



Hawassa University
ሀዋሳ ዩኒቨርሲቲ

**DESIGN OF A MORPHOLOGICAL GENERATOR AND ANALYZER FOR SIDAAMU
AFOO VERBS USING FINITE-STATE TRANSDUCER**

MASTER'S PROGRAM (MSC)

KITAW AYELE GESUMA

HAWASSA UNIVERSITY

INSTITUTE OF TECHNOLOGY

FACULTY OF INFORMATICS, DEPARTMENT OF COMPUTER SCIENCE

HAWASSA, ETHIOPIA

NOVEMBER, 2021

**DESIGN OF A MORPHOLOGICAL GENERATOR AND ANALYZER FOR SIDAAMU
AFOO VERBS USING FINITE-STATE TRANSDUCER**

KITAW AYELE GESUMA

**A THESIS SUBMITTED TO THE
DEPARTMENT OF COMPUTER SCIENCE,
HAWASSA UNIVERSITY, INSTITUTE OF TECHNOLOGY, SCHOOL OF
GRADUATE STUDIES
HAWASSA UNIVERSITY
HAWASSA, ETHIOPIA
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE IN COMPUTER SCIENCE**

NOVEMBER, 2021

Declaration

I hereby declare that this MSc Specialty or equivalent thesis is my original work and has not been presented for a degree in any other university, and all sources of material used for this thesis/dissertation have been duly acknowledged.

Name:- Kitaw Ayele Gesuma

Signature: _____

This MSc Specialty or equivalent thesis has been submitted for examination with my approval as thesis advisor.

Name:- Tesfaye Bayu (PhD)

Signature: _____

Place and Date of Submission: _____

SCHOOL OF GRADUATE STUDIES

HAWASSA UNIVERSITY

ADVISORS' APPROVAL SHEET

This is to certify that the thesis entitled “**Design of a Morphological Generator and Analyzer for Sidaamu Afoo Verbs Using Finite-State Transducer**” was submitted in partial fulfillment of the requirements for the degree of Master's Degree with specialization in COMPUTER SCIENCE for, the Graduate Program of the Department COMPUTE SCIENCE, and has been carried out by Kitaw Ayele (ID NO. GPCoScR/0009/11), is conducted under my supervision. Therefore I recommend that the student has fulfilled the requirements and hence hereby can submit the thesis to the Department.

Tesfaye Bayu (PhD) _____

Name of major advisor

Signature

Date

Name of co-advisor (if there is any)

Signature

Date

SCHOOL OF GRADUATE STUDIES

HAWASSA UNIVERSITY

EXAMINERS' APPROVAL SHEET

We, the undersigned, members of the Board of Examiners of the final open defense by Kitaw Ayele Gesuma have read and evaluated his/her thesis entitled **“Design of a Morphological Generator and Analyzer for Sidaamu Afoo Verbs Using Finite-State Transducer**, and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirements for the degree.

<u>Tesfaye Bayu (PhD)</u>	_____	_____
Name of Major Advisor	Signature	Date
_____	_____	_____
Name of Internal Examiner-I	Signature	Date
Andargachew Mekonnen	_____	_____
Name of Internal Examiner-II	Signature	Date
<u>Wondwossen Mulugeta (PhD)</u>	_____	_____
Name of External examiner	Signature	Date
_____	_____	_____
SGS Approval	Signature	Date

Final approval and acceptance of the thesis is contingent upon the submission of the final copy of the thesis to the School of Graduate Studies (SGS) through the Department/School Graduate Committee (DGC/SGC) of the candidate's department.

ACKNOWLEDGEMENT

First of all, I would like to praise the almighty God for his mercy and grace for letting me accomplish this thesis work.

Secondly, my heartfelt gratitude should go to my advisor Dr. Tesfaye Bayu for his close follow-up in my work. I am thankful for his time spent and irreplaceable efforts exert to support my thesis work. This long journey without his advice and professional support might not have come true. His support has been from the beginning while I start the selection of my thesis topic, for all of which, his support was greatly appreciated.

Thirdly, I am grateful for respect and love for my family who are my lifeline and I owe them for what I have now. And their advice empowers me to work hard and gives me the way to look better into the invisible world. They were also beside me in all circumstances and times of need. Special my elder brothers Adane Ayele and Alemu Ayele played a great role in my success. My deepest gratitude should also go to my lovely Dad and Mom for their closest follow-up and appreciation.

Fourthly, I would like to thanks Mr. Mathewos W/giorgis, Ass.Pro, Department of Sidaamu Afoo of Hawassa University, for his earnest support related to Sidama Afo linguistic. He was never hesitant about my repeated request and quests. My gratitude must go to Mr. Damitew Dawit from the same Department has also helped me a lot, My heartfelt gratitude must go to him.

Last but not least, I would like to thanks Hawassa University for the sponsorship and opportunity that I was given to be here.

Table of Contents

ACKNOWLEDGEMENT	V
Table of Contents	VI
LIST OF TABLES	IX
LIST OF FIGURES	X
LIST OF ACRONYMS AND ABBREVIATIONS.....	XI
ABSTRACT.....	XIII
CHAPTER ONE: Introduction	1
1. Background.....	1
1.1 Research motivation.....	3
1.2 Statement of the problem	3
1.3 Research Questions.....	6
1.4 Objective	6
1.4.1 General objective	6
1.4.2 Specific objective.....	6
1.5 The significance of the study	7
1.6 Scope and Limitation of the study	8
1.6.1 Scope.....	8
1.6.2 Limitation.....	8
1.7 Research outcome/deliverable	8
1.8 Research design	9
1.9 Organization of the thesis	9
CHAPTER TWO: Literature review.....	10
2. Introduction.....	10
2.1 CONCEPTS AND TERMINOLOGIES	10
2.1.1 Introduction.....	10
2.1.2 MORPHEME.....	11
2.1.3 Affixes.....	12
2.1.4 Clitics	15
2.1.5 Morphotactic	16
2.1.6 Types of morphology	17
2.1.7 Morphophonemic	18
2.1.8 Epenthesis	19
2.1.9 Metathesis	20
2.1.10 Assimilation	22

2.1.11	Computational morphology	23
2.1.12	Morphological generator and analyzer.....	23
2.2	Related work	28
2.2.1	Local languages.....	28
2.2.2	Non-Ethiopian languages	30
CHAPTER THREE: Overview of Sidaamu Afoo language.....		31
3.2	Phonological overview.....	31
2.1	Morphological Overview of Sidaamu Afoo.....	35
2.1.1	Verb morphology	39
2.1.1.1	mono-consonantal	40
2.1.1.2	Mono-syllabics.....	40
2.1.1.3	Disyllabic verbs.....	41
2.1.1.4	Converb.....	42
2.1.1.5	Copula (Verb to be) /ikkate gurda/.....	42
2.1.1.6	The verb of presences	43
2.1.1.7	Possession verb	44
2.1.1.8	Verb of Imperative	44
2.1.1.9	Verb of jussive/hajajote gurda duuchaho/.....	45
2.1.1.10	Negative marker in verbs	45
2.1.1.11	Negative of tense aspects	46
2.1.1.12	Negative of imperative and jussive.....	46
2.1.1.13	Negative jussive verb.....	47
2.2	Verbal inflection	47
2.2.1	Gender suffix	49
2.2.2	Number suffix in verb	50
2.2.3	Tense/Aspect.....	50
2.3	Summary	53
2.4	Order of the suffixes	53
2.5	Stem Expansion	54
2.5.1	Causative.....	54
2.5.2	Auto-benefactive.....	56
2.5.3	Passive.....	56
CHAPTER FOUR: Research Design and Methodology		57
4.1	Introduction.....	57
4.2	Methodology	57

4.3	Implementation tools and techniques.....	58
4.4	Source of data	58
4.5	Lexicon design.....	59
4.6	Irregular verb.....	59
4.7	Alteration rule	60
4.8	Tags.....	60
4.9	Continuous class	61
4.10	Rules design.....	66
4.10.1	Epenthesis Sound.....	66
4.10.2	Nasal metathesis rule	66
4.10.3	Replacement rule.....	67
4.11	Graphical Representation of whole design.	68
4.12	Evaluation of Morphological System	69
CHAPTER FIVE: Implementation and Discussion.....		69
5.1	Word generation by doubling the last consonant.....	71
5.2	Nasal metathesis sound ‘n’ and ‘m’ for non-geminate words.....	72
5.3	Imperative verb result	74
5.4	Jussive verb result.....	75
5.5	Replacement.....	76
5.6	Morphological Analyzer	77
5.7	Morphological Generator.....	78
5.8	Overall evaluation.....	80
CHAPTER SIX: Conclusion and Future work		80
6	Conclusion	80
6.1	Future work.....	82
REFERENCES		83
APPENDIX I: MORPHOLOGICAL ALTERNATION RULE:		87
APPENDIX II: WEB CRAWILLING CODE		91
APPENDIX III: SIDAAMU AFOO ROOT WORD LIST OF VERBS		94

