



Addis Ababa University
College of Technology and Built Environment
Dep't of Computer Engineering
Addis Ababa University

A Top- Down Chart Parser for Ge'ez Sentences

Nega Meareg W/gebriael

A Thesis Submitted to the College of Technology and Built Environment

Dep't of Computer Engineering

Addis Ababa Ethiopia

December 2024

Addis Ababa University
College of Technology and built Environment
Dep't of Computer Engineering

Nega Meareg W/gebriael

Surafel Lemma Abebe (PhD)

This is to certify that the thesis prepared by Nega Meareg W/gereal, titled: “A Top-Down Parser for Geez language” and submitted in partial fulfillment of the requirements for the Degree of Master of Computer Engineering complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

<u>Name</u>	<u>Signature</u>	<u>Date</u>
1. _____	_____	_____
2. _____	_____	_____

Signed by Advisor

1. _____

Dedication

To my beloved father, “Megabe Tibebat Meareg”, who’s natural love and support have been my backbone through the most challenging times? This work is a testament to resilience and perseverance, completed despite the mental health struggles I faced during my studies, the devastation of the war in Tigray, and the disruptions caused by the COVID-19 pandemic.

Acknowledgement

First and most I would like to express my deepest respect to my advisor, Dr. Surafel Lemma, for his valuable guidance, on-time feedback, patience, throughout the years we passed. His expertise and encouragement have been instrumental in shaping this work and ensuring its completion.

I am also deeply thankful to my friends, whose continuous fellowship and motivation kept me through difficult times, and to the dedicated staff who provided the resources and support for this work. Each of you has played an important role in making this accomplishment possible, and I am forever grateful for your contribution.

Contents

List of Figure	i
List of Tables	ii
Acronyms	iii
Abstract	iv
Chapter one.....	1
1.1: Introduction.....	1
1.2 Motivation	2
1.3 Statement of the problem	2
1.4 Objective	3
1.4.1 Specific objectives	4
1.5 Methodology	4
1.5.1 Literature Review	4
1.5.2 Data Collection	5
1.5.3 Design and Development.....	5
1.6 Scope	5
1.7 Application of Results	5
1.8 Organization of the Thesis	6
Chapter Two: Literature Review.....	7
2.1 Introduction.....	7
2.2.1 Geez Language.....	7
2.2 Geez Language Word classes	9
2.2.1 Grammar (ሰዋስጡ) structure of Geez language	9
2.2.2 Major parts of speeches	10
2.2.2.2 Adjectives (ቅፅል)	11
2.2.2.3 Verb (ግስ)	12
2.2.2.4 Adverbs	13
2.2.2.5 The Stems of verbs.....	13
2.2.3 Minor parts of speech	13
2.2.3.1 Pronoun (መረሕያን/ተውላጠ ስም)	13

2.2.3.2 Demonstrative pronoun	14
2.2.3.3 Possessive pronouns.....	14
2.2.3.4 Conjunctions	15
2.2.3.5 Punctuation marks.....	15
2.3 Geez language phrases	15
2.3 .1 (Noun phrases) NP ምስረታ ስማዊ ሓረግ.....	15
2.3.2 Verb Phrases VP (ምስረታ ግሳዊ ሓረግ)	16
2.3.3 Sentences formulation in Geez language	16
2.4 Grammar formalism.....	19
2.4.1 Context Free Grammar.....	20
2.4.2 Transition Network Grammar	20
2.4.3 Augmented Transition Network	21
2.4.4 Context Sensitive Grammar	21
2.4.5 Probabilistic Context Free Grammar	22
2.4.6 Unification based Grammar	23
2.5 parsing.....	23
2.5.1 Approaches of Sentences Parsing	24
2.5.2. Unsupervised approach	25
2.5.3 Rule-Based approaches	25
Summary.....	26
Chapter Three: Related Work.....	27
3.1 Introductions	27
3.1.1 Amharic language parsing	27
3.1.2 Afan Oromo Language Parsing	28
3.1.3 Tigrigna Language parsing	29
3.1.4 Arabic Language parsing.....	29
Summary	30
Chapter Four: - Design and Implemetnaion of Geez language Parser.....	31
4.1 introductions	31
4.2 System Architecture	31

4.3 Components of the system Architecture	32
4.3.1 The grammar rule	32
4.3.2 Tokenization Process	33
4.3.3 Lexicon	33
4.3.4 Word Classifier	33
4.3.5 Input sentences	33
4.3.6 Parsed Sentences	34
4.3.7 The parser function	34
4.4 Top-Down Parsing Algorithm (psedocod)	34
4.5 Mathematical Formulation	35
4.6 Data collection	35
4.7 System Imlementation	36
4.7.1 System Specification	36
4.8 Experimental Result and Discussion.....	36
4.9 Summary	38
Chapter Five.....	39
5. Conclusion and Recommendation	39
References	40
Appendix	44
Parsed Sentences	56

List of Figure

Figure2. 1: Geez Script arrangement (a) earlier Geez script (b) recent Geez script (C) Derived Ge'ez script	8
Figure2. 2: Geez language numbering representation	8
Figure2. 3: Similar words and letters	9
Figure2. 4: Root or main verbs in Geez language	12
Figure2. 5: sentence construction of geez language	17
Figure2. 6: Example 2 Sentence construction of geez language	18
Figure2. 7: Example 3 Sentence construction of geez language	18
Figure2. 8: Example 4Sentence construction of geez language	18
Figure2. 9: Example 5 Sentence construction of geez language	19
Figure2. 10: Augmented Transition Network.....	21
<i>Figure4.1: Architecture design of the parser</i>	<i>31</i>
Figure5.1: The Chart Parser Function.....	38
Figure5. 3: Parsed Sentences	56
Figure5. 4: Example of the parsed sentence	57
Figure5. 5: Sample sentence	57
Figure5. 6:Tree representation of the parsed sentences	57
Figure5. 7Sample Parsed sentence.....	58
Figure5. 8Tree representation of the parser	58
Figure5. 9Sample parsed sentence	58
Figure5. 10: Tree representation of parsed sentence	59
Figure5. 11: Tree representation of the parsed sentence.....	59
Figure5. 12: Examples of some parsed sentence	59
Figure5. 13: Examples some parsed sentence.....	59
Figure5. 14: Tree representation of the parser	60
Figure5. 15: Some examples of parsed sentence	60

List of Tables

Table2. 1: Example of Inflections in Geez numerals.....	11
Table2. 2: Geez language adjectives.....	12
Table2. 3: Geez pronouns	14
Table2. 4: possessive pronouns in Geez language.....	14
Table5. 2: Parsed sentences by the parser.....	37
Table5. 3: Parsed sentences by the language expert.....	37

Acronyms

NLP- Natural Language processing

NL- Natural Language

NLU-Natural Language Understanding

NER- Named Entity Recognition

NP- Noun Phrase

VP- Verb Phrase

CFG- Context Free Grammar

TNG- Transition Network Grammar

PCFG- Probabilistic Context Free Grammar

SVO-Subject Verb Object

VSO- Verb Subject Object

OVS- Object Verb Subject

ADJ- Adjectives

N- Nouns

S- Sentences

Abstract

Parsing is the process of breaking down a sentence into its part of speech or words like verb, noun, preposition, adjective and so on. Parsing plays an important role in enhancing the performance of numerous natural languages processing (NLP). Here this work is designed to parse geez language on the approach of a Top-Down chart Parser for Geez language sentences using the context free grammar rule (CFG).

We reviewed various parsing approaches for different languages, to achieve our objective. However, Geez language parsing remains challenging due to the absence of annotated dataset. To address this gap, we collected a dataset for sentence parsing from the well-known book *Mezumere Dawit*. Given the lack of the lack of pre-existing labeled data; we collaborated with a language expert (Amanuel), an instructor in the Geez Department at Aksum University, to ensure linguistic accuracy. The expert prepared the dataset in a format suitable for established grammatical rules for sentence construction based on verb and noun phrase structures, and manually parsed the sentences.

This study presents a Top-Down parsing approach for Geez language sentences; addressing the challenges posed by the language's unique morphological and syntactic structures. Using a dataset of 500 sourced from the book '*Mezumere Dawit*' the parser correctly parsed 470 sentences, achieving a parsing accuracy of 94%. The results were validated through the manual parsing, where 490 sentences were parsed manually, with 470 sentences matching the parser's output. The performance of the parser was evaluated using standard metrics, including precision, recall, F1 score, and Experimental results show the effectiveness of the proposed method in parsing Geez language sentences. This study contributes a foundational step towards computational processing of Geez language, with potential applications in machine translation and historical analysis.

Key words: - CFG (Context Free Grammar), NLP (Natural language Processing), Top-Down Parser.

Chapter one

1.1: Introduction

Natural language (NL) is a language that humans interact for their daily activities, such languages are English, French, Arabic, Amharic, Ge'ez, and many others. These languages have their own complexity, grammar, syntax, Semantics, and scripts to give meaningful sentences. Natural language understanding (NLU) is a method of understanding natural languages in a machine readability form, process it, and understand its semantics (meaning), syntax (structure), and pragmatics (context).[1].

Natural language processing (NLP) is part of computer science and Artificial Intelligence (AI) concerned with enabling computers to understand, interpret, process, and generate human language in a meaningful way. Natural language Processing (NLP) addresses the gap between human communications and machine understanding. There are different applications or tasks of Natural language Processing (NLP): some of them are language modeling, speech recognition, Named Entity Recognition (NER), Machine Translations, Part of Speech Tagging, and Parsing, Text to Speech Conversion. [2]

Parsing is the process of analyzing or identifying the structure of a sentence or piece of text into a well-defined grammatical structure of a sentence or a text. Top-Down parsing is a technique of parsing that constructs a parse tree from the root (sentence) down into the leaves (words) of a sentence. [2], [3]

Geez language is not widely spoken in Ethiopia and Eritrea today. Instead, it is a liturgical language primarily used in religious contexts by the Ethiopian Orthodox Tewahco, Eritrean Tewahdo Church, and Related communities. However, Tigrigna, Amharic, and Tigre which evolved from Geez are widely spoken in Ethiopia and Eritrea. It has its own script, number, and writing system. The modern language of Ethiopia Amharic and Tigrigna are derived from this language. Ones in a while, it serves as the official language of the Aksumite kingdom. Now a day its service is limited in the religious area especially both in the Ethiopian and Eritrea Orthodox church. Some local and international higher institutions are also used for academic purposes, among them are Addis Ababa university, Bahrdar university, Debremarkos university,

Aksum university, Kidiste Silasie Cathedral university, and there are such universities abroad in Germany, United states of America, Canada included this language on their academic programs. There are also online services that provide educational materials and explore Geez language. The Ethiopian Orthodox Twahdo Church also has prominent instructors that teach students specialized in teaching programs in Gonder and Axum. [4], [5].

1.2 Motivation

It is known that Ge;ez language is the source of culture, arts, scripts, heritages, religious and ethical values of Ethiopians and Eritreans [20].The Scripts are the origin of sematic languages of Ethiopian languages such as , Amharic, Tigrigna, and Tigre from Eritrea. It is a source for the religious, arts, culture, scripts, heritages, many books and arts are written in Geez language. [20].

While I was thinking to do my thesis, I am trying to investigate different works; I noticed that the work of Amharic Sentences parsing (Abdurahman), recommended that his work should be further investigated on Ge'ez language, because it is an ancient and a basement for other sematic languages like Amharic, Tigrigna and Tigre spoken in Eritrea. Besides that, there are no such works on, and there is no more such work is available on Natural language Processing (NLP)

1.3 Statement of the problem

Parsing is a fundamental task of natural language processing (NLP), enabling machines to analyze and understand sentences structures. While Extensive research has been conducted on parsing thechiques for widely spoken languages such as [3], [6], [7], [7], [8], [9], [10], [11], [12], [13,], similarly there are also some works for Geez language on Natural Lnaguage Processing (NLP) ,[14], [15], [16], [17], [18], [19], [20].

Despite the rich lingusic heritage of the Geez language, there has been limited research on parsng approaches tailored to its unique grammar. This research gap highlights the need for effective parsing thechniques that can handle the language's morphological and syntactic complexities.among various thecniques; this study specifically adopts the Top-Down paring for Geez language sentnces. The selection of Top-Down parsing is justified by its suitability for languages with well defined and hierarchical grammatical structrues, such as Geez language. Top-Down parsing, which begins from the start symbol and precedes by by expanding

production rules, aligns well with the rule-based nature of Geez grammar. Additionally, it allows for intuitive grammar rule design and easier error detection during parsing.

While modern languages such as Amharic and Tigrinya have evolved from Geez, the study of Geez itself remains critical for both linguistic scholars and those working in the field of natural language processing (NLP). Despite its relevance, geez has not received the same degree of computational attention as other languages, particularly in sentence parsing. Parsing, a fundamental task in NLP, involves analyzing the structure of a sentence and determining the syntactic relationships between its components.

Top-down parsing, a technique widely used in syntactic analysis, constructs a parse tree from the root (sentence) down to the leaves (words or word-like components). This method has been successfully applied to many languages, but its application to Geez presents unique challenges due to the language's distinct syntactic and morphological properties. For instance, geez features a rich system of inflection, with words undergoing various modifications based on tense, aspect, and agreement. Additionally, geez syntax is relatively flexible, making it prone to ambiguities that must be addressed in the parsing process.

This thesis aims to explore the use of top-down parsing for Geez, focusing on the development of a parsing model that accommodates the language's morphological richness and syntactic flexibility. Through this research, we seek to address the limitations of current parsing methods for Geez and contribute to the development of computational tools for this ancient language. The primary research problem addressed by this thesis is the lack of effective parsing methods for the Geez language, particularly in the context of top-down parsing. Some unique challenges of the language are morphological complexity, Flexible word order and limited resources.

1.4 Objective

The general objective of this study is to design and implement a top-down chart parser for Ge'ez language sentences.

1.4.1 Specific objectives

- To investigate and document the unique syntactic, morphological, and orthographic properties of Geez language, including its inflectional patterns, word-order flexibility, and the nature of its syllabic script.
- To study different types of grammar rules, and formalism and select the one that handle for Geez language sentences parsing
- To construct accurate and annotated dataset of Geez language sentences sourced from classical texts (e.g Mezumure Dawit).
- Collaborate with language experts (such as instructors from Aksum University's Geez department) to validate grammatical rules, refine parsing strategies, and ensure the linguistic accuracy of the parser.
- To assess the parser using standard metrics such as accuracy, precision, recall, and F1-score to quantify its effectiveness and identify areas for improvement.

1.5 Methodology

The methodology follows for the research encompasses a comprehensive approach to the development of a Top-Down Chart Parser the Geez language. It includes key steps such as clearly defining the research approach for Top-Down Parsing methodology used for analyzing Geez language sentences, explaining why this approach is chosen over alternatives, like the Bottom-up or statistical parsing methods. The other point is describing how the grammar rule formalism of the Geez language was constructed. Beyond these, to look at the appropriate tools and techniques for parsing Geez language sentences and set evaluation metrics for the parsing process.

1.5.1 Literature Review

The literature review explores existing works of different types of parsing methodologies, techniques, and strategies. It examines linguistic studies on the Geez language, discusses challenges and strategies for applying NLP techniques to languages with limited corpora like Geez language, and exploring its unique grammatical structures and challenges for the studying of the parsing Geez language. In addition, the literature review looks at state-of-the-art parsing

techniques applied to other sematic languages, and high light how these approaches are applied in under resourced language.

1.5.2 Data Collection

The dataset for this study will be collected from the book known as“Mezumeur Dawit”, this is a very well-known book having diversified sentences that can represent the parsing process for the language. We will consider the language experts input for the research, because the rules and the tagging process will be conducted based on their knowledge.

1.5.3 Design and Development

The design and development method are focused on the architecture design and the development process of the parser. Grammar design is the main part of the components along with other modules.

1.6 Scope

This research focuses on the development and evaluation of a top-down parsing system for analyzing sentence structures in the classical Geez language. The study is limited to the syntactic level of natural language processing, emphasizing the formalization and application of grammar rules for Geez.

1.7 Application of Results

The application result of the research listed below

- Digital preservation and formalization of Geez grammar, ensuring its accessibility for future generations.
- The parsing system can be integrated into digital platforms for teaching Geez to students, clergy, or linguists.
- The parsing results and annotated data can serve as training data for AI models, advancing research in Geez NLP.
- The methodology can be adapted to other under-resourced languages, providing a framework for computational analysis.

1.8 Organization of the Thesis

The thesis is organized as follows; chapter two discusses literature review in the field of natural language processing, description of the Geez language, grammar formalism, and parsing approaches and strategies. Chapter three discusses the relevant work of parsing for both Ethiopian and non-Ethiopian languages. Chapter four covers the system design; it provides the detailed coverage of each component of the system. On chapter five the implementation of the designed system, it includes details of the development process, experimentation, and results obtained during testing. The final chapter six discusses the conclusion and future work.

Chapter Two: Literature Review

2.1 Introduction

The literature review gives a detailed analysis of existing research and theories that inform current studies. It explores the fundamental concepts of natural language parsing, with a particular focus on rule-based approaches suitable for under-resourced languages such as Geez. The chapter begins with an overview of the Geez language and its linguistic characteristics, emphasizing its syntactic and morphological complexities

2.2.1 Geez Language

Geez language is an ancient Ethiopia and Eritrea language; it has its own writing system, syntax, and structure and grammar rule. There are different types of writing systems, these are Alphabets (English, Russian, Greek), Abjads (Arabic, Hebrew), Abugidas or alpha syllabaries (Devnagari, Thai, geez, Amharic), Featural alphabets (Hangal), syllabaries (Japanese, Cherokee), and Logographic system (Chinese character) [22]. Among all these writing systems geez language uses abjad and abugida. The word order or structure of Geez language follows **SVO**, **VSO** and **OVS** (subject verb and object, verb, subject and object, object, verb and subject). At the beginning the writing system was from right to left but later changed from left to right. There are 26 basic letters that represent the syntax of the language. [23].

	ግዕዝ	ካዕብ	ሣልስ	ራብዕ	ሐምስ	ሳድስ	ሳብዕ		ግዕዝ	ካዕብ	ሣልስ	ራብዕ	ሐምስ	ሳድስ	ሳብዕ
ጀ	አ	ኡ	ሊ	ላ	ሌ	ለ	ሐ	ጀ	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ጁ	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ	ጁ	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ጂ	ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጘ	ጂ	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
ጃ	ደ	ዱ	ዲ	ዳ	ዴ	ድ	ደ	ጃ	መ	ሙ	ሚ	ማ	ሜ	ሞ	ሠ
ጄ	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ	ጄ	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
ጅ	ወ	ዉ	ደ	ደ	ደ	ወ	ደ	ጅ	ረ	ሩ	ሪ	ራ	ሪ	ሪ	ሪ
ጆ	ዘ	ዛ	ዘ	ዛ	ዘ	ዘ	ዘ	ጆ	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
ጇ	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ	ጇ	ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
ገ	ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ	ገ	ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
ገ	የ	ዩ	ዪ	ያ	ዮ	ይ	የ	ገ	ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
ገ	ከ	ከ	ከ	ከ	ከ	ከ	ከ	ገ	ነ	ኑ	ኒ	ና	ኑ	ን	ኖ
ገ	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ	ገ	አ	ኡ	ሊ	ላ	ሌ	ለ	ሐ
ገ	መ	ሙ	ሚ	ማ	ሜ	ሞ	ሠ	ገ	ከ	ከ	ከ	ከ	ከ	ከ	ከ
ገ	ነ	ኑ	ኒ	ና	ኑ	ን	ኖ	ገ	ወ	ዉ	ደ	ደ	ደ	ወ	ደ
ገ	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ	ገ	ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
ገ	ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ	ገ	ዘ	ዛ	ዘ	ዛ	ዘ	ዘ	ዘ
ገ	ፈ	ቀ	ፈ	ቀ	ፈ	ፍ	ፎ	ገ	የ	ዩ	ዪ	ያ	ዮ	ይ	የ
ገ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ	ገ	ደ	ዱ	ዲ	ዳ	ዴ	ድ	ደ
ገ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ገ	ገ	ገ	ገ	ገ	ገ	ገ	ገ
ገ	ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ገ	ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
ገ	ረ	ሩ	ሪ	ራ	ሪ	ሪ	ሪ	ገ	አ	ኡ	ሊ	ላ	ሌ	ለ	ሐ
ገ	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ	ገ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ	ጸ
ገ	ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ	ገ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ	ፀ
ገ	አ	ኡ	ሊ	ላ	ሌ	ለ	ሐ	ገ	ፈ	ቀ	ፈ	ቀ	ፈ	ፍ	ፎ
ገ	ጥ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ	ገ	ጥ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ

(a) (b)

ሷ	ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ	ሿ	ፈ	ቀ	ፈ	ቀ	ፈ	ፍ	ፎ
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(c)

Figure2. 1: Geez Script arrangement (a) earlier Geez script (b) recent Geez script (C) Derived Ge'ez script

Geez language also has its own number system; and they are used in the Ethiopian Calendar. The figure below shows the representation of the number system of the language.

