-*f* " to/for the public"

As compound adpositions consisting of two parts, adpositional prefixes and post positions put after nouns. The postpositions can either be single adposition that stand by their own or an adposition not separated from a noun.

Examples **sanduqa *gubba-rra*** "On top side of the box"



3.4.5 Conjunctions in Affan Oromoo

Conjunctions in Afan Oromo are coordinating or subordinating. They coordinate words, phrases, clause and sentences. A list of Afan Oromo coordinating conjunctions is found in Hamiid [49] together with a detailed discussion on such coordinating and subordinating conjunctions.

This paper adopts the current trend that conjunctions and adpositions appear in the same categories, the adposition Grammatical Categories. One problem that arises by categorizing adpositions and conjunctions into different categories is the problem pertaining to distinguish conjunctions from adpositions. The problem in distinguishing the two mainly arises from the fact that the same words are mostly used as both adpositions and conjunctions. However, in cases where it is possible to separate adpositions from conjunctions, they are parsed separately. That is when the parser is able to distinguish between the two sub categories a distinct category is given to both of them.

3.4.6 Numerals

These are words representing numbers. They can be cardinal or ordinal numbers. A list of the Afan Oromo Cardinal numbers is found in Hamiid [49]. In Afan Oromo, the ordinal numbers are formed from the cardinal numbers by suffixing the suffix {*–ffaa}***.**

Example	Cardinal	gloss	ordinal	gloss
	tokko	one	tokko*offaa*	first
	sadi	three	sada*ff*aa	Third

Like English, compound Afan Oromo numerals are put separately. The following are examples to illustrate this.

Example D*hiba lamma*, "two hundred"
Dhiba lammaffaa "two hundredths"
*Dhiba lammaa-fi shan "*two hundred and five"

In Afan Oromo, there are also numerals that indicate distribution. These numerals are called distributive numerals.

Example ***'sadi sadi'*** "three three "

There are also special numerals in Afan Oromo that correspond to the English "half", "quarter" etc.

Examples of these include '***walakkaa'*** "half" and '***siisoo'*** "one third".

3.4.7 Interjections

Like English, Afan Oromo has many words or phrases used to express such emotions as sudden surprise, pleasure, annoyance and so on. Such Afan Oromo words are called interjections. These Afan Oromo interjections can stand-alone by themselves outside a sentence or can appear anywhere in a sentence.

Examples	***ashuu!***	" wonderful!"
	wayyoo	" my goodness"
	ani bade!	**"**my goodness"

A long list of Afan Oromo interjections is found in Hamiid [49].

Based on the above lexical categories, the next section explores the types of phrases found in Affan Oromoo. The idea of headedness discussed in this chapter of this paper may indicate that the types of phrases found in the language depend on the lexical categories of the language. Moreover, Baye [45] and Sag and Wasow [48] divide the types of phrases based on the lexical categories. Thus this paper will depend on this classification for the purpose of the problem under consideration but keeping the idea of headedness in mind. Moreover, this paper depends entirely on Baye [45] and Sag and Wasow [48] for the analysis of Afan Oromo phrasal categories.

3.5 Phrasal Categories

As it is indicated above phrasal categories depend on the lexical categories of a language. They use the lexical categories as the head of their phrases. Thus all phrasal categories are hierarchical in nature. This hierarchical nature of categorization is very important for it enable us to classify feature structures in more subtle way that will allow intermediate level categories of various sorts. For example, verbs may be classified as intransitive or transitive; and transitive verbs may further be sub classified as strict transitive (those taking a direct object and nothing else) or ditransitive. Thus the hierarchical system let us talk about the properties shared by two distinct types by associating a feature or a constant with their common super type. But before talking about the phrases in Afan Oromo, let's define the word phrase.

3.5.1 A Phrase

A phrase can be defined as a syntactic combination of a word with one or more other words. A phrase is constrained or restricted by two things: in terms of the constituents' and the lexical categories like nouns, verbs, etc. Thus, we can determine the number of phrases by

the number of words. A question of how to check whether a structure is a phrase can be answered using the following four guiding principles Baye [45]. These are:

1. If the constituents of the phrase can be moved together to another place without separation.
2. If the phrase can replaced by a pronoun (for noun phrase).
3. If one of the constituents of that phrase is missed, the meaning of that phrase will be corrupted.
4. If an insertion of other word in between that phrase affects the meaning.

Based on the type of lexical categories in Afan Oromo, there are five phrase types in the language. They will be reviewed in the following subsections.

3.5.2 Noun phrases

A noun phrase is made of one noun and one or more other lexical categories including the noun itself. For example, in the phrase '**mana citaa** "thatched house", there are two nouns which make the noun phrase: '**mana'** "house" and '**citaa'** "thatched".

Thus, noun phrase and phrases in general must meet the above criteria to be called a phrase. In the following sentence '**Tolaan mana citaa qaba'** "Tola has owned a thatched house", '***mana citaa'*** "thatched house" is a noun phrase. But to check whether it is really a phrase or not, we can see the above criteria. The following arrangement is impossible for the above reasons.

A. **Qaba Tolaan mana citaa**. "legal movement"
B. * **citaa Tolaan mana qaba**. (Illegal movement because of the above reason 1 and 4.)
C. *** Mana Tolaan citaa qaba.** (illegal because of rule 1&4)

The above sentences with asterisks have illegal phrase construction because of the above rules. Thus we checked that "**mana citaa"** is a phrasal structure.

As indicated above, nouns can appear in a number of positions, such as in the positions of the three nouns in '**Tolaan kitaaba Haawwiif bite'** "Tola bought Hawi a book". These same positions allow sequences of a noun followed by an article, as in '**Tolaan kitaabicha Haawwiif kenne'** "Tola gave Hawi the book.". Since the position of the article can also be filled by demonstratives ('***kun'***, '**sun'**, etc.), possessives **('koo', 'kee', 'keessan',** etc), or quantifiers (e.g. '**xiqqoo'**), the more general term "Determiner" abbreviated as (DET) is used.

Moreover, each constituent in a phrase has its own positions and functions. For example, '**mana'** and '**citaa**' are both constituents of the phrase **'mana citaa'**. A phrase is usually headed by one word. The head word is the core component of a phrase. Without a head a phrase can't be built. On the other hand a head can stand-alone by itself. A head word can determine not only phrase type but also lexical categories. If the head is a noun, then the phrase is a noun phrase, etc Sag and Wasow [48]; Levine and Green [50].

NP has a lot of constituents in Afan Oromo. As indicated above one of the constituents is the determiner. Consider the following example,

A) **[Namni tokko]**$_{NP}$ **[saree ajjeese]**$_{VP}$ " A man killed a dog"

B) **[Namichi]$_{NP}$ [saree ajjeese]$_{VP}$** "the man killed a dog"

We can see that NP consists of determiners of type articles '**tokko'** "a" in (A) and '**–ichi'** "the" in (B). However, the position of these determiners in Afan Oromo is different from English in that determiners come after the nouns they modify. Not only determiners but also all modifiers for nouns come after it in the language. A NP may also consist of two nouns like **'mana citaa'** "thatched house". In Afan Oromo the order of words especially the head word and the modifiers and specifies are different from the word in English. For example, a NP in Afan Oromo consists of one noun word as head word and another noun plus an adjective as modifiers and specifies like in the following example.

'Mana citaa bareedaa' 'a beautiful thatched house'.

In addition to the above developments Afan Oromo has a NP which may appear as accusative, nominative, genitive, dative and instrumental. This type of existence of nouns

in a different form for different function is called case. In the following subsection it will be reviewed in brief. However, for detailed treatment of case in Afan Oromo, readers are referred to Abebe [42].

3.5.2.1 Accusative and Nominative Case

The accusative case form is the basic form of nouns and pronouns in Afan Oromo. Abebe [46]; Baye [43]; Gragg [51]. This means that nouns and pronouns in direct object position do not have overt case marker, as shown in the following sentences. While the nominative case (words in their subject position form) are inflected for agreement in terms of case.

A. [Tulluu - n] $_{NP}$ **[mana]**$_{NP}$ ijaar-e
" Tullu built (a) house."

B. Tulluu-n **[*farda adii*]** $_{(NP)}$ yaabbat-e
" Tulluu rode (a) white horse."

C. Tulluu-n **[*intala*-tii]**$_{NP}$ beellam –e
" Tulluu dated the girl."

D. Tulluu- n **[*intala –tii diimttuu*]** $_{(NP)}$ beellam- e
" Tulluu dated the white girl."

E. ***nam- ni*** of jaalat -a
"Man loves himself"

F. ***fard -i*** marga dheed-e
"A horse grazed grass."

In the above example, the phrase **'Tulluu-n**' is an NP as nominative case (subject case) and **'mana'** [house] is an NP as accusative case. Thus in the above example one can see that NP as subject has case marker, i.e. **'-n', 'ni', and 'i'** while NP as object form has no case marker in a sentence.

The object NPs '***mana'*** house' (head noun) in (a), '***farda adii***' "white horse" (a head noun and modifying adjective) in (b), '***intala-it'*** "the girl" and '***mucaa'*** " "the boy" (head noun in (c) & (d) respectively, and ***intala-ittii diim- ttuu*** "the white girl"(a head noun along

with singulative marker and modifying adjective) in (e) are not all overtly marked for accusative case.

Similarly, personal pronouns '***ana***' "me", '***nuu'*** "us", '***sii'*** 'you' (second person singular), '***isin'*** "you" (2pl), '***isa'*** "him", '***ishii'*** "he", '***isaan"*** "them" in object position do not affix accusative case marker.

It can be noted from (a-f) in the above examples that nominative case in Afan Oromo nouns is marked by '-ni', '-i', '-n' and 'φ' (empty set). Abebe [46] generalized these subject markers as in the following case.

I. '-ni' occurs after a noun which ends with a short vowel that is dropped, (e.g. in E above),
II. –i' occurs after a noun that ends with a short vowel which is preceded by consonant cluster; the short vowel of the stem is again deleted (e.g. F),
III. '-n' occurs after a noun that ends with a long vowel(e.g. A—D), and
IV. 'φ' occurs after a noun that ends with a consonant (e.g. G).

An adjective modifying a head noun in external argument position attaches similar suffixes as the nouns in the above examples for subject markers.

A. ***nam-ni furdaa-n*** dhibee hin-danda'-u "A fat man cannot resist disease"
B. [***fard-i gurraach-I***] NP collee -dha " Black horse is smart"

As can be observed in (A and B), subject marker in the nominative case is copied onto the modifying adjectives. The forms of the nominative marking suffixes on the adjectives are phonologically conditioned in the same way as they are made on nouns. Detailed treatment of case in Afan Oromo is found in Abebe [46]. The same is true for personal pronouns in Afan Oromo.

There are personal pronouns such as '***nu'*** "us" and '***sii'*** "you" which do not seem to fit into the rules in above. '***sii'*** "you" (accusative or object case)' and '***ati*** ' "you"

(Nominative) are different forms from one another and may be considered suppletive. On the other hand, '***nu'*** "us" and '***nuhi/nuti"*** "we" share some common phonetic form that have been summarized from above in the above rules. In general NP constituents are:

1. ***A noun as head word***
2. ***Specifiers like adjectives, adposition, etc***
3. ***Quantifiers like numbers***

Furthermore, Afan Oromo simple noun phrase is head final. A more detailed discussion of noun phrase will be presented in the sub topic "Sentence in Afan Oromo". The last point to make about Afan Oromo Sentence is that it has discontinuous morpheme to indicate negative markers. For example, Abbabaan ***hin***ddhuf***n***e

3.5.3 Verb Phrases

Before the discussion of the verb phrases (VP), it is necessary to introduce the word complement here. In a simple language, a complement is a word or a phrase that the head word can take as its constituents to make it grammatical. Some words do not take complements like '**Abbabaan dhufe'** "Abebe came"; some take one complement like **'Abbabaan muka cabse'**; and still others take two complements **'Abbabaan konkoolataa naaf bite'**. Based on this definition, Afan Oromo verb phrases can be captured by dividing them into three categories. These are intransitive, strictly transitive and ditransitive verbs.

'Abbabaan dhufe' "Abebe came".
'Abbabaan teechuma cabse' "Abebe broke chair"
'Abbabaan konkoolataa naaf bite' "Abebe bought me a car".