LIST OF TABLES

Table 3.1: The phoneme inventory of Sidaamu Afoo adopted from [50].	31
<i>Table 3.2: The vowel phonemes, adopted from the document [50].</i>	31
<i>Table 3.3: Tense suffix for all person.</i>	53
Table 5.1 Grammatical geminate consonant 1prp word generation	71
Table 5.2: Nasal metathesis ‘n’ sound insertion for grammatical information.....	73
Table 5.3: Imperative negative marker suffix.....	75
<i>Table 5.4: positive jussive attachment generation</i>	75
Table 5.5: last consonant replacement and experimental result presentation	76

LIST OF FIGURES

Figure 4.1: Diagrammatic view of Morphological generator and analyzer	68
Figure 5.1: FST graph word-formation process from the root word.	72
Figure 5.2: experimental output of the morphological analyzer.....	78
Figure 5.3: Morphological generated word forms from the single root word	79

LIST OF ACRONYMS AND ABBREVIATIONS

1prn	First person neutral
-1prp	First person plural
-1prs	First person singular.
-2prp	Second person plural
-2prs	Second person singular.
-3prp	Third person plural
-3prs	Third person singular.
ACC	Accusative
-adj	Adjective
C	Consonant
CAUS	Causative class
-Ctense	Continuous tense (/harinshote aan nota/)
dcause	Double causative
DV	Derived verbs
-F	Feminine
FSA	Finite state Automata
FST	Finite state transducer
IMP	Imperative class
INF	Infinitive
M	Masculine gender.
MT	Machine translation
N	Noun

NLP	Natural language processing
NLTK	Natural language toolkit
NMSC	Negative marker suffix class
Nom	Nominative
NP	Noun phrase
NUM	Number class
-Pperfect	Present perfect (/taaga/)
-pre/fut	Present and Future tense (/xa woyim albira/)
PRSN	Person class
PVS	Politeness verb suffix
-Spast	Simple past (/tene mule/)
TENIN	Tense inflection class
V	Vowel
-V	Verb class
/ /	Expression Sidaamu Afoo words
‘ ’	Equivalent English translation

ABSTRACT

Sidaamu Afoo is an official language of the newly formed sidaama national state. It is one of the under-resourced languages of Ethiopia. There are very small attempts of study in the morphological perspective of the language. The morphological study is a low-level activity that paves the way for the development of high-level NLP applications. This study was interested in the design and implementation of a morphological generator and analyzer for the language verbs. The finite state transducer is used to design the verb morphological generator and analyzer system. Lexicon formalism is used for designing method and foma programming language as implementation toolkit. Forty-one distinct linguistic rules were used to form those variate word forms. The experimental result shows that out of 74,934 total generated words, 94.8% of words are correctly generated and analyzed by the system. Only 5.2% of words generated are invalid and wrongly analyzed. The factors of the problem are optional replacement rules and compiler ignorance of the glottal consonant of the language. Strict demarcation boundary for optional replacement rules and alternative techniques for glottal sound management left for future work.

Keywords: *Sidaamu Afoo , Finite state transducer, Morphological study.*

CHAPTER ONE: Introduction

1. Background

Language is the means of communication for human beings. The communication can be oral or in written form to convey information. Now a day, technology, integrating different community members from different parts of the world without communication barriers. Different actors playing their roles in their discipline to realize the expected communication platform for everybody. So language is one of the strong bonds between these variant communities. The study of language and adding technological value to it is very important.

There are two (genetic and typological) broad world language classification methods that have subcategories within them. The genetic classification method classifies the language based on the evolutionary history of the language that passed through to get present features to describe with. The typological classification method takes the structural characteristics of the language to categorize into the respective category. Typological language classification can be isolating, agglutinative, and inflected/fusional. Isolating language has no morphologically analyzable words means that the words have only one morpheme. A morpheme in which the smallest meaning (lexical or grammatical) bearing component of the word. An agglutinative language is characterized by the words that are the result of a combination of the morphemes that contain corresponding grammatical meaning within it. Inflected/fusional languages have no one-to-one grammatical meaning correspondence to constituent morphemes.

There are 86 languages spoken in Ethiopia having millions of native speakers, a few of which are Amharic, Afaan Oromo, and Sidaamu Afoo [1]. According to the 2007 National census, Sidaamu Afoo has around 2.9 million native speakers settled in the southern part of the country. The information specified in the census above is too far to describe the current population size.

The Joshua Project website assumed that the current population size is above 4.9 million [2]. According to typological world language classification Sidaamu Afoo is an agglutinative language with inflectional and derivational morphological affixation [3] [4]. Each word in the language is morphologically rich and formed by reduplication and affix (prefix, suffix, infix, and circumfix) addition mechanisms. The language belongs to the Cushitic linguistic family and uses the Latin alphabet for writing since the insurance of national linguistic policy in 1992.

This study is interested in the morphology of the language. Morphology is all about the structure of words. Linguist uses morphological analysis to derive grammatical detailed information about words that enables the user to understand what the word means. The morphological generator creates new words from the base form attaching different morphemes to it. A morpheme can be free or bound, based on the meaning that they give. Free morphemes can stand alone and have lexical meaning. While bound morphemes cannot stand alone and needs free morpheme to exist and change the grammatical meaning of the word [5]. Such morphological information is useful for different natural language processing applications, like in machine translation, speech synthesis, semantic analysis, POS tagger, spelling correction, dictionary development, grammar checking, and information extraction [6] [7]. The computer needs a program to understand the word and related information to support the above tasks. A morphological analyzer is a special type of software that disintegrate and helps to analyze the given word into its constituent parts (morpheme) and the generator works in the reverse order. This study will investigate a finite state transducer approach for Sidaamu Afoo morphological generator and analysis.

1.1 Research motivation

The researcher is one of the community members of this language speaker and needs to contribute something to the language to make the language more valuable in his discipline. The first reason that motivates the researcher to begin a study on this topic is, there is only one morphological analyzer work done on the complex language using a supervised memory-based learning approach with small data set. First of all, this doesn't assure the stability of the morphological analyzer for the language. The second reason is paving the way for those who are interested in working on the Sidaamu Afoo language incorporating morphological generator and analyzer concept in their study. A morphological generator and analyzer is a fundamental tool for many natural language processing applications. Hence, the opportunity to make machine translation among local languages that may facilitate smooth communication between different ethnic groups of Ethiopia or foreign languages becomes familiar for easy information exchange. The researcher believes that the avoidance of communication barriers in different ethnic groups may calm down the political turmoil.

1.2 Statement of the problem

The study on the given language may contribute something to other languages or may not at all [8]. According to the author, the nature of each language based on its orthography, phonology, and morphology makes unique variations among the different languages. Based on their features languages are classified into different language families. Most horns' of Africa languages are classified into the Afro-Asiatic family. They are morphological complex with a large number of cross-border speakers. Those horns' of Africa languages

from Ethiopia, Eritrea, Djibouti, and Somalia lack well-prepared computational resources, such that they are not computationally developed to the anticipation [9]. Some Ethiopian languages are undertaken morphological analyzer studies by different scholars at different times. For example, the Afaan Oromo morphological analyzer is developed for the application of natural language processing using a corpus-based approach [10] [11]. An automatic morphological analyzer for Amharic text using an unsupervised and auto-segmental approach was designed and developed [12]. The Supervised memory-based learning approach was used for Sidaamu Afoo as the first morphological analyzer implementation [13]. Supervised memory-based learning approach tries to get similarities between new instance and previously stored data which is probabilistic method [14]. There is some challenging notion related with supervised learning approaches. The first one probabilistic model can be easily affected by overfitting which leads to poor system performance. Second computational time consumption is too high. The Author explained that he used a limited number of data set to test his Sidaamu Afoo morphological analyzer system. The corpus-based approach needs a large amount of data set to develop an effective morphological analyzer for the language but a limited data set doesn't guaranty a stable morphological analyzer for the language. Finite state transducer has many plenty room for those drawbacks happen in supervised learning approach. By the nature finite state transducer approach has language specific rule defining component which increase the accuracy of the system performance in addition to less computational complexity. In previous work there is no morphological generator component which is integral part of morphological study. Finite state transducer has beauty of working in both direction (generator and analysis) with small implementation modification. On the other side, Sidaamu Afoo has its unique word-formation structure and orthographic representation. All

words are represented with appropriate alphabetic symbols that obey all rules required by the language. It supports consonant germination and the vowels can be long or short base on constituent sounds for grammatical purposes. The misrepresentations of the required alphabets bring loss of information or it may lead to wrong meaning. For example */gowa/* meaning in English 'sew' and */gowwa/* meaning in English 'fool' */goowa/* meaning in English 'neck' or */lalo/* 'cattle' and */laalo/* 'fruits' has a completely different meaning. So to avoid such kind of problem spelling correction, needs a successful language morphological analyzer. The morphological generator and analyzer support the spelling correction system to help native speakers to convey their thought through the correct script. The Sidaama national regional state working language is Sidaamu Afoo but all civil servants who are working for are not Sidaamu Afoo speakers. The situation leads to the migration of skilled employees from the region due to the lack of language knowledge. As a solution machine translation application plays an important role. That is why a morphological generator and analyzer is important right now for Sidaamu Afoo. The lack of computational resources for the language leads the researcher to select a finite state base rule approach to conduct experimental research on the language. The finite state rule-based approach provides the accurate result of generation and analysis even there are difficulties during linguistic hand-constructed rules [15]. The reasons are to choose a finite state rule-based approach for this specific language: 1) Sidaamu Afoo has no sufficient resource or very small resource for computational language applications. 2) To handle the complexity of the language and easy management of data scattering. This approach needs a clear understanding of the language to explain easily. There is good study progress that has been done on the Sidaamu Afoo linguistic aspects. But the computation facets of the language are at a very low level. In the computational aspects, the most fundamental language-related

activities that take the language to some extent of expectation have not been done. For this reason, the researcher has proposed to put a milestone by developing a morphological generator and analyzer for the language to flourish in the natural language processing (NLP) area. Building a morphological generator and analyzer lays the foundation for the language as it is the lowest level text processing technique and provides support for further development of NLP applications in this language.