-	ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ	ጇ	ገ	ገ
አልቦ	አሐዳ	ከልሌቀ	ሠላሰቀ	አርባዕቀ	ሐምሰቀ	ሰድስቀ	ስብዓቀ	ስመንቀ	ተሰዓቀ	አሠርቀ
0	1	2	3	4	5	6	7	8	9	10
ጁ	ፀ	ፁ	ፊ	ፋ	ፈ	ፈ	ፈ	ፈ	ፈፈ	
20	30	40	50	60	70	80	90	100	1000	
አሰራ	ሠላላ	አርባዓ	ሃምሳ	ሰድሳ	ስብዓ	ስመንያ	ተሰዓ	ምዕት	አልፍ	
ፈፈፈ	ፈፈፈፈ									
አላላት	ትልልታት									
1000 000	100 000 000									

Figure2. 2: Geez language numbering representation

There are also similar sound letters in Geez language, their sounds are the same but, they are differing in their shape. The figure below shows their symbols and the reason why they differ from one another. [18]

Letters	Known Name	Reason
ሀ	ሃሌታው "ሀ"	Since it is the beginning of the Ge'ez word ሃሌ
ሐ	ሐመሩ "ሐ"	Since it is the beginning of the Ge'ez word ሐመር
ኀ	ብዙኃት "ኀ"	When the word ብዙኃት written it is used.
ሰ	አሳቱ "ሰ"	When the Ge'ez word አሳት written it is the one used.
ሠ	ንጉሡ "ሠ"	When the Ge'ez word ንጉሡ written it is the one used.
አ	አልፋው "አ"	The word አልፋ is always written using it
ዐ	ዐይን "ዐ"	The shape is like Eye and the Ge'ez word ዐይን is written using it
ጸ	ጸሎት "ጸ"	The Ge'ez word ጸሎት written it is the one used.
ፀ	ፀሐይ "ፀ"	The shape is like sun and used to write the Ge'ez word ፀሐይ

Figure2. 3: Similar words and letters

2.2 Geez Language Word classes

2.2.1 Grammar (ሰዋስው) structure of Geez language

Grammar (ሰዋስው) is the structure of the language that governs the clauses, phrases and words of any language. [24] [25]. Words are the basic part of any language and in Geez also. There are a lot of words that combine and give meaning to full sentences. Each word that we use has its own part of speech, in Geez there are eight parts of speech, these are Nouns/ስም/, verbs/ግስ/, adjectives/ቅፅል/, adverbs/ተውሳክግስ/, pronoun/መራሕያን/, preposition/መስተዋድድ/, conjunction/መስተጻምር/, and interjection/ቃለ አጋኖ/.

Many grammar texts also define parts of speech in terms of form and structure classes.[24]. The form structure is organized in terms of the main part of speeches Noun/ስም/, Verbs/ግስ/, adjectives/ቅፅል/ and adverbs/ተውሳክ ግስ/, these are the words that give the meaning a full sentences. The structure classes are organized in the form of pronouns/መራሕያን/, preposition/መስተዋድድ/, conjunction/መስተጻምር/, and interjection/ቃለ አጋኖ/, these words are used to indicate grammatical relationship and structure of There are also open and closed word classes, [24] the open word classes are the major classes constructed from the major parts of speech Noun, verbs, prepositions and adverbs and these are the content words that give the

lexical or the content meaning of the sentences. The other category is the closed word classes which are composed of the pronouns, prepositions, conjunctions and interjections which the relationships of the sentences. Therefore, the form classes are called open word classes, because new words are entering constantly into the language.

2.2.2 Major parts of speeches

2.2.2.1 Nouns/ስም/

A noun is the name that represents a person, place, animals, thing, feeling and idea. [24] In Geez there are different types of nouns in general concert and abstract, common and proper, collective and countable nouns. In Geez language most nouns end with the sixth letter/ sades fidel/, but there are also nouns ends with other letters.

In general, in Geez language there are two ways of forming plural form of a noun

1. Pattern replacement: ደብር dabr.....አድባር..ad bar
2. Addition of an ending: አመት.....አመታት

Plurals formed by adding pattern replacement are known as “Broken” plurals or “Internal” Plurals and those that are formed by addition an ending are known as “External” plurals.[24].in Geez language we use አ፣አ.....ት፣ል፣ን፣ው፣ት to inflect a singular noun to plural.

Table2. 1: Example of Inflections in Geez numerals

Geez language Nouns		
Adding	Root word	Inflected
አ	ልብ	አልባብ
አ.....ት	ባሕር	አብሕርት
	ገብር	አገብርት
	ጥብ	አጥብት
	ነቅፅ	አንቅፅት
ት	ገዳም	ገዳማት
	እም	እማት
ል	ኪሩብ	ኩሩቤል
	ሱራፊ	ሱራፊል
ን	ጸድቅ	ጸድቃን
ው	እኑ	አኅው
	አብ	አበው

In Geez language plural formations can be occur by changing the letter/fidel/ to fourth /ራብዕ/ and sixth/ሳድስ/ letters,for example ደብተራ to ደባትር ፣መድሎት to መዳልው:- we can also form plural noun at the end of the noun is the letter/fidel/ third/ሣልስ/,fifth/ኅምስ/ and seventh/ሳብዕ/ using ወ and የ as fourth/ራብዕነት/ ዊ and ያ with ት and ን. For example, ደዌ to ደዌያት፣አረጋዊ to አረጋዊያን፣ዘማሪ to ዘማርያን።

2.2.2.2 Adjectives (ቅዕል)

Adjectives are words that describe, identify and define a noun or pronoun. [24][25]. nouns describe the nature of things, but adjectives describe the behavior or characteristics like shape, size, color and type of things.in Geez language adjectives are inflected to form plural form by prefixing “አለ፣አ” at the beginning and suffixing ን፣ዊ፣ይ፣ያ፣ት፣ሙ፣ው..... at the end. Some examples of adjectives are shown below.

Table2. 2: Geez language adjectives

Geez language adjectives			
Singular	Plural	Prefix	Suffix
ፍንወ	ፍንዎን	ን
ሰማያዊ	ሰማያውያን	ያን
መኑ	እለመኑ	እለ.....	
ውእቱ	ውእቶሙ		...ሙ
አብ	አበው		...ው

2.2.2.3 Verb (ግስ)

A verb is a word that is used to describe the action, state and forming the main part of the predicate sentences. [25][25]. Geez language verbs can be in a perfect form (past form) or imperfect form (present and future). Perfect verbs indicate a completed action; these actions may be past perfect, past continuous and past participle with relative pronoun of (H). The imperfect verbs include present, continuous and future actions. All perfect verbs end with first order and the imperfect verbs end with the sixth order when the noun is he (ውእቱ) [24]. The morphology of verbs starts with perfect verbs. There are also rules to change the imperfect verbs which are expressed by the root verbs. Root verbs (ግስ አርእስት) are regular verbs which describe the time behavior of an event. They can be 7 or 8, and each have their own characteristics.[15] [18].

ተራ ቁጥር	የግለዝ ግስ አርእስት	የፈደል ደረጃ	Pronunciation (አነባቡብ)
1	ቀተለ	(ቀ)ግለዝ (ተ)ግለዝ (ለ)ግለዝ	/kətələ/
2	ቀደሰ	ግለዝ ግለዝ ግለዝ	/kəddəsə/
3	ገባረ	ግለዝ ሳድስ ግለዝ	/gəbirə/
4	አለመረ	ግለዝ ሳድስ ግለዝ ግለዝ	/ʔəʔməra/
5	ባረከ	ራብዕ ግለዝ ግለዝ	/barəkə/
6	ሚመ	ጎምስ ግለዝ	/semə/
7	ብከለ	ሳድስ ሳድስ ግለዝ	/bihilə/
8	ቆመ	ሳብዕ ግለዝ	/komə/

Figure2. 4: Root or main verbs in Geez language

2.2.2.4 Adverbs

A word that describes the characteristics or qualities of a verb is referred to as an adverb. In Geez, adverbs can place either before or after the verb they modify. [24]

2.2.2.5 The Stems of verbs

Ethiopian scholars refer [5] [25] refer stems as **አዕማደ** /ʔəʕimad/ meaning pillars shafts that uphold the roof of a building. Stems serve as the foundation structures of verbs, supporting their conjugations. According to the scholars [5] [24] geez has five distinct patterns each functioning independently.

Perfective stems.....**ገቢር**

Causative stems.....**አገብር**

Causative-reciprocal stems.....**አስተጋብር**

Reflective stems.....**ተገብር**

Reciprocal Stems.....**ተጋብር**

2.2.3 Minor parts of speech

2.2.3.1 Pronoun (**መራሕያን/ተውላጠ ስም**)

A pronoun is a word that replaces a noun or noun phrase. Unlike any other languages, geez has 10 pronouns [24] [26], Amharic has 9 pronouns [25], English has 7 pronouns [27]. In geez pronouns can serve as substitutes for nouns, verbs like “to be” and adjectives. There are different types of pronouns such as personal, reflexive, relative, reciprocal, demonstrative, interrogative, indefinite, and possessive pronouns. In Ge’ez pronouns /**መራሕያን**/ can be classified in gender or sex and numbers like singular and plural. These are shown in the table below

Table2. 3: Geez pronouns

	Pronouns/መራሕያን		ፆታ/Sex			Plural /ቡዙሕ	Singular /ነፀላ/
	ግእዝ	English	Masculine/ ተባዕታይ	Feminine /ተንስታይ/	Common /የወል/		
1 st person	አነ	I			✓		✓
	ንሕነ	We			✓	✓	
2 nd person	አንተ	You	✓				✓
	አንቲ			✓			✓
	አንትሙ		✓			✓	
	አንትን			✓		✓	
3 rd person	ውእቱ	He/it	✓				✓
	ይእቲ	She/it		✓			✓
	ውእቶሙ	They	✓			✓	
	ውእቶን			✓		✓	

2.2.3.2 Demonstrative pronoun

A demonstrative pronoun replaces a person, place or thing and can act as a subject, an object, or the object of a preposition. Some examples of demonstrative pronoun in geez are, ዝንቱ/ዝ/, ዛቲ/ዛ/, እሎ/እሎ/እሎንቱ, እሎን፣እላ፣እላንቱ, ዝኩ፣ ውእቱ ዝኩቱ፣ ዝስኩ, ይእቲ፣ እንታኩቲ ፣እንትኩ, ውእቶሙ፣እልኩ፣እልኩቱ,, ውእቶን፣እልኩን፣እልኩቶን

2.2.3.3 Possessive pronouns

Possessive pronouns are words that indicate ownership, or possession shows that something belongs to someone or something. In geez, possessive markers or suffixes are used for this purpose. The table below shows the possessive pronoun of geez language. [18]

Table2. 4: possessive pronouns in Geez language

	singular	English	Plural	English
	Geez		Geez	
1 st person	ዚአየ	Mine	ዚአነ	ours
2 nd person	ዚአከ	Yours	ዚአክሙ	Yours
	ዚአኪ		ዚአክን	
3 rd person	ዚአሁ	His	ዚአሆሙ	Theirs
	ዚአህ	Hers	ዚአሆን	

2.2.3.4 Conjunctions

A conjunction is a word to join clauses, sentences or words within the same clauses. In geez conjunctions such as ወ, አው, ዳዕሙ, አላ, and ባሕቱ are commonly used. Some examples of conjunctions ማርያም ወማርታ መጹ, (Maryam and Marta come), አብርሃም ወይስሐቅ ወያዕቆብ፡ ፡ , (Abraham, yeshak and Yaekob). [18].

2.2.3.5 Punctuation marks

In geez, there is no question mark; interrogatives in geez are placed at the end of a word or a sentence, distinct intonation, typically at a low level, where the style of pronunciation itself indicates a question. Some examples are listed here, ሁ, ኑ, ኡ, ኢ, ት, ኣ:- ኣይት:- ሶበኑ: (when?), ተአምሩኑ: (Do you know?),አንትሙሁ: (Are you?): ተአምረኒኢ: (Do you know me?). [18].

2.3 Geez language phrases

2.3 .1 (Noun phrases) NP ምስረታ ስማዊ ሓረግ

ተውላጠ ሰም+ ስም = ኣንትሙ.<PN> ሰብኣ ሰገል<N>

መስተአምር(Prep)+ ስም = ክልከሆሙ <Prep>ኣርድኣት.<N>

ቅዕል (adj) +ስም= ደቂቀ<adj> ህፃናት.<N>

ስም+ስም = ኅጢአት.<N> ሰኩናየ.<N>= ድንጋጫ.<N> ሞት.<N>

ስም+ስም+ስም = አምላክ <N>አማልክት.<N> እግዚአብሔር.<N>

(prep)መስተአምር+ ስም+ ስም = ወበኩሉ ምድር ስብሐቲሁ

ስም+ (Conj)መስተዋድድ+ ስም+ ስም = ወመዓቶሙኒ ከመ ሕምዝ አርዌ ምድር

ተውላጠ ስም+ ግሳዊ ቅዕል+ ስም = ኩሎሙ ገበርተ ዐመፃ

ስም+መስተዋድድ+ ስም+ መስተዋድድ+ ስም = ዕዕ:እነተ:ትክልት:ጎበ:ሙሐዘ:ማይ

ቅዕል+ ቅዕል+ ስም = ደቂቀ ዕንላ እመሕያው

ተውሳጠ ስም+ መስተካምር+ ቅፅል+ ስም = እሎንቱ ስለስቱ ደቂቀ ህፃናት

ተውሳጠ ስም+ መስተካምር+ ዓረፍተ ነገር() =ዘንተ ኩልክሙ እለ ትነብሩ ውስተ ዓለም

ስም + ቅፅል + ስም = ወልሳኖሙኒ በሊገ መጥባሕት

2.3.2 Verb Phrases VP (ምስረታ ግሳዊ ሓረግ)

ግስ + ስም + ስም = ያነብብ መዓልተ ወሌሊተ

ግስ + ስም+ መስተዋድድ + ስም= ትውሀብ፡ፍሬሃ፡በበ፡ግዜሃ

ግስ + ቅፅል + መስተዋድድ= ኢይትሥኡ፡ ረሲዓን፡እምደይን

ተውሳክ ግስ + ግስ= ለምንት፡አንገለጉ

ግስ + መስተዋድድ + ቅፅል = ተጋብኡ፡ምስሌሆ፡ጎቡረ

ግስ + መስተዋድድ + ስም + ግስ + ስም = ይቤልዋ፡ለነብስየ ዕያድጎነኪ፡አምላክኪ፡

ግስ + መስተዋድድ + ግስ + ስም+ መስተዋድድ = አእምሩ፡ከመ፡ተሰብሐ፡እግዚአብሔር፡በጻድቁ፡

ስም + መስተዋድድ + ተውሳክ ግስ + ግስ = እግዚአብተስፋ፡ባሕቲትከ፡አጎድረከኒ

ስም + ግስ + ተውሳክ ግስ = ቃልየ፡አጽምእ፡እግዚአ

ስም + ምስተዋድድ + ግስ+ ተውሳክ ግስ = ስመ፡ጎቤከ፡እጸሊ፡እግዚአ፡

መስታዋድ + ግስ + ስም = በጽባሕ፡እቀውም፡ቅድሜክ

ስም+መስተዋድድ+ መስተዋድድ+ ስም = ወይከውን፡ከመ፡ዕፅ፡እነተ፡ትክልት፡ጎበ፡ሙሐዘ፡ማይ

2.3.3 Sentences formulation in Geez language

Sentences are constructed from the noun phrase and verb phrase, here the language experts define some rules to create geez language sentences.

Noun phrase + verve phrase = Sentences

(ኣንተ) + (ወእምኩሉ ምንዳቤየ አድጎነኒ) = ኣንተ ወእምኩሉ ምንዳቤየ አድጎነኒ

(እስመ:አንተ:) + (እግዚአብሔር:ባሕርይ:ቅዱስ:አኅዳርከኒ)=
 እስመ:አንተ:እግዚአብሔር:ባሕርይ:ቅዱስ:አኅዳርከኒ::

(አምላክየ) + (ወረዳኢያ ጎይልየ ወጸወንየ ወመድጎንየ:) = አምላክየ ወረዳኢያ ጎይልየ ወጸወንየ
 ወመድጎንየ::

(ጻዕረ ሞት) + (አጎዘኒ) = አጎዘኒ ጻዕረ ሞት::

(እግዚአብሔር) + (ይስማዕከ በዕለተ ምንዳቤክ) = ይስማዕከ እግዚአብሔር በዕለተ ምንዳቤክ

(ንጉሥ) + (እግዚአ በጎይልከ ይትፌሣሕ) = እግዚአ በጎይልከ ይትፌሣሕ ንጉሥ

አብነት (Examples)