The structures which appear as constituents of VP are all types of adverbs, adpositional phrase and noun phrase. A more detailed description will be presented in the sub topic of "Sentences in Affan Oromoo".

3.5.4 Adjective Phrase

Adjectives are specifiers for noun phrase. They usually come after the noun (usually the head word) they specify. For example, '**mana guddaa' "**big house". In adjectives nouns can act as adjectives like **'mana citaa'** "thatched house" or verbs as adjectives like **'mana gubate'** "burnt house". Again a further detail is in the next subtopic "Sentences in Afan Oromo".

3.5.5 Adverb Phrase

Adverb phrase is made up one adverb as head word and one or more other lexical categories including adverbs itself as modifiers and specifiers in Afan Oromo. For example, in Afan Oromo it is possible to have two adverbs in adverb phrase like in a phrase '**kaleessa galgala**' "yesterday night". As indicated above adverbs and their phrases are used to modify verbs. Hence they precede verbs in a sentence. In general adverb phrase can have one adverb as head word and noun phrase, another adverb, etc as constituent. See the detailed discussion in Baye [45].

3.5.6 Adpositional Phrase

Adpositional phrases are combination of nouns and adposition. They usually specify verb phrase. This phrasal category sometimes is called adpositional objects (Baye 1986).

A. 'Inni ***gara*** mana deeme' "He went to the house"
B. 'Lammaan kophee Caaltuu-***f*** bite'. "Lemma bought a pen to Chaltu".

Adpositions in an adpositional phrase can be either stand independently like in (A) or affixed to the adpositional object like in (B) above.

3.6 Sentences

3.6.1 Afan Oromo Simple Sentences

A simple Afan Oromo sentence consists of a noun phrase NP, which is the subject, followed by a verb phrase VP that comprises the predicate.

'***Namichi saree jaalata'*** "The man loves dog."

Baye [45] classifies simple sentences into four, namely: declarative sentences, interrogative sentences, negative sentences and imperative sentences. Declarative sentences are used to convey ideas and feelings that the speaker has about things, happenings, feelings, etc, that could be physical, mental, real or imaginary.

Example: '**Haawwin abokatoo taate**.' "Hawi became a lawyer"

A sentence that questions about the subject, the complement, or the action the verb specifies, is called an interrogative sentence.

Example**: 'Haawwin yoom dhuftee?'** "When did Hawi come?"

In order to construct interrogative sentences, Afan Oromo sentences usually involve such interrogative pronouns as '**eenyu'** "who", '**maal'** "what", '**essa'** "where", '**meeqa'** "how many/how much", and '**yoom'** "when". These interrogatives can then be combined with prepositions to produce some more interrogative prepositional phrases like '**eenyu irra'** "from whom"**, 'maalif'** "why", etc.

Negative sentences simply negate a declarative statement made about something.

Example: '**Tolaan laqana nyaatee**.' "Tola ate his lunch"

'Tolaan laqana hin-nyaanee .' "Tola did not eat his lunch"

In the above example, the verb in both sentences is '**nyaatee'** "ate". It is negated by the negation prefix /hin-/ and the suffix /-nee/ indicating that the subject of the sentence is masculine, third person singular. Simple imperative sentences convey instructions and mostly their subject is a second person pronoun that is usually but implied by the suffix on the verb.

The combination of zero or more Noun Phrase and one or more verb phrases make up a sentence in Afan Oromo. However, a sentence is considered as a special kind of phrase which consists of noun phrase (NP) and the verb phrase (VP). Thus we can talk in terms of phrases when talking about sentences.

Before proceeding, we need to reflect for a moment on the traditional terminology ***parts of speech (POS)***. There are certain feature structure that are appropriate for certain POS (lexical categories), but not for others. For example, CASE is appropriate only for nouns, adjectives and pronouns in Afan Oromo. While the feature AUX(ILLARY) is specifiable only for verbs (to distinguish helping verbs from all others). Like wise, the features PER(SON) and NUM(BER) are used for nouns, verbs, and determiners. Thus, we have to guarantee in the parser to be developed that the right feature go with the right lexical categories. Moreover, it should be noted that the lexical categories discussed in this chapter are important, since they serve as the head of a phrase in a sentence of Afan Oromo.

As discussed so far the head will always tell as its value a lexical category in Afan Oromo. Head does the same job assigned to the POS; but it also does more here, namely it provides us a way to account which features are appropriate for which POS. Moreover, as Sag and Wasow [48] say, HEAD enable us to introduce complex features: features within features. Furthermore, it will be of immediate use in providing us with a simple way to express the relation between a headed phrase and its head daughter. To explain these terms, let's describe a tree structure which is common to almost every discipline. Any sentence can be represented in upside down tree diagram. For example, the sentence '**Namichi saree jaalata'** "The man loves dog".

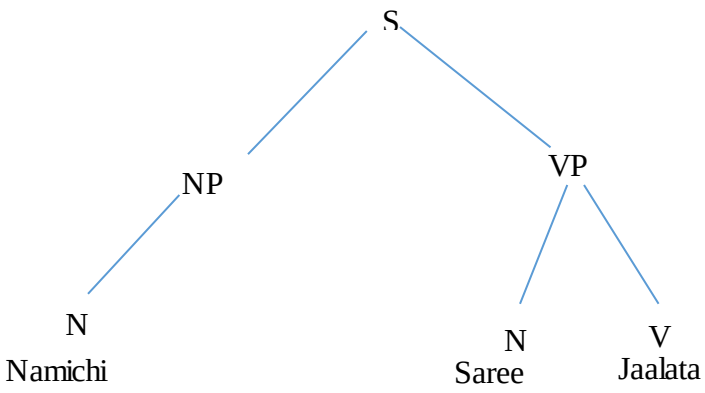


Figure 3.1: Sample Structure of simple Sentence.

A tree is said to consist of nodes connected by branches. A node (for e.g. S in above example) above another one (for e.g. NP or VP) in a branch is said to dominate it. The nodes at the bottom of the tree, that is, those that do not dominate anything else, are called terminal (or leave) nodes. A node right above another node on a tree is said to be its mother node and to immediately dominate it. A node right below another one is said to be its daughter. Two daughters of the same level are sisters to one another.

Keeping the above simple definition of daughter in mind, now let's come back to the idea of ***headed phrase*** and ***head daughter***. Grammar rules in Afan Oromo (and in any other language for that matter) require that the mother and one of the daughters bear identical (unified) values both for POS and features. The constituent on the right hand side of the rule that carries the unifying (matching) feature value is always the head daughter.

The general principle governing all trees built by headed rules as stated by Sag and Wasow [48] say that in any headed phrase, the head value of the mother and the head value of the daughter must be unified (or have identical value). Moreover, they state that phrase structure rules are not different in kind from word structures, except they are governed by grammar rules rather than lexical entries. Thus, we can say a sentence (or a phrase) is grammatically correct (or well formed) based on grammar rules. For example, Sag and Wasow [48] say that a sentence is well formed just in case each local sub tree within it either

I. Contains a lexical entries, or

II. Satisfies some grammar rules and principles.

The schematic representation of a sentence as forwarded by Sag and Wasow [48] is:

S ⟶ PHRASE PHRASE

HEAD NOUN HEAD VERB

AGR AGR

The structure says that a sentence consists of a noun phrase and verb phrase, noun and verb as head, unified by their agreement in a sentence, respectively see Sag and Wasow [48]. It should be noted that this is a general representation of sentence of all types. But the purpose of this paper is to develop a parser for simple statement which describes physical or ideal; realistic or abstract ideas, actions and emotions. Such kind of sentences in Afan Oromo ends with a period. Moreover, the paper will also include the feature of type agreement for all grammatical categories and past tense for verbs in developing parser again for the same reason.

3.6.2 Afan Oromo Complex Sentences

Complex sentences in Afan Oromo are those sentences that are composed of complex phrases such as main clause (MC), subordinate clause (SC). Intern, each MC and SC be composed of NP, VP, or AdjP. The pattern of combination could take the form of a complex NP and a simple VP, a simple NP and a complex VP, or both complex NP and VP. It is worth doing to see the structure of complex phrases before analyzing how they combine with each other to construct complex sentences.

A complex MC is one that contains an embedded sentence with in it. The phrase, for instance, '**Huccuun adii Tolaan kan bitee'** "The white cloth that Tola bought" is a complex MC/NP whose head is '**huccuu'** "cloth". This head has been combined with the complement '**adii'** "white" in order to produce the simple NP '**huccuu adii' "**a white cloth". This simple NP, in

turn, has combined with the dependent clause/subordinate clause '**Tolaan kan bitee'** "that Tola bought" to produce the above complex NP. The presence of the relativizer, '**kan'** "that" in it indicates that the clause is a subordinate clause and it cannot stand alone. The structure of this complex NP can be depicted by a parse structure tree as follows.

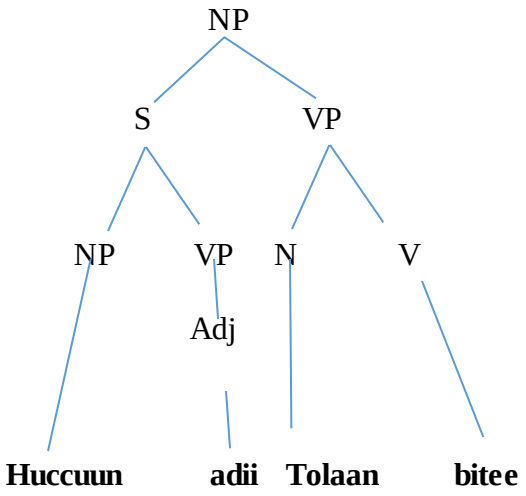


Figure 3.2: The Structure of a Complex Noun Phrase

One can understand from this tree diagram that the relative clause '**Tolaan kan bitee'** "that Tola bought" modifies the NP '**Huccuun adii'** "white cloth". The object of the verb '**kan' 'bitee'** "that (he)" is 3rd person singular masculine and its person level is exactly the same as that of '**Huccuun adii'**. Hence, '**Tolaan kan bitee'** "that Tola bought" is referred to as a relative clause Adugna [52].

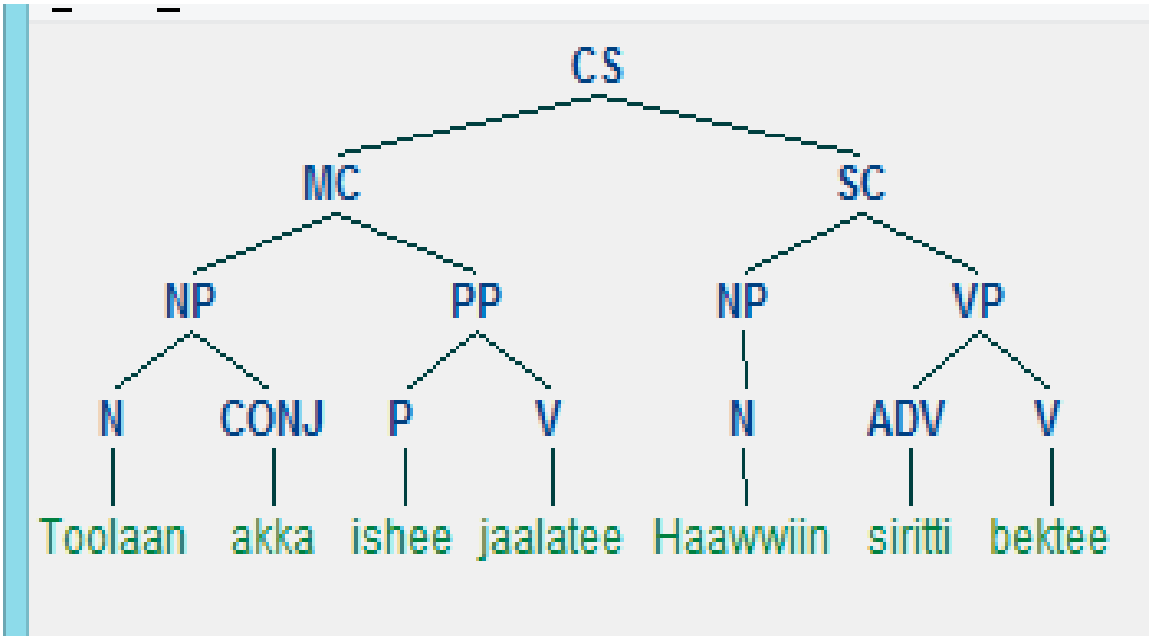
Likewise, a sentence is complex if it contains more than one verb or a sentence within it. That is, like a complex NP/MC, a complex VP/SC also contains an embedded sentence that plays the role of a compliment or a modifier.