1.3 Research Questions

1. How to approach computational aspects of the verbs and what are the challenges in the language while using FST?
2. To what extent is FST important for Sidaamu Afoo verbs morphological generator and analyzer design and implementation?

1.4 Objective

1.4.1 General objective

The general objective of this study is to design a Morphological generator and analyzer for Sidaamu Afoo verbs.

1.4.2 Specific objective

The specific objective of the study to achieve the general objective includes:

- To collect documents to identify word-formation structure in the language.
- To develop a lexicon for the experimental purpose.

- To define relevant language-specific rules that govern the morphotactic structure of the words.
- To evaluate the developed morphological generator and analyzer system using FST.
- To report the experimental result.

1.5 The significance of the study

Many NLP applications rely on the morphological analyzer. Therefore the development of the Sidaamu Afoo morphological analyzer helps other NLP tasks to be done for the language. So this study lays a foundation for proceeding development of higher-level natural language processing applications. For example for machine translation application development for language to any other languages, because it plays a vital role. The importance of the study compresses the demand of individuals to the community. The existence of a morphological analyzer for the language paves the way for the community to experience new computational linguist applications to make their life simple. Individuals who want to develop different natural language applications for Sidaamu Afoo will use this morphological analyzer as a constituent of their work. For anybody, the existence of applications enables them to do something or to have interaction with others regardless of communication barriers. This motivates individuals to do more. Generally, the study takes the development of language one step up.

1.6 Scope and Limitation of the study

1.6.1 Scope

The scope of the study is limited to the design and development of a morphological generator and analyzer for Sidaamu Afoo in a finite state transducer approach. This study focuses only on the verb part of speech.

1.6.2 Limitation

There is a different subclass in the part of speech in the language within closed or open classes like an adjective, adverb, and so on. Due to the shortage of time, all parts of speech are not covered in the study.

1.7 Research outcome/deliverable

The study provides different outcomes from those the first one is the word-formation structure identification for the language. The second one is the scripted lexicon and defined rules for language. The third outcome of the study is that the efficient and effective morphological generator and analyzer for the Sidaamu Afoo language using a finite state transducer approach. The effectiveness of the study will be measured by comparison between source word forms and expertise evaluation morphological generator and analyzer outputs in addition to comparison with the words in the collected document.

1.8 Research design

The nature of this study is experimental. To experiment anticipated objective of the study whether achievable accordingly or not, encompasses different experimental activities. The system is designed base on the FST approach. The input of the system is a surface word for morphological analysis and root word with relevant tags for morphological generator and internal the system looks for morphemes. The information related to each morpheme is grasped by the system and determined based on the rule. The system generates a different word form with detailed information of the morpheme according to their morphotactic.

1.9 Organization of the thesis

The rest of this work is organized as follows: chapter 2 Literature review, here is concepts, terminology, and approaches to design and implement morphological analyzer and generator. Chapter 3 overall language related discussion and morphological structure about Sidaamu Afoo. Chapter 4 Methodology and design issue considered in this work. Chapter 5 Morphological generator and analyzer implementation using foma software and discussion. Conclusion and future work discussed in Chapter 6.

CHAPTER TWO: Literature review

2. Introduction

The morphological analysis deals with detailed information in which looking into every component involved to construct a word. The reverse is true for a morphological generator that constructs different word forms from a single root word by attaching a variety of morphemes to the root or stem. The morphological analyzer is all about the extraction of each feature in the word with a corresponding grammatical meaning, which depicts the part of what it means for the word. It is an integral part of many high-level natural language processing applications like MT, information retrieval, dictionary development, compression, spelling check, and so [16]. A morphological analyzer is a software that acts on the word to provide the relevant information regarding the input word. A morphological generator is also software that constructs variants of words from the same root. This study focuses on Sidaamu Afoo verb structure, formation, and morphological generator, and analyzer implementation.

2.1 CONCEPTS AND TERMINOLOGIES

2.1.1 Introduction

Theoretical morphology is the base foundation of morphological study and computational morphology illustrates the use of computers for language-related tasks [17]. The word morphology is made up of two different Greek words Morph means ‘form/shape/structure’ and logy means ‘study [18] [19]. Morphology deals with the internal constituent form/shape/structure of a word. The building block of the word form/structure is the morpheme. A morpheme is the least indivisible part of a word that contains some

information to convey stored in the word. Word is a fundamental component of a language that can exist alone [20]. For example word **/basure/** meaning in English ‘cats’ is made up of two morphemes **/basu/** meaning in English ‘cat’ and **/-re/** English equivalent suffix ‘-s’ morpheme **/-re/** ‘-s’ shows that the word is plural.

2.1.2 MORPHEME

The idea of morpheme was discussed early by Baudoin de Courtenay in 1859 and Sweet in 1876 to refer smallest meaningful linguistic unit [21]. As stated above the morpheme is the part of the word that holds lexical or grammatical information of the word. Based on the number of the morpheme that participated in the word-formation process the word can be simple or complex [20]. For example **/goxi/** meaning in English ‘sleep’, this word cannot further be disintegrated. This means it is a simple word because there is only one morpheme of the word itself. The complex word contains two or more morphemes to appear as a word. For example **/hiraancho/** meaning in English ‘seller’, this word contains two morphemes **/hira-ancho/** ‘sell-er’. This shows that this word is complex. According to immunity that they have to stand alone morphemes can be classified into two. These are free morpheme and bound morpheme [22]. Free morphemes can stand alone and have lexical meaning. While bound morphemes cannot stand alone and needs free morpheme to exist and change the grammatical meaning of the word. For example, a **horse** is a free morpheme such that it can give full meaning to the reader. If the word changed to **horses** the word contains both free and bound morpheme **horse - s**. as described above horse can stand alone but morpheme ‘-s’ cannot. The function of ‘-s’ in the word is to show the plural number of the **horse**. So without free morpheme, bound morpheme has no meaning. Some examples of bound

morpheme in English like *un* (*un-happy*), *il* (*il-legal*), *in* (*in-sufficient*), *re* (*re-write*), and so on. A hyphen before bound morpheme ‘-s’ tells us that it needs a free morpheme to exist. The specific morpheme (root or affix) can express itself differently without meaning change called allomorph [19]. For example *ir-regular*, *il-legal*, *im-possible* these three prefixes are negative allomorphs. The reasons for the existence of allomorphs are the first one is phonological alternation, the second is a morphological process, and the third is the nature of word formation in the language. Sometimes one morpheme may have many properties to express the meaning of a word which is called a portmanteau morpheme [19]. E.g *jumps* morpheme ‘-s’ indicates that present tense and 3prs. Based on the ability that morpheme has to create new words from another word class can be classified into derivational and with a similar class inflectional [23].

Derivational morphemes can create a new word in addition to changing its class.

For example, the verb *send* can be changed to noun *sender* by the addition of ‘-er’ noun affix to the verb.

Inflectional morpheme doesn’t change the class of word but it brings a change in case, number, aspect, and tense. For example ‘-ed’ (*work-ed*), ‘-s’ (*cat-s*).

Bound morphology is further classified into sub-categories affixes and clitics [18].

2.1.3 Affixes

Affixes are bound morphemes that attach themselves with the other words to yield grammatical meaning. Morphological rich words can have many affixes but the way how

they align one to another is the critical issue [19]. According to the position that affixes attach to the stem word they are classified into four basic and two special forms of affixes.

Prefixes

Prefixes are the one type of affix that comes at the beginning of the word to produce grammatical meaning.

For example: *re*-create = recreate

Im-Possible= impossible

Here is '*re*-' and '*im*-' are prefixes of the words. The hyphen (-) indicates the position of the stem word.

Suffixes

This type of affixes attached at the end of the given words to yield grammatical meaning as obvious.

For example: work-*ed*= worked

Girl-*s*= girls

Here are '*-ed*' and '*-s*' are suffixes of the words. The hyphen (-) indicates the position of the stem word.