1. አንተ (ስም) (N) ወእምኩሉ ምንዳቤየ (መስተዋድዳዊ ማረግ) () መእሰሪ ዓንቀፅ

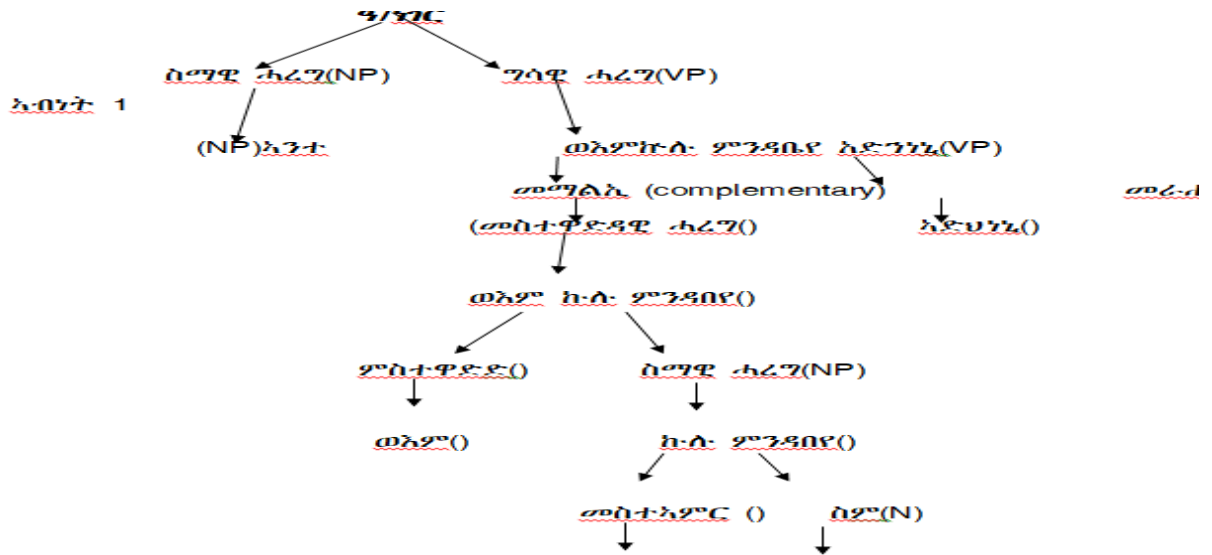


Figure2. 5: sentence construction of geez language

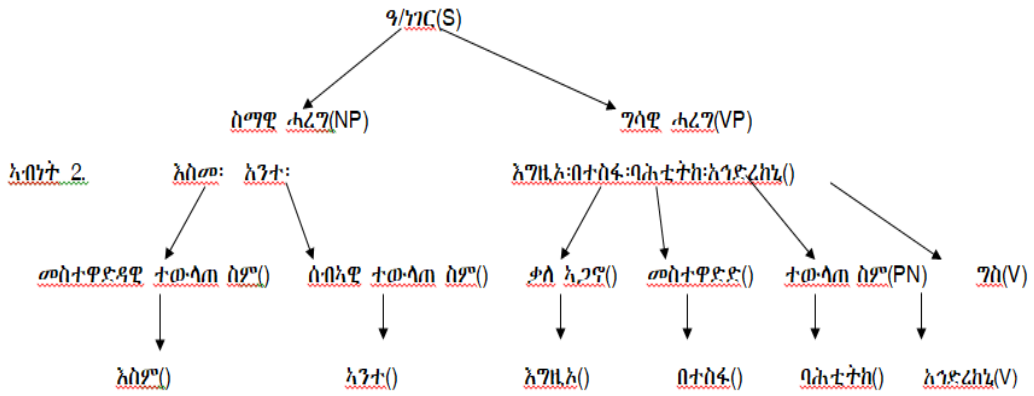


Figure2. 6: Example 2 Sentence construction of geez language

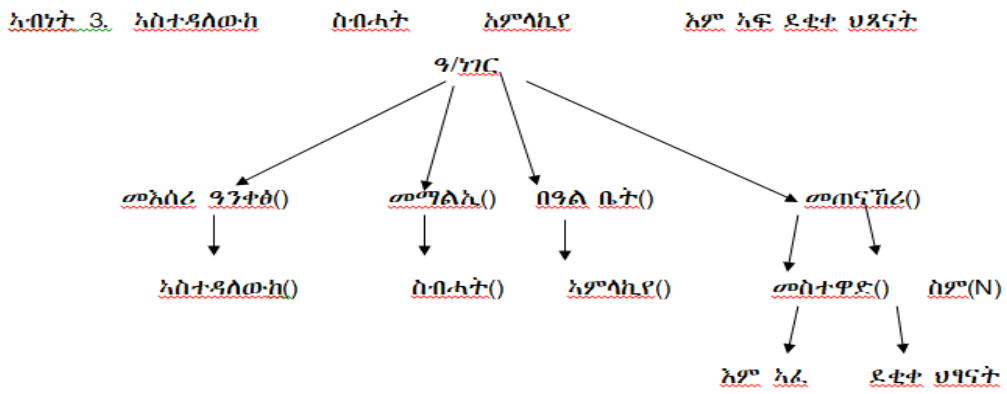


Figure2. 7: Example 3 Sentence construction of geez language

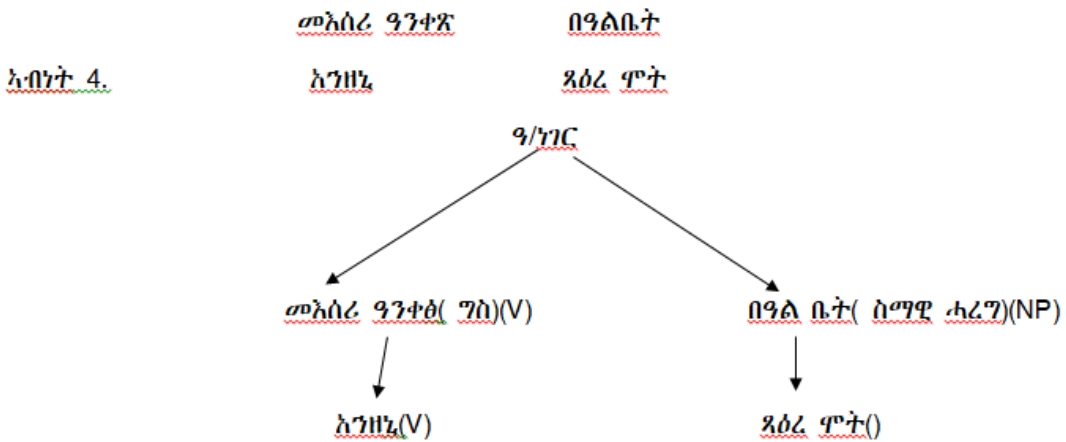


Figure2. 8: Example 4 Sentence construction of geez language

አበነት5.

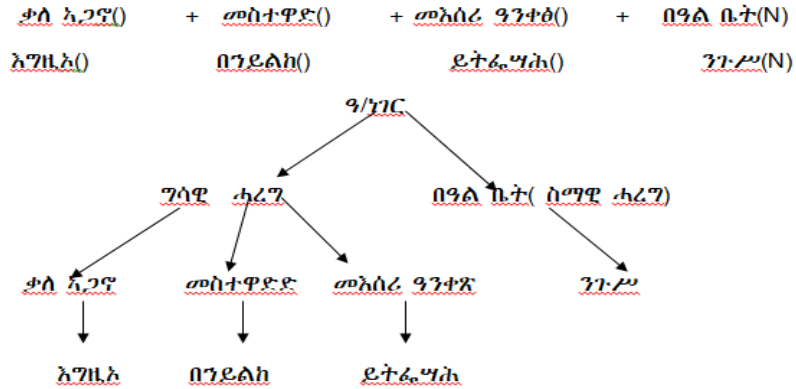


Figure2. 9: Example 5 Sentence construction of geez language

ስርዓተ ንግግር ዓ/ነገር በገቢር ግስ

1. ባህር ሙሴ መእሰሪ ዓንቀጽ + ተስሐቢ. + በዓል ቤት
2. ሙሴ ከፈለ ባህር በዓል ቤት + መእሰሪ ዓንቀጽ + ተስሐቢ.
3. ባህር ከፈለ ሙሴ ተስሐቢ. + መእሰሪ ዓንቀጽ + በዓል ቤት

ስርዓተ ንግግር ዓ/ነገር ተገብሮ ግስ

ባሕር ተከፍለ በዓል በት + መእሰሪ ዓንቀጽ

ተከፍለ ባሕር መእሰሪ ዓንቀጽ + በዓል ቤት

2.4 Grammar formalism

A parser needs grammar that explains how sentences are built from their components and how the meaning of a sentence is derived from the information obtained in its parts. When a parser processes an input word or sentence in a specific language, it produces a description of the possible structural relationship between words or sequences. It also gives structural representations of the input, as explained by the grammar rules.

In some parsers the formulation of such representations occurs simultaneously with the derivation of structural analyses based on the grammar. It is widely known grammatical structure plays an important role in developing meaning, making the identification of the grammatical structure of natural language sequence an important step in understanding the sentence's meaning. However, many sentences, including those in geez, can be ambiguous, carrying multiple meanings. To effectively handle such ambiguity, grammar formalism capable of managing these differences is important for natural language processing. [29]

There are different types of grammar formalism, among those are listed below, Context Free Grammar, Transformational Grammar, Transition Network grammar, Unification Based Grammar, and Probabilistic Context Free Grammar.

2.4.1 Context Free Grammar

Context free grammar (CFG) is defined as four-tuple (V, Σ, R, S) where:

- V :-A finite set of nonterminal symbols (abstract symbols representing language constructors)
- Σ :- A finite set of terminal symbols (the actual characters or tokens of the language).
- R : - A finite set of production rules, each of the form of $A \longrightarrow \alpha$, Where $A \in V$ and α , is a string made up of symbols from $V \cup \Sigma$.
- S :-The start symbols a special non terminal from which derivation begins. [33].

The production rules identify how nonterminal symbols can be replaced by sequence of terminals and nonterminal, this recursive Context free grammar (CFGs) is crucial in parsing techniques such as Top-Down Parsing, it provides a formal framework for constructing parsers that can systematically analyze the syntactic structures of sentences. Starting from the symbol, production rules are applied to derive strings in the language. These derivations can be represented as parse trees, which visually map the structure of the generated sentence or words. Even though Context Free Grammar (CFGs) is highly versatile it has some limitation; it cannot capture all language behavior, especially those requiring context sensitive information.

2.4.2 Transition Network Grammar

Transition network grammar (TNG) formalism defines the rules by using nodes and labeled arcs in a transition network. One of the nodes is identified as a start symbol. Beginning at the starting state an arc can be navigated if the current word in the sentence is in the group on the arc. If the arc is succeeded, the current word is modified to the next word. Simple transition networks are sometimes known as Finite State Machine (FSM) and have the same expressive rule as regular grammar. But they have some limitations to fully describe all languages that can be explained by context free grammar (CFG) [31]. To be more descriptive rule as the context free grammar (CFG) the transition network grammar should allow arc labels to denote to other networks and

word groups. Grammar formalism obtained by this rule is known as recursive transition network [32].

2.4.3 Augmented Transition Network

Augmented Transition Network is another kind of transition Network Grammar for writing natural language grammar. This grammar formalism denotes the grammar in the assumption if there is a path from the beginning state to the ending state such that the labels on the arc match the words of the sentences, a sentence is in the language defined by the network [32]. Augmented Transition Network can invoke sub-networks recursively, allowing them to represent nested and hierarchical structure that basic networks cannot. While a basic Transition Network Grammar relies purely on state transition to decide if a sentence is in the language, an Augmented Transition Network introduces extra capabilities (recursion, registers, and conditions) that enable it to capture more complex syntactic phenomena. This makes Augmented Transition Network significantly more powerful and better suited for parsing natural languages with complex, hierarchical structures.

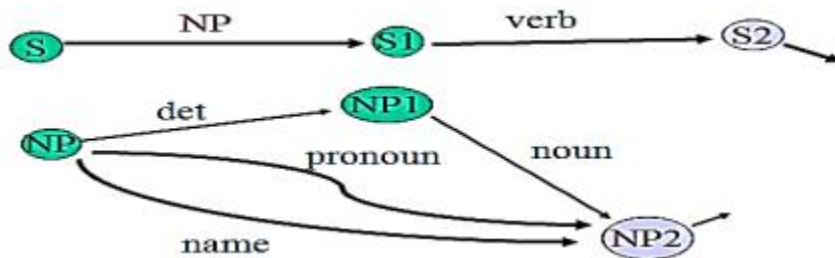


Figure2. 10: Augmented Transition Network

2.4.4 Context Sensitive Grammar

Context sensitive grammar is defined as a four-tuple (V, Σ, R, S) ,

Where:

- V :- is a finite set of nonterminal symbols,
- Σ :- is a finite set of terminal symbols (disjoint from V).
- R :- is a finite set of production rules, unlike Context-Free Grammars (CFG), in Context Sensitive Grammar each rule is of the form;

$$\alpha A \beta \longrightarrow \alpha \gamma \beta$$

Here, $A \in V$ is a terminal, α and β are strings (which can be empty) composed of terminals and/or non terminals, and γ is a non empty string of symbols (ensuring that the production is non-contracting).

- S : - is the start symbol, a designated nonterminal from which derivations begin.

All production rules in a Context Sensitive Grammar (CSG) are non-contracting, meaning that the length of the string on the right-hand side is always less than or equal to the length of the string on the right hand side. This condition prevents the derivation process from reducing the string length, which is essential for ensuring that the grammar defines a Context-Sensitive language. [30].

2.4.5 Probabilistic Context Free Grammar

Probabilistic context free grammar (PCFG) is grammar formalism each production rule is related with probability [8]. Probabilistic context free grammar has five tuples, $G = (N, \Sigma, R, S, Q)$ where

N = a finite set of non-terminal symbols

Σ = a finite set of terminal symbols

R = a finite set of production rules of the form $A \rightarrow \alpha$ (where A is an element of N and α is an element of $(\Sigma \cup N^*)$)

$S \in N$ is a start symbol and

$Q: R \rightarrow [0, 1]$ is a function that assigns a probability to each member of R

Therefore, the probability of $P(y)$ of a parse tree y is defined as the product of probabilities of all rule applications in the derivation of y .

$$P(y) = \prod_{i=1}^{|R|} q_i^{\text{count}(i,y)} \text{ where}$$

Y is a parse tree generated using PCFG

$\text{Count}(i, y)$ = the number of times that the i^{th} production rule $r_i \in R$ is used in the derivation of y

q_i is the probability of rule i . [7,34]

2.4.6 Unification based Grammar

Unification based grammar rules focus on the case, gender, tense, and values that are denoted in the lexical entries of the words. The feature structures of these sentences can be extracted using unification operation [35]. These feature structures are processed through the operation of unification, where the entire grammar is defined as a set of constraints applied to these structures.

2.5 parsing

Parsing is the process of breaking down a sentence into its parts of speech, such as nouns, adjectives, verbs, prepositions, etc., and assigning each word its respective category within in a sentence of a given language. It also involves grouping words into phrases. Additionally, parsing identifies the structure of sentences based on a set of predefined rules. In natural language processing parsing can occur at either the word or sentence level. Word- level parsing involves tokenizing a word into the smallest meaningful units of words. [8]. before proceeding further let's look at how sentence parsing operates using sample Geez language texts. Generally, the parsing process when an input word or string is given. After the input is tokenized or breaks down by the tokenization process, the parser analyzes the tokens, matches them to the grammar and parses the input text or string according to the grammar rule given.

Using the given information, the parser builds a parse tree for each input sentence. The parse tree visually represents the derivation of the input texts or strings based on the grammar rule. It recognizes the language's grammatical rules and their application. Each node in the parse tree corresponds either to a word in the input or a non-terminal in the grammar. [35].

The choice of a Top-Down Parser is particularly suited to Geez language processing for different reasons: The first reason is its alignment with Expert-Defined Grammar. Geez, being a classical language with unique syntactic and morphological properties, benefits from a rule based approach. A Top-Down Parser naturally aligns with the structured, expert-defined grammar rules

provided by language experts. It begins with the start symbol and recursively applies grammar rules to generate the expected structure, mirroring thorough linguistic analysis of Geez. The other reason is it offers greater transparency, following researchers to trace how each grammatical rule is applied. This is particularly important for a language like Geez, where the explicit, handcrafted grammar needs to be carefully validated and refined in collaboration with language experts. Geez also shows hierarchical sentence structures and, at times, recursive patterns. The Top-Down Parser is inherently recursive, which makes it well-suited to handling such nested and recursive linguistic phenomena. This approach allows the parser to naturally construct parser trees that represent the complex, multi-layered structure Geez sentences.

2.5.1 Approaches of Sentences Parsing

Parsing is a process of constructing syntactic analysis of a given sentence based on a given grammar [6]. Parsing can serve as an intermediate stage for semantic analysis of natural language processing applications such as Information Retrieval (IR), Information Extraction (IE), and question answering. Based on the algorithms design, approach of sentences parsing can be classified in two types: these are stochastic and rule-based approach sentence parsing [37].

2.5.1.1 Stochastic approaches

A stochastic approach of sentence parsing works based on probability and statistics. This approach follows the Bayes (Network) theorem and the Markov assumption model to determine the sequence of words in each sentence [6]. The stochastic approach also can be classified as a supervised approach and unsupervised approach.

2.5.1.2 Supervised approach

Supervised approach sentence parsing uses, labeled data, and each sentence paired parse tree for analyzing the syntactic structure of the sentence [5]. This method adopts machine learning techniques to learn the relationships between sentences and their grammatical structures.

Key components of supervised approach are

Labeled dataset: -Collect a dataset of sentences annotated with their parse trees. Datasets like the Penn Treebank are commonly used for this purpose. Each sentence in the dataset should have a corresponding syntactic structure (e.g. consistency or dependency tree).

Feature Extraction: Extract relevant features from the sentences that can help the model learn patterns.

Model Selection: choose suitable machine learning for parsing, for example Transition-based parsers and Graph-based parsers.

2.5.2. Unsupervised approach

The unsupervised sentence parsing approach uses unlabeled data to analyze and understand the structure of sentences parsing [5]. This approach is particularly useful when annotated datasets are scarce or expensive to obtain. Some basic components of the unsupervised sentence parsing approach are listed here.

Data Collection: Data can be collected from different sources like books, articles, or Web pages.

Future Representation: Represent sentences using features that hold their syntactic properties.

Clustering and Probabilities Models: use clustering techniques to group similar structures or words.

2.5.3 Rule-Based approaches

The rule-based approach for sentence parsing involves using a set of grammar rules to analyze the sentence. This approach has basic components to accomplish its functionality; these are grammar definition, Context Free Grammar (CFG), Phrase structure rules, Lexicon, Parsing algorithm. The rule-based approach is composed of three different types of sentence parsing: Top-Down Parsing, Bottom-up parsing, and Chart parsing.

2.5.3.1 Top-Down parsing

A Top-Down parser begins with the start symbol (S) and works to rewrite it into a sequence of terminal symbols that matches with the word classes in the input sentences. At any given point, the parse state can be represented as a symbol list which shows the operations performed up to that level. For example, the parser starts in the first state (S). Applying the grammar rule $S \rightarrow NP VP$, the symbol list becomes (NP VP), then using the rule $NP \rightarrow N VP$, the symbol list updates to N VP, and these activities continue recursively. The parser proceeds in this manner until it reaches terminal state symbols, at which point it checks whether the word classes in the input sentence match the rewritten terminal sequence. [32].

2.5.3.2 Bottom-Up Parsing

A Bottom-Up parser strategy begins with the words in a sentence and works backward using the rewritten rules to reduce the sequence of symbols on its right-hand side. [38]. this method starts by assigning agrammatical category to each word and progress until it reaches the start symbol. The process treats the resulting highest –level sequence as a new string to operate on. Additionally, this method groups words into their respective categories for example (Adj VP, V-->VP) according to the grammar rule.[8].

2.5.3.3 Chart Parsing

Chart parsing addresses the limitations of simple parser interns of completeness and efficiency by applying dynamic programming techniques. Dynamic programming enhances efficiency by storing and reusing intermediates results when needed. In syntactic parsing, this approach is referred to as Chart Parsing. The “Chart” is a data structure that stores partial parsing results, allowing them to be reused throughout the process. This method ensures that a constituent is built only ones and then stored in a table for later use as a sub-constituent of larger structures.

The table used to store partial results is called Well- Formed –Substring-Table [18], it records adjacent word sequences within a sentence and tracks the syntactic constituents identified. Ones the table is constructed it maintains a record of all detected constituents.

Summary

This chapter began with an overview of the Geez language, one of the ancient sematic languages of Ethiopia and Eritrea. We explore the grammar of Geez, including word classes, phrases, and sentence structure. Geez follows **SVO**, **VSO** and **OVS** (Subject Verb and Object, Verb, Subject and Object, Object, Verb and Subject) word order. Next, we discussed the different types of grammar formalism and parsing methods. Different grammar formalisms and parsing techniques, approaches reviewed by different scholars are discussed in detail.

Chapter Three: Related Work

3.1 Introductions

Significant progress has been made in natural language processing (NLP) for different languages. Sentence parsing plays a critical role in enhancing the performance of numerous applications, such as Machine Translation, Grammar Checking, Question Answering, Semantic analysis, and more. Additionally, parsing is essential for identifying phrasal categories, analyzing correct grammatical structures, and understanding the relationships between words in a sentence. Automatic parsing has been one of the most extensively studied areas in NLP, with a substantial amount of research dedicated to it over the past few decades. This chapter reviews papers relevant to this research area. To provide a clear understanding of the current study, the chapter discusses related works for both Ethiopian and non-Ethiopian language.