'Toolaan akka ishee jaalatee Haawiin siritti bektee'.

"Hawi knew that Tola loves her"

The dependent clause here is **Toolaan akka ishee jaalatee** "As Tola loves her…" **akka ishee jaalatee** is the part that made the clause dependent. As this clause indicates the reason

for ‘Hawi’s knowledge of being loved, it is used as an adverbial clause of reason. The structure of this VP can be depicted by a tree diagram as follows:



(CS
(MC
(NP (N Toolaan))
(PP (P akka))
(VP (PR ishee) (V jaalatee)))
(SC (NP (N Haawwiin)) (VP (ADV siritti) (V bektee)))) [0.00041472000000000004]

Figure 3.3: The Structure of a Complex Verb Phrase

To sum up, simple sentences are composed of simple noun phrase and simple verb phrases while complex sentences can consist of a complex NP and a simple VP, a simple NP and a complex VP, or a complex NP and a complex VP. Only complex sentence that are composed of a simple NP and a complex VP are considered in this study due to time and resource constraints.

CHAPTER FOUR

DATA PREPARATION AND PCFG EXTRACTION

4.1 Introduction

This chapter may be considered as the core of this study. Based on the assumptions and approaches discussed in chapter two and the syntactic property of Afan Oromo reviewed in chapter three, this chapter discusses the PCFG that has been employed by the automatic complex sentence parser for Afan Oromo developed in this study.

The chapter begins by discussing the approaches taken to design the parser, and then moves to the second section where the sample text used in the study is analyzed. The third and fourth section discusses a simple manually annotated morphological analyzer and part of speech tagger respectively. The fifth section deals with the extraction of a probabilistic context free grammar rule from the tagged corpus. Finally, under the sixth section the conversion of the rules to the Chomsky Normal Form (CNF) is presented.

4.2 The Design Approach of the Parser

The use of statistical methods has greatly furthered progress in many areas of language processing. Disambiguation, database classification of documents, speech recognition and grammar learning are just some of the areas that statistics has helped. Statistics is the ideal tool for analyzing certain regularities of a language frequently occurring phenomena Chomsky [34].

Currently, few or no efforts are being put to make large Afan Oromo corpora electronically available, and this will not give researchers in the area of NLP the opportunity to extract the

statistical information that can help them understand language. That is why this study relay on manually annotated and tagged sentences.

As mentioned in the second chapter, the approach considered in the design of Afan Oromo complex sentence parsing system is the PCFG bottom up chart parsing. As it was mentioned in chapter two, CFGs are powerful enough to describe most of the structure in natural languages, and they are important in that they are restricted enough so that efficient parsers can be built to analyze sentences Allen [19]. PCFGs, which define a language as a probability over strings, are the probabilistic version of CFGs and they have been used in a variety of applications [40].

Compared to the CFGs, PCFGs are appear to be more useful for sentence parsing as they allow handling frequent parsing problems like structural ambiguity especially as grammar gets large, predicting the parse space, and ungrammatical sentence analysis.

Assuming that a sentence $w_{1,n}$ (where $w_{1,n}$ is a sequence of words $w_{11}, \ldots, w_{1n}$) has $T(n)$ possible parse trees (possible structures), and t_i is the i^{th} parse tree that $0 < i \leq T(n)$, then the probability of the i^{th} parse tree of the sentence $w_{1,n}$ is the product of the probabilities of all rules used in the parsing and is given by Yao and Lua, [53]

$$P(t_i) = \prod_{j=1}^{n} P(R)^j \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots Equation\ 1$$

And the probability of the sentence $w_{1,n}$ is the sum of the probabilities of all possible parse trees that

$$P(w_{1,n}) = \sum_{t_i=1}^{T(n)} P(t_i) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots Equation\ 2$$

In statistical meaning, $P(t_i)$ shows the possibility of the i^{th} parse tree of all possible parses, while $P(w_{1,n})$ shows how grammatical a sentence $w_{1,n}$ could be in the language. The larger the $P(t_i)$ is, the more reasonable the parse is. What this means is that to find the best parse

of all possible parses of a sentence is nothing but to find $max_{0<t_i\leq T(n)}\{P(t_i)\}$. Therefore, $P(w_{1,n})$ and $max_{0<t_i\leq T(n)}\{P(t_i)\}$ are two important facts in syntax analysis that the first one gives the reasonability of a sentence in PCFGs, while the second one gives the most possible parse of all. Accordingly, the focus is on the $P(t_i)$ values, to get the most probable parse structure.

The following section describes the sample corpus on which this algorithm was applied and the data pre-processing tasks that were done on it.

4.3 The Sample Corpus

For the purpose of this study, 300 Afan Oromo simple and complex sentences that were collected from widely used grammar books, and newspapers of the language were considered. 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 process of each sentence in the sample corpus. However, the researcher admits that the sample size considered is still very small.

The sentences were selected from books entitled "Seerluga Afan Oromo" by professor Baye Yimam [44] and "**NATOO: Yaadrimee Caasluga Afan Oromo**" by Berkesa Adugna [52] , all of them were prepared with the purpose of serving as references for teaching Afan Oromo language at tertiary and secondary levels, respectively and **"Labsii Walii-gala Mirgoota Namummaa"** a human right law published in June 17, 1956 GC. The books were chosen as references after discussion with linguistic advisors. Whereas the articles from the human right law were chosen because it is incorporated under NLTK corpora as a reference for natural language processing. The sentences were selected in such a way that sentences constitute two or more of the phrase classes and embed one of the various clause types (i.e. relative clause, reason clause, result clause and time clause) for complex sentence and one noun phrase and one verb phrase for simple sentence all discussed in chapter three.

The sentences were then hand tagged for later parsing purpose of this study. Some of the parses of the sample sentences were given in Baye [45] while the remaining sentences were

hand tagged and parsed by the researcher and a Teachers Training College Afan Oromo lecturer according to the phrase structure rules of the Afan Oromo language. Comments and suggestions were then taken from the other Afan Oromo language expert and the linguistic advisor for the thesis. The probability calculations for the words in the sentences, grammar rule induction, and probability assignment to the grammar rules, were conducted using 240 sentences (80% of the sample corpus). These sentences were picked randomly as training set while the rest 60 sentences (20%) from the corpus were used as a test set.

4.4 The Morphological Pre-processing

Although 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 Allen [19]. In Afan Oromo, for instance, a word may consist of a single stem but, it might have multiple morphemes. Afan Oromo is an inflectional languages and, therefore, most of the words in the language consist of a stem plus an affix(s) (e.g., '**barat-a'** "studen" sing. masculine; '**barat-oota'** "The students" pl ; '**barat-u'** "student" sing. Feminine etc.).

As it was pointed out in chapter two, such a preprocessor (i.e., a morphological analyzer) is one of the most essential NLP systems in the development of part of speech tagging and sentence parsing systems. Even though there is a single attempt regarding Afan Oromo automatic morphological analyzer by Abeshu [42], it was difficult to the researcher in order to incorporate the prototype because the resources were not available in any of the archives. Hence attempts were made to incorporate a manually annotated stem and affix(s) only for this research purpose.

4.5 The Part Of Speech Tagger

As it was mentioned in chapter two, Getachew [20] and Nedjo [54] has developed a prototype simple automatic parts of speech tagger (POS tagger) for Afan Oromo language.

Getachew [20], in his study, used the Viterbi algorithm with HMM, whereas Nedjo [54] used Maximum Entropy Markov Model. The researcher of this sturdy able to incorporate HMM POS tagger for the purpose of this study.

Word Code Table:

This table is used to store words from the sample text and a corresponding word code is given sequentially for each word. The 3029 words have been taken from 300 complex sentences.

Category Code Table:

This table is used to store the word categories identified based on universal part-of-speech tag set. The universal tag sets are:

ADJ,J	An adjective
AdjP	Adjectival Phrase
ADV	An adverb
ADVC	An adverb not separated from a conjunction
AdvP	Adverbial Phrase
AUX	Auxiliary verbs and all their other forms
COMP	Complement
CONJ	A conjunction
ITJ	Interjections
JC	A conjunction not separated from an adjective
JNU	A numeral used as an adjective
JP	An adjective not separated from a preposition
JPN	A noun not separated from a preposition and that function as an adjective
N	Noun in all forms
NC	A conjunction not separated from a noun
NP	A preposition not separated from a noun
NP	Noun Phrase
NUM	Number

NV	Verbal nouns
PP	Prepositional Phrase
PREP	A preposition
PUNCT	Punctuation
REL	Relative clause
V	Verb in all forms except auxiliary, compound and all forms of auxiliary and compound verbs
VC	A verb prefixed or suffixed by a conjunction
VCO	**C**ompound verbs
VP	Verb Phrase
X	to represent unknown

This table is used to store the 28 word categories identified with a corresponding category code. The total number of categories identified are 27, the 27th tag being for all punctuations and an additional 1 for all unidentified words. Some minor changes such as the replacement of 'J' by 'Adj', which both represent the adjectives class in the POS tagger and the parser, respectively, have taken place.

Lexical Probabilities Table:

The lexical probabilities table stores the probabilities of words given categories (tags) for each word in a given corpora. This is written as p (word\tag) or p (word\category) or shortly as p (w_i\C_i). For instance, p (Haawwii\N) denotes the (lexical) probability of Haawwii "wish", which is a common name in the language, to be a noun, and p (Haawwii \ADV) denotes the (lexical) probability of Haawwii to be an adverb in a given pre-tagged corpus. The lexical generation probability is estimated simply by counting the number of occurrence of each word by a category. Mathematically this is given by:

$$P(W_i \backslash C_i) = \frac{\text{number of times Wi appears in category } C_i}{\text{Total number words with category } C_i} \quad \textit{.....Equation 3}$$

Such probability values of the lexical tables were recalculated using the new data entered in the word category code table.

Transition Probabilities Table:

Transition probabilities are denoted by p ($C_i \backslash C_{i-1} \ldots C_n$) and refer to the probability of a tag given one or more previous tags. For instance, the probability of a noun to be preceded by an adjective is given by p (C_i = Noun\$C_{i-1}$ = Adjective). Depending on the value of n (which tells us the maximum number of categories being considered), we can have bigram (n = 2), trigram (n = 3) or in general an n-gram transitional probabilities. If (n = 2), the model is a bigram model and it is denoted by p ($C_i \backslash C_{i-1}$). If (n = 3), the model is a trigram model and it is denoted by p ($C_i \backslash C_{i-1} C_{i-2}$). These models assume that the probability of the occurrence a particular category depends solely on the one or more categories immediately preceding it. In practice, given a database of texts tagged with part of speech, the bi-gram or transitional probabilities can be estimated simply by counting the number of times each pair of categories occurs compared to individual category counts. Mathematically this is written as:

$$P(C_i = \propto / C_{i-1} = \beta) = \frac{\text{count of the number of times } \propto \text{ and } \beta \text{ occur together in the corpus}}{\text{Number of times } \beta \text{ occurs in the corpus}} \quad \ldots\ldots \text{Equation 4}$$

Where $\propto$ and β are parts of speech codes. Such probability values of the transitional probability table were recalculated using the categorical sequence of the newly entered data into the word code table.

4.6 Extraction of A Probabilistic Context Free Grammar

In the process of extracting the PCFGs, the sentences that are used as a training set were first manually hand parsed and represented in the following manner.

‘Ani huccuu adii Tolaan bitee kaleesa argee.’