Circumfixes

Circumfixes are affixes that come at the beginning and end of the word at the same time.

For example, /*Bela*/ 'he ate' /*al-bela-hum*/ 'I haven't eaten' in Amharic word which means 'I didn't eat'

Here are '*al-*' and '*-hum*' as prefix and suffix occurs at the same time. The Italic format shows that the prefix and suffix of a word. This kind of occurrence is called a circumfix affix.

Infixes

Infixes are affixes that occur somewhere except at the beginning and end of the word but in the middle.

For example, */It/* 'you eat' */i-n-t/* 'have you eaten' in Sidaamu Afoo which means

The two (Interfixes and Transfixes) special kinds of affixes most of the time exhibit in Germanic and Semitic languages [24].

Interfixes

Interfixes are affixing they link two independent words to compound them in a Germanic language.

e.g tag reise = tag-e-reise 'day's journey'

tag - licht = tag-es-licht 'day light'

Transfixes

Transfixes are the type of special affixes that brings not only discontinues of affixes but also discontinues of roots. This kind of affixes occurs only in Semitic languages. The roots are made up of consonants and transfixes are made up of vowels that operate consonants of the root. There is no specific position for transfixes to happen in a word that means it can occur anywhere regarding the type of transfixes. For example,

Root word	Gloss
<i>/ktb/</i>	'write'
<i>/katab/</i>	'he write'
<i>/jiktib/</i>	'he will write'
<i>/mak'tuub/</i>	'written'
<i>/mak'taba/</i>	'bookshop'

2.1.4 Clitics

Clitics has the property of a word and affixes. In another way, it is intermediate between affixes and words. According to its nature clitics are somehow complicated in that they act in the syntactical structure of the words. A good example of the clitics in English words like 've in I've and 's (phonological attach) in genitive grammatical cases in another way it is a short form of the words [24]. The difference is that affixes are attached to the single lexical category but clitics are not limited to a single word but up to the combination (phrase level) of words [25]. The similarity of affix and clitic they both categorized in the bound morpheme. The Sidaamu Afoo language has proclitic (the negative clitics) and others are enclitics (conjunctive enclitics and noun-phrase) [3].

A language can be expressed by different properties that can attest to the given language but the morphology is uniquely identified the language-specific behavior, this hinders the generalization [26]. According to the authors based on the morphological aspects of the language, it is difficult to discuss cross-linguistic validity. The mechanism to identify the

word-form in the various language is based on language-specific criteria about phonological and morphosyntactic aspects.

The rule can be discovered by using a machine learning approach to analyze or syntheses the morphology of the language [27]. Machine discover rules based on the pre-stored examples.

Having a morphological analyzer for agglutinative language paves way for other NLP. Stating all language-related rules in detail and design a morphological analyzer based on those rules assure the correctness of the analysis output [28].

2.1.5 Morphotactic

Languages are classified into different categories based on the morphological expression meaning and the way how their word constituents are organized to each other to yield anticipated words [26]. Traditional the above criteria lead to name the categories of the languages as isolating, agglutinating, and inflected/fusional. Morphotactic is the set of rules that governs the arrangement/order of the morphemes that participate in the formation of a word [29] [30]. According to the authors, morphotactic is a language-specific component of morphology. In another word, morphemes should place in the right position in restricted order to attest to the validity of the complete word [31]. Morphemes are the basic meaning-bearing component and they are not come together randomly to form the word. Understanding the order of morpheme in the word-formation process paves the way for a morphological analyzer to identify the way how the morphemes are constructed for the complex word [23].

2.1.6 Types of morphology

Traditionally linguistic morphology is classified into inflectional, derivational, and compounding categories [32]. Morphology can be classified as inflectional and derivational [33] [16]. The term morphology comes from the Greek word *morpho* equivalent with ‘form’ and *logy* which stands for ‘study’ for the English term. Morphology deals with the internal word-formation structure (morpheme) that contains grammatical meaning within. For example, word mother and player. The word mother provides only one meaning without subdivision (female parent) but the word player constitutes stem action verb play and the nominalizing suffix ‘-er’. In general morphological studies classified Inflectional and Derivational categories. **Inflectional morphology** focused on (case, number, aspect, tense) the word form attaching appropriate morphemes on the stem. For example, jumping, jumps, jumped, the word forms have jump stem words in common but based on the type of morpheme attached with different grammatical meanings. **Derivational morphology** mainly focused on the new word formation attaching derivational morpheme to the base word. For example, manage-ment this word contains manage verb and ment derivational morpheme. The word manage (verb) changes to word management (noun) due to ment derivational morpheme attached to the base word. In Sidaamu Afoo whether inflectional or derivational morphology is highly accompanied by affixes. There is only one prefix (di-) and many suffixes in the language [3]. It is common, that word reduplication (whole or partial morphemes), and compounding in the language. For example, /lede lede/ ‘again and again’ and /tutuga/ ‘repeatedly throwing’ this example shows that complete reduplication and half reduplication respectively. The reduplication pattern can be partial at (beginning, middle, and end) morpheme, and complete word reduplication.

Compounding is the word-formation process through joining two or more standalone words to form a new lexeme [32]. For example, the word snowball is the combination of two words that snow and ball. In Sidaamu Afoo compounding the word-formation process is common. For example

Compound word	Glosses
<i>/Galado-weese/</i> [galadiweese]	‘type of plant’.
<i>/Haak’-boce/</i> [haqqiboce]	‘kind of bird’.
<i>/Hase- k’amale/</i> [basiqamale]	‘kind of bird’

2.1.7 Morphophonemic

Morphophonemic deals with the relationship between morphology and phonology in such a way it identifies the phonological alternation in the morphemes [19]. It is the combination of two basic components of language phonology the study of sound structure and morphology the study of the structure of word-formation. Due to the close relationship between these two language pillar aspects whether regular or irregular alternation of phonetic affects the way of pronunciation in morpheme. Eg. Plural allomorphs in English: ‘-s’ in *cats*, ‘-z’ in *shoes*, ‘-ez’ in *churches*. Sidaamu Afoo is classified in the highland east Cushitic language. This language family exhibit three fundamental morphophonemic rules. From these rules epenthesis (addition of one or more sounds to a word), nasal metathesis (the transposition of sounds), and assimilation (Sound change to be more similar to other nearby sounds) are the major ones.

2.1.8 Epenthesis

According to the morphophonemic rule epenthesis perspective, in Sidaamu Afoo no more than two consonants come together. If there are consecutive two consonants within the word indicates that germination or obstruent sounds. In the verb inflection of the language, epenthetic vowels are inserted between consonants to the word based on the three conditions. The first one is when mon-syllabic verb stem followed /-tV/ initial consonant of the suffix in such a way the 'i' vowel comes in the middle of two consonants to preserve the linguistic rule.

For example

Verb	Syllabic Structure	mon-syllabic	Glosses
/š-tu/	[šitu]	/š-/	'She/they killed'
/y-tu/	[yitu]	/y-/	'she said'

The second condition is that the epenthetic vowel is inserted into the word when three consecutive consonants that appear in the 'i' sound must be inserted between the second and the third consonant to keep linguistic rule secure. For example,

Verb	CC-C	Verb stem	Gloss
/Kubb-tu/	[kubbitu]	/kibb-/	'jump'

The word /kubb-tu/ cannot appear like /kubbt/ this is not allowed in Sidaamu Afoo. To avoid situation 'i' sound must be insert between geminating last 'b' and 't' sound. So the correct form should be /kubbite/ the correct representation of the word.

The third condition that when the single causative ‘-s’ and the double causative ‘-siis’ comes next to the stem-final obstruent’s.

Verb	single causative	Glosses	Double causative	glosses
/gat/	/gatis-/	‘Leave it’	/gatisiis-/	‘cause to be left’
/bat/	/batis-/	‘pay’	/batisiis-/	‘cause to be payed’

2.1.9 Metathesis

Metathesis is the phonological process that shows the transposition of consecutive consonants within a given word. The phenomena happen in the Sidaamu Afoo when the stem of the verb ends without stress for those which have /-n/ as an initial suffix. For example, /ninke inte/ ‘We ate’ the word /inte/ constructed from two sounds /?it/ ‘eat’ and ‘-ne’ and expected to be /itne/ but the language never allow such king of phonological syntax. For that reason, the word is represented in the /inte/ manner to facilitate speech convenience for communication. There are two types of metathesis in the language. These are Nasal Metathesis and glottal metathesis.

2.1.9.1 Nasal metathesis

Nasal metathesis is the most common type of metathesis in the language. The sound ‘-n’ of 1prp, 3prn.N polite, and 3prs.M.pres.perf reshuffled with stem-final obstruent’s. Here is an exceptional case that appears that disobey the nasal metathesis due to stem-final glottal stop in the word. for example, /la’-/ meaning in English ‘look’, /sa’-/ meaning in English ‘pass’, /e’-/ meaning in English ‘enter’, and /ba’-/ meaning in English ‘vanish’. The glottal stop

phoneme resists transposition possibility due to the behavior that exhibits like a sonorant.