3.1.1 Amharic language parsing

There are some works regarding sentences parsing for Amharic Language, and one of them is automatic sentence parsing for Amharic text [7]. This paper uses probability context free grammar formalism to construct the sentences and used inside outside algorithm with a bottom-up chart parsing strategy. The paper conducted an experiment from sample data prepared one type of sentences, which was combination of four words and achieved an accuracy of 85%. The sample data was limited to 100 sentences [7], and they were all the same type of sentences, declarative type of sentences. The sentences were tagged automatically using the POS tagger developed by Mesfin [21] and manually parsed by the language experts.

The second work for Amharic Language parsing is an integrated approach to automatic complex sentence parsing for Amharic text. [8]. this work is focused on the Amharic complex sentences and the approach they follow to solve the problem is very similar [7]. The grammar formalism is probability context free grammar, and the algorithm is inside outside algorithm. Beyond this the paper tries to integrate the work with some morphological, Lexicon and syntactic properties of the language. The experimental result shows 98.7% on the training set and 94% on the test set.

A hybrid approach to Amharic base phrase chunking and parsing [9] is also additional work for Amharic language parser; this paper focuses on retrieval of different kinds of phrases by

grouping syntactically interrelated words using Hidden Markov Model (HMM). This study considers simple and complex Amharic sentences for the experiment and achieves 93.5%.

Another paper on the Amharic language parsing is a top-down chart parser for Amharic sentences [6]. Context Free Grammar Formalism (CFG) is used for grammar formalism of the language. This work covered all types of sentences so far not covered on the previous works. The experimental result conducted for the different types of sentences scored, 90%, 85%, 80%, and 75%.

3.1.2 Afan Oromo Language Parsing

An Automatic Sentence Parser for the Oromo Language using the chart algorithm was developed by Driba Megarsa.[39]. The parser was developed Head-driven Phrase Structure Grammar (HPSG) formalism, merged into a left-right table. Besides the study included a supervised learning algorithm to handle unknown and ambiguous words and integrated a morphological analyzer to break down words into their origin forms and associated morphemes.

The dataset for this study was collected from Oromo Textbooks, and they were around 352 sentences. These sentences were classified as a training and test set (85% and 15%) respectively. Two experiments were conducted in two phases, first on the training, and next on the test dataset. The result was evaluated by comparing with the manually parsed sentences. To improve accuracy, errors were distinguished and corrected, and succeeded by repeating training cycles until a satisfactory result was obtained.

The performance of the parser was evaluated using the percentage of correctly parsed sentences from the sample datasets. The parser shows high accuracy, scoring 95% correctness on the training and testing sample datasets. The algorithm used on this parser was efficient on predicting the unknown words in the input sentences.

3.1.3 Tigrigna Language parsing

The Development of Automatic Parser for Tigrigna Language using the Bottom-Up Probabilistic chart parser is one work for the Tigrigna language Parser. [12]. Simple and complex Tigrigna sentences were conducted on this research, and Probability Context Free Grammar Rule was used on the formalism of the grammar rule. This work conducted an experiment on three different parts; the first test was performed on the training dataset, the second one conducted on the test dataset and the third one on different dataset not included on the sample dataset. The accuracy achieved for the simple sentences' dataset were 95%, 94% and 85% respectively. Similarly, the accuracy achieved for the complex sentences' dataset were 91%, 90% and 80% respectively.

3.1.4 Arabic Language parsing

Al-Tanni et.al. [37], Developed a Top-Down parser specifically designed for parsing simple Arabic sentences within a limited a limited domain of Arabic grammar. The study began by driving grammar rules that describe the structure of grammatical sentences. Based on these rules, the authors implemented a parser capable of assigning grammatical structures to input sentences.

The dataset used in the study consisted of 70 sentences collected from Arabic documents, with sentence length ranging from 2 to 6 words. The parser operates in three main stages: Sentence decompositions, grammar identification, and parsing. Initially, the system uses automatic categorization to break the sentence into its core components. Following this it identifies the relevant grammar rules to analyze and determine the structural composition of the sentence.

There is also one work on the Arabic language known as an Efficient Transition Network Parser, [38], focusing on analyzing and extracting the attributes of the language. Top-Down Parsing technique was used through a Recursive Transition Network (RTN). Additionally, a morphological analyzer was incorporated to handle words not directly found in the lexicon. The datasets used for this work were 90 sentences, and their word length was limited to six words, and collected from a grade level textbook. The sentences were constructed to ensure gender and number agreement, enabling the evaluation of the parser's ability to correctly identify syntactic structures of the Arabic language.

Summary

This review gives an insight view on the sentences parsing research in Ethiopian languages and highlights advancement of parsing approaches. Different studies apply diverse parsing techniques such as Bottom-up probabilistic chart parsing; Top-Down parsing, and Hybrid models, showing that no single method is universally effective for Semitic and Cushitic languages. Most studies rely on small datasets (70-352 sentences), limiting generalizability. Expanding annotated dataset and using diverse sentence types improve parser robustness. Using grammar structures specific to each language such as HPSG (Head-driven Phrase Structure Grammar) for Afan Oromo or PCFG (Probabilistic Context Free Grammar for Amharic and Tigrigna) is essential for accurate parsing.

4.3 Components of the system Architecture

4.3.1 The grammar rule

The grammar rule defines the formal Context Free Grammar (CFG) used by the chart parser to analyze and parse Geez sentences. This grammar acts as the backbone of the parser by specifying the syntactic rules and patterns that Geez sentences follow. In addition to this the grammar rule provides a rule based structure to interpret the syntactic composition of sentences, allow the parser to decompose sentences into hierarchical constituents (e.g noun phrase, verb phrase), and enables the parser to validate whether a given sentence adheres to the syntactic rules of Geez.

A Context Free Grammar (CFG) is composed of production rule that defines how complex structures (non terminals) break down into simpler ones (terminals). The grammar likely follows this structure:

- Start symbol (Root Rule)- defines the highest-level structure of the sentence (e.g a complete Geez sentence)
- Example: $S \longrightarrow NP VP$, this means a sentence (S) consists of a Noun Phrase (NP) followed by a Verb Phrase (VP)
- Phrase level rules: describe how main syntactic components are structured. Noun Phrase (NP) includes nouns, modifiers, and Verb phrase (VP) contains verbs, Object, or adverbs.
- Example $NP \longrightarrow N | N$ and $VP \longrightarrow V | NP | V$
- $N \rightarrow$ "ግዳ" | "ዕረፍት" |
- $V \rightarrow$ "ሐፀነኒ" | "ሜጣ" |
- When a Geez sentence is input into the parser the sentence is split into individual words (tokens).
- The chart parser applies the Context Free Grammar Rule (CFG) rules recursively in a Top-Down manner, and it tries to match the sentence against the grammar, building a parse tree if the sentence confirms to the defined rules.

4.3.2 Tokenization Process

Tokenization is the process of splitting a sentence into its individual components (tokens). This step is important because the parser cannot analyze the sentence directly as a raw text string. Instead, it needs tokens to match against the Context Free Grammar (CFG) rules. The parser takes Geez sentence as a string. For Example “ወንበ ማየ ዕረፍተ ሐፀነኒ”, the sentence is tokenized using python’s split () method (sent=PhraseToBeparsed.split()) this method splits words by white space, producing [ወንበ, ማየ, ዕረፍተ, ሐፀነኒ,]. These tokens are then passed to the NLTK chartparser, which compares them against the defined Geez Context Free Grammar (CFG)..

4.3.3 Lexicon

The purpose of the lexicon is to serve as a dictionary or repository of words with their part of speech. Its function is to generate tagged corpus from the lexicon generator module and supports the morphological analysis of input tokens. Its interaction with the components is to give information to the word classifier and enrich the parsing process by providing context and details for words in the input sentences.

4.3.4 Word Classifier

The purpose of the word classifier is to assign grammatical word orders, such as noun, verb, and adjective to each token in the input sentences. The function of the word classifier is to utilize the lexicon for determining the part of speech for each token. Its function is working closely with the tokenization process and lexicon to ensure accurate classification and gives categorized tokens to the parser for syntactic analysis.

4.3.5 Input sentences

The input sentence serves as the raw data to be analyzed and parsed by the system. Its function is to represent the initial state that providing sentences that need parsing and supplies to the tokenization process for preprocessing. It interacts with the system as an initial step for all downstream processes.

4.3.6 Parsed Sentences

This part provides the final syntactic structure of the input sentences after analysis. Its function represents the output of the parsing process, formatted as a tree or another hierarchical format, and shows how tokens are organized grammatically according to the grammar rule. The interaction is after the parser generated the processing inputs through the grammar rules, lexicon and the word classifier.

4.3.7 The parser function

The parser is the main component responsible for building the syntactic structure of the input sentence based on the grammar rules. It verifies whether the sentence conforms to the defined grammatical rules and produces a structured representation of the sentence. The functions of the parser are taking tokens and their corresponding classifications (parts of speech) as input, uses the grammar rules (CFG) to validate and build parse tree or equivalent structure of the sentence, and produces a syntactic structure (parse tree) as the output. Its interaction is almost with all components of the system that integrated for parsing process.

4.4 Top-Down Parsing Algorithm (psedocode)

```
Input: Sentence S, Context-Free Grammar (CFG)
Output: Parse Tree(s)

1. Initialize the parser with the start symbol of the CFG.
2. Tokenize the input sentence S.
3. For each production rule in the CFG:
    a. Match the leftmost non-terminal symbol with a production.
    b. If a match occurs:
        i. Expand the production.
        ii. Continue parsing recursively.
    c. If no match is found, backtrack and try the next rule.
4. If the entire input is consumed and the derivation is complete:
    a. Construct and return the parse tree.
5. If no valid derivation is found, return an error (unparsable).
```

Figure4.2: The Chart Parser algorithm (psedocode)

The algorithm implements a syntactic parsing process using the chart parsing using the chart parsing technique, widely used method for analyzing the grammatical structure of sentences. It takes a sentence or phrase and a predefined grammar as an input and outputs a corresponding tree. As its core, the algorithm uses the NLTK (Natural Language Toolkit) implementation of chart parsing, which efficiently handles ambiguous or complex sentences by storing intermediate results.

4.5 Mathematical Formulation

Formalize the parsing process with mathematical expressions where Context Free Grammar (CFG) is formally defined as:

$$G = (N, \Sigma, P, S)$$

- N = Non-terminal symbols (e.g, NP, VP)
- Σ = Terminal Symbols (Geez words)
- P = Production Rules (e.g, $S \rightarrow NP VP$)
- S = Start symbol

Given an input sentence $w = (w_1, w_2 \dots w_n)$, the parser checks if:

$$S \Rightarrow^* w$$

Where, $S \Rightarrow^*$ represents derivation through Context Free Grammar (CFG) rules.

4.6 Data collection

The data for developing a Top-Down sentences parser for Geez language was collected from the book “Mezumure Dawit”. It is a liturgical text written in classical Geez, a foundational source of the language. It provides:

- Rich representation of authentic Geez grammar and syntax
- It preserves classical linguistic structure and diverse sentences types
- Being a well known source of reference

Nearly 3000 sentences were collected from the book. The selection process randomly picked from each chapter. For our study we use only 500 of them. The following table shows name of phrases and their tagged names

Table5. 1: Name of Phrases and Tagged Name

SNO	Name of Phrases	Taged Name
1	Noun Phrase	NP
2	Verb Phrase	VP
3	Preposition	Prep
4	Adjectives	adj
5	Adverbs	adv
6	Conjunctions	Conj
7	Pronoun	Pron

4.7 System Imlementation

The parser is implemented using python on the Google colab platform, ensuring easy accss to computational resources.

4.7.1 System Specification

- Operating System: Windows 10
- Memory: 4 GB
- Storage: 500GB
- Development Enviroment Python 3.10 on Google Colab

4.8 Experimental Result and Discussion

The parsing process needs a gold standard tree for different evaluation metrics, unfortunately there is no such standard set for Geez language sentence parsing. We collect around 3000 sentences, but only 500 sentences were tagged by the language expert. The linguistic validation and manual parsing for this research were conducted in collaboration with a Geez language expert from Department of Geez, Aksum University. The expert holds extensive academic and practical experience in Geez linguistics, with a focus on Syntax, grammar, and historical texts. his expertise was instrumental in:

- **Validating the grammar rules:** - Ensuring the Context-Free Grammar (CFG) used in the parser aligns with the authentic syntactic structures of the Geez language.
- **Manual Parsing:**-providing a gold standard by manually parsing 500 sentences from the Mezumur Dawit text for comparison with the algorithm’s output.

Based on the rules obtained from the experts, we try to parse the tagged sentences and get the following results.

Table5. 2: Parsed sentences by the parser

Number of Sentences	Correctly parsed	Not Parsed	Percentage
500	470	30	94 %

The following table shows the language experts parsed sentences result.

Table5. 3: Parsed sentences by the language expert

Number of Sentences	Match with the parser	Not Matched with parser
500	490	10

Given:

- Gold standard total (total constituents in gold standard): 490
- Number of correctly parsed trees (correct constituents): 470
- Total number of sentences: 500

Precision = (Number of Correct Constituents) / (Total Number of Constituents in Parser Output)

Precision=470/500

Precision=0.94

Precision measures how many sentences the parser identified as correct according to the expert. The parser produces very few incorrect outputs; there is a 94% chance that it is accurate.

Recall = (Number of Correct Constituents) / (Total Number of Constituents in Gold Standard)

Recall=470/490

Recall =0.959 (95%)

Recall measures how many of the correct sentences from the expert’s dataset were successfully parsed by the algorithm.This means the parser is able to capture all valid Geez sentence

structures 95%. This indicates that the grammar and algorithm are including different types of sentences.

F1-score = $2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$

$2*(0.94*0.959)/(0.94+0.959) = \text{F1Score}=0.9499$

The F1 score is the harmonic mean of precision and recall, offering a balanced measure of the parser's overall performance. The parser performs consistently well in both accuracy (precision) and coverage (recall), With an F1 score of 94%.

Accuracy=Correct parsed tree /Total Sentences

$(470/500) * 100\% = 94\%$

Accuracy refers to the percentage of all sentences (Correct and incorrect) that were parsed correctly. 94% of the sentences were correctly parsed only 6% of the outputs were errors. This reflects strong overall performance across the dataset.

4.9 Summary

This chapter covers the design of Geez language parser using a recursive descent algorithm. It outlined the system architecture, formal grammar, tokenization, parsing algorithm, mathematical formula, the system specification, data collection and experimental result.

Chapter Five

5. Conclusion and Recommendation

Developing a parser for under resource languages like Geez is challenging, with the help of the language expert and some literature review we are trying to implement a Top-Down Parser for Geez language. By giving a structured methodology, which includes tokenization, word classification, and syntax tree generation, the parser successfully demonstrated the ability to analyze and parse Geez sentences, following the grammatical rules of the language.

The evaluation metrics applied, including accuracy, precision, recall, and F1-score, gave a valuable insight for the performance of the parser and indicating areas of strength and opportunities for improvements. The parser's capability to cope up different sentence structures and word forms was critical in ensuring that it could parse the complexities found in Geez language. This indicates that the parser needs more work to enhance the system's performance; these include modifying the lexicon to cover a wide range of vocabulary, refining the error handling mechanism and exploring the combination of machine learning techniques to improve the overall performance of the parser. Besides expanding the parser to handle more complex structure of the language is a valuable direction for future work.

As a conclusion this work brings a beginning cornerstone for sentence parsing in Geez language and opens the door for further research and technological advancement in the field of natural language processing for the Semitic languages.

Based on the result so far obtained we recommend the following points

1. Expanding the data set: - Future researchers should consider the data set should include more diverse sentences from different sources to improve the generalizability of the parsing model
2. Improve the parsing algorithm:- Top down approach can be integrated in to machine learning
3. Developing a strong lexicon and Morphological analyzer
4. Creating a Geez Treebank
5. Extending to other Semitic languages.

References

- [1]. W. Shuly, Natural Language Processing. University of Haifa Press, Israel, 2009.
- [2]. Manning, C. and H. Shutze, “Foundations of Statistical Natural Language Processing”, MIT Press, Cambridge, 1999.
- [3]. A.Tayal, M. Raghuwanshi and L. Malik, Syntax parsing: implementation using Grammar rules for English Eanguage”, Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference, IEEE Computer Society Washington, DC, USA, Jan 2014.
- [4]. አባ ኪዳነ ማርያም ጥዑመ ልሳን, 14ቱ መዝገብ ቅዳሴ, ኢ.አ ኢትዮጵያ: አኩቲት አሳታሚዎቻ, 2009.
- [5]. ዘርአዳዊት, አድሐና, ልሳናተ ሴም (ግእዝ፣ትግራይ፣አማርኛ) ንጽጽራዊ መዝገብ ቃላት, አዲስ አበባ፣ ኢትዮጵያ: ሜጋአሳታሚና ማከፋፈያ ኃ/የተ/የግ/ማኅበር, 2009 ፣ መምህረ ልሳነ ግዕዝ ወትርጓሜ መጻሕፍት አዲስ ኪዳን ቅድስትሥላሴ መንፈሳዊ ኮሌጅ.
- [6]. Abdurehman Dawud, “A top-down chart parser for Amharic sentences”, Addis Ababa University, 2015
- [7]. Atelach Alemu., “Automatic Sentence Parsing for Amharic Text: An Experiment Using Probabilistic Context Free Grammars”, unpublished, master's Thesis School of Information Studies for Africa, Addis Ababa University, 2002.
- [8]. Daniel Gochel, “An Integrated Approach to Automatic Complex Sentence Parsing for Amharic Text”, Unpublished, Master Thesis at School of Information Studies for Africa, Addis Ababa, 2003.
- [9]. Abeba Ibrahim, “A Hybrid Approach to Amharic Base Phrase Chunking and Parsing”, master's thesis, Addis Ababa University, Unpublished 2013
- [10]. Diriba Megersa, “Automatic Sentence Parser for Oromo Language Using Supervised Learning Technique”, master's thesis, Addis Ababa University, Unpublished 2002.
- [11]. Abyot Bayu, “Design and Development of Word Parser for Amharic Language”, Addis Ababa University, 2000.

- [12]. Yayneshe Medhin Assefa, “Development of Automatic Parser for Tigrigna Sentences: Using Bottom-Up probabilistic Chart Parser”, Addis Ababa University, 2017
- [13]. Ahmad Al-Taani, Mohammed Msallam, and Sana Wedian, “A Top-Down Chart Parser for analyzing Arabic Sentences”, The International Arab Journal of Information Technology, Vol. 9, No. 3, March 2012
- [14]. Desta Berihu Weldegerigis, “Design and Implementation of Automatic Morphological Analyzer for ge’ez verbs”, Addis Ababa University, 2010.
- [15.] Gebrearegawi Gebremariam Gidey, “Design and Development of Automatic Morphological Synthesizer for Ge’ez Language Verbs” Aksum University, 2019.
- [16]. Yibeltal Chanie, Abebe Belay, “Designing a Stemmer for Ge’ez text Using Rule Based Approach”, Debre Markos University, 2017.
- [17]. Samson Tesfaye, “A Hybrid approach for Machine Translation from Ge’ez to Amharic Language”, St. Mary University, 2020
- [18]. Tadesse Kassa, “Morpheme-Based Bi-Directional Ge’ez -Amharic Machine Translation”, Addis Ababa University, 2018.
- [19]. Dawit Mulugeta “Geez to Amharic Automatic Machine Translation: A Statistical Approach”, Addis Ababa University, 2015.
- [20]. Hagos Gebremedhin Gebremeskel, “Ge’ez POS Tagger Using Hybrid Approach”, Nankai University, college of Software Engineering, December 2019
- [21]. Mesfin Getachew, Automatic part of speech tagging for Amharic: An experiment using Stochastic Hidden Markov (SHM) approach, Master thesis, Addis Ababa University, 2001
- [22] Coulmas, Florian, Writing Systems an Introduction to Their Linguistic Analysis, Deutch's Institute für Japanstudien, Tokyo: Cambridge University Press, 2003.
- [23]. Yitayal, Abate, "Morphological Analysis of Ge’ez Verbs Using Memory Based Learning," A master's Thesis summited to Addis Ababa University, Addis Ababa, 2014.