(CS
(MC (NP (N **Ani**))(VP(NP (N huccuu) (Adj **adii**))(NP(N Tolaan)(V bitee))))
(SC(VP (Adv **kaleesa**) (V **argee**))))

Following this, the CFG rules were extracted from these manual parses of the training set. Then, in order to gather statistical information about the grammar rules observed, counting the number of occurrences of each rule, which was found to be the simplest way for the purpose, was carried out in the manually parsed training sentences. Later, these statistical figures were used to estimate the probability of each rule being used (See also Allen, [19]). Finally, the probability values were assigned to each of the extracted rule using the formula:

$$P(Rj/C) = \frac{\text{Count of the number of times Rj occurs in the PCFG table}}{\text{Count of the total occurrence of the category C on the LHS in the PCFG}} \quad \text{.....Equation 5}$$

4.7 Chomsky Normal From (CNF) Representation

The representation of the PCFG in the Chomsky Normal Form (CNF) is carried out for the purpose of simplicity. After the probabilistic context-free grammar was extracted, the next thing that needed to be done was to convert it into Chomsky Normal Form (CNF). This conversion was done for the purpose of simplicity. Any CFG rules can be easily re-written in the CNF, and this restriction will not result in the loss of any real expressive power. It means each rule

$R^i \in R$ is in one of the following two forms,

$$R^i: N^i \rightarrow w^j \quad \text{or} \quad R^{i''}: N^i \rightarrow N^j N^k$$

where $w^j \in W$ is a terminal symbol and $N^i, N^j, N^k \in N$ are non-terminal symbols.

A grammar in CNF would also be easier to manipulate because of the binary structure it attains.

Rules of the form A → BCDE (p), where p is the probability, are replaced by a set of rules [3]:

The PCFGs extracted for the Afan Oromo language are converted to CNF. This conversion was in line with the phrase structure rules of the language and in actual sense it related to representing it in the exact structure more than converting it to the CNF. That is, since all clauses considered were 4-word sentences, most of the rules were already in the CNF.

Table 4.1 below depicts an example of the extracted CNF rules with their respective probability values.

LHS	RHS1	RHS2	Probability
CS	**MC**	**SC**	**1**
MC	**NP**	**VP**	**1**
SC	**NP**	**VP**	**1**
NP	**R1**	**ADJ**	**0.006**
R1	**N**	**ADJ**	**1**
NP	**NP**	**ADP**	**0.005**
NP	**N**	**E**	**0.647**
NP	**N**	**N**	**0.004**
NP	**N**	**P**	**0.026**
VP	**ADV**	**V**	**0.5**
VP	**V**	**E**	**0.5**

Table 4.0-1:- An example of the CNF rules with their respective probability values.

4.8 Conclusion

In this chapter, the PCFG parsing approach implemented and the sample text used in this study were described. Using a manually annotated words and tagged text the extraction of a probabilistic context free grammar rule from the tagged corpus as well as some of the conversion of the rules to the Chomsky Normal Form (CNF) was presented.

CHAPTER FIVE

PARSING ALGORITHM AND EXPERIMENTATION

5.1 Introduction

This chapter discusses the parsing algorithms used to develop the prototype Afan Oromo complex sentence parser, the experiments conducted, and the analysis and discussions made based on the findings of the experiments. Besides, the design of the parser, which comprises the input output interface, the probabilistic rule base and the chart parsing module, is presented in this chapter.

The next section presents a discussion on the Inside-Outside algorithm based on which this parser was designed. The third section of this chapter introduces and discusses PCFG parsing as it was implemented by the Inside-Outside algorithm. The fourth section describes the design of the parser while the fifth section presents a report on the experimentation, the results observed and the solutions given.

5.2 The Parsing Algorithm

Ambiguity resolution is a crucial problem facing computational models of natural languages [19]. A given sentence, for instance, can have many possible syntactic parses that can be obtained, and are collectively known as its parse space. The probabilities of specific parse trees of a given sentence can be found using a chart parsing algorithm, where the probability of each constituent is computed from the probability of its sub constituents and the probability of the rules used. Thus, the probability of an entry E (also called node N(i,j) in this thesis) of category C using a rule i with n sub-constituents corresponding to entries E_1, …, E_n:

$$P(E) = P(Ri/C) * P(E_1)\ldots * P(E_n)\ldots\ldots\ldots\ldots\ldots \quad \textit{Equation 6}$$

Together with the chart parsing algorithm, a step that computes the probability of each entry when it is added to the chart can be implemented. The standard algorithm for computing this

probability is called the Inside-Outside algorithm. For computing $P\ (w_1,\ n)$ and $max_{0<t_i \leq T(n)}^{\{P(t_i)\}}$, either of the two probability calculations are required. That is, the inside probability or the outside probability, for each node in the parse structure. For a word sequence $w_{k,l}$ derived directly or indirectly from a non-terminal node $N_{k,l}^{j} \in N$, the inside probability is:

$$\beta_j(k,l) = P(\ w_{k,l} \mid N_{k,l}^{j}\ ) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots Equation\ 7$$

And the outside probability is:

$$\alpha_j(k,l) = P(\ w_{1,k-1}, N_{k,l}^{j}\ , w_{l+1,n}\ ) \quad \ldots\ldots\ldots\ldots\ldots\ldots Equation\ 8$$

Where $\beta_j(k,l)$ denotes the probability of the word sequence $w_{k,l}$ in which all the words are inside the node $N_{k,l}^{j}$ and $\alpha_j(k,l)$ denotes the probability of the words outside the node $N_{k,l}^{j}$, Yao and Lua, [53]

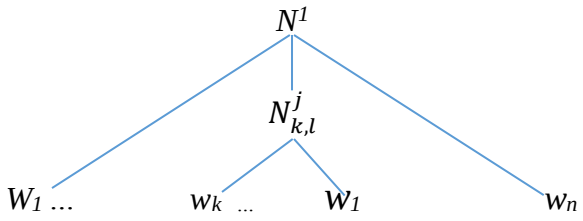


Figure 5.0-1:- The words outside and inside $N_{k,l}^{j}$.

The above figure shows the words, $w_1, \ldots, w_k\text{-}1$ and $w_{l+1} \ldots w_n$ are outside $N_{k,l}^{j}$ while the words $w_k \ldots w_l$ are inside $N_{k,l}^{j}$. The probability of the best parse tree of a sentence which is $max_{0<t_i \leq T(n)}^{\{P(t_i)\}}$ can, therefore, be obtained by calculating the inside probability or the outside probability. The parser developed in this study is based on this algorithm with a modified parse chart proposed by Yao and Lua [53]) to assist the parsing. The algorithm used for finding the best (the most probable) parse structure for a given sentence is given in section 5.4.

5.3 PCFG Parsing

In this study, a parser based on the Inside-Outside algorithm was developed to implement PCFG parsing on Afan Oromo complex sentences. Its parse mechanism, parse chart, and configuration are introduced in this section.

5.3.1 Parse Tree

Parsing a sentence involves finding a possible legal structure (or structures) for the sentence and this usually results in a tree, often referred to as a *parse tree*. Since the PCFG rules in this study are restricted to the CNF, each possible parse tree of a sentence is a binary tree. Generally, in a given parse tree, level 1 is always a set of terminal nodes, which are words, and level 2 is a set of non-terminal nodes, which are the POS tags of the corresponding words. In CNF, this is written in the form $N^j \rightarrow w^j$, and each node in level 2 only has a descendant in level 1.

For a non-terminal node N^i, if there is a rule $N^i \rightarrow w^j$, then N^i is supported by the grammar rule $N^i \rightarrow w^j$ and if there is a rule $N^i \rightarrow N^j N^k$, then N^i is supported by the grammar rule $N^i \rightarrow N^j N^k$. Assuming that each word in an n-word sentence $w_{1,n}$ has one POS tag. $T(n)$ denotes the maximum number of structure trees that $w_{1,n}$ probably have, and each non-terminal node of the trees is supported by a grammar rule, then T(n) can be calculated as follows:

$$T(n) = 1 \text{ , } n = 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad \textit{Equation 9}$$

$$T(n) = \sum_{i=1}^{n-1} T(i)T(n-i) \text{ , } \quad n > 1 \quad \ldots\ldots\ldots \textit{Equation 10}$$

This indicates that the value of $T(n)$ increases exponentially as the number of words in a sentence increases though practically there are trees in the parse tree space that involve nodes that are not supported by grammars rules and, therefore, the number of all parse trees of a sentence is less than $T(n)$.

5.3.2 Parse Chart

As it was indicated earlier, this study adopted the parse chart that was designed by Yao and Lua [53] to implement the Inside-Outside algorithm. This parse chart was based on equation 9 and 10 in section 5.3.1 and is a matrix but only top-right half would be used since it is a symmetrical matrix. The size of this chart depends on the number of words in the sentence being parsed. It means that for an *n*-word sentence the chart was an *n*×*n* matrix.

Note that the process of analyzing Afan Oromo complex sentences starts prior to passing the input sentence into the parse chart. Thus a 5-word long input sentence would be tagged into 6 POS tags, a 6-word long sentence into 7 POS tags, a 7-word long sentence into 8, and so on. This was only because of the heuristics used to deal with movements of verbal affixes during structural representation of Afaan Oroomoo complex sentences. This is illustrated by the example '***Toolaan ishee akka jaalatee, Haawwiin siritti bektee.'*** "Hawi knew that Tolla really loved her."

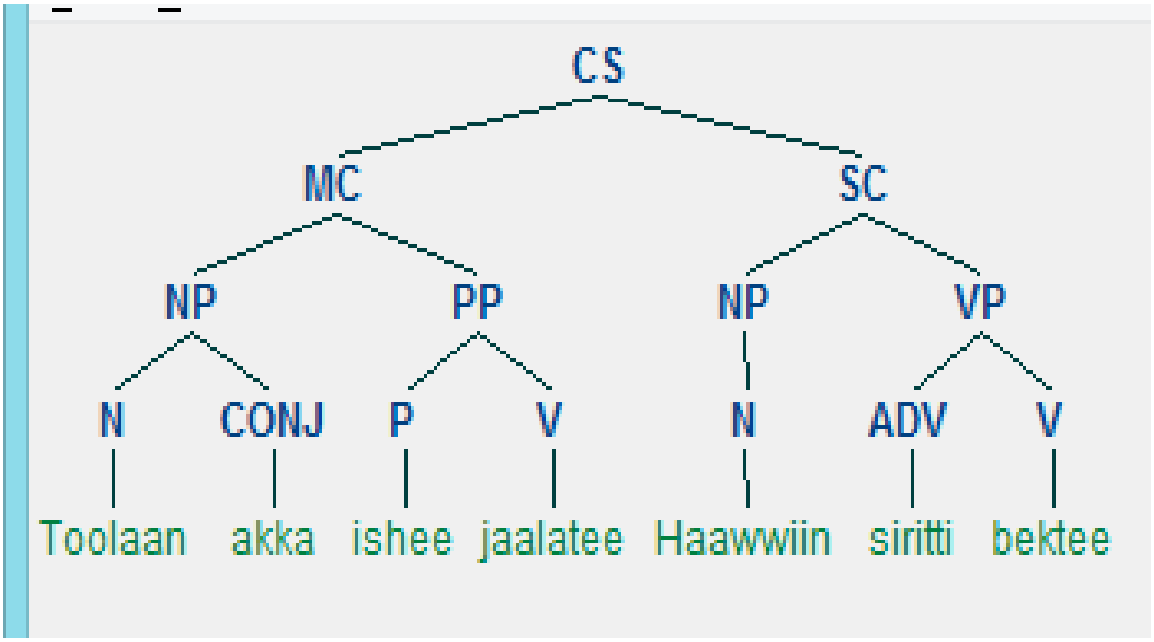


Figure 5.0-2:-Tree structure diagram of the complex sentence

As it is illustrated by the above figure, the underlined verbal affix/adposition '**_akka' "**that", detached itself from '**jaalatee'** "that he loved her", and took the position before the preposition. All the Affan Oromoo complex sentences that were considered in this study were formed by embedding clauses that involve such relativizers, which were then analyzed into verbs and verbal affixes (called ***Complements*** COMP) and denoted by ADP.

The element $N_{(i,j)}$ $(i, j \in [1, 7])$ in the figure denotes a non-terminal node and the element $N_{(1,7)}$ is the starting symbol if each $N_{(i,j)}$ is supported by a grammar rule. The non-terminal $N_{(i,j)}$ $(i, j \in [1, 8])$ in the diagonal supported by the rule $N_{(i,i)} \rightarrow w_i$ is the POS of the word w_i . Figure 5.3 below shows an example of 8-level-chart, which can parse 7-word sentences.