The rule that governs the situation looks like follows.

When /C-n/ converter into /-n C/

For example,

Verb stem	Glosses	1prp or 3prn.Pol	Glosses
<i>/hig-/</i>	‘turned’	<i>/hingummo/</i>	‘we turned’
<i>/it-/</i>	‘eat’	<i>/intummo/</i>	‘we ate’
<i>/tug-/</i>	‘throw’	<i>/tungummo/</i>	‘we threw’
<i>/af-/</i>	‘know’	<i>/anfummo/</i>	‘we know’

2.1.9.2 Glottal metathesis

Here the stem-final sonorant is reshuffled with the sound ‘-d’ in the word. The sound disintegrated into two components such /-‘/ and /d/. And the ‘-d’ will be deleted and the /-‘/ will be remain. The remained part metathesizes with the stem-final sonorant. For example

Verb	Glosses	metathesized verb	Glosses
<i>/tum-/</i>	‘pound’	<i>/tu’m-a/</i>	‘to pound for oneself’
<i>/gan-/</i>	‘hit’	<i>/ga’n-a/</i>	‘to hit for oneself’
<i>/fan-/</i>	‘open’	<i>/fa’n-a/</i>	‘to open for oneself’

2.1.10 Assimilation

Assimilation is the behavior of sound influenced by or influence neighborhood sounds to be in some manner. The assimilation can be avoidance (removal of sound from the word), of sound in partial or complete assimilation.

For Example,

Original word	Glosses	Assimilated sound
<i>/dunkeemmo/</i>	'we carry'	<i>[duŋkeemmo]</i> <i>n+k -> ŋk</i>
<i>/hajanjeemmo/</i>	'we order'	<i>[hajañjeemmo]</i> <i>n+j -> ñj</i>
<i>/hanbeemmo/</i>	'we forget'	<i>[hambeemmo]</i> <i>n+b -> mb</i>
<i>/kul-neemmo/</i>	'we tell'	<i>/kulneemmo/ [kulleemmo]</i> <i>l+n-> ll</i>
<i>/kad- kada/</i>	'jumped over'	<i>[kakkada]</i> <i>d+k -> kk</i>
<i>/go-wi/</i>	'hate'	<i>/go-bbu/</i>
<i>/gow-tu/</i>	'she fool'	<i>w+t -> bb</i>
<i>/daa-hi/</i>	'carry'	<i>/daa-ku/ /daah-tu/ -> h+ t ->kk</i>

There are morphophonemic phenomena in the Sidaamu Afoo language like glottal stops.

Eg.

Word	Glosses	Morphophonemic form	Glosses
<i>/Saa/</i>	'cow'	<i>/sa'a/</i>	'to pass'
<i>/Meeda/</i>	'to shave'	<i>/me'e/</i>	'how many'

2.1.11 Computational morphology

Computational morphology is the sub-branch of computational linguistic. The core notion of the topic is the computer-aided word structure analysis and generation based on the compiled theories. It enables a computer to automatically reveal stored information within words [34] [35]. Morphological analysis disintegrates the components of the word into a part and the generation is the reverse process. The computer extracts the grammatical feature of the word to deduce the morphological information. The morphological analyzer accepts surface form as input and produces lexical form as output. The reverse is true for the morphological generator. Eg. The input word *cats* as the surface form then morphological analyzer disintegrates it into morphemes *cat-s* to output as lexical form. The reverse inputs as lexical form and provide surface form is the process of generation. Eg input lexical form **cat-s** will form **cats** as surface form as generator output.

2.1.12 Morphological generator and analyzer

A morphological analyzer is a computer program that is responsible to disclose all required information within the input token based on grammatical detail of morpheme [36] and the morphological generator works in the reverse way. The system disintegrated all components of the complex word into constituent morphemes (stem and affix) and provide corresponding information based on the parts. It is the primary requisite of many other natural language processing applications like machine translation (MT) spelling checkers etc. The three most known morphological analysis techniques are morpheme-based (an item and arrangement approach), word-based morphology (word and paradigm approach), and lexeme-based morphology (an item and process approach) [37].

Knowledge required for morphological analysis and generator

The experts need to know in detail about different perspectives of the language. For the morphological analysis, the structure of the word-formation process is the core issue to interpret the given word. A computer program produces output that what it was oriented to do accordingly. To the minimum, the linguistic expert must be aware of orthographic representation of the language and phonological alternation that occur during the affixation process [38]. According to the authors, syntactic or semantic properties of affixation and the base form of the word is must be known. Based on this knowledge the computer system design and implementation should equip the above capability to handle the situation.

2.1.13 Approaches to Morphological Processing

Finite State Automata

This method is used to accept or reject strings that are found in the language. The transition occurs because of the inputs at each state which leads the current state to process input and transform the state to the next. The transition follows all rules that govern the system. It is a mathematical model of computation with five tuples $(Q, \Sigma, q_0, A, \delta)$.

Q stands for a finite set of states

Σ stands for finite input alphabets

q_0 element of Q stands for the initial state

A subset of Q stands for the set of accepting states

$\delta: Q \times \Sigma \rightarrow Q$ stands for the transition function

For the given state R and input value I,

The transition function defined $\delta(R, I)$ as a state to move to the next appropriate state.

Two-level morphology

Two-level morphology approach is designed by Kimmo Koskenniemi as a computational model in 1983. The basics of the model are Morphotactic (which deals with the order of morpheme in the word) and morphophonemic (alternative form of the morpheme related to phonological aspects). The bidirectional and separation of the application program from language description (rule and lexicon) are the main features of the model [39]. What assumption was, the usage of rules in parallel rather than a sequential manner to describe phonological alternation in finite-state terms. The model contains two representations and a rule component. The two representations are lexical (morphophonemic) word-form that describes the chain of morpheme in the word and surface representation which depicts the actual spelling of a valid word. The rule component of the model plays the integration role for these two representations.

Finite State Transducer

It is an improved version of the finite-state automation model based on the two-level morphology principles. The difference between the finite state automata (FSA) and the transducer is that the transducer transition is accompanied by input and output tape which is the only input label in FSA. For example the transition T labeled according to FST ($T = \{a:a, b:b, c:c, \dots\}$) and ($T = \{a, b, c, \dots\}$) if it is FSA respectively.

FST is defined as the collection of 7 tuples $(q_0, Q, F, \Sigma, \Lambda, \delta, \lambda)$ where:

q_0 element of Q stands for the initial state

Q stands for a set of finite-state

F subset of Q stands for a set of the final state

Σ stands for a set of input alphabets

Λ stands for a set of output alphabets

δ stands for a set of transition functions ($Q \times \Sigma \rightarrow Q$) that map two states and an input alphabet to state.

λ stands for transduction function ($Q \times \Sigma \rightarrow \Lambda$) that mapping state and input alphabet to output alphabet.

Stemmer

Stemmer is the mechanism of getting stem from input word by removing affixes and substitute if necessary based on the rule. This approach is efficient for regular words but it cannot handle irregular word-formation processes [40].

Corpus-based Approach

Corpus is a tremendous collection of text documents. The word in the document will be analyzed to generate anticipated output statistically. Corpus base morphology analysis has its limitation because it requires a large amount of data for the process [40]. Machine required to understand the structure of the word extracting the necessary information from a huge amount of corpus automatically to segment into the parts if unsupervised method applied to train it. With the little divergence the semi-supervised method needs certain involvement of person to function properly but the rest part of the system function

automatically. The knowledge of the expert is required to annotate the corpus to train the machine when the method is supervised [41].

Rule-based approach

This approach needs full knowledge of the linguistic expert to implement. With this approach, morphological analyzer development has been implemented for many years for many languages based on the linguistic theory [22]. The factors that make the rule-based morphological analysis approach prominent, the first one is less storage requirement and a small amount of corpus to implement it. The second reason unlike the statistical approach there are no complicated computational difficulties, once the rules are set which speed up the morphological analysis process. The third reason is accuracy due to predefined rule sets to govern the last output of the morphological analyzer. The fourth reason is that it is straightforward to correct any handicaps that happen within the morphological analysis system.

Direct Acyclic Word Graph (DAWG)

It is a data structure that stores sorted characters of words using edges and nodes. The edges of the graph represent the characters of the words and nodes represent the branching point or terminal. The words can be separated into the base and affix parts to represent on the graph. This approach minimizes the usage of memory space of the words for which have similar affixes in a common.

Paradigm approach

This approach explores all possible word forms for the given stem. The words are organized in the table base on their feature and morphological behaviors. The table depicts the

relationship between stem and all word forms. The root word follows a certain pattern (paradigm) to generate word form and let to differentiate its categories.

Suffix stripping

For agglutinative language, the word is formed from the stem by the addition of different suffixes. If the suffixes are known in advance it is possible to analyze the word morphological disintegrating suffixes and root words.