[24]. አባ ኪዳነ ማርያም ጥዑመ ልሳን, 14ቱ መዝገብ ቅዳሴ, ኢ.አ ኢትዮጵያ: አኩቲት አሳታሚዎች, 2009.

[25]. ባዩ ይማም, የአማርኛ ሰዋስው፣የተሻሻለ ሁለተኛ እትም, አዲስ አበባ ፣ ኢትዮጵያ: ካልቸር ኤንድ አርት ሶላይቲ ኦፍኢትዮጵያ, 2000 አንደ ኢትዮጵያ.አቆጣጠር.

[26] ደሴ በቀለ, , ትንሳኤ ግእዝ, አዲስ አበባ፣ ኢትዮጵያ: በኢትዮጵያ ኦርቶዶክስ ተዋህዶ ቤተ ክርስቲያን በሰንበት ት/ቤቶችማደራጃ መምሪያ ማህበረ ቅዱሳን, 2002፣ በአዲስ አበባ ዩንቨርስቲ የስነ ቋንቋ መምህር.

[27]. Andrea, DeCapua, Grammar for Teachers, A Guide to American English for Native and . Non-Native Speakers, USA, and American: Springer Science + Business Media, LLC, 233 Spring Street, New York, 2008.

[28]. A. Smith, “Novel Estimation Methods for Unsupervised Discovery of Latent Structure in Natural Language Text”, A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy, Baltimore, Maryland October 2006

[29]. Key M., “Functional Grammars”, in proceedings of the fifth Annual Meeting of the Barclay linguistic Society, 1979.

[30]. Allen, J., “Natural Language Understanding, 2nd Ed.”, the Benjamin/ Cummings Publishing Company, Inc., California, 1995.

[31]. Christopher D. Manning, “Foundations of Statistical Natural Language Processing”, The MIT Press Cambridge, Massachusetts London, England, 2000.

[32]. Bala Sundara Raman L, Ishwar S, Sanjeeth Kumar, and Ravin Dranath, “Context free Grammar for Natural Language Constructs – An Implementation for Venpa Class of Tamil Poetry”, Tamil Internet, Chennai, India, 2003.

[33]. T. L. Booth and R. A. Thompson, “Applying Probability Measures to Abstract Language”, IEEE Transactions on Computers C-22, pp 442-450, 1973.

- [34]. Woods William A., "Transition Network Grammars of Natural language Analysis", Communication of the ACM Reprinted in Barbara J. Grosz, Karen, 1970.
- [35]. Nabil Khoufi, Chafik Aloulou and Lamia Hadrich Belguith, "ARSYPAR: A Tool for Parsing the Arabic language based on Supervised Learning", The International Arab Conference on Information Technology (ACIT), 2013.
- [36]. Hafte Abera, Hidden Markov Model Based Tigrigna Speech Recognition", Unpublished Master Thesis, Information Science, Addis Ababa University, Addis Ababa, 2009.
- [37]. A. Al-Taani, M. Msallam and S. Wedian, "A Top-down Parser for Analyzing Arabic Sentences", The international Arab journal of information technology, Vol.9, No.2, March 2012
- [38]. B. Bataineh and E. Bataineh, "An Efficient Recursive Transition Network Parser for Arabic Language", Proceedings of the World Congress on Engineering, London, 2009.
- [39]. Diriba Megersa, "Automatic Sentence Parser for Oromo Language Using Supervised Learning Technique", Unpublished Master Thesis, School of Information Studies for Africa, Addis Ababa University, Addis Ababa, 2002
- [40]. Bird, Steven, Ewan Klein, and Edward Loper. Natural language processing with Python: analyzing text with the natural language toolkit." O'Reilly Media, Inc.", 2009

Appendix

- ወ<Prep>ኢየጎድሩ<V>:እኩያን<N>:ምስሌክ<prep>::
2. ወ<Prep>ኢይነብሩ<V>:ዐማፅያን<N>:ቅድመ<Prep>:አዕይንቲክ<N>::
3. ጸላእክ<V>:እግዚአብሔር<N>:ኩሎሙ<adj>:ገበርተ<V>:ዐመፃ<N>::
4. ብፁእ<Adj>:ብእሲ<N>:ዘኢሖረ<V>:በምክረ<V>:ረሲዓን<N>::
5. ወዘኢቆመ<V>:ውስተ<Prep>:ፍኖተ<N>:ኃጥአን<N>::
6. ወዘኢነበረ<V>:ውስተ<Prep>:መንበረ<N>:መስተሣልቃን<N>::
7. ዘዳእሙ<N>:ሕገ<N>:እግዚአብሔር<N>:ሥምረቱ<V>::
8. ወዘሕጎ<N>:ያነብብ<V>:መዓልተ<N>:ወሌሊተ<N>::
9. ወይከውን<V>:ከመ<Prep>:ዕዕ<N>:እንተ<Prep>:ትክልት<V>:ነበ<Prep>:ሙሐዘ<V>:ማይ<N>::
10. እንተ<Prep>:ትውሀብ<V>:ፍሬሃ<N>:በበ<Prep>:ጊዜሃ<N>::
11. ወኩሎ<Adj>:ዘገብረ<V>:ይፌጽም<V>::
12. አክ<V>:ከመ<Prep>:ዝ<Adj>:ኃጥአን<N>:አክ<V>:ከመ<Prep>:ዝ<Adj>::
13. ዘዳእሙ<Conj>: ከመ<Prep>:መሬት<N>:ዘይግሕፎ<V>:ነፋስ<N>:እምገጸ<N>:ምድር<N>::
14. ወበእንተ<Prep>:ዝ<Adj>:ኢይትሥኡ<V>: ረሲዓን<N>:እምደይን<N>::
15. ወኢኃጥአን<V>:ውስተ<Pre>:ምክረ<V>:ጸድቃን<N>::
16. ለምንት<Adj>:አንገለጉ<V>:አሕዛብ<N>::
17. ወሀዝብኒ<N>:ነበቡ<V>:ከንቶ<N>::
18. ወተንሥኡ<V>:ነገሥተ<N>:ምድር<N>::
19. ወመላእክትኒ<N>:ተጋብኡ<V>:ምስሌሆ < Pre >:ነቡረ<Adv>::
20. ንበትኮ<V>:እምኔነ < Pre>:ማእሰሪሁ<N>::
21. እግዚአ<Adv>:ማ<Adv>:በዝጉ<V>:እለ < Pre>:ይሣቅዩኒ<V>::
22. ቡዙኃን<N>:ቆሙ<V>:ላዕሌየ < Pre>::
23. ቡዙኃን<N>:ይቤልዋ<V>:ለነብስየ<N>::
24. ኢያድላከኒ<V>:አምላክከኒ<N>::
25. አንተሰ<Pro>:እግዚአ<Adv>:ምስካይየ<V>:አንተ<V>::
26. ሶበ < Pre | >:ጸዎፅክዎ<V>:ለእግዚአብሔር<N>:ስምዐኒ<V>:ጽድቅየ<N>::
27. ወእምንዳቤየ<N>:አርሐበ<V>:ሊተ<Adv>::
28. ተሣሀለኒ<V>:ወስምዐኒ<V>:ጸሎትየ<N>::
29. ደቂቀ<N>:ዕንለ<N>:እመሕያው<N>:እስክ < Pre>:ማእዜነ < Adv>:ተከብዱ<V>:ልበኩሙ<N>::
30. ለምንት<Adv>:ታፈቅሩ<V>:ከንቶ<N>:ወሀ < V>:ሐሰተ<N>::
31. አእምሩ<V>:ከመ < Prep>:ተሰብሐ<V>:እግዚአብሔር<N>:በጸድቁ<N>::
32. እግዚአብሔር<N>:ይስምኦኒ<V>:ሶበ < Prep>:ጸራሕኩ<V>:ነቤሁ < Prep>::
33. ቡዙኃን<N>:እለ < Prep>:ይቤሉ<V>:መነ<N>:ያርእየነ<V>:ሠናይቶ<N>::
34. ተዐውቀ<V>:በላእሌነ < Prep>:ብርሃነ<N>:ገጽከ<N>:እግዚአ<Adv>::
35. ወወደይክ<V>:ትፍሥሕተ<N>:ውስተ < Prep>:ልብነ<N>::
36. እምፍሬ<N>:ሥርናይ<N>:ወወይን<N>:ወቅብዕ<N>:በዝጎ<V>::
37. በሰላም<N>:ቡቱ < Pro>:እሰክብ<V>:ወእነውም<V>::

38. እስመ < Prep >: አንተ < Pro >: እግዚአብሔር < Adv >: በተስፋ < Prep >: ባሕር፣ ትክክል < Pro >: አገድረክኒ < V >::
39. ቃልየ < N >: አጽምእ < V >: እግዚአብሔር < Adv >::
40. ወአጽምአኒ < V >: ቃለ < N >: ስእለትየ < N >::
41. ንጉሥየኒ < N >: ወአምላክየኒ < N >::
42. ስመ < N >: ንቤክ < Prep >: እጸሊ < V >: እግዚአብሔር < Adv >::
43. በጽባሕ < N >: ስምዐኒ < V >: ቃልየ < N >::
44. በጽባሕ < N >: እቀውም < V >: ቅድሜክ < Adj >: ወአስተርኢ < V >: ለክ < Adv >::
45. እስመ < Adj >: ኢኮንከ < V >: አምላክ < N >: ዘዐመጸ: የፈቅር::
46. ወእያገድሩ < V >: እኩያን < N >: ምስሌክ < Prep >::
47. ወእይነብሩ < V >: ዐማጽያን < N >: ቅድመ < Prep >: አእይንቴክ < N >::
48. ጸላእክ < V >: እግዚአብሔር < Adv >: ኩሎሙ < Adj >: ገበርተ < V >: ዐመፃ < N >::
49. እግዚአብሔር < Adv >: በመግትክ < N >: ኢትቅሥፈኒ < V >::
50. ወበመቅሠፍትክ < N >: ኢትገጸኒ < V >::
51. ወፈውሰኒ < V >: እስመ < Prep >: ተሀውካ < V >: አዕዕምትየ < N >::
52. ነፍስየኒ < N >: ተሀውከት < V >: ፈድፋድ < Adv >::
53. ወአንተኒ < Pro >: እግዚአብሔር < Adv >: እስክ < Prep >: ማእዜኑ < Adj >::
54. ተመየጥ < V >: ወባልሓ < V >: ለነፍስየ < N >::
55. ወአድንነኒ < V >: በእንተ < Prep >: ምሕረትክ < N >::
56. እስመ < Prep >: አልቦ < V >: በውስተ < Prep >: ሞት < N >: ዘይዜከረክ < V >::
57. ወበሲአልኒ < N >: መነ < Adj >: የአምነክ < V >::
58. ሰራሕክ < V >: በምንዳቤየ < N >::
59. ወአንጽብ < N >: ኩሎ < Adj >: ሌሊተ < N >: ዐራትየ < N >::
60. ወበአንብእየ < N >: አርሓስክ < V >: ምስካብየ < N >::
60. ወበአንብእየ < N >: አርሓስክ < V >: ምስካብየ < N >::
61. እግዚአብሔር < Adv >: አምላክየ < N >: ብክ < Adj >: ተወክልኩ < V >: ኢትግደፈኒ < V >::
62. ወአድንነኒ < V >: አእምኩሎሙ < Adj >: እለ < Adj >: ሮዱኒ < V >: ወባልሓኒ < V >::
63. ከመ < Prep >: እይምሥጥየ < V >: ከመ < Prep >: አንበሳ < N >: ለነፍስየ < N >::
64. እንዘ < Prep >: አልቦ < V >: ዘያድገን < V >: ወዘይባልሕ < V >::
65. እግዚአብሔር < Adv >: አምላክየ < N >: እመሰ < Prep >: ከመ < Prep >: ገለ < Adj >: ገበርኩ < V >::
66. ወእመኒ < Prep >: ቦ < V >: ዐመፃ < N >: ውስተ < Prep >: እደውየ < N >::
67. ወእመኒ < Prep >: ፈደይክዎሙ < V >: ለእለ < Prep >: ይፈድዩኒ < V >: እኩየ < N >::
68. ለያውድቁኒ < V >: ጸላእትየ < N >: እራቅየ < N >::
69. ወይዴግና < V >: ጸራዊ < N >: ለነፍስየ < N >: ወይርከባ < V >::
70. ወይከዳ < V >: ውስተ < Prep >: ምድር < N >: ለሕይወትየ < N >::
71. እግዚአብሔር < Adv >: እግዚእነ < N >: ጥቀ < Adv >: ተሰብሐ < V >: ስምክ < N >: በኩሎ < Adj >: ምድር < N >::
72. እስመ < Prep >: ተለዐለ < V >: ዕበየ < N >: ስብሐቴክ < N >: መልእልተ < N >: ሰማያት < N >::
73. አምአፈ < N >: ደቂቅ < N >: ወህፃናት < N >: አስተዳሎክ < V >: ስብሐተ < N >: በእንተ < Prep >: ጸላኢ < N >::
74. ከመ < Prep >: ትንሥቶ < V >: ለጸላኢ < N >: ወለገፋኢ < N >::
75. እስመ < Prep >: ንጌኢ < V >: ሰማያተ < N >: ግብረ < V >: አጸቢዒክ < N >::
76. ውርን < N >: ወከዋክብት < N >: ዘለሊክ < V >: ሣረርክ < V >::
77. ምንትኑ < V >: ውእቱ < Pro >: ሰብእ < N >: ከመ < Prep >: ትዝክሮ < V >::

- 78.ወምንትነት<V>:ዕንብ<N>:እመሕያው<N>:ከመ< Prep>:ተሐውጸ<V>::
- 79.ሕቀ<Adv>:አሕጸጸኮ<V>:እመላክቲክ<N>::
- 80.ክብረ<N>:ወስብሐት<N>:ከለልኮ<N>::
- 81.እገረ<N>:ለክ<Adv>:እግዚአ<Adv>:በኩሉ<Adj>:ልብየ<N>::
- 82.ወ< Prep>እነግር<V>:ኩሉ<Adj>:ስብሐቲክ<N>::
- 83.እትፌሳሕ<V>:ወእትሐሠይ<V>:ብክ<Pro>::
- 84.ወ< Prep>እዜምር<V>:ለስምክ<N>:ልዑል<N>::
- 85.ሶበ< Prep>:ገብኩ<V>:ጸላእትየ<N>:ድኅሬሆሙ< Prep>::
- 86.ይድወዩ<V>:ወይትሀጉሉ<V>:እምቅድመ< Prep>:ገጽከ<>::
- 87.እስመ< Prep>:ገበርክ<V>:ሊተ<Adv>:ፍትሕየ<N>:ወበቀልየ<N>::
- 88.ወነበርክ<V>:ዲበ< Prep>:መንበርክ<N>:መኩንነ<N>:ጽድቅ<N>::
- 89.ገሠጽኮሙ<V>:ለአሕዛብ<N>:ወተሀጉሉ<N>:ረሲዓን<N>::
- 90.ወ< Prep>ደምስስክ<V>:ስሞሙ<N>:ለዓለመ<N>:ዓለም<N>::
- 91.በ< Prep>እግዚአብሔር<N>:ተወከልኮ<V>:እፎ<Adv>:ትብልዋ<V>:ለነፍስየ<N>::
- 92.አዐይል<V>:ወስተ< Prep>:አድባር<V>:ከመ< Prep>:ፆፍ<N>::
- 93.እስመ< Prep>:ናሁ<Adv>:ኃጥአን<N>:ወሰቁ<N>:ቀስቶሙ<N>::
- 94.ወ< Prep>አስተዳለው<V>:አሕጸቲሆሙ<N>:ወስተ< Prep>:ምጉንጳቲሆሙ<N>::
- 95.ከመ< Prep>:ይንድፍዎሙ<V>:ለርቱአን<N>:ልብ<N>:በጽሚት<Adv>::
- 96.እስመ< Prep>:ናሁ<Adv>:ዘእንተ<Adj>:ሰራዕክ<V>:እሙንቱ<Pro>:ነሠቱ<V>::
- 97.ወጸድቅሰ<N>:ምንተ<Adv>:ገብረ<V>::
- 98.እግዚአብሔር<N>:ወስተ< Prep>:ጽርሐ<N>:መቅደሱ<N>::
- 99.እግዚአብሔር<N>:ወስተ< Prep>:ሰማይ<N>:መንበሩ<V>::
- 100.አድሕነረ<V>:እግዚአ<Adv>:እስመ< Prep>:ጎልቀ:<V>ኔር<N>::
- 101.ወውጎደ:<V>ሃይማኖት:<N>እምዕንለ:<N>እመሕያው<N>::
- 102.ከንቶ:<N>ትናገሩ:<V>አሐዱ:<Adj>ምስለ:< Prep>ካልኩ<Adj>::
- 103.በከናፍረ:<N>ጉሕለት:<N>ልበ:<N>ወበልብ:<N>ይትናገሩ:<V>::
- 104.ይሤርወን:<V>እግዚአብሔር:<N>ለከናፍረ:<N>ጉሕሉት<N>::
- 105.ወለልሳን:<N>እንተ:< Prep>ታዐቢ:<V>ነቢብ<V>::
- 106.እለ:< Prep>ይብሉ:<V>ናዐቢ:<V>ልሳናቲነ<N>::
- 107.እስከ:< Prep>ማእዜኑ:<Adj>እግዚአ:<Adv>ትተስዐኒ:<V>ለግሙራ<Adv>::
- 108.እስከ:< Prep>ማእዜኑ:<Adv>ትመይጥ:<V>ገጽከ<N>እምኔየ< Prep>::
- 109.እስከ:< Prep>ማእዜኑ:<Adv>አነብር:<V>ሀዘነ:<V>ወስተ:< Prep>ነብስየ<N>::
- 110.ወትጼእረኒ:<V>ልብየ:<N>ኩሉ:<Adj>አማረ<Adv>::
- 111.እስከ:< Prep>ማእዜኑ:<Adv>ይትዓበዩ:<V>ጸላእትየ:<N>ላዕለየ< Prep>::
- 112.ነጽረኒ:<V>ወስምዐኒ:<V>እግዚአ:<Adv>አምላኪየ<N>::
- 113.ይበል:<V>እብድ:<N>በልቡ:<N>አልቦ:<V>እግዚአብሔር<N>::
- 114.ጎሰሩ:<V>ወረኩሱ:<V>በምግባሪሆሙ<N>::
- 115.አልቦ:<V>ዘይገብራ:<V>ለሠናይት:<N>አልቦ:<V>ወኢአሐዱ:<Adj>::
- 116.እግዚአብሔር:<N>ሐወጸ:<V>እምሰማይ:<N>ላእለ:< Prep>እንለ:<N>እመሕያው<N>::
- 117.ከመ:< Prep>ይርአይ:<V>እመ:< Prep>ቦ:<V>ጠቢብ:<N>ዘየጎሥሥ<V>ለእግዚአብሔር<N>::
- 118.እግዚአ:<Adv>መኑ:<Adv>የጎድር:<V>ወስተ:< Prep>ጸላሎትክ<N>::
- 119.ወመኑ:<Adj>ያጸልል:<V>ወስተ:< Prep>ደብረ:<N>መቅደስክ<N>::