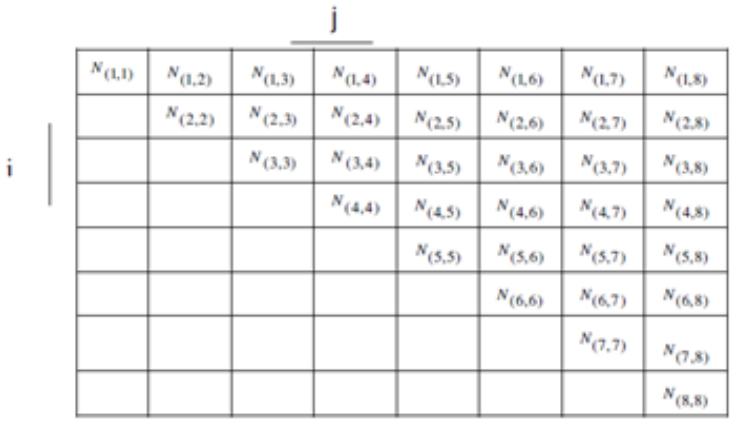


Figure 5.0-3:- An 8-level-chart

Therefore, in this study, for an **n**-word sentence $w_{1,n}$**,** if each non-terminal node is supported by a grammar rule, there are **n+1** terminal nodes in level 1 which is the diagonal, ***n*** non-terminal nodes in level 2, **n-1** non-terminal nodes in level 3,…, and **1** non-terminal node (the starting symbol) in level **n**. in the above chart, let$P\ (N_{(i\,j)})$ denote the probability of node $N_{(i,j)}$, then each node $N_{(i,,j)}$ **(*i*=*j*, and *i*, *j*∈[1, 8])** in level 1 is determined by $P(N_{(i,j)} = P(N_{(i,j)} \rightarrow w_k$, which is completely handed by the tagger, and each node $N_{(i,j)}$ in the non-diagonal levels is determined by the equation

$$P(N_{(i,j)}) = \prod_{k=1}^{j-1} P(N_{(i,j)} \rightarrow N_{(i,k)}\, N_{(k+1,j)})\, P(N_{(i,k)}) P(N_{(k+1,j)}) \quad \ldots\ldots\ldots Equation\ 11$$

The calculations of all these non-non-diagonal nodes (e.g. $\mathbf{N_{(1,2)}}$, $\mathbf{N_{(1,3)}}$, and $\mathbf{N_{(1,5)}}$) is indicated by charts **5(b), 5(c),** and **5(d)** on figure 5.4

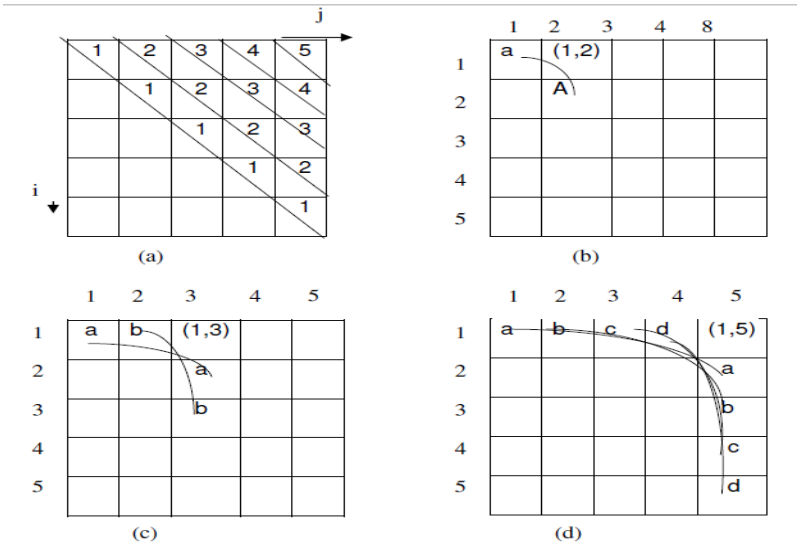


Figure 5.0-4:-The calculation of probabilities of non-terminal nodes

In the above charts, (a) shows a 5 level chart, (b) indicates that a node in level 2 depends on "aa" in level 1, (c) shows that a node in level 3 depends on "aa" and/or "bb" in level 1 and 2, and (d) shows that a node in level 5 depends on "aa", "bb", "cc", and/or "dd". This parse chart gives the parse tree space for a given input sentence, and calculating the value of $N_{(i,j)}$ based on the chart would result in the calculation of the inside probability of ($N_{(i,j)}$). The tree with the maximum probability $max_{0<t_i \leq T(n)}^{\{P(t_i)\}}$ can then be obtained from the structure that consisted of rules with the maximum probabilities. This would also result in the calculation of $P(w_{1,n})$, if there is an interest to determine how grammatical a sentence is in the language.

5.4 The Design of the Parser

5.4.1 Preprocessing the Parser's Input

A sentence is input from a file for parsing. For example, if we take the sentence:

'Jabbilee adii Haawwiin kaleesa bittee argee#'

"I saw the white calf that Hawi bought yesterday."

Here the # symbol indicates the end of the sentence. When this sentence passes through the morphological analysis process, each word is analyzed into a stem and affix(s), and outputs the following format that the tagger would use as an input:

'Jabbi adii Haawwii kaleesa bittee argee'

Tagging each stem with the appropriate POS outputs the following format:

Jabbi\N adii\Adj Haawwii\N kaleesa\Adv bittee\V argee\V .\PUNCT

Each tagged stem will then be re-synthesized with its respective affix using a hyphen (-) as a separator between a stem and its affix(s) as shown below.

Jabbi-lee\N adii\Adj Haawwii-n\N kaleesa\Adv bittee\V argee\V .\PUNCT

Following this, each word will go through a post tagger morphological process to see if the category of each tagged stem changes when reunited with the affix(s). This is a very important stage in the development of the parser not only for the reason just mentioned, but also it simplifies the complex process of parsing Afaan Oromo complex sentences. In other words, Afan Oromo verbs that take affixes (such as {-een}, {-wan}, {-(o)ota}, {-yyii} and {-lee}), and, therefore, often called *Relativizers*, are identified and handled at this stage. Finally, the input sentence would attain the following format, and will be passed to the sentence parsing module:

Jabbi-lee\N adii\Adj Haawwii-n\N kaleesa\Adv bittee\V argee\V #\PUNCt

The input preprocessor module algorithm is provided below.

For each sentence in the file
Get one sentence at a time
For each word in the sentence
Get the stem of each word
Call the HMM POS tagger
Get the tagged stems of the sentence
Call the Morphological Synthesizing Function to update the Category output of the tagger
Pass the final tagged string of words to the parser

Figure 5.0-5:-An algorithm for preprocessing an input sentence to the parser

5.4.2 The Input & Output Interface

A sample interface was designed for the sentence parser application program. When Afan Oromo complex sentence parser application program runs, the Afan Oromo sentence parser window is displayed. In fact, this is the main interface that will be displayed when the application program runs for the first time and it persists until the program is exited. This window has four main buttons, just below the menu bar. Below is a brief description of each of these buttons.

The POS Tagger button

Clicking this button displays the tagged stems of the input words.

The Tagged Sentence button

Clicking this button displays the final tagged input sentence isolating and tagging the complement independently among the other words.

The Parse button

This button allows a user to parse sentences from a saved file one by one.

Therefore, the prototype developed performs its function as follows.

The morphological analysis component takes in the input sentence and outputs a string of stems as given below:

'Mootumaan, Biyyattin sadarkaa guddaa dinagdee irratti galmeesiftee ibsee.'

"The government described the greatest economic achievement that the country has scored"

The tagger then takes in the above string of stems as an input, tags each stem with the appropriate POS tag and outputs the following format that the parser would use as an input.

'Mootumaan\NOUN Biyyattin\NOUN sadarkaa\ADV guddaa\ADJ dinagdee\NOUN irratti\ADP galmeesiftee\VERB ibsee\VERB .\PUNCT'

However, before the above output of the tagger is passed to the parser, each tagged stem word would be synthesized with its affixes (if any) and the resulting word's category will be searched from a table that updates the categories of inflected stems. Thus in passing through this process the above Afan Oromo complex sentence, for example, would have the following format.

Mootumaa-n\NOUN Biyya-ttin\NOUN sadarkaa\ADV guddaa\ADJ dinagdee\NOUN irratti\ADP galmeesiftee\VERB ibsee\VERB .\PUNCT'

In the above example, those words in the input sentence that are affected by the above processes are indicated by underline. This was the actual output of the POS tagger assisted by the morphological analysis. The parser then extracts each word and each POS tag from the tagged sentence and stores them in one dimensional array variables. The output of the parser gives the probability of the selected parse structure, the parse result, and the grammar rules used in parsing. For the example sentence shown so far, the outputs include:

The probability of the parse structure: 0.000034129851158584

The parse result:

(CS (SC (NP (N Mootumaa-n)(,)))
(MC-VP
(S
(NP (N Biyya-ttin) (ADJP (ADV sadarkaa) (ADJ guddaa)))
(VP (NP (N dinagdee) (ADP irratti)) (V galmeesiftee))) (VP(V ibsee))))

The rules applied in parsing the above sentence include:

CS → SC MCP-VP

SC → NP N NP → N

MC-VP → S VP S →NP VP NP → N ADJP ADJP → ADV ADJ

VP →NP V NP → N ADP V →E

The rules used in parsing are the grammar rules used and here those rules containing terminal nodes (for example, N **Mootuma-n**) are not displayed since such lexical rules are not included in the PCFG table. This is because the parser inputs pre-tagged sentences and lexical information is handled by the component systems that preprocess sentences before the parser.

The Probabilistic Rule Base

The PCFG rules have been converted to CNF by inducing the hand parsed sentence using python. The rules which are in CNF, were extracted from the sample corpus and are represented in the same database as the lexical probability table and others in the following format.

PCFG-CNF			
LHS	RHS1	RHS2	Probability
S	NP	VP	1
NP	Adj	R1	0.006
R1	Adj	N	1
NP	AdjP	NP	0.005
S'	S	COMP	1
COMP	COMP	E	1
VP	Adv	V	0.114
VP'	S'	VP	0.262
VP	V	E	0.332
NP	Adj	N	0.185

Table 5.1 :- Parsing result on Training Set before making no error correction

PCFG Representation

In the above table, E is an empty production. The field LHS stores the left hand side of the rules while RHS1 and RHS2 represent the first and the second right hand side rules, respectively. The probability field represents the associated probability values for each of the grammar rules. (For a complete list of the PCFG rules extracted from the corpus see Appendix 5). Word information needed in the parsing is stored in the Word Code and Lexical Probabilities tables that were implemented by the tagger and stored in the same database as the PCFG table.

The chart parsing module

The chart parsing module is the PCFG Inside-Outside algorithm assisted by the parse chart that Yao and Lua [53] designed based on equation 7 and 8 in order to implement the Inside-Outside algorithm. For each non terminal node N(i,j) in the parse chart where i is different from j and i, j = [1,n], its value is determined using the formula given in equation 11.

The following is the Inside-Outside algorithm that is used in this study to implement PCFG parsing.

For each POS tag in the chart matrix diagonal

Check if a POS tag occurs alone or together with the next one at the right hand side of a rule:

If it occurs in both cases , get the highest probability of these rules and store its left hand side on the POS matrix, store the rule in the chart entry table and store the probability value in the Probability array

Else store its left hand side on the POS matrix, store the rule in the chart entry table and store the probability value in the Probability array

Else, store 0 in the Probability array and leave the chart matrix value empty

Set the lexical probabilities of each of the words to their corresponding categories.

Get the maximum of $P(t_i)$ and assign it to be the best parse.

Figure 5.6:- The Parse Chart Procedure to Implement the Inside-Outside Algorithm

This algorithm was also coded in the development of the prototype and was used to calculate the inside probabilities of each parse in the parse tree space. In calculating the probabilities, the algorithm used equation 9 and 10 that adds a step to construct the parse bottom up. During parsing, the categories of words in a sentence are fed one by one into the diagonal (i.e., the first level) of the chart. Here, the parse tree space is constructed bottom to top and the probability of each parse is calculated as well.

For each sentence to be parsed
Extract the words and tags from the output of the tagger.
Insert the POS tags into the diagonal of the chart matrix
Call the parse chart procedure that implements the Inside-Outside algorithm to calculate the probability of each node in the chart matrix.
Get the highest probability from the Probability array.
Get the rules that produced the highest probability from the chart entry table.
Generate the parse structure and display the output.