2.2 Related work

2.2.1 Local languages

Many Ethiopian languages have many speakers studied well concerning linguistic aspects. Most of the computational aspects are not studied and developed for variate of NLP applications for local languages. But there were some attempts by different graduate students at different times as final thesis. Michael Gasser developed the python programs based on finite-state approach for three (Amharic, Afaan Oromo, and Tigrigna) Ethiopian languages as morphological analyzers and generators [9].according to the author report overall performance of this work for Amharic and Tigrigna, eight errors for Tigrigna verbs and 96% accuracy, two errors on Amharic verbs and 99% accuracy, nine errors on Amharic noun and adjective and 95.5% accuracy was achieved. Saba Amsalu and Dafydd Gibbon implemented a morphological generator and analyzer for Amharic language using finite-state transduce approach for all kinds of parts of speech. The tool that they used as an implementation environment is Xfst software. For 468 verbs in the corpus, 94% were analyzed, for 650 nouns 85% were correctly analyzed and for 76 of adjective recall 88%

correct analysis obtained [42]. Memory-based learning approach for the development of a morphological analyzer in Amharic language was attempted. The experimental result shows that the promising achievement that paved the way for the other NLP applications development [43]. This study was conducted on two separate algorithms IB1 and IGtree and got 93.6% and 82.3% experimental results respectively. A supervised memory learning approach was used as the first morphological analyzer attempt for Sidaamu Afoo [13]. The tremendous electronic afaan Oromo information production without vast computational study, motivates a researcher to start the first morphological analyzer design and development process [10]. The tool used by the research to achieve his objective was Linguistica-beta2 (linguistica is a Python library for unsupervised learning tool for affix pattern identification) and achieved 92.8% from his experiment. The rule-based approach is used to implement a morphological analyzer for Ge'ez verbs [22]. The methodology that the researcher preferred in his implementation is CV-based and two-level morphology. The experimental performance of the system was evaluated against expertise annotation. Experimental achievement according to verb feature extraction scored 92.05% and 73.98% verb level analysis. For the first time, a finite-state transducer was used to develop a morphological analyzer for wollayta language [44]. Lexicon lemma is obtained by semi-automatic. Morphotactic was implemented by hand in lexc formalism. The morphological rule implemented in twlc formalism the precision and recall is (94.85%), (94.11%) respectively. Verbs are inflected in wollayta for aspects, number, person, mood, gender, and case. Morphological analysis recognized 83.13% of words and unrecognized 16.87 words which has glottal sound.

2.2.2 Non-Ethiopian languages

Morphological generator and analysis is the base for many NLP applications [45]. For that reason development progress from the early time up to today is ongoing. With different approaches to address the issue of many natural language processing problems. The Arabic language is the one that many researchers worked on it intensively just following various approaches [46] [47] [48]. Morphological analyzer for Korean language using a modified version of CYK algorithm which is capable of analyzing the spontaneous speech of the language [49]. Morphological analysis for Marathi language combining paradigm-based and finite-state approach in which the researcher achieved 97.18% of correct word analysis [50]. The combination of two-level formalism and unification-based formalism approach was followed to implement morphological analysis for the Basque language [51]. Morphological analyzer and generator based on finite-state transduce implementation for kannada language achieved 90% for nouns and 85% for the verbs [45]. Morphological generator and analyzer developed for the language of hind as an integral part of machine translation to Punjabi language [52].

CHAPTER THREE: Overview of Sidaamu Afoo language

3.2 Phonological overview

The Sidaamu Afoo language consonant phonemes are based on the manner verse place of articulation which is found in the language described as follows.

Table 3.1: The phoneme inventory of Sidaamu Afoo adopted from [50].

	Labial	Alveolar	Post-Alv	Velar	Glottal
Plosives		t	c	k	ʔ
	b	d	j	g	
Ejectives	p'	t'	c'	k'	
Implosive		ɓ			
Fricatives	f	s	ɣ		h
Nasals	m	n	ɲ		
Lateral		l			
Vibrant		r			

Except /h/ consonants in Sidaamu Afoo, all are geminating consonants.

The vowel representation of the language is depicted in the blow.

Table 3.2: The vowel phonemes, adopted from the document [50].

	Front	Central	Back
High	i/ii		u/uu
Fron	e/ee		o/oo
Low		a/aa	

The language syllables structure obeys the (C)V(C)(C) general template for the formation of the word but the most common one is CV. This does not mean all words start with a CV but some words word begin with VC syllables structure.

For Example:

Word	Glosses
<i>/Hara/</i>	'to go'
<i>/giwa/</i>	'reject'

The words that start with VC syllables structure.

For Example

Word	Glosses
<i>/ani/</i>	'I'
<i>/ati/</i>	'you'

/uduunne/ '*materials*' CV is dominant but not absolute.

Another characteristic of Sidaamu Afoo is that tone language means the stressed sound or high tone makes different meanings with similar morphemes.

For Example:

Word	Glosses
<i>/mule/</i>	'kidney'
<i>/mulè/</i>	'not far'

/ ʔ symbol on the top of the later 'e' shows that lower tone. There are the glottal phonemes in the language like /'L/ /'M/ /'N/ /'y/. In Sidaamu Afoo language, the single word conveys much information due to its morphological complex structure. For example, a single word

tells us about the subject, object, tense, number, gender, and extra information comprised within it. In Sidaamu Afoo language, the word never begins with a geminated consonant and it ends with the vowel but not with the consonant [53].

There is twenty four [‘,B,C,CH,D,DH,F,G,H,J,K,L,M,N,NY, PH,Q,R,S,SH,T, W,X,Y] language-specific and five borrowed [P,TS,V,Z,ZH] consonants which total of twenty nine [‘,B,C,CH,D,DH,F,G,H,J,K,L,M,N,NY,P,PH,Q,R,S,SH,T,TS,V,W,X,Y,Z,ZH] and five vowel [A,E,I,O,U] in upper and lower alphabets [54]. When these alphabetic sounds (phonemes) are created in the language brings different phonotactics in the word-formation process. Based on the occurrence location phonotactic variation can be expressed as geminate, simplex, short, and long that pay the way to have different meanings in the word. The phonemes can be a beginning, middle, or end of the word according to their occurrence location. According to the linguistic rule the consonants at beginning of a word never became geminated. But the simplex and geminate representation bring meaning variation in words. The geminated word has a different meaning from the simplex one.

For example:

Word	Glosses
<i>/soora /</i>	‘fattening’
<i>/soorra/</i>	‘ exchange’
<i>/woga /</i>	‘begging’
<i>/wogga /</i>	‘year’

The short and long vowel occurrence also create the meaning variation like that of simplex and geminate in the consonants. For example:

Word	Glosses
<i>/hara/</i>	‘to go’
<i>/haara/</i>	‘to take’

The meaning variation comes true because of the short and long-form appearance of vowel ‘a’ in the word.

The long and short vowels can appear at different positions such that at the beginning, middle, and end of a word.

For example:

Word	Glosses
<i>/uulla/</i>	‘land’,
<i>/wiisa/</i>	‘plowing’
<i>/ollaa/</i>	‘village’.

Single sound variation at same place for different words change meaning of it.

For example:

Word	Glosses
<i>/gaara/</i>	‘mound’
<i>/gaaro/</i>	‘conflict’

This is not always true because sometimes with sound variation there may no meaning change at all.

For example:

Word	Glosses
<i>/hunsha/</i>	‘grief’
<i>/unsha/</i>	‘grief’

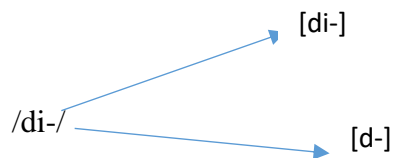
2.1 Morphological Overview of Sidaamu Afoo

Morphology is the study of the internal structure of a word-formation process. How the morphemes interact with one another to bring the expected meaning of the word is called morphotactic. And the alternation of sound in the word-formation process (all morpheme). The morphological structure of the Sidaamu Afoo reveals a variety of morphological structures that exhibit in the language. The word-formation process can be one of the following categories. The first one affixation process and the second one is non- affixation (compounding, reduplication, tone change, and blending). Few prefixes exist in Sidaamu Afoo language the most popular one is /di-/ which denotes the negative meaning of a word. But all words that start with /di/ don’t mean negative words. For example /diriirsa/ ‘laying’ this is the word in Sidaamu Afoo as the other words but the beginning of the word is /di/ that doesn’t guaranty the word to be negative.

The negative indicator prefix in Sidaamu Afoo (di-) can exist in an alternative manner that shows the same meaning. For example:

Word	Glosses
<i>/ageemmo/</i>	‘I will drink’
<i>/di-ʔageemmo/</i>	‘I will never drink’
<i>/d-ageemmo/</i>	‘I will never drink’
<i>/uffanno/</i>	‘she will blow’
<i>/di-ʔuffanno/</i>	‘she will never blow’
<i>/d-uffanno/</i>	‘she will never blow’

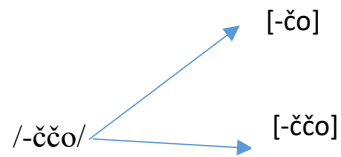
These two words convey the same meaning as the phonological representation.