- 120.ዘየላውር፡<V>በንጹሕ፡<N>ወይገብር፡<V>ጽድቀ<N>።
- 121.ወዘይነብብ<V>፡ጽድቀ፡<N>ለቡ<V>።
- 122.ወዘኢንሕለወ፡<V>በልሳኑ<N>።
- 123.ወዘኢገብረ፡<V>እኩየ፡<N>ዲቦ፡< Prep>ቢጹ<N>።
- 124.ወዘእያጸዐለ፡<V>እዝማዲሁ<N>።
- 125.እቀበኒ፡<V>እግዚአ፡<Adv>እስመ፡< Prep>ከያክ<Pro>ተወክልኩ<V>።
- 126.እብሎ፡<V>ለእግዚአብሔር፡<N>እግዚእየ፡<N>አንተ<Pro>።
- 127.እስመ፡< Prep>ኢትፈቅደ፡<V>ለሠናይትየ<N>።
- 128.ለቅድሳን፡<N>እለ፡< Prep>ውስተ፡< Prep>ምድር<N>።
- 129.ተሰብሐ፡<V>ኩሉ፡<Adj>ሥምረትክ፡<N>በላዕሌሆሙ< Prep>።
- 130.በዝን፡<V>ደዋሆሙ፡<N>ወእምዝ፡<Conj>እስተፋጠኑ<V>።
- 131.እግዚአ፡<Adv>ስምዐኒ፡<V>ጸድቅየ<N>።
- 132.ወአጽመዐኒ፡<V>ጸሎትየ፡<N>ዘኢኮነ፡<V>በከናፍረ፡<N>ጉሕለት<N>።
- 133.እምቅድመ፡< Prep>ገጽክ፡<N>ይወዕኔ፡<V>ፍትሕየ<N>።
- 134.አዕይንትየኒ፡<N>ርእያ፡<V>ጽድቀክ<N>።
- 135.ሐወጽከኒ፡<V>ሌሊተ፡<N>ወፈተንክ፡<V>ለልብየ<N>።
- 136.እመከርከኒ፡<V>ወኢተረከበ፡<V>ዐመፍ፡<N>በላዕሌየ< Prep>።
- 137.ከመ፡< Prep>ኢይንብብ፡<V>አፉየ፡<N>ግብረ፡<V>ዕንሰ፡<N>እመሕያው<N>።
- 138.በእንተ፡< Prep>ቃለ፡<N>ከናፍርክ<N>።
- 139.አፈቀርክ፡<V>እግዚአ፡<Adv>በነይልየ<N>።
- 140.እግዚአብሔር፡<N>ነይልየ፡<V>ወጸወንየ፡<V>ወመድኅንየ<V>።
- 141.አምላክየ፡<V>ወረዳኢያ፡<V>ነይልየ፡<V>ወጸወንየ፡<V>ወመድኅንየ<V>።
- 142.አምላክየ፡<V>ወረዳኢያ፡<V>ወእትዌክል፡<V>ቦቱ<Pro>።
- 143.ወእምእመንየ፡<V>ወቀርኑ፡<V>ሕይወትየ፡<V>ወምስካይየ<>።
- 144.እንተ፡< Prep>ሶቦ፡< Prep>ጽዋዕክም፡<V>ለእግዚአብሔር፡<N>እድኅን፡<V>እምፀርየ<N>።
- 145.አንዘኒ፡<V>ጸዕረ፡<N>ሞት<N>።
- 146.ወውሐዘ፡<V>ዓመፍ፡<N>ሆከኒ<V>።
- 147.ዐገተኒ፡<V>ጸዕረ፡<V>ሲኦል<N>።
- 148.ይስማዕክ፡<V>እግዚአብሔር፡<N>በዕለተ፡<N>ምንዳቤክ<N>።
- 149.ወይቁም፡<V>ለክ፡<Pro>ስሙ፡<N>ለአምላክ፡<N>ያዕቆብ<N>።
- 150.ወይፈኑ፡<V>ለክ፡<Adv>ረድኤተ፡<N>እመቅደሱ<N>።
- 151.ወእምጽዮን፡<N>ይትወከፍክ<V>።
- 152.ወይዘክር፡<V>ለክ፡<Adv>ኩሎ፡< Prep>መሥዋዕትክ<N>።
- 153.ወያጥልል፡<V>ለክ፡<Adv>ቀርባከክ<N>።
- 154.የሀብክ፡<V>እግዚአብሔር፡<N>ዘከመ፡< Prep>ልብክ<N>።
- 155.ወይፈጽም፡<V>ለክ፡<Adv>ኩሎ፡< Prep>ሥምረተክ<N>።
- 156.ወንትፌሣህ፡<V>በአድኅኖትክ<N>።
- 157.ወነዐቢ፡<V>በስመ፡<N>እግዚአብሔር፡<N>አምላክነ<Adj>።
- 158.ወይፈጽም፡<V>ለክ፡<Adv>ኩሎ፡<Adj>ሥምረትክ<N>።
- 159.ወንትፌሣሕ፡<V>በአድኅኖትክ<N>።
- 160.ወነዐቢ፡<V>በስመ፡<N>እግዚአብሔር፡<N>አምላክነ፡<Adj>።
- 161.ወይፈጽም፡<V>ለክ፡<Adv>እግዚአብሔር፡<N>ኩሎ፡<Adj>ስእለትክ፡<N>።
- 162.እግዚአ፡<Adv>በነይልክ፡<N>ይትፌሣሕ፡<V>ንጉሥ፡<N>።

- 163.ወብዝጎ፡<Adv>ይትሓሠይ፡<V>በአድጎኖትከ<N>።
- 163.ወብዝጎ፡<N>ይትሓሠይ፡<V>በአድጎኖትከ<N>።
- 164.ፍትወተ፡<N>ነፍሱ፡<N>ወሀብከ<V>።
- 165.ወስእለተ፡<N>ከናፍሪሀ፡<N>ኢከላእከ<V>።
- 166.እስመ፡< Prep>በጸሕኮ፡<V>በበረከት፡<N>ሠናይ<N>።
- 167.ወአንበርከ፡<V>አክሊለ፡<N>ዲቦ፡< Prep>ርእሱ፡<N>ዘእምዕንቁ፡<N>ክቡር<Adj>።
- 168.ሐዩወ፡<V>ሰአለከ፡<V>ወገብከ<V>።
- 169.ለነዋኅ፡<Adj>መዋዕል፡<N>ለዓለመ፡<N>ዓለም<N>።
- 170.ዐቢይ፡<N>ክብሩ፡<N>በአድጎኖትከ<N>።
- 171.አምላክዮ፡<N>አምላክዮ፡<N>ነጽረኒ፡<V>ወለምንት፡<Adv>ጎደገኒ<V>።
- 172.ርሑቅ፡<Adj>እምእድጎኖትዮ፡<V>ቃለ፡<N>ጎጥአትዮ<N>።
- 173.አምላክዮ፡<N>ጸራሕኮ፡<V>ጎቤከ፡< Prep>ዕለትዮ፡<N>ወኢሰማዕከኒ<V>።
- 174.ወሌሌትዮኒ፡<N>ቅድሜከ፡< Prep>ወጎይልከኒ<N>።
- 175.ኪያከ፡<Pro>ተወከልከ፡<V>አበዊነ<N>።
- 176.ተወከሉከኒ፡<V>ወአድጎንኮሙ<V>።
- 177.ጎቤከ፡< Prep>ጸርሑ፡<V>ወድጎኑ<V>።
- 178.ኪያከ፡<Pro>ተወከሉ፡<V>ወኢተጎፍሩ<V>።
- 179.አንሰ፡<Pro>ዕዪ፡<N>ወአኮ፡<V>ሰብእ<N>።
- 180.እግዚአብሔር፡<N>ይፊእዮኒ፡<V>ወአልቦ፡<V>ዘዮጎጥአኒ<V>።
- 181.ውስተ፡< Prep>ብሔር፡<N>ሥዑር፡<N>ሀዮ፡<Adv>ያጎድረኒ<V>።
- 182.ወጎብ፡< Prep>ማዮ፡<N>ዕረፍት፡<N>ሐፀነኒ<V>።
- 183.ሜጣ፡<V>ለነፍሰዮ፡<N>ወመርሐኒ፡<V>ፍኖተ፡<N>ጽድቅ<N>።
- 184.በእንተ፡< Prep>ስመ፡<N>ዘአሀ<Pro>።
- 185.እመኒ፡< Prep>ሐርኮ፡<V>ማእከለ፡< Prep>ጽላሎተ፡<N>ሞት<N>።
- 186.ኢይፈርሆ፡<V>ለእኩይ፡<N>እስመ፡< Prep>አንተ፡<Pro>ምስሌዮ< Prep>።
- 187.ለእግዚአብሔር፡<N>ምድር፡<N>በምልክ<N>።
- 188.ዓለምኒ፡<N>ወኩሎሙ፡<Adj>እለ፡< Prep>ይብሉ፡<V>ውስቴታ< Prep>።
- 189.ወውእቱ፡<Pro>በባሕር፡<N>ሣረራ<V>።
- 190.ወበአፍላግኒ፡<N>ውእቱ፡<Pro>አጽንክ<V>።
- 191.መኑ፡<Adv>የዐርግ፡<V>ውስተ፡< Prep>ደብረ፡<N>እግዚአብሔር<N>።
- 192.ወመኑ፡<Adv>ይቀውም፡<V>ውስተ፡< Prep>መካነ፡<N>መቅደሱ<N>።
- 193.ዘንጹሕ፡<V>ልቡ፡<N>ወንጽሕ፡<V>እድዊሀ<N>።
- 194.ጎቤከ፡< Prep>እግዚአ፡<Adv>አንቃዕደኮ፡<V>ነፍሰዮ፡<N>አምላኪዮ<N>።
- 195.ወኪያከ፡<Pro>ተወከልከ፡<V>እግዚአ፡<Adv>ወኢይትጎፈር፡<V>ለዓለም<N>።
- 196.ወኢይትሥሐቁኒ፡<N>ጸላእትዮ<N>።
- 197.እስመ< Prep>ኩሎሙ፡<Adj>እለ፡< Prep>ይሴፈው፡<V>ኢይትጎፈሩ<V>።
- 198.ለይትሐፈሩ፡<V>ኩሎሙ፡<Adj>እለ፡< Prep>ይኤብሱ፡<V>ዘልፈ፡<Adv>።
- 199.ፍኖትከ፡<N>እግዚአ፡<Adv>አምረኒ<V>።
- 200.ወአሠረ፡<N>ዘአከ፡<Pro>ምሀረኒ<V>።
- 201.ወምርሐኒ፡<V>በጽድቅከ<N>።
- 202.ፍታሕ፡<V>ሊተ፡<Adv>እግዚአ፡<Adv>እስመ፡< Prep>አንሰ፡<Pro>በየውዑትዮ፡<N>አሐውር<V>።
- 203.ወተወከልከ፡<V>በእግዚአብሔር፡<N>ከመ፡< Prep>ኢይደክም<V>።

- 204.ፍትነኒ:<V>እግዚአ:<Adv>ወስመከረኒ<V>::
- 205.ፍትን:<V>ልብዮ:<N>ወኩልያትየ<N>::
- 206.ከመ:< Prep>ምሕረትከ:<N>ቅድመ:< Prep>አዕይንትየ:<N>ወእቱ<Pro>::
- 207.ወተፈሳሽከ:<V>:በአድንኖትከ<N>::
- 208.ወኢብርከ:<V>ወስተ:< Prep>ዐውደ:<N>ከንቱ<N>::
- 209.ወኢቦአከ:<V>ምስለ:< Prep>ዐማዕያን<N>::
- 210.ፀላእከ:<V>ማኅበረ:<N>እኩያን<N>::
- 211.ወኢይነብር:<V>ምስለ:< Prep>ጽልሕዋን<N>::
- 212.ወአጎሳብ:<V>በንብጽሕ:<N>እደውየ<N>::
- 213.ወአዕውድ:<V>ምሥዋዒከ:<N>እግዚአ<Adv>::
- 214.ከመ:< Prep>እስማዕ:<V>ቃለ:<N>ስብሐቲከ<N>::
- 215.ወከመ:< Prep>ንሣር:< Prep>ኩሎ:<Adj>መንበርክ<N>::
- 216.እግዚአ:<Adv>አፍቀርከ:<V>ሥነ:<N>ቤትከ<N>::
- 217.እግዚአብሔር:<N>ምእመና:<N>ለሕይወትየ:<N>ምንትነ:<Adv>ያደንግፀኒ<V>::
- 218.ሶበ:< Prep>ይቀርቡኒ:<V>እኩያን:<N>ይበልፀኒ:<V>ሥጋየ<N>::
- 219.ጸላእትየስ:<N>እለ:< Prep>ይሣቅዩኒ:<V>እሙንቱ:<Pro>ደከሙ:<V>ወወድቁ<V>::
- 220.እመኒ:< Prep>ጸብአኒ:<V>ተዓይን:<N>ኢይፈርሀኒ<V>ልብየ<N>::
- 221.አሐተ:<Adj>ሰአልክዎ<V>ለእግዚአብሔር:<N>ወኪያሃ:<Pro>ኢየጎሥሥ<V>::
- 222.ከመ:<Prep>አጎድር:<V>ቤቶ:<N>ለእግዚአብሔር:<N>በኩሎ<Adj>መዋእለ:<N>ሕይወትየ<N>::
- 223.ጎቤከ:< Prep>እግዚአ:<Adv>ጸራሕከ:<V>አምላኪየ:<N>ወኢትጸመመኒ<V>::
- 224.ስምዐኒ:<V>እግዚአ:<Adv>ቃለ:<N>ስእለትየ:<N>ዘጸረሕከ:<V>ጎቤከ< Prep>::
- 225.ሶበ:< Prep>አንሥከ:<V>ዕደውየ:<N>በጽርሓ:<N>መቅደስከ<N>::
- 226.ኢትስሐባ:<V>ምስለ:< Prep>ኃጥአን:<N>ለነፍስየ<N>::
- 227.ወኢትግደፈኒ:<V>ምስለ:< Prep>ገበርተ:<V>ዐመፃ<N>::
- 228.እለ:< Prep>ይትናገሩ:<V>ሰላመ:<N>ምስለ:< Prep>ቢጸሙ<N>::
- 229.አምጽኦ:<N>ለእግዚአብሔር:<N>ውሎደ:<N>አማልክት<N>::
- 230.አምጽኦ:<V>ለእግዚአብሔር:<N>ዕጉሎ:<N>ሐረጊት<N>::
- 231.ስገዱ:<V>ለእግዚአብሔር:<N>በዐጸደ:<N>መቅደሱ<N>::
- 232.ቃለ:<N>እግዚአብሔር:<N>ላዕለ:<N>ማያት<N>::
- 233.አምላክ:<N>ስብሐት:<N>አንጉድጉደ<V>::
- 234.እግዚአብሔር:<N>ላዕለ:< Prep>ማያት:<N>በኩሉ<Adj>::
- 235.ቃለ:<N>እግዚአብሔር:<N>በዐብይ:<Adj>ስብሐት<N>::
- 236.ወይቀጥቅጦ:<V>እግዚአብሔር:<N>ለአርዘ:<N>ለባኖስ<N>::
- 237.አአከተከ:<V>እግዚአ:<Adv>እስመ:< Prep>ተወከፍከኒ<V>::
- 238.ወስላተ:<V>ጸላዕየ:<N>ኢረሰይከኒ<V>::
- 239.እግዚአ:<Adv>አውባእካ:<V>እምሲኦል:<N>ለነፍስየ<N>::
- 240.ወአድንንከኒ:<V>እምእለ:< Prep>ይወርዱ:<V>ወስተ:< Prep>ግብሩ<N>::
- 241.ዘምሩ:<V>ለእግዚአብሔር:<N>ጸድቃኑ<N>::
- 242.ወግነዩ:<V>ለዘድረ:<N>ቅሱሳቱ<N>::
- 243.ኪያከ:<Pro>ተወከልከ:<V>እግዚአ:<Adv>ወኢይትጎፈር:<V>ለዓለም<N>::
- 244.ወበጽድቅከ:<N>አንግፈኒ:<V>ወባልሐኒ<V>::
- 245.አጽምዕ:<V>እዘነከ:<N>ጎቤየ:< Prep>ወፍጡነ:<Adv>አድሕነኒ<V>::

- 246.ከነት፡<V>አምላክየ፡<N>ወመድጎንየ<N>።።
- 247.ወቤተ፡<N>ጸወንየ፡<N>ከመ፡<>ታድጎነኒ<V>።።
- 248.እስመ፡< Prep>ኔይልየ፡<Adj>ወጸንየ፡<Adj>አንተ<Pro>።።
- 249.በእንተ፡< Prep>ስምከ፡<N>ምርሐኒ፡<V>ወሌስየኒ<N>።።
- 250.ቤዝወኒ፡<V>እግዚአብሔር፡<N>አምላክ፡<N>ጽድቅ<N>።።
- 251.ሶበ፡<Prep>ተጋብኡ፡<V>ጎብረ፡<Adv>ላዕሌየ< Prep>።።
- 252.ቡፁግን፡<Adj>እለ፡< Prep>ተጎድጉ፡<V>ሎሙ፡<Adv>ጎጢአቶሙ<V>።።
- 253.ወእለ፡< Prep>ኢሐሰበ፡<V>ሎሙ፡<Adv>ኩሎ፡<Adj>ጌጋየሙ<N>።።
- 254.ቡፁፀ፡<Adj>ብእሲ፡<N>ዘኢጎሎቁ፡<V>ሎቱ፡<Adv>እግዚአብሔር<N>።።
- 255.ወዘአልቦ፡<V>ጽልሐት፡<N>ወስተ፡< Prep>ልቡ<N>።።
- 256.እስመ፡< Prep>አርመምኩ፡<V>በልያ፡<V>አእዕምትየ<N>።።
- 257.እምጎበ፡< Prep>እጸርሕ፡<V>ኩሎ፡<Adj>አሚረ<Adv>።።
- 258.እስመ፡< Prep>መዓልተ፡<N>ወሌሊተ፡<N>ክብደት፡<V>እዴከ፡<N>ላዕሌየ< Prep>።።
- 259.ወተመየጥኩ፡<V>ለሕርትምና፡<N>ሶበ፡< Prep>ወግአኒ፡<V>ሦክ<N>።።
- 260.ተፈሥሐ፡<V>ጸድቃን፡<N>በእግዚአብሔር<N>።።
- 261.ወለራትግን፡<Adj>ይደልዎሙ፡<V>ክብር<N>።።
- 262.ግነዩ፡<V>ለእግዚአብሔር፡<N>በመሰንቆ<N>።።
- 263.ወበመዝሙር፡<M>ዘዐሠርቱ፡<Adj>አወታሪሁ፡<N>ዘምሩ፡<V>ሎቱ<Adj>።።
- 264.ወሰብሕዎ፡<V>ስብሐተ፡<V>ሐዲስ<Adj>።።
- 265.ሠናየ፡<Adj>ዘምሩ፡<V>ወየብቡ፡<V>ሎቱ<Adj>።።
- 266.እስመ፡< Prep>ጽድቅ፡<N>ቃሉ፡<N>ለእግዚአብሔር<N>።።
- 267.ወኩሎ፡<Adj>ግብሩ፡<N>በሃይማኖት<N>።።
- 268.ያፈቅር፡<V>እግዚአብሔር፡<N>ጽድቀ፡<N>ወምጽዋተ<N>።።
- 269.እባርኮ፡<V>ለእግዚአብሔር፡<N>በኩሎ፡<Adj>ጊዜ<N>።።
- 270.ወዘልፈ፡<Adv>ስብሐተ፡<V>ወስተ፡< Prep>አፉየ<N>።።
- 271.በእግዚአብሔር፡<N>ትክብር፡<V>ነፍተስየ<N>።።
- 272.ይሰምዑ፡<V>የዋሃን፡<Adj>ወይትፈሥሐ<V>።።
- 273.አዕበይዎ፡<V>ለእግዚአብሔር፡<N>ምስሌየ< Prep>።።
- 274.ወናልዕል፡<V>ስም፡<N>ጎብረ<Adv>።።
- 275.ጎሠሥክዎ፡<V>ለእግዚአብሔር፡<N>ወተሰጥወኒ<V>።።
- 276.ወእምኩሎ፡<Adj>ምንዳቤየ፡<N>አድጎነኒ<V>።።
- 277.ግፍገሙ፡<V>እግዚአ፡<Adv>ለእለ፡< Prep>ይገፍዑኒ<V>።።
- 278.ጽብኦሙ፡<V>እግዚአ፡<Adv>ለእለ፡< Prep>ይጸብኡኒ<V>።።
- 279.ንሣእ፡<V>ወልታ፡<N>ወኩናተ፡<N>ወተንሥእ፡<V>ወስተ፡< Prep>ረዲአትየ<N>።።
- 280.ምላሳ፡<V>ሰይፈከ፡<N>ወዕግቶሙ፡<V>ለእለ፡< Prep>ሮዳኒ<V>።።
- 281.በላ፡<V>ለነፍስየ፡<N>አነ፡<Pro>ወእቱ፡<Pro>ረዲአኪ<V>።።
- 282.ለይትጎፈሩ፡<V>ወይጎሰሩ፡<V>ኩሎሙ፡<Adj>እለ፡< Prep>የጎሥሥዎ፡<V>ለነፍስየ<N>።።
- 283.ለይግብኡ፡<V>ድጎሪሆሙ፡< Prep>ወይትጎፈሩ።።<V>እለ፡< Prep>
>መከሩ፡<V>እኩየ፡<N>ላዕሌየ< Prep>።።
- 284.ለይኩኑ፡<V>ከመ፡< Prep>ጸበል፡<N>ዘቅድመ፡< Prep>ገጸ፡<N>ነፋስ<N>።።
- 285.ይነብብ፡<V>ኃጥእ፡<N>በዘቀያስሕት፡<V>ርእሶ<N>።።
- 286.ወአልቦ፡<V>ፍርሀተ፡<N>እግዚአብሔር፡<N>ቅድመ፡< Prep>አዕይንቲሁ<N>።።
- 287.እስመ፡< Prep>ጸልሐወ፡<N>በልሳነ<N>።።