Figure 5.7: The Overall Implementation of the Parsing Algorithm

The parser is represented diagrammatically in the figure below.

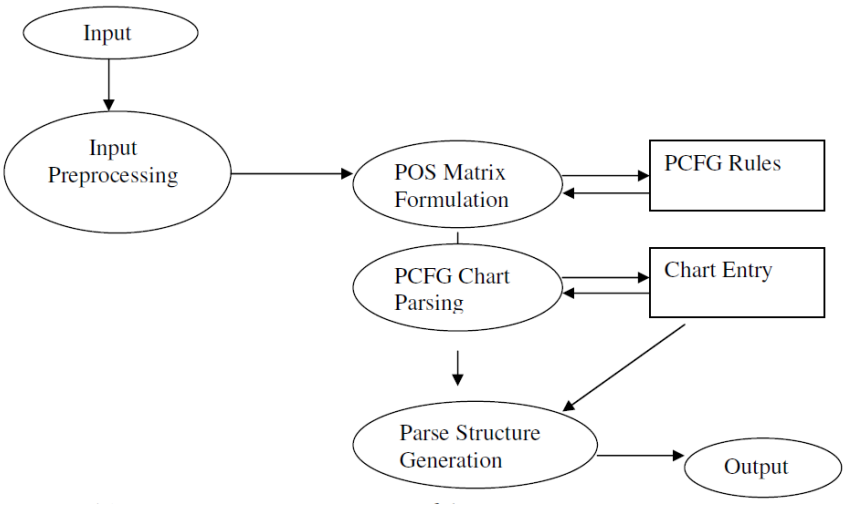


Figure 5.8:-Diagrammatic Representation of the Parser

5.5 The Experiment

The sample text selected, which was discussed in chapter four, was used for experimentation. Each word in the corpora had been morphologically analyzed manually, each sentence were also hand tagged and hand parsed by Afan Oromo language instructor from department of Afan oromo in Arsi University and by the researcher himself, with comments and suggestions from the linguistic advisor and other experts of the language at Arsi University of Language Studies and some other experts of the language. The 300 sample sentences were selected randomly based on

the way they attain their complex features (i.e., being composed of a simple NP and a complex VP) and the distribution of the different phrase structures, using judgment sampling technique.

As it was made clear from the very beginning of this study, the main objective of this study was to parse Afan Oromo complex sentences automatically using the PCFG bottom up chart parsing approach and the Inside Outside algorithm that gets its POS inputs from a POS tagger. Hence, the experimentation began with checking the boost in accuracy (if any) that the POS tagger exhibited..

To that end, when the tagger was first trained**,** and then run on the same data, the accuracy obtained was 76.3%. The errors observed were more of human made errors (errors made during the manual morphological analysis and tagging), and the preparing the lexical and transitional probability tables. After reviewing the manually performed activities and the lexical and transitional probability calculations, and making corrections where necessary, the tagger attained 89.7% accuracy on the training set. This was higher than what the tagger attained on its original training set, i.e., 84%.

Tested with the Test Set, the tagger originally had 66.6% accuracy and this grew up into 80% in this study. One source of error observed was the accidental conflicting category that the morphological analysis and the bi-gram propose for a given word. This source of error was left unsolved due to shortage of time. Also, errors in this phase of testing the tagger could be attributed to the small size of the sample corpus considered.

Generally speaking, the improvement on the POS tagging module (or on the input preprocessing module in general) could mainly be attributed to the use of the morphological preprocessing analysis prior to the application of the HMM tagging, the category checking mechanism after the tagged stems were synthesized with their affixes, the statistical category guessing mechanism that fully relies on the transitional probabilities, and the slight increment in the corpus used (from 261 words in the original tagger to 883 stems in this study).

5.5.1 Experiment on the Training Set

All the manually carried out tasks, i.e., the morphological analysis, tagging and parsing as well as the probability calculations for the words in the sentences, the grammar rule induction, and probability assignment to the grammar rules, which were all discussed in chapter 4, were conducted using 240 sentences randomly picked from the sample corpus and saved as Training Set. The first experiment was conducted on these 240 sentences.

One major source of error was the existence of two relatively more probable (than the other rules) rules that have the same RHS. These rules were **S →NP VP** with a probability of 1.0 and **VP →NP VP** with a probability of 0.524. Since the former rule has more probability than the latter, it dominates and **S** frequently appeared at a node where **VP** should have come.

The parser was then retrained and the test redone again on the Training Set. The final results obtained before and after such corrections were made were then reported under results of the experiment sub-section 5.5.3.1.

5.5.2 Experiment on the Test Set

The second experiment was done on the remaining 60 sentences from the original corpus and saved as Test Set. The findings on this unseen part of the corpus were reported under section 5.5.3.2.

5.5.3 Results of the Experiment

The following sections discuss the results on the obtained on the experiments carried on the Training Set and Test Set. It also reports the result (before making no correction and after making corrections to the lexicon, grammar and the algorithm) on the training set.

5.5.3.1 Result on the Training Set

The result obtained when the Parser was trained and run on the same data, Training Set, is shown in the following table.

Data/set	No of sentences	No of erroneously parsed sentences	Accuracy
Training Set	240	80	66.6%

Table 5.-0-2:- Parsing result on Trainin g Set before making no error correction

As one would expect the accuracy achieved should have been higher than 66.6% as the parser was trained and tested on the same data, i.e. on the Training Set. But, most of the errors at this time were as mentioned earlier emanated from the rule confusion of S→NP VP and VP→ NP and VP, and this was solved considering the two rules as ones that use different RHS. The other reason for the low accuracy experiment was due to the human made errors the accuracy was not as high as it was expected. The final accuracy obtained on this set after human made errors were identified and corrected is shown in the table below.

Data/set	**No of sentences**	**No of erroneously parsed sentences**	**Accuracy**
Training Set	240	48	80.0%

Table 5. 3:-Parsing result on Training Set after making some error correction

5.5.3.2 Result on The Test Set

The test on the unseen part of the training corpus provided the result shown in the following table.

Data/set	**No of sentences**	**No of erroneously parsed sentences**	**Accuracy**
Test Set	60	17	71.6%

Table 5.4:-Parsing result on Test Set

Another test was attempted to be conducted using some 10 simple sentences that were used by the previous study, i.e. parsing Afan Oromo simple sentence [7]. Although the PCFG tables of both the previous and the present studies share most of the rules, the probability values associated a given rule differs from table to table. For instance, in the case of PCFG rules extracted from the simple sentence corpus, the probability of the rule VP →N V(0.199) is higher than VP →V E(0.066) whereas in the PCFG rules table that was extracted from complex sentences, the probability of VP→V E (0.332) is higher than VP→N V(0.007), which is totally a reverse case. Hence, in this test almost every sentence considered were erroneously parsed.

A third test was conducted just by embedding the manually annotated morphological analysis module and HMM post tagger into the previous simple sentence parsing for Afan Oromo text. The test was conducted on 10 sentences which included sentences that were erroneously parsed by the simple sentence parser. Accordingly, nine of the ten simple sentences that were left unparsed in the previous studies were successfully parsed.

5.5.4 Solution to Identified Problems

Errors that were encountered during the various experimentation phases of this study start from such human errors as wrong manual morphological analysis, tagging and parsing of the data used the sample corpus. These and the other sources of errors such as miscalculation of the lexical, transitional and PCFG rule probabilities in the database were corrected through iterative approach. Besides, errors were encountered when words that did not have their stem in the database or any of their inflectional forms in the sample corpus used were mis-tagged or left untagged by the tagger. To deal with this untagged words, a module that was implemented by former studies to guess the word categories for untagged words, which the tagger usually tags as UNC, was adopted.

In this study, therefore, new words or stems were disambiguated by using bi gram lexical co-occurrence probabilities. This solved the problem to some extent. However, since there was

no absolute way of handling mis-tagged words, the accuracy report on the Test Set was very low. The major problem of mis-parsing was the dominance of **S →NP VP** over another rule with the same RHS but relatively lower probability, **VP→NP VP.** This problem was alleviated by putting an apostrophe on the **VP** at the RHS of the second one, assigning a SC and MC for subordinate and main clause and, thereby considering the two confusing rules as ones with different RHS. The inadequacy of rules at the rules library, i.e. at the PCFG table, often referred to as *under generation* and errors during the extraction of rules and their probabilities were additional sources of nuisance during the experiments, which were both minimized through iterative approach.

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

The thesis has tried to describe a way of integrating the ideas and outputs of already studied Afan Oromo NLP systems but in a different approach in order to solve a bit further problem in the area of the development of an automatic sentence parser for Afan Oromo. To this end, a POS tagger and a simple sentence parser for Afan Oromo were used as the springboard. The Inside-Outside algorithm together with a chart parse module that was originally proposed by Yao and Lua [53] in order to parse Chinese sentences, were also used to implement the Probabilistic Context Free Grammars PCFGs parsing, and efforts have also been made to develop a prototype.

Parsing is the process of discovering analyses of sentences, that is, consistent sets of relationships between constituents that are judged to hold in a given sentence, and, concurrently, what the constituents are, since constituents are typically defined inductively in terms of the relationships that hold between their parts. There are two ways in which the process of parsing sentences of such kind can be carried out: manual and automatic. The manual process is tiresome, prone to error, expensive, and obviously, the problem would worsen as the volume of information gets huge and complex. Thus, the second way of parsing sentences, automatic sentence parsing, avoids such complexities and plays important roles in Natural Language Understanding systems.

As the end goal, this study was to develop an automatic complex sentence parser for Afan Oromo, important concepts and terms in relation to parsing were reviewed and areas where the outputs of sentence parser are useful were illustrated. Besides, the rule based and the stochastic approaches, which are the major approaches to automatic sentence parsing in particular and to NLP in general, and other strategies, were briefly described and reviewed. The knowledge base of a sentence parser and the necessary components such as the lexicon

and the grammatical formalisms that are needed to store information that would assist the parsing process were also discussed in some detail.

Following this, literature in the area of the Afan Oromo writing system, lexical and grammatical categories was reviewed and discussed. This was because the knowledge of the grammar of the language is the core component in designing the parser. Thus, features of the language that were considered in designing the various components of the parser were made clear. Almost all lexical and phrasal categories, sentence formalisms, typical characteristics of complex sentences, and features of the language that were considered in designing the various components of the parser were also discussed.

Next, the developed sample corpus that was used in this study and some of the major problems the researcher faced in the process of getting the necessary sample and the steps taken to deal with the problems were presented. That is, since there is no corpus developed so far for studies on Afan Oromo complex sentence parsing, 300 complex sentences were collected from two widely used Afan Oromo grammar books, published articles and newspapers. Then, each of the words in the corpus were morphologically analyzed, tagged and parsed manually. The lexical and transitional probabilities of each of the words in the ***Training Set***, which was a subset of the sample corpus used as a training set, were calculated. The grammar rules were extracted, the probability of each rule in the ***Training Set*** were calculated, the resulting PCFG rules were converted into Chomsky Normal Form (CNF) for simplicity, and represented in a table called PCFG-CNF.

The thesis then presented the algorithms and modules required by the parser to access the knowledge base and parse input sentences with appropriate lexical categories. For this purpose, an interface, which would allow a user communicate with the system was created and a prototype was developed using python Tkinter.

Experiments were conducted in two phases, one on the Training Set and the other on the Test ***Set***. Only one parameter, the percentage of correctly tagged and parsed words and sentences in the sampled text, was used to measure the performance boost of the original purely statistical part-of-speech tagger, and a simple sentence parser, and a newly developed complex sentence Afan Oromo parser in this study. The results achieved based on the small

sample were high, 80.0% on the training set and approximately 71.6% on the test set. Before achieving such accuracy, the experiment was repeatedly done on both the Training Set and the Test Set identifying errors and making corrections. Most of the identified errors were human made errors during the preprocessing of the parser's input, conflicting PCFG rules, low probability and absence of some rules. Finally, a discussion on the possible causes of errors and their solutions was made before closing up the thesis.

Although the accuracy of the parser developed in this study was somewhat above average, it may not have an immediate practical application for the parser was not trained on large quantities of data that endorsed all features of Afan Oromo (complex) sentences.