The reason why the /di-/ is substituted by /d-/ in the second word is according to the rule of the language. It is unusual having different consecutive vowels in the Sidaamu Afoo language if it happens the strongest sound assimilates the weak sound.

The phonological alternation happens when the singular indicator suffixes /ččo/ followed by different consonant sounds.

Word	Glosses
<i>/lukki-chcho/</i>	‘hen’
<i>/kin-cho/</i>	‘stone’



In sidaaamu afoo the consonant sound ‘*k*’ cannot have an immediate fellow of consonant sound ‘*č*’ for this reason the separating ‘*i*’ vowel comes in between two. But consonant sound ‘*n*’ can be immediately by the ‘*č*’ so it doesn’t need ‘*i*’ vowel to split these two consonant sounds. What makes the difference in both conditions is the linguistic rule guides the co-occurrence of the sound together in stressed or normal form. Here is ‘*n*’ can be followed by immediate ‘*č*’ but two /čč/ cannot happen because of the phonological linguistic rule of the language. It is obvious to have /čč/ after ‘*k*’ which is not followed immediately by ‘*č*’ according to the rule. The reason why the phonological aspect of the language is discussed here is that the phonological alteration brings a change in the morphological structure of a word. There is the variate of the suffixes which shows plurality in the language with different alternatives. Some suffixes may replace others without any meaning change. For Example, Suffix /-uwa/ and /-ubba/ can replace one another.

Words	Glosses	equivalent suffix
<i>/siqq-uwa/ [siqquwa]</i>	‘sticks’	<i>/siqq-ubba/ [siqqubba]</i>
<i>/gob-ubba/ [gobbubba]</i>	‘countries’	<i>/gobb-uwa/ [gobbuwa]</i>

This phenomenon is not always true.

Word	Glosses	inequivalent word form
<i>/lukk-uwa / [lukkuwa]</i>	‘hens’	<i>/lukk-bba/ [lukkubba]</i>

/lukkubba/ is not familiar with the language speakers. This creates inconvenience for the audience.

But in the first example, both /uwa/ and /ubba/ suffixes show the plural number of the sticks.

In the second case, the first part /lukkuwa /show the plural form of a hen, and the second part /lukkubba/ is not usual.

But there is so many all morpheme exist to show plurality of the word. For example,

Word	Glosses
<i>/min-na/ [minna]</i>	‘houses’
<i>/saa-da/ [saada]</i>	‘animals’
<i>/boot-ta/ [bootta]</i>	‘oxen’
<i>/k’aakk’-ulle/ [kaakkulle]</i>	‘children’
<i>/Pogee-yye/ [ogeeyye]</i>	‘professionals’
<i>/moor-aano/ [mooraano]</i>	‘thieves’
<i>/geer-ra/</i>	‘elders’

The above examples show the plural suffixes allomorphs. Some words in Sidaamu Afoo to express pluralism may not take affixes as attachment but it may change morphophonemic nature of the original word. For example

Singular form	Glosses	plural form	Glosses
<i>/beetto/</i>	‘boy’	<i>/ooso/</i>	‘boys’
<i>/beetto/</i>	‘girl’	<i>/seenne/</i>	‘girls’

In morphological aspects of language the word contain several numbers of consecutive suffixes attachment which carries grammatical information within the given word.

For example:

Word	Glosses
<i>/gobba-ʔya/ [gobba'ya]</i>	'my land'
With one suffix <i>/-ya/</i> and stem word <i>/gobba/</i>	
<i>/gobba-ʔya-ti/ [gobba'ya-ati]</i>	'it is my land'
with two suffixes <i>/-ya-ti/</i> and stem word <i>/gobba/</i>	
<i>/mund-ee-nke-nni/ [mundeenkenni]</i>	'by our blood'
With three suffixes <i>/-ee-nke-nni/</i> and stem word <i>/mund/</i>	
<i>/dand-in-anni-kki-ri-čči-nni/</i>	
<i>[dandiinannikkirichchinni]</i>	'With impossible things'

With six suffixes */-in-anni-kki-ri-čči-nni/* and stem word */dand/*

Regardless of the number of attached suffixes they bear corresponding meaning in the word.

For example, the first suffix in the chain indicates the identity of the person in the word, and the next suffix shows being.

2.1.1 Verb morphology

In Sidaamu Afoo verbs occupy the last position in the sentence. The attempt to substitute the position of a verb with other parts of speech creates inconvenience for sentence

formation. Sidaamu Afoo verbs are inflected in various ways unlike other parts of speech. They inflected for number, gender, tense, aspect, mood, and person. The Sidaamu Afoo verb formation is illustrated in mono-consonant to multisyllabic structure variation.

For example:

2.1.1.1 mono-consonantal

Verb stem	CV structure	glosses
/š-/	C-	‘kill’
/y-/	C-	‘say’

2.1.1.2 Mono-syllabics

Verb stem	CV structure	glosses
/abb-/	VC1C1-	‘bring’
/af-/	VC-	‘know’
/ag-/	VC-	‘drink’
/ass-/	VC1C1-	‘do’
/irb-/	VC1C2-	‘rub’
/dod-/	CVC-	‘run’
/kaad-/	C1V1V1C2-	‘deny’
/giw-/	C1VC2	‘reject’

<i>/kubb-/</i>	C1VC1C1	‘jump’
<i>/sirb/</i>	CVC1C2	‘sing’
<i>/hay š š -/</i>	C1VC2C3C3	‘wash’

2.1.1.3 Disyllabic verbs

Here is verb with two syllabics. they can be simple or reduplicated verbs. For example

Verb stem	CV structure	glosses
<i>/egenn-/</i>	VC1VC2C2	‘know’
<i>/gombis-/</i>	CV1C2C3V2C	‘turnover’
<i>/hasaw-/</i>	CVC2VC3	‘talk’
<i>/dadil-/</i>	CV1CV2C2	‘worry’
<i>/gangan-/</i>	CVC2C3VC2	‘hit repeatedly’
<i>/kukkub-/</i>	CVCCVC2	‘jump repeatedly’

Sometimes disyllabic verbs are the part of a derived word in reduplication. Fro example

Verb stem	root	CV structure	Glosses
<i>/Kukkub-/</i>	<i>/kubb-/</i>	CVC2C3V	‘jump relatedly’

There are trisyllabic verb structure exist in the language for example:

Verb Stem	CV Structure	Glosses
<i>/fokkifat-/</i>	C1V1C2C2V2C3V3C4	‘Be ashamed’
<i>/dugunuun-/</i>	C1V1C2V2C3V3V3C4	‘bow one’s head’

2.1.1.4 Converb

The verb inflections that containing the stem and the subject suffix of the verb. This kind of verb becomes before the main verb. For example:

Person	Verb	glosses
Iprs.N	<i>/kibb-e/</i>	‘ I collide ’
2prs.M	<i>/kibb-i-te/</i>	‘you collide’
3prs.M	<i>/kibb-é/</i>	‘He collide’

There is the suffix in the language that expresses the politeness of the speaker while talking with somebody */-nni/*. For example

/sainni/ ‘polite, he passed’

2.1.1.5 Copula (Verb to be) */ikkate gurda/*

The three grammatical alternants that exist in the language which never conjugated like other tenses. They are representing 1st,2nd, and 3rd person singular with */-ho/* and */-te/* suffix to show masculine and feminine gender respectively. But the plural form can use any of the above suffixes. The */-it/* suffix in the verb represents genitive cases within a given sentence.

For example:

Word	Glosses
<i>/seeda-ho/</i>	‘I/you/he (am/are/is) tall’ masculine gender
<i>/seeda-te/</i>	‘I/you/she (am/are/is) tall’ feminine gender
<i>/ama-sii-ti/</i>	‘his mother’
<i>/ama-’ya-ti/</i>	‘my mother’
<i>/ama-see-ti/</i>	‘her mother’
<i>/ama-nkee-ti/</i>	‘our mother’

2.1.1.6 The verb of presences

The verb /no-/ or /heer-/ in the language depicts the existence of somebody or someone.

Presences verb	Glosses
<i>/No-ommo/</i>	‘I am/was present’
<i>/No-otto/</i>	‘you are/were present’ for the second person masculine gender.
<i>/no-otta/</i>	‘you are/were present’ for the second person feminine gender.
<i>/heer-eemmo/</i>	‘I shall live/I live’
<i>/heer-atto/</i>	‘you live’ for second person masculine gender.
<i>/heer-itta/</i>	‘you lived’ for second person feminine gender.