- 288.ሰበ፡< Prep>ትረክቦ፡<V>ጎጢሎቱ፡<N>ይጸልካ<V>።
- 289.ቃለ፡<N>አፉሁ፡<N>ዐመፍ፡<N>ውኑሕሉት<N>።
- 290.ወኪፈቀደ፡<V>ይለቡ<V>ከመ፡< Prep>ያሠኒ<V>።
- 291.ውጎለየ፡<V>ዐመፍ፡<N>በውስተ፡< Prep>ምስካቡ<N>።
- 292.ወቆመ፡<V>በፍኖት፡<N>እንተ፡< Prep>በኩሉ፡<Adj>ኪኮነት፡<V>ሠናይተ<N>።
- 293.ወአይትሀክያ፡<V>ለእኪት<N>።
- 294.እግዚአ፡<Adv>በሰማየ፡<N>ሣህልክ<N>።
- 295.ኢትቅናእ፡<V>ላዕለ፡< Prep>እኩያን<Adj>።
- 296.ወኢትቅናእ፡<V>ላዕለ፡< Prep>ገበርተ፡<V>ዐመፍ<N>።
- 297.እስመ፡< Prep>ከመ፡< Prep>ሣዕር፡<N>ፍጡነ፡<Adv>ይየብሱ<V>።
- 298.ወከመ፡< Prep>ሐመልማል፡<N>ሐምል፡<N>ፍጡነ፡<Adv>ይውድቁ<V>።
- 299.ተወከል፡<V>በእግዚአብሔር፡<N>ወግበር፡<V>ሠናየ<N>።
- 300.ወያጎድረክ፡<V>ዲበ፡< Prep>ምድር<N>።
- 301.ወይርዕየክ፡<V>ዲበ፡< Prep>ምድር<N>።
- 302.ተፈሣሕ፡<V>በእግዚአብሔር፡<N>ወይውሀበክ፡<V>ስእለተ፡<N>ልብክ<N>።
- 303.ክሥት፡<V>ለእግዚአብሔር፡<N>ፍኖተክ<N>።
- 304.ተወከል፡<V>ቦቱ፡<Pro>ወውእቱ፡<Pro>ይገብር፡<V>ለክ<Pro>።
- 305.እግዚአ፡<Adv>በመከትክ፡<N>ኢትቅሰፈኒ<V>።
- 306.ወመቅሰፍትክ፡<N>ኢትገሥጸኒ<V>።
- 307.እስመ፡< Prep>አሕጸክ፡<V>ደጉጸኒ<V>።
- 308.ወአጽናእክ፡<V>እዴክ፡<N>ላዕሌየ< Prep>።
- 309.ወአልቦ፡<V>ፈውስ፡<N>ለሥጋየ፡<N>እምገጸ፡<N>መዓትክ<N>።
- 310.ወአልቦ፡<V>ሰላም፡<N>ለአዕዕምትየ፡<N>እምገጸ፡<N>ጎጥከትየ<N>።
- 311.እስመ፡< Prep>ኖኅ፡<N>እምሥእርተ፡<N>ርእሰየ፡<N>ጌጋይየ<N>።
- 312.ከመ፡< Prep>ጾር፡<N>ክቡድ፡<N>ከብደ፡<V>ላዕሌየ< Prep>።
- 313.እቤ፡<V>አዕቅብ፡<V>አፉየ፡<N>ከመ፡< Prep>ኪይስሐት፡<V>በልሳንየ<N>።
- 314.ወአንበርኩ፡<V>ዐቃቤ፡<V>ለአፉየ<N>።
- 315.ሰበ፡< Prep>ይትቃወሙኒ፡<V>ጎጥክን፡<N>ቅድሜየ< Prep>።
- 316.ተጸመምኩ፡<V>ወአትሓትኩ፡<V>ርእሰየ<N>።
- 317.ወአርመምኩ፡<V>ለሠናይ<N>።
- 318.ወተሓደሰ፡<V>ቁስልየ<N>።
- 319.ወሞቀኒ፡<V>ልብየ፡<N>በውስጥየ<N>።
- 320.ጸኒሓ፡<V>ጸናሕክዎ፡<V>ለእግዚአብሔር<N>።
- 321.ስምዐኒ፡<V>ወተመየጠኒ<V>።
- 322.ወሰምዐኒ፡<V>ቃለ፡<N>ስእለትየ<N>።
- 323.ወአውዕአኒ፡<V>እምዐዘቅተ፡<N>ሕርትምና<N>።
- 324.ወእጽቡር፡<V>እምዐዘቅተ፡<N>ሕርትምና<N>።
- 325.ወአቀሞን፡<V>ውስተ፡< Prep>ኩኩሕ፡<N>ለእገርየ<N>።
- 326.ወአጽንኦን፡<V>ለመከየድየ<N>።
- 327.ወወደየ፡<V>ውስተ፡< Prep>አፉየ፡<N>ስብሓተ፡<N>ሓዲሰ<Adj>።
- 328.ስብሓቲሁ፡<N>ለአምላክነ<N>።
- 329.ይርአዩ፡<V>ቡዙጎን፡<Adj>ወይፍርሁ<V>።
- 330.ወይትወክሉ፡<V>በእግዚአብሔር<N>።

- 331.ቡፁእ፡<Adj>ብእሲ፡<N>ዘስመ፡<N>እግዚአብሔር፡<>Nትውክልቱ<V>።
- 332.ቡፁእ፡<Adj>ዘይለቡ፡<V>ላዕሊ፡< Prep>ነዳይ፡<N>ወምስኪን<N>።
- 333.እምዕለት፡<N>እኪት፡<Adj>ያድኅኖ፡<V>እግዚአብሔር<N>።
- 334.እግዚአብሔር፡<N>የዐቅቦ፡<V>ወየሐይዎ፡<V>ወያብዕዖ፡<V>ዲቦ፡< Prep>ምድር<N>።
- 335.ወኢያገብኦ፡<V>ወስተ፡< Prep>እደ፡<N>ጸላኢሁ<N>።
- 336.እግዚአብሔር፡<N>ይፈድኦ፡<V>ወስተ፡< Prep>አራተ፡<N>ሕማሙ<N>።
- 337.ወይመይጥ፡<V>ሎቱ፡<Adv>ኩሎ፡<Adj>ምስካቢሁ፡<N>እምደዌሁ<N>።
- 338.አንሰ፡<Pro>እቤ፡<V>እግዚአ፡<Adv>ተሳላላሊ<V>።
- 339.ከመ፡< Prep>ያፈፈቅር፡<V>ሆላ፡<Adj>ነብ፡< Prep>አንቅዕተ፡<N>ማያት<N>።
- 340.ከማሁ፡< Prep>ታፈቅር፡<V>ነፍስየ፡<N>ነብ፡< Prep>እግዚአብሔር<N>።
- 341.ጸምአት፡<V>ነፍስየ፡<N>ነብ፡< Prep>አምላኪየ፡<N>ሕያው<Adj>።
- 342.ማእዜ፡<Adv>እበጽሕ፡<V>ወእሬኢ፡<V>ገጽ፡<N>ለአምላክየ<N>።
- 343.ሲሰየ፡<N>ኮነኒ፡<V>አንበዕየ፡<N>መዓልተ፡<N>ወሌሊተ<N>።
- 344.እስመ፡< Prep>ይብሉኒ፡<V>ኩሎ፡<Adj>አሚረ፡<Adv>አይቴ፡<Adv>ውእቱ፡<Pro>አምላክክ<N>።
- 345.ዘንተ፡<Adj>ተዘከረየ፡<V>ተከዕወት፡<V>ነፍስየ፡<N>በላዕሌየ< Prep>።
- 346.እስመ፡< Prep>እበውእ፡<V>ወስተ፡< Prep>መካን፡<N>ማኅደረ፡<N>ስብሕት፡<N>ቤተ፡<N>እግዚአብሔር<N>።
- 347.በቃለ፡<N>አሚን፡<N>ወበትፍሥህት፡<N>ደምፁ፡<V>እለ፡< Prep>ይገብሩ፡<V>በዓለ<V>።
- 348.ለምንት፡<Adv>ትቴክዚ፡<V>ነፍስየ፡<N>ወለምንት፡<Adv>ተሀውከኒ<V>።
- 349.እመኒ፡< Prep>በእግዚአብሔር፡<N>ከመ፡< Prep>እገኒ፡ሎቱ<Adv>።
- 350.እምሕዝብ፡<N>ወፁኣን፡<V>እምጽድቅ<N>።
- 351.እምብእሲ፡<N>ዐማዒ፡<V>ወጽልው፡<V>ባልሐኒ<V>።
- 352.እስመ፡< Prep>አንተ፡<Pro>አምላኪየ፡<N>ወነይልየ<N>።
- 353.ለምንት፡<Adv>ተነድገኒ፡<V>ወለምንት፡<Adv>ትኩዝየ፡<V>አንሰሰ፡<V>።
- 354.ሶቦ፡< Prep>ያመነድቡኒ፡<V>ፀርየ<N>።
- 355.ፈኑ፡<V>ብርሃንክ፡<N>ወጽድቀክ<N>።
- 356.እማንቱ፡<Pro>ይምርሓኒ፡<V>ወይሰዳኒ፡<V>ደብረ፡<N>መቅደስክ<N>።
- 357.ወወስተ፡< Prep>አብያቱክ፡<N>አግዚአ<Adv>።
- 358.እበውእ፡<V>ነብ፡< Prep>ምሥዋዒሁ፡<N>ለእግዚአብሔር<N>።
- 359.ነብ፡< Prep>አምላኪየ፡<N>ዘአስተፍሥሐ፡<V>ለውርዙትየ<N>።
- 360.እገኒ፡<V>ለከ፡<Adv>አምላክየ፡<N>በመሰንቆ<N>።
- 361.ለምንት፡<Adv>ትቴክዚ፡<V>ነፍስየ፡<N>ወለምንት፡<Adv>ተሀውከኒ<V>።
- 362.እመኒ፡< Prep>በእግዚአብሔር፡<N>ከመ፡< Prep>እገኒ፡<V>ሎቱ<Adv>።
- 363.መድኅኔ፡<N>ገጽየ፡<N>አምላክየ<N>።
- 364.እግዚአ፡<Adv>ሰማዕነ፡<V>በእዘኒነ<N>።
- 365.ወአበዊነሂ፡<N>ዜነውነ<V>።
- 366.ግብረ፡<N>ዘገበርከ፡<V>በመዋዕሊሆሙ፡<N>በመዋዕለ፡<N>ትካት<N>።
- 367.እዴክ፡<N>ሀረወቶሙ፡<V>ለፀር፡<N>ወተክልከ፡<V>ኪያሆሙ<Pro>።
- 368.ሣቀይከሙ፡<V>ለአሕዛብ፡<N>ወሰደድኮሙ<V>።
- 369.ዘአዐ፡<V>በኩናቶሙ፡<N>ወረስዋ፡<V>ለምድር<N>።
- 370.ወመዝራእቶሙኒ፡<N>ኢያድኅኖሙ<V>።
- 371.ዘእንበለ፡< Prep>የማንክ፡<N>ወመዝራዕትክ፡<N>ወብርሃነ፡<N>ገጽክ<N>።

- 372.እስመ:< Prep >ተሳላላሁ<V>::
- 373.አንተ:<Pro>ውእቱ:<Pro>ንጉሥየ:<N>ወአምላክየ<N>::
- 374.በአዝዝከ:<V>መድኅኒቶ:<N>ለያዕቆብ<N>::
- 375.ብከ:<Pro>ንወግአሙ:<V>ለኩሎሙ:<Adj>ፀርነ<Adj>::
- 376.ጉሥኦ:<V>ልብየ:<M>ቃለ:<N>ሠናየ<N>::
- 377.ወአነ:<Pro>አየድዕ:<V>ግብርየ:<N>ለንጉሥ<N>::
- 378.ከመ:< Prep>ቀለመ:<N>ጸሐፊ:<Adj>ዘጠበጠበ:<V>ይጽሕፍ:<V>ልሳንየ<N>::
- 379.ይህኒ:<V>ላሕዩ:<N>እምውሉደ:<N>ዕንለ:<N>እመሕያው<M>::
- 380.ተክዕወ:<V>ሞገስ:<N>እምከናፍሪክ<N>::
- 381.በእንተ:< Prep>ዝ:<Pro>ባረከከ:<V>እግዚአብሔር:<N>ለዓለም<N>::
- 382.ቅንት:<V>ሰይፈከ:<N>ንያል:<N>ውስተ:< Prep>ሐጃክ<N>::
- 383.በሥንክ:<N>ወበላሕይክ<N>::
- 384.አርትዕ:<V>ተሠራሕ:<V>ወንገሥ<V>::
- 385.በእንተ:< Prep>ጽድቅ:<N>ወርትዕ:<N>ወየውሁት<N>::
- 386.ወይመርሐክ:<V>ስብሐተ:<V>የማንክ<N>::
- 387.አሕጻክ:<V>ስሐል:<N>(፩):<V>ንያል<N>::
- 388.አምላክነ:<N>ንይልነ:<N>ወጸወንነ<N>::
- 389.ወረዳኢነ:<V>ውእቱ:<Pro>ለምንዳቤነ:<V>ዘረከበነ:<V>ፈድፋደ<Adv>::
- 390.በእንተ:< Prep>ዝ:<Pro>እንፈርህ:<V>ለእመ:< Prep>አድለቅለቀት:<V>ምድር<N>::
- 391.ወለእመኒ:< Prep>ፈለሱ:<V>አድባር:<N>ውስተ:< Prep>ልበ:<N>ባሕር<N>::
- 392.ደምፀ:<V>ወተሐምጉ:<V>ማያቲሆሙ:<N>::
- 393.ወአድለቅለቁ:<V>አድባር:<N>እምንይሉ<N>::
- 394.ፈልግ:<N>ዘይውሕዝ:<V>ያስተፌስሕ:<V>ሀገረ:<N>እግዝአብሔር<N>::
- 395.ቀደሰ:<V>ማኅደር:<N>ልዑል<Adj>::
- 396.እግዚአብሔር:<N>ውስተ:< Prep>ማእከላ:< Prep>ኢትትሀወክ<V>::
- 397.ወይረድኦ:<V>እግዚአብሔር:<N>ፍጽመ<N>::
- 398.ኩልክሙ:<Adj>አሕዛብ:<Adj>ጥፍሐ:<V>እደዊክሙ:<N>::
- 399.ወየብቡ:<V>ለእግዚአብሔር:<N>በቃለ:<N>ትፍስሕት<N>::
- 400.እስመ:< Prep>ልዑል:<Adj>ወግሩም:<Adj>እግዚአብሔር<N>::
- 401.ወንጉሥ:<Adj>ዐብይ:<Adj>ውእቱ:<Pro>ሲበ:< Prep>ኩሎ:<Adj>ምድር<N>::
- 402.አግረረ:<V>ለነ:<Adv>አሕዛብ:<Adj>ወሕዝበ:<Adj>ታሕተ:<N>እገሪነ<N>::
- 403.ወንረየነ:<V>ሎቱ:<Adv>ለርስቱ<N>::
- 404.ሥኖ:<N>ለያዕቆብ:<N>ዘአፍቀረ<V>::
- 405.ፀርገ:<V>እግዚአብሔር:<N>በይባቤ<N>::
- 406.ወእግዚእነ:<N>በቃለ:<N>ቀርን<N>::
- 407.ዘምሩ:<V>ለአምላክነ:<N>ዘምሩ<V>::
- 408.ዘምሩ:<V>ለንጉሥነ:<N>ዘምሩ<V>::
- 409.ዐብይ:<V>እግዚአብሔር:<N>ወበዝገን:<V>አኩሎት<N>::
- 410.በሀገረ:<N>አምላክነ:<N>በደብረ:<N>መቅደሱ<N>::
- 411.ይዔዝዝ:<V>ተፍሥሕተ:<V>ለኩሎ:<Adj>ምድር<N>::
- 412.አድባረ:<N>ጽዮን:<N>በገቦ:<N>መስዕ<N>::
- 413.ሀገሩ:<N>ለንጉሥ:<N>ዐብይ<Adj>::
- 414.እግዚአብሔር:<N>የአምር:<V>ክበዲሃ:<Adj>ሶበ:< Prep>ተመጠውዋ<V>::