It is possible to conclude that this thesis was an attempt to indicate the possibility of using probabilistic approach particularly HMM on NLP of Afan Oromo and to use the statistical approach for automatic sentence parsing in addition to rule based or Hybrid approach.

It was in the mind of the researcher that Ethiopian students and researchers would strengthen such a practice that would lead towards the ultimate materialization of higher levels and more demanding research endeavors such as conceptual parsing and machine translations, which all are tasks of NLP.

6.2 Recommendations

There are many shortcomings in this research. These limitations are pointed out below and are active research areas, which should be addressed by interested individuals in the area. The efforts of those researchers might enable efforts of coming up with an efficient sentence parser for Afan Oromo language. Thus, the following could be recommended as possible research areas.

1. In this study, the bi gram lexical co-occurrence assisted by a manually annotated morphological analyzer, which was developed by assuming the input and output formats of a previously conducted study, was used to guess for unknown words. This technique works pretty well for the various inflections of a given stem in the database. Although this was one step forward to earlier studies, some words that were absolutely new to the

database (i.e., those none of their inflectional forms exist in the database) were still tagged wrongly. Thus future researches could explore for a better result combining the both bi gram and tri gram lexical co-occurrence and the integrated system of the morphological analysis systems studied.

2. Future researchers can prepare corpus in which both simple and complex Afan Oromo sentences have proportional representation, extract the PCFG rules and then test the performance of the approach and techniques used in this study in parsing both simple and complex Afan Oromo sentences.
3. Replicate this work using a large data and incorporating all types of sentences with all attributes like case, number, gender, person, tense, and definiteness and so on to increase the coverage of the current system in parsing various sentence types. Based on this, the performance of PCFG on the NLP for Afan Oromo could be explored.
4. Conduct similar researches in other local languages such as, Tigrigna, and so on by adopting the procedures followed in this study.
5. Noun Phrase recognition, conceptual parsing, word sense disambiguation and machine translations are other possible future research areas worth conducting as a continuation of a full-fledged Afan Oromo sentence parser.
6. It could commence the need to develop processed Afan Oromo corpus (i.e., morphologically analyzed, hand tagged, parsed, etc data) for purposes of experimentation and researches on statistical natural language processing.
7. The grammar rules extracted and the corresponding probabilities are still static values. Both the grammar induction and the corresponding probability computations could be made dynamic, i.e. to recalculate the probability values and add new grammar rules, during the parsing of sentences.

REFERENCES

[1] Mirzanur Rahman, Sufal Das and Utpal Sharma, "Parsing of part-of-speech tagged Assamese Texts," *International Journal of Computer Science,* vol. 6, no. 1, pp. 28-34, 2009.

[2] Abebe Abeshu. "Analysis of Rule Based Approach for Afan Oromo Automatic Morphological Synthesizer," *STAR Journal,* pp. 94 - 97, 2013.

[3] Danel Gochel Agonafer, "An Integrated Approach to Automatic Complex Sentence Parsing For Amharic Text," Unpublished Msc Thesis , ADDIS ABABA, 2003.

[4] Kwon, Yong-uk Park and Hyuk-chul, "Korean Syntactic Analysis using Dependency Rules and Segmentation," *Proceedings of the Seventh International Conference on Advanced Language Processing and Web Information Technology (ALPIT2008),* vol. 7, no. 1, pp. 59-63, 2008.

[5] Edward, Michael Liddy, Encyclopedia of Library and Information Science, 2nd ed., Marcel Decker.

[6] Warner, Amy J, "Natural Language Processing.," *Annual Review of Information Science and Technology,* vol. 22, pp. 79-107, 1987.

[7] Diriba. MEGERSA, "An Automatic Sentence Parser for Oromo Language Using Supervised Learning Technique," Unpublished Masters Thesis, Addis Abeba, 2002.

[8] Abera N, "Long vowels in Afan Oromo: A generic approach," Master's thesis, School of graduate Studies Addis Ababa University, Addis Ababa, Ethiopia, 1988.

[9] Grage G. & Kumsa T, "Oromo dictionary," African studies center. Michigan State University, Michigan, USA, 1982.

[10] Girma Debele Dinegde ,Martha Yifiru Tachbelie, "Afan Oromo News Text Summarizer," *International Journal of Computer Applications*, vol. 103, no. 4, pp. 1-6, 2014.

[11] M. Volk, "Parsing German With GSPG: The Problem of Separable-Prefix Verbs," MA Thesis., 1988.

[12] Wilson, Kyongho Min & William H., "Are Efficient Natural Language Parsers Robust?," School , Sydney ,Australia , 2005.

[13] Dostert, B. H and F.B. Thompson, "Syntactic Analysis in REL English.," in *Papers in Computational Linguistics*, Budapest, 1976.

[14] Thant, Win Win, Tin Myat Htwe, and Ni Lar Thein, " Parsing of Myanmar sentences with function tagging," University of Computer Studies, Yangon, Myanmar, 2012.

[15] Getachew Mamo, Million Meshesha, "Parts of Speech Tagging for Afaan Oromo," *International Journal of Advanced Computer Science and Applications,Special Issue on Artificial Intelligence*, no. 53230, 2013.

[16] Abraham Tesso Nedjo, Degen Huang, Xiaoxia Liu, "Automatic Part-of-speech Tagging for Oromo Language Using Maximum Entropy Markov Model (MEMM)," *Journal of Information & Computational Science 11:10*, vol. 11, no. 10, p. 3319–3334, 1 July 2014.

[17] Debela. Tesfaye, "Designing a Rule Based Stemmer for Afaan Oromoo Text," Masters Thesis, Addis Ababa University, 2013.

[18] Y. Mao, "Natural Language Processing Module (Part of Speech Tagging and Sentence Parsing) Laboratory Manual," Cognitive Science In Context (CSIC), 10 10 1997. [Online]. Available: http://www.csic.cornell.edu/201/natural_language. [Accessed 15 8 2015].

[19] Allen J, Natural Language Understanding, 2nd ed., California: The Benjamin/ Cummings Publishing Company, 1995.

[20] Abiyot Bayou, "Developing Automatic Word Parser for Amharic Verbs and Their Derivation," Master Thesis at School of Information Studies for Africa, Addis Ababa, 2000.

[21] Merlo paola, Parsing with Principle and Classes Information, Boston: Kluwer Academic, 1996.

[22] Reyle, U and C. Rohrer, Natural Language Parsing and Linguistic Theories, Boston: Reidel Publishing Company, 1988.

[23] Y. Mao, "Natural Language Processing Module (Part of Speech Tagging and Sentence Parsing) Laboratory Manua," 1997. [Online]. Available: http://www.csic.cornell.edu/201/natural_language/. [Accessed 27 9 2015].

[24] Tamas E. Doskocs, "Natural Language Processing in Information Retrieval," *Journal of the American Society for Information Science,* vol. 37, no. 4, pp. 191 - 196, 1986.

[25] B. Prichett, Grammatical Competence and Parsing Performance, Chicago: The University of Chicago, 1992.

[27] Shieber, Stuart M., Yves Schabes, and Fernando CN Pereira, "Principles and Implementation of Deductive Parsing," *The Journal of Logic Programming*, vol. 12, pp. 1-37, 1995.

[28] Stuart J. Russel, Peter Norving, Artficial Intelligence: A Modern Approch, 3rd ed., Prentice Hall, 2010.

[29] Rus, Mihai Lintean, Vasile, "Naive Bayes and Decision Trees for Function Tagging," in *In Proceedings of the International Conference of the Florida Artificial Intelligence Research Society*, Key West, FL, 2007.

[30] Daniel Schutzer, Artificial Intelligence: An Applications-Oriented Approach, New Uork: Van Nostrand Reinhold Company., 187.

[31] Salton, G and Michael J. McGill, Natural Language Processing, New York: McGraw-Hill, 1983.

[32] Spark jones, & Bonnie Lynn Webber, Readings in Natural Language Processing., Los Altos, USA: Morgan Kaufnann, 1986.

[33] Gazdar, Gerald; Mellish, Chris., "Natural Language Processing in Prolog.," susx university, 1996. [Online]. Available: http://www.cogs.susx.ac.uk/local/books/nlp-in-prolog/ch01/chapter-01-sh-1.6.html#sh-1.6... [Accessed 14 May 2016].

[34] Naom Chomsky, Aspects of the Theory of Syntax, Cambridge, Massachusetts : MIT Press, 1965.

[35] William Woods, "Transition Network Grammars of natural Language Analysis," in *Communication of the ACM*, Karen, 1970.

[36] Kay, Martin, "Functional grammar," in *n Proceedings of the Fifth Annual Meeting of the*, UK, 1979.

[37] Ralph Grishman, "Natural Language Processing," *Journal of the American Society*, vol. 35, no. 5, pp. 291-296, 1984.

[38] Liang HUANG,Yinan PENG,Huan WANG,Zhenyu WU, "PCFG Parsing for Restricted Classical Chinese Texts," National University of Singapore, Singapore, 1998.

[39] Blaheta, D. and Johnson, "Assigning function tags to parsed text," in *In Proceedings of the 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics.*, 2000.

[40] Berwick, Robert C., and Amy S. Weinberg, The Grammatical Basis of Linguistics Performance: Language Use and Acquisition, London: MIT press, 1989.

[41] Biber, Douglas, Susan Conrad, and Randi Reppen, "Corpus Linguistics: Investigating Language Use," Cambridge University Press, New York, 1998.

[42] Abebe Keno, "Case Systems in Oromo," MA Thesis, Addis Ababa University, 2002.

[43] Baye Yimam, "Oromo Substantives: Some Aspects of Their Morphology and Syntax," MA Thesis. Addis Ababa University., Addis Ababa , 1981.

[44] Baye Yimam, Seerluga Afaan Oromoo, Adiis Ababa: Adiis Ababa University press, 2003.

[45] Baye Yimam, 'The Phrase Structure of Ethiopian Oromo," PhD Desertation : University of London, London, 1986.

[46] Askale Lemma, "Seerluga Afaan Oromoo," Unpublished Hand out for Oromo Syntax, AAU, Addis Ababa, 1997.

[47] Tilahun Gamta, Oromo-English Dictionary, Addis Ababa : Addis Ababa. University Press., 1989.

[48] Sag, Ivan A., et al., Syntactic Theory: A Formal Introduction, Standford : Centre for the Study of Language and Information, 1999.

[49] Hamid Muudee, Hamiid Muudee's English-Oromo Dictionary Vol.1, Atlanta: Sagalee Oromoo Publishing, Inc, 1995.

[50] Levine, R.D. and G.M. Green., Studies in Contemporary Phrase Structure Grammar, Cambridge : Cambridge. University Press, 1999.

[51] G. Gragg, Oromo Dicationary: East Lansing, Michigan: Michigan State University, 1982.

[52] Adugna Berkesa, NATOO: Yaadrimee Caasluga Afaan Oromoo, Addis Ababa, 2010.

[53] Yao, Yuan, and Kim Teng Lua, A Probabilistic Context-Free Grammar Parser for Chinese, Singapore: National University of Singapore . Department of Information Systems and Computer Scienc, 1998.

[54] Abreham Teso Nedjo, "Aoutomatic Part-of-speach Tagging for Oromo Laguage Using Maximum Entropy Markov Model (MEMM)," *Jornal of Inforamtion and Computational Science,* vol. 11, no. 10, pp. 3319 - 3334, 1 July 2014.

[55] Owens Jonathan, A Grammar of Harar Oromo (North Eastern Ethiopia), Bumbers : Bumbers. Buske., 1985.

[56] Satta, Mark Jan, Nederhof Giorgio, "Parsing Non-Recursive Context-Free Grammars," in *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL ANNUAL'02)*, Philadelphia, Pennsylvania, USA, 2002.

APPENDICES

APPENDIX A: THE SAMPLE TEXT

1. Biqiltoonnis\NOUNta'an\CONJbineeldonni\NOUN jiraachuuf\ADJ,\COMAnyaata\NOUN isaan\PRON\ barbaachisa\VERB.\PUNCT
2. Xurii \NOUNqaama\NOUNkeessaa\baasuuf, bishaan\NOUNtumsa\VERBguddaa\ADV godha\VERB.\PUNCT
3. Akaakuu\ADJ nyaataa \NOUNqaamaaf \NOUNbarbaachisan\ADJ filachuun\VERB\ fayyaa \NOUNkeenyaaf\PRON gaarii\ADVdha\ADP.\PUNCT
4. Guddinaa \NOUNfi\CONJ jabina\NOUN qaamaa \NOUNargachuuf\ nyaata\NOUN walmadaale \ADJargachuun\ADV dansaa\VERBdha\ADP.\PUNCT
5. Waan arganne hunda nyaachu osoo hin taane,nyaata madaalamaa soorachuutu bu'aa qaba.
6. Ani \PRONkaleesa\NOUN malee\CONJ haar'a \NOUNnyaata\NOUN\hin\nyaane\VERB.\PUNCT
7. Marartun\PROPN gara\ADJ gabaa\NOUN demtee\VERB,\COMA Midhaan\NOUN bitte\VERB.\PUNCT
8. Bulchaan\PROPN\ dhengada\ADV sare\PROPN gamadaa\PROPN ajjesse\VERB.\PUNCT
9. Yoo\ADPdheebotte\NOUN,\COMA bishaan\NOUN Amboo\PROPN dhugi\VERB.\PUNCT
10. Barumsi\NOUN waan \CONJitti\ADP cimeef\ADJ,\COMAaddaan\ADVkute\VERB.\PUNCT
11. Isheen\PRON hoojii\NOUN ishii\PRON waan\CONJ beektuuf\NOUN,\COMAmana\NOUN barumsaa\NOUN iraa\ADP haafte\VERB.\PUNCT
12. Yoo \ADPdhufuu\NOUN baattellee\ADJ,\COMA xalayaa\NOUN naaf\PRON barreessi\VERB.\PUNCT
13. BoonaaN\PROPN biyyaa\NOUN alaatii\ADJ akka\ADP dhufeen\NOUN,\COMA hiriyyoota\NOUN isaaf\PRONdubbii\NOUN godhe\VERB.\PUNCT
14. Yoo\ADP finfinnee \PROPNdeemteef\NOUN,\COMA meeshaa\NOUN naa\PRON bitta\VERB.\PUNCT
15. Bokkaa\NOUN cimaa\ADJ waan\CONJ roobeef\NOUN,\COMA lagni\NOUN guutee\ADJ riqicha\NOUN cabse\VERB.\PUNCT
16. Namni \NOUNkamiyyu\ADJ taanaan\NOUN ,\COM\maqaa\NOUN mataa\ADJ isaa\PRON qaba\VERB.PUNCT
17. Gargaarsi \NOUNmootummaa\COMNOUN duubaan\ NOUNjiraannaan\NOUN umanni \NOUNmisoomaaf\NOUN seexaa\NOUN cimaa \ADJqaba\VERB.\PUNCT

18. Namni\NOUN barate\NOUN tokko\ADV haalaa\NOUN fi \CONJbeekumsa\NOUN isaa\PRON haawwii\NOUN ummataa\COMNwajjin \ADJdeemsisuu \ADVqaba\VERB.\PUNCT
19. Hiyyeessi\NOUN jabaate\ADJ yoo \ADPhojjate\NOUN,\COMA hattuma\NOUN keessaa\ADJ bahuu\ADV ni \ADPdanda'a\VERB.\PUNCT
20. Kaayyoon\NOUN barataa \NOUNtokkoo\ADJ inni\ADP guddaan\ADJ barumsa\NOUN isaatti \PRONjabaatee\ADJ,\COMAbiyya\ isaa \PRONguddisuu\ADVta'uu\ADP qaba\VERB.\PUNCT
21. Maqaa\NOUN moggaassuun\ADV nama\NOUN hin\ADPdhibu\VERB,\COMA akka\ADP hawwan\NOUN sanatt\ADVi bakkaan\NOUN gahuutu\ADV nama\NOUN dhiba\VERB malee\ADP.\PUNCT
22. Gaafa\ miilkiin\NOUN sif \PRONhin\ADP tolle\ADJ har'a \NOUNcaraa\NOUN gaarii \ADJmiti\ ADPjetta\VERB.\PUNCT
23. Qorumsi\NOUN ga'e \ADJjedhanii\NOUN batattisuu\NOUN irra\ADP laf-jala\NOUN itti\ADPqophaa'uutu\ADV gaarii\VERBdha\AUX.\PUNCT
24. Namni\NOUN abdii\NOUN fi\CONJ akeeka\NOUN qabu\ADV tattaaffii \NOUNcimaa\ADV godhee\NOUN bakka\NOUN yaadee\ADV hin\ADP hanqatu\VERB.\PUNCT
25. Mucaan \NOUNobbo\ADP magarsaa\ PROPNeega\CONJ kashlabbaayee\NOUN manabarumsaa\NOUN hafee\NOUN,\COMA ganda\NOUN keessa\ADJ jooraa\NOUN oola\ADV ture\VERB.\PUNCT
26. Eda\NOUN robaa\NOUN waan\CONJ buleef\NOUN,\COMA lagi\NOUN caanco\PROPN guteetu\ADV danbali'e\VERB.\PUNCT
27. Galataan\PROPN mana\NOUN ajjeeruu\NOUNdhaaf\ADP bisii\NOUN maraa\ADVjira\VERB.\PUNCT
28. Gadaan\NOUN baranaa\NOUN horata\NOUN moo\CONJ Birmajjii\NOUNdha\VERB?\QUESTION MARK
29. Yemmu\ADPwaraabesi\NOUN yuusu\ADV,\COMAsareen\NOUN dute\VERB.\PUNCT
30. Gamadaan\PROPN kaleessa\ADV sangaa\NOUN gurgure\VERB.\PUNCT
31. Yemmun \ADP ani \PRONxule\NOUN gahu\NOUN,\COMABantiin\PROPN achii\ADV hin\ADP jiru\VERB.\PUNCT
32. Margeen \PROPNyeroo\ADVhunda\ADV mana\NOUN barumsaa\NOUN deemti\VERB.\PUNCT
33. Biyya\NOUN misoomsuuf\ADJ tokkummaan\ADV haa\ADP kaanu\VERB.\PUNCT
34. Yemmuu\ADP deemtu\NOUN,\COMA na\PRON dubbisii\ADV darbii\VERB.\PUNCT
35. Dhagaan\PROPN gaara\NOUN sana\ADJ iraa\ADP kokolaatee\NOUN,\COMA Farda\NOUN ajeese\VERB.\PUNT
36. Situ\PRON na \PRONarge\VERB malee\CONJ ani\PRON\ si\PRON hin \ADPagarre\VERB.\PUNCT

37. Jaartiin\NOUN mataa\NOUN arrii\ADJ,\COMA nama\NOUN walitti\ADV naqaxe\VERB.\PUNCT
38. Huccuu \NOUNhaphii\ADJ uffatee\NOUN,\COMA qorra\NOUN keessa \ADJciisa\VERB.\PUNCT
39. Otto\ADP loon\NOUN hin \ADPgalin\NOUN,\COMA hattun\NOUN mooraa\ADV guutte\VERB.\PUNCT
40. Erga\ biftuun\NOUN lixxee\AD ,\COMAbokaan\NOUNrobe\VERB.\PUNCT

Appendex B

Sample Output

Sent: Toolaan akka ishee jaalatee Haawwiin siritti bektee

Parser: <ViterbiParser for <Grammar with 21 productions>>

Grammar: Grammar with 21 productions (start state = CS)

CS -> MC SC [1.0]

MC -> NP PP VP [1.0]

SC -> NP VP [1.0]

VP -> P PR V [0.4]

VP -> PR V [0.4]

VP -> ADV V [0.2]

NP -> N [1.0]

PP -> P [1.0]

V -> 'jaalatee' [0.4]

V -> 'demee' [0.12]

V -> 'bektee' [0.48]

N -> 'Toolaan' [0.45]

N -> 'Haawwiin' [0.4]

N -> 'dinagdee' [0.15]

ADV -> ADV ADV [0.2]

ADV -> ADV [0.3]

ADV -> 'siritti' [0.5]

PR -> PR [0.4]

PR -> 'ishee' [0.6]

P -> P [0.5]

P -> 'akka' [0.5]

Inserting tokens into the most likely constituents table...

Insert: |=.....| Toolaan

Insert: |.=.....| akka

```
   Insert: |..=....| ishee
   Insert: |...=...| jaalatee
   Insert: |....=..| Haawwiin
   Insert: |.....=.| siritti
   Insert: |......=| bektee
Finding the most likely constituents spanning 1 text elements...
   Insert: |=......| N -> 'Toolaan' [0.45]          0.4500000000
   Insert: |=......| NP -> N [1.0]                  0.4500000000
   Insert: |.=.....| P -> 'akka' [0.5]              0.5000000000
   Insert: |.=.....| PP -> P [1.0]                 0.5000000000
  Discard: |.=.....| P -> P [0.5]                  0.2500000000
  Discard: |.=.....| P -> P [0.5]                  0.2500000000
   Insert: |..=....| PR -> 'ishee' [0.6]            0.6000000000
  Discard: |..=....| PR -> PR [0.4]                 0.2400000000
   Insert: |...=...| V -> 'jaalatee' [0.4]          0.4000000000
   Insert: |....=..| N -> 'Haawwiin' [0.4]          0.4000000000
   Insert: |....=..| NP -> N [1.0]                  0.4000000000
   Insert: |.....=.| ADV -> 'siritti' [0.5]         0.5000000000
  Discard: |.....=.| ADV -> ADV [0.3]               0.1500000000
   Insert: |......=| V -> 'bektee' [0.48]           0.4800000000
Finding the most likely constituents spanning 2 text elements...
   Insert: |..==...| VP -> PR V [0.4]               0.0960000000
   Insert: |.....==| VP -> ADV V [0.2]              0.0480000000
Finding the most likely constituents spanning 3 text elements...
   Insert: |.===...| VP -> P PR V [0.4]             0.0480000000
   Insert: |....===| SC -> NP VP [1.0]              0.0192000000
Finding the most likely constituents spanning 4 text elements...
   Insert: |====...| MC -> NP PP VP [1.0]          0.0216000000
   Insert: |====...| SC -> NP VP [1.0]             0.0216000000
Finding the most likely constituents spanning 5 text elements...
Finding the most likely constituents spanning 6 text elements...
```

Finding the most likely constituents spanning 7 text elements...

Insert: |=======| CS -> MC SC [1.0] 0.0004147200

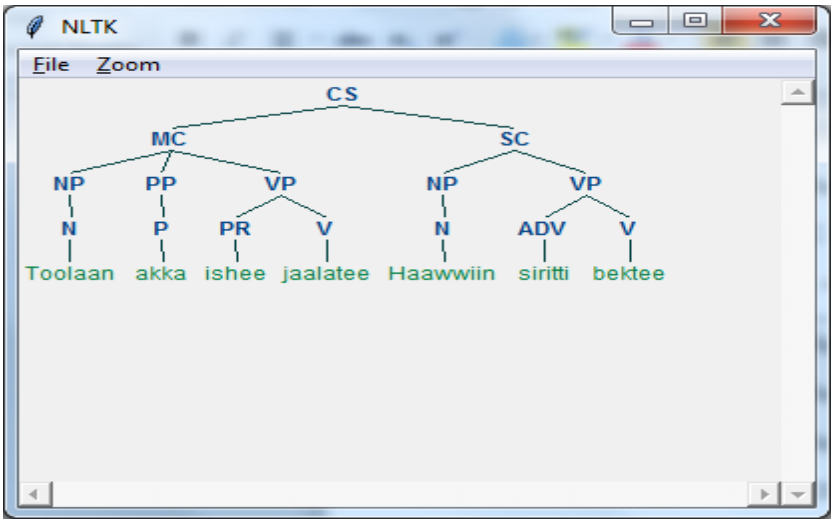
Time (secs) # Parses Average P(parse)

0.5830 1 0.00041472000000

---

n/a 1 0.00041472000000

Draw parses (y/n)? y



CS =complex sentence

MC= Main Clause

SC=Sub Clause

Print parses (y/n)? y

(CS

(MC

(NP (N Toolaan))

(PP (P akka))

(VP (PR ishee) (V jaalatee)))

(SC (NP (N Haawwiin)) (VP (ADV siritti) (V bektee)))) [0.00041472000000000004]