415. እስመ፡<Prep>ናሁ፡<Adv>ነገሥተ፡<Adj>ምድር፡<N>ተጋብኡ፡<V>ወመጽኡ፡<V>ጎብረ<Adv>።
416. እመንቱስ፡<Pro>ዘንተ፡<Adv>ርአዮሙ፡<V>አንከሩ፡<V>።
417. ደንገ፡<V> ወፈርሁ፡<V>፤ ወአጎዘሙ፡<V>ረሲዓን<N>።
418. ወሐሙ፡<V>በሀየ፡<Adv>ከመ፡< Prep>እንተ፡< Prep>ትወልድ፡<N>።
419. ስምዑ፡<V>ዘንተ፡<Adv>ኩልክሙ፡< Adj>እለ፡< Prep>ትነብሩ፡<N>ውስተ፡<Prep>ዓለም፡<N>
420. በበ፡< Prep>በሐውር፡<N>ደቂቀ፡<N>ዕንለ፡<N>እመሕያው፡<N>።
421. አብልዕትኒ፡<N>ወነዳይኒ፡<N>።
422. አፋዮ፡<N>ይነግር፡<V>ጥበበ<N>።
423. ወጎሊና፡<N>ልብየ፡<N>ምክረ<N>።
424. አጽምዕ፡<V>ምሳሌ፡<N>በእዘንየ<N>።
425. ወእከሥት፡<V>በመዝሙር፡<V>ነገርየ<N>።
426. ለምንት፡<Adv>እፈርህ፡<V>እምዕለት፡<N>እከት፡<N>።
427. ጎጢአት፡<N>ሰከኖየ፡<N>ዐገተኒ፡<V>።
428. እለ፡< Prep>ይትአመኑ፡<V>በጎይሎሙ፡<N>።
429. ወይዜጎሩ፡<V>በብዝኅ፡<N>ብዕሎሙ፡<N>።
430. አምላክ፡<N>አማልክት፡<N>እግዚአብሔር፡<N>ነበበ፤ <N>ወጸውዓ፡<V>ለምድር<N>።
431. እምሥራቀ፡<N>ፀሐይ፡<N>እስከነ፡< Prep>ዐረብ<N>።
432. ወእምጽዮን፡<N>ሥነ፡<N>ስብሐቲሁ፡<N>።
433. እግዚአብሔርሰ፡<N>ገሃደ፡<Adv>ይመጽእ<V>።
434. ወአምላክነሂ፡<N>እያረምም<V>።
435. እሳት፡<N>ይነድድ፡<V>ቅድሜሁ፡< Prep>።
436. ዐውዶሂ፡<V>ዐውሎ፡<N>በዙጎ፡<Adj>።
437. ይጼውኦ፡<V>ለሰማይ፡<N>በላዕሉ፡< Prep>።
438. ወለምድርኒ፡<N>ከመ፡< Prep>ይከንን፡< Prep>ሕዝብ<N>።
439. አስተጋብኡ፡<V>ሎቱ፡<Adv>ጸድቃኑ፡<N>።
440. እለ፡< Prep>ይገብሩ፡<V>መሥዋዕተ፡<N>ዘበሕጉ፡<N>።
441. ይነግራ፡<V>ሰማያት፡<N>ጽድቀ፡<N>ዚአሁ፡<Pro>።
442. እስመ፡< Prep>እግዚአብሔር፡<N>መከንን፡<N>ውእቱ፡<V>።
443. ተሳህለኒ፡<V>እግዚአ፡<Adv>በከመ፡< Prep>ዕባየ፡<N>ሳህልክ<N>።
444. ወበከመ፡< Prep>ቡዝኅ፡<V>ምሕረትክ፡<N>ደምስስ፡<V>ጎጢአትየ<N>።
445. ጎዕበኒ፡<V>ወአንጽሐኒ፡<V>እምጎጢትየ<N>።
446. ወእምስበሳየ፡<N>አንጽሐኒ፡<V>።
447. እስመ፡< Prep>ለልየ፡<Pro>አአምር፡<V>ጌጋይየ<N>።
448. ወጎጢአትየኒ፡<N>ቅድሜየ፡< Prep>ውእቱ፡<V>በኩሉ፡< Prep>ግዜ<>።
449. ለክ፡<Adv>ለባሕቲ፡<Pro>አበስከ፤ <V>ወእኩየኒ፡<N>በቅድሜክ፡< Prep>ገበርኩ፡<V>።
450. ከመ፡< Prep>ትጽድቅ፡<V>በነቢብክ፡<N>ወትማእ፡<V>በኩነኔክ<N>።
451. እስመ፡< Prep>ናሁ፡<Adv>በጎጢአት፡<N>ተፀነሰክ፡<V>።
452. ወበዐመ፡<N>ወለደተኒ፡<V>እምየ<N>።
453. እስመ፡< Prep >ናሁ፡<Adv>ጽድቀ፡<Pro>አፍቀርክ<V>።
454. ዘኢይትነብብ፡<V>ጎብኡ፡<Adj>ጥበብክ፡<N>አይዳዕከኒ፡<V>።
455. ትንዘህኒ፡<V>በአዛብ፡<N>ወእንጽሕ<V>።

- 456.ተጎዕበኒ:<V>እምበረድ:<N>ወእጸዐዱ.<V>::
- 457.ታሰምዐኒ:<V>ትፍሥሕተ:<N>ወሐሣተ<N>::
- 458.ወይትፊሥሐተ:<V>አዕዕምጸተ:<N>ጸድቃን<V>::
- 459.ሚጥ:<V>ገጽክ:<N>እምጎጥአትየ<N>::
- 460.ወደምስስ:<V>ሊተ:<Adv>ኩሎ:<Adj>አበሳየ<N>::
- 461.በ:< Prep>ንጹሐ:<N>ፍጥር:<V>ሊተ:<Adv>እግዚአብሔር:<Adv>::
- 462.መንፈሰ:<N>ርቱዐ:<Adj>ሐድስ:<V>ውስተ:< Prep>ከርሥየ<N>::
- 463.ኢትግደፈኒ:<V>እምቅድመ:< Prep>ገጽክ<>::
- 464.ወመንፈስክ:<N>ቅዱስ:<Adj>ኢታውዕእ:<V>እምላዕሌየ< Prep>::
- 465.ለምንተ:<Adv>ይዜጎር:<V>ጎዶል:<Adj>በአክዩ<N>::
- 466.ወይዲምፅ:<V>ኩሎ:<Adj>አሚረ<Adv>::
- 467.ጎጠአተ:<N>ጎለየ:<V>ልብክ<N>::
- 468.ከመ:< Prep>መላጸ:<N>በሊጎ:<Adj>ገበርክ:<V>ሕብለ<N>::
- 469.አብደርክ:<V>እኪተ:<N>እምሠናይት<N>::
- 470.ወትዲምፅ:<V>እምትንብብ:<V>ጽድቀ<N>::
- 471.አፍቀርክ:<V>ኩሎ:<Adj>ነገረ:<N>ልሳን:<N>መስጠመ<N>::
- 472.በእንተ:<Conj>ዝ:<Pro>ይነሥተክ:<V>እግዚአብሔር:<N>ለዝሉፉ<Adv>::
- 473.ይመልጎክ:<V>ወያፈልሰክ:<V>እምቤትክ<N>::
- 474.ወሥርወከኒ:<N>እምድረ:<N>ሕያዋን<N>::
- 475.ይርአዩ:<V>ጸድቃን:<N>ወይፈርሁ<V>::
- 476.ይሥሐቁ:<V>ላዕሌሁ:< Prep>ወይበሉ<V>::
- 477.ነዋ:<Adv>በእሲ:<Adj>ዘኢረሰዩ:<V>ለእገዝአብሔር:<N>ረዳኢሁ<N>::
- 478.ወተአመነ:<V>በብዝነ:<V>በዕሉ<N>::
- 479.ወተኃለየ:<V>በከንቱ<N>::
- 480.ይብል:<V>አብድ:<N>በልቡ:<N>አልቦ:<V>እግዚአብሔር<N>::
- 481.ጎስሩ:<V>ወረኩሱ:<V>በጌጋዮመ<N>::
- 482.አልቦ:<VV>ዘይገብራ:<V>ለሠናይት<N>::
- 483.እግዚአብሔር:<N>ሐወጸ:<V>እምሰማይ:<N>ዲበ:< Prep>ዕንለ:< Prep>እመሕያው<N>::
- 484.ከመ:< Prep>ይርአይ:<N>እመ:< Prep>ቦ:<V>ጠቢብ:<N>ዘየጎሥሥ:<V>ለእግዚአብሔር<N>::
- 485.ኩሎ:<Adj>ዐረየ:<V>ወጎብረ:<Adv>ዐለወ<V>::
- 486.አልቦ:<V>ዘይገብራ:<V>ለሠናይት:<N>አልቦ:<V>ወኢአሐዱ:<Adj>::
- 487.ወኢየአምሩ:<V>ኩሎመ:<Adj>ገበርተ:<V>ዐመፀ<N>::
- 488.እለ:<Adj>ይውጎጥዎመ:<V>ለሕዝብየ:<N>ከመ:< Prep>ብልዐተ:<V>እክል<N>::
- 489.ወለእግዚአብሔር:<N>ኢጸውእዎ<V>::
- 490.ወበሀየ:<Adv>ፈርሁ:<V>ወገረሞመ:<V>ዘኢኮነ:<V>ግሩመ<N>::
- 491.እስመ:< Prep>እግዚአብሔር:<N>ዘረወ:<V>አዕዕተሆመ:<N>ለመደልዋን<Adj>::
- 492.እግዚአብሔር:<Adv>በስምክ:<N>አድጎነኒ<V>::
- 493.ወበጎይልክ:<Adv>ፍታሕ:<V>ሊተ<Adv>::
- 494.እግዚአብሔር:<Adv>ስምዐኒ:<V>ጸሎትየ<N>::
- 495.ወአጸምፅ:<V>ቃለ:<N>አፉየ<N>::
- 496.እስመ:< Prep>ነኪራን:<Adj>ቆመ:<V>ላዕሌየ< Prep>::
- 497.ወጎዶላን:<Adj>ጎሥሥዋ:<V>ለነፍስየ<N>::

- 498.ወኢረሰይዎ:<V>ለእግዚአብሔር:<N>ቅድሚያም.< Prep>::
- 499.ናሁ:<Adv>እግዚአብሔር:<N>ይረድስኒ.<V>::
- 500.ወእግዚአብሔር:<N>ያድገና:<V>ለነፍስየ.<N>::

Grammar Rule Function

```

from nltk.parse import ChartParser
grammarGeez = nltk.CFG.fromstring("""
S -> VP NP | Conj Prep N V | N N N | adj V N | N V N | NP VP | NP Prep VP | Prep VP | N N | F
VP -> V Prep | Prep adj V | V Prep V | V | V Prep adv | Prep adv | V Prep | V N | V V | V Prep
V -> "አኮ" | "ዘይግሕፎ" | "ኢይትሥኡ" | "ወኢኃጥአን" | "ምክረ" | "አንገለጉ" | "ነበቡ" | "ወተንሥኡ"
Prep -> "ከመ" | "ከመ" | "ከመ" | "ወበእንተ" | "ውስተ" | "ምስሌህ" | "እምኔነ" | "እለ" | "ላዕሌየ" | "ሶበ
NP -> adj N VP | Conj Prep | N N | N V N | N V | N | adv adv | N VP | Pro adv | N N N Prep ac
Conj -> "ዘዳእመ"
adj -> "ዝ" | "ዝ" | "ለምንት" | "ቅድሚካ" | "ኩሎ" | "መኑ" | "ኩሎ" | "ብከ" | "እምኩሎ" | "እ
N -> "ኃጥአን" | "መሬት" | "ነፋስ" | "እምገጸ" | "ምድር" | "ረሲዓን" | "እምደይን" | "ጸድቃን" | "አሐዛብ"
adv -> "ነቡረ" | "እግዚአ" | "ሚ" | "ሊተ" | "ማእዜነ" | "ለከ" | "ፈድፋድ" | "ሕቅ" | "እፎ" | "በጽሚት
Pro -> "አንተሰ" | "ቡቱ" | "አንተ" | "ባሕቲትከ" | "ውእቱ" | "ብከ" | "እሙንቱ" | "አማረ"

```

```

def getTree(phraseToBeparsed, grammar, squareBrackets):
    t=""
    sent= phraseToBeparsed.split()
    rd_parser=nltk.ChartParser(grammar)
    for tree in rd_parser.parse(sent):
        t+=str(tree)+"\n"
    if squareBrackets ==1:
        t=t.replace("(","[").replace(")","","]")
    return t

```

Parsed Sentences

```

print("102.ወ መኑ ያጸልል ውስተ ደብረ መቅደስከ")
print(getTree("ወ መኑ ያጸልል ውስተ ደብረ መቅደስከ", grammarGeez, usesquareB

102.ወ መኑ ያጸልል ውስተ ደብረ መቅደስከ
[S
 [VP [Prep ወ] [VP [adj መኑ] [VP [V ያጸልል] [VP ]]]]
 [NP [Prep ውስተ] [NP [N ደብረ] [NP [N መቅደስከ]]]]]

```

Figure5. 1: Parsed Sentences

Syntax Tree Generator

```
[S
 [VP [Prep ወ] [VP [adj መኑ] [VP [V ያጸልል] [VP ]]]]
 [NP [Prep ውስተ] [NP [N ደብረ] [NP [N መቅደስ]]]]]
```

(C) 2011 by [Miles Shang](#), see [license](#).

Options

Help

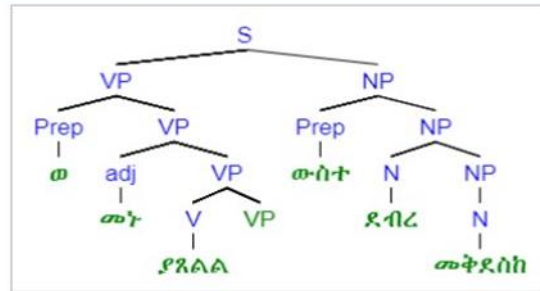


Figure5. 2: Example of the parsed sentence

```
print("109.እብሎ ለ እግዚአብሔር እግዚአብሔር እንተ")
print(getTree("እብሎ ለ እግዚአብሔር እግዚአብሔር እንተ", grammarGeez, usesquareBrackets))
```

```
109.እብሎ ለ እግዚአብሔር እግዚአብሔር እንተ
[S
 [VP [V እብሎ] [Prep ለ]]
 [NP [N እግዚአብሔር] [NP [N እግዚአብሔር] [NP [Pro እንተ]]]]]
```

Figure5. 3: Sample sentence

Syntax Tree Generator

```
S
 [VP [V እብሎ] [Prep ለ]]
 [NP [N እግዚአብሔር] [NP [N እግዚአብሔር] [NP [Pro እንተ]]]]]
```

(C) 2011 by [Miles Shang](#), see [license](#).

Options

Help

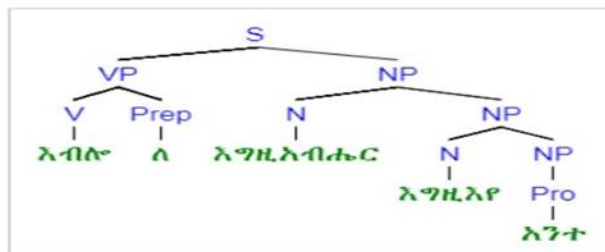


Figure5. 4:Tree representation of the parsed sentences

```
print("9. ወመላእክትኒ ተጋብኡ ምስሌሆ ኅቡረ")
print(getTree(" ወመላእክትኒ ተጋብኡ ምስሌሆ ኅቡረ", grammarGeez, usesquareBrackets))
```

4 ወመላእክትኒ ተጋብኡ ምስሌሆ ኅቡረ
 [S [NP [N ወመላእክትኒ] [V ተጋብኡ]] [VP [Prep ምስሌሆ] [adv ኅቡረ]]]

Figure5. 5Sample Parsed sentence

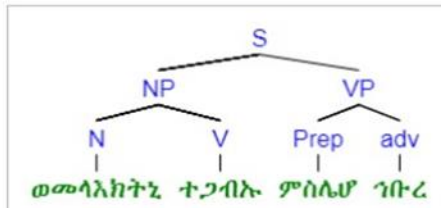


Figure5. 6Tree representation of the parser

```
print("36.ጸላእክ እግዚአ ኩሎሙ ገበርተ ዐመፃ")
print(getTree("ጸላእክ እግዚአ ኩሎሙ ገበርተ ዐመፃ", grammarGeez, usesquareBrackets))
```

4.ጸላእክ እግዚአ ኩሎሙ ገበርተ ዐመፃ
 [S [VP [V ጸላእክ] [adv እግዚአ] [VP [adj ኩሎሙ] [V ገበርተ]]] [NP [N ዐመፃ]]]

Figure5. 7Sample parsed sentence

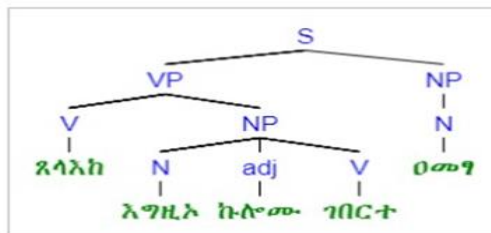
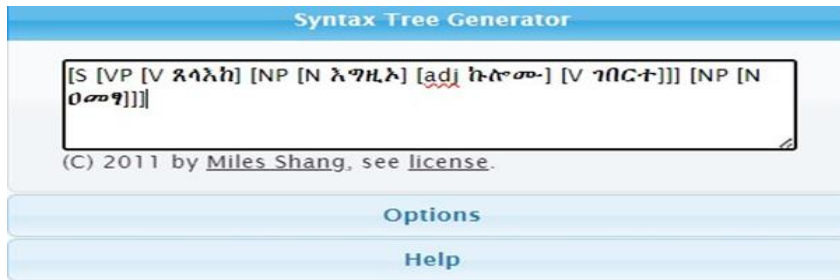


Figure5. 8: Tree representation of parsed sentence

Syntax Tree Generator

[S [NP [N ወዘሕን]] [VP [V ያነብብ] [NP [N መዓልተ] [N ወሌሊተ]]]]

(C) 2011 by [Miles Shang](#), see [license](#).

Options

Help

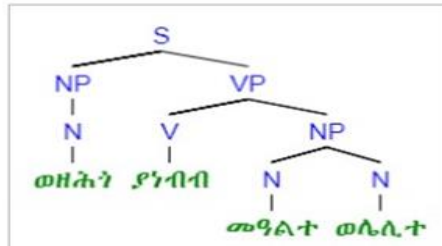


Figure5. 9: Tree representation of the parsed sentence

Syntax Tree Generator

[S [VP [Prep ሰበ] [V ጸዋዕክዎ]] [NP [N ለእግዚአብሔር] [VP [V ስምዐኔ] [N ጽድቅየ]]]]

(C) 2011 by [Miles Shang](#), see [license](#).

Options

Help

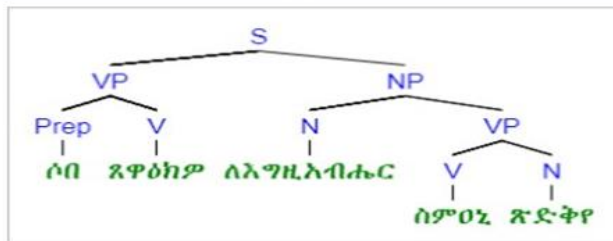


Figure5. 10: Examples of some parsed sentence

```
print("130.ይሰማዕክ እግዚአብሔር በዕለተ ምንዳቤክ")
print(getTree("ይሰማዕክ እግዚአብሔር በዕለተ ምንዳቤክ", grammarGeez, usesquareBrackets))
```

```
130.ይሰማዕክ እግዚአብሔር በዕለተ ምንዳቤክ
[S [VP [V ይሰማዕክ] ] [NP [N እግዚአብሔር] [NP [N በዕለተ] ] [NP [N ምንዳቤክ]]]]]
```

Figure5. 11: Examples some parsed sentence

[S [VP [V ደስግዕስ]] [NP [N እግዚአብሔር] [NP [N በዕለተ] [NP [N ምንጻቤክ]]]]]

(C) 2011 by Miles Shang, see license.

Options

Help

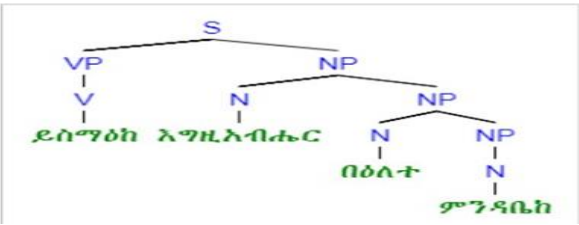


Figure5. 12: Tree representation of the parser

```

print("81.እግዚአብሔር ውስተ ጽርሐ መቅደሱ")
print(getTree("እግዚአብሔር ውስተ ጽርሐ መቅደሱ", grammarGeez, usesquareBrackets))

```

81.እግዚአብሔር ውስተ ጽርሐ መቅደሱ

Figure5. 13: Some examples of parsed sentence