2.1.1.7 Possession verb

Possession verbs doesn't show the past time in the language but the current time possession indicated by /no-/ verbs and object suffixes. What makes /no-/ verb different in presence and possession is that /o/ sound is elongated here which is not exist in presence verb. For example,

<i>Person</i>	<i>Verb with object suffixes</i>	<i>Glosses</i>
1prs.N	<i>/Noo-'e/</i>	'I have'
2prs.N	<i>/Noo-he/</i>	'you have'
3prs.M	<i>/Noo-si/</i>	'he has'
3prs.F	<i>/Noo-se/</i>	'she has'

2.1.1.8 Verb of Imperative

The verb that pass commanding information to someone. In the language suffixes /-i/ for singular and /-e/ for plural depicts imperative information in addition to final consonants of the stem germinated. For example

Person	Verb with suffix	glosses
2prs.N	<i>/ag-i/</i>	'you drink'
	<i>/tug-i/</i>	'you throw'
2prp.N	<i>/ag-e/</i>	'you drink'
	<i>/tug-ge/</i>	'you throw'

As discussed in the introductory part of this document Sidaamu Afoo is highly related to stressed sound to convey grammatical information. The sound that brings a change in 3prs.M perfect stressed suffix and 2prs.M imperative unstressed variation in the meaning. For example,

Person	verb with suffix	Type of verb	Glosses
3prs.M	<i>/sirb-í/</i>	‘perfect’	‘He sang’
2prs.N	<i>/sirb-i/</i>	‘imperative’	‘you sing’

2.1.1.9 Verb of jussive/hajajote gurda duuchaho/

The jussive verbs are similar with imperative one in the mood but unlike imperative jussive verbs can be used for all person. Imperative verbs are used to give command to the second person. For example

Person	jussive verb with suffix	Glosses
1prs.N	<i>/abb-o/</i>	‘let me bring’
3prs.M	<i>/abb-o-na/</i>	‘let him bring’
3prs.F	<i>/abb-i-t-o-na/</i>	‘let her/them bring’
1prp.N	<i>/abb-i-n-o/</i>	‘let us bring’

2.1.1.10 Negative marker in verbs

Negative marker is applicable in all types of verb forms such that tens/aspect and imperative to jussive.

2.1.1.11 Negative of tense aspects

All conjugated forms of tenses and aspects use the similar negative prefix */di-/* to the verb.

The present continuous tense takes the negative prefix */di-/* at the front of the presence verb */no/* which is conjugating part of it. For example,

Person	positive form	glosses	negative form	glosses
1prs.M	<i>/gan-oommo/</i>	‘I have hit’	<i>/di-gan-oommo/</i>	‘I haven’t hit’
1prs.M	<i>/kad-oommo/</i>	‘I have kicked’	<i>/di-kad-oommo/</i>	‘I haven’t kicked’
1prs.M.prC	<i>/kad-anni noommo/</i>	‘I am kicking’	<i>/kad-anni di-noommo/</i>	‘I am not kicking’

2.1.1.12 Negative of imperative and jussive

The negative marker in the imperative and jussive verbs little bit different from the tense prefix */di-/* which is represented by the suffix for the second person singular and plural form. the suffix which depicts the second person negative is */-tooti/* or */-ooti/* and */-tinoonte/* or */-inoonte/* for the singular and plural respectively.

For example,

Negative of imperative verb	glosses
<i>/abb-i-toot/</i>	‘do not bring’ 2prs.N
<i>/ag-ooti/</i>	‘do not drink’ 2prs.N

/abb-i-tinoonte/

‘do not bring’ 2prp.N

/ag-i-noonte/

‘do not drink’ 2prp.N

2.1.1.13 Negative jussive verb

Negative of all jussive verbs represented by suffix */-nke/*. For example

Person	jussive verb with suffix	glosses
1prs.N	<i>/agoo-nke/</i>	‘let me not drink’
3prs.M	<i>/agoo-nke/</i>	‘let him not drink’
3prs.F	<i>/aggoo-nke/</i>	‘let her not drink’

2.2 Verbal inflection

The person can be represented as first, second and third with the number and gender. First person singular */ani/* and plural */ninke/*. Second person singular */ati/* and plural */ki’ne/*. Third person singular */isi/* or */ise/* and plural */insa/*. The first person singular is not identified by gender. */ani/* this word represents first person masculine and feminine gender. The plural form of the first person singular is also neutral */ninke/*. In the second person the same phenomena happen which means second person singular */ati/* have no gender related information. The word */ati/* doesn’t indicate whether the second person have masculine or feminine gender. The neutrality of gender is the same like that of first person plural form except word */ki’ne/*. Unlike first and second person the third person have gender variation at third person singular */isi/* for masculine gender representation and */ise/* to indicate the

feminine gender. But third person */insa/* plural is neutral as the prior plural person class. These the above pronoun words occur in morphological rich Sidaamu Afoo words as suffix attached to the stem of verb in the sentence. For example:

Person	Inflected verb	Glosses
1prs.M	<i>/kubb-ummo/ [kubbummo]</i>	‘ I jumped’
	<i>/kubb-oommo/ [kubboommo]</i>	‘I had jumped’
1prs.F	<i>/kubb-umma/ [kubbumma]</i>	‘I jumped’
	<i>/kubb-oomma/ [kubboomma]</i>	‘I had jumped’
2prs.M	<i>/kubb-itto/ [kubbitto]</i>	‘you jumped’
	<i>/kubb-ootto/ [kubbootto]</i>	‘you had jumped’
2prs.F	<i>/kubb-itta/ [kubbitta]</i>	‘ you jumped’
	<i>/kubb-ootta/ [kubbootta]</i>	’you had jumped’
3prs.M	<i>/kubb-i/ [kubbi]</i>	‘he jumped’
	<i>/kubb-ino/ [kubbino]</i>	‘he had jumped’
3prs.F	<i>/kubb-itu/ [kubbitu]</i>	‘she jumped/
	<i>/kubb-itino/ [kubbitino]</i>	‘she had jumped’
1prp	<i>/kubb-inummo/ [kubbinummo]</i>	‘we jumped’
	<i>/kubb-inoommo/ [kubbinoommo]</i>	‘we had jumped’
2prp	<i>/kubb-itini/ [kubbitini]</i>	‘you jumped’

/kubb-itinoonni/ [kubbitinoonni] ‘you had jumped’

3prp /kubb-i/ or /kubb-itu/ [kubbi] or [kubbitu] ‘they jumped’

/kubb-ino/or/kubb-itino/ [kubbino]or[kubbitino] ‘the had jumped’

2.2.1 Gender suffix

The word may contain suffixes that point to a certain gender (feminine and masculine) in the word. But it is not common to have gender reference suffix in all person position. For example ‘-o’ indicates masculine gender in the 1st and 2nd person. ‘-a’ suffix represents the feminine gender for the above persons. For the 3rd person ‘-t’ shows the feminine gender and there is no known suffix for the masculine gender representation for it.

For example:

Person	gender suffix	Glosses
1prs.M	/higumm-o/	‘I turned’
1prs.F	/higumm-a/	‘I turned’
2prs.M	/higi-tto/	‘you returned’
2prs.F	/higi-tta/	‘you returned’
3prs	/hig-i/or /hig-u/	‘They returned’

2.2.2 Number suffix in verb

There are verb suffixes to indicate the numbers in the word. Most of the time the suffixes indicate plural number rather than singular one. There is no suffixes to indicate number in the third person. For example:

Person	stem	glosses	singular	plural
1prs.M	<i>/shaf/</i>	‘shake’	<i>/shaf-ummo/</i>	<i>/shanf-ummo/</i>
1prs.F	<i>/shaf/</i>	‘shake’	<i>/shaf-umma/</i>	-
2prs.M	<i>/shaf/</i>	‘shake’	<i>/shaf-itto/</i>	<i>/shaf-ini/</i>
2prs.F	<i>/shaf/</i>	‘shake’	<i>/shaf-itta/</i>	-
3prs.M	<i>/shaf/</i>	‘shake’	<i>/shaf-i/</i>	<i>/shaf-i/shaf-u/</i>
3prs.F	<i>/shaf/</i>	‘shake’	<i>/shaf-u/</i>	-

In the above example /-n/ for the first and second person appears to show plurality which does not exist in the singular form of the word. So it is possible to say that /-n/ shows a plurality of the word. In the third person, there is no /-n/ which means there is no demarcation for the singular and plural form of the word regarding the specified person.

2.2.3 Tense/Aspect

The tense can be expressed in the past time and non-past time. The past time includes present perfect tense, simple past tense, and past perfect tense. Non-past time includes present and future time. Sidaama language has suffixes to express past and non-past time.

Person	Stem	present perfect	Simple past	past perfect
1prs.M	<i>/agi/</i> 'drink'	<i>/ag-oo-mm-o/</i>	<i>/ag-u-mm-o/</i>	ag-oo-mm-o-nka
1prs.F	<i>/agi/</i> 'drink'	<i>/ag-oo-mm-a/</i>	<i>/ag-u-mm-a/</i>	ag-oo-mm-a-nka
2prs.M	<i>/agi/</i> 'drink'	<i>/ag-oo-tt-o/</i>	<i>/ag-i-tt-o/</i>	
2prs.F	<i>/agi/</i> 'drink'	<i>/ag-oo-tt-a/</i>	<i>/ag-i-tt-a/</i>	
3prs.M	<i>/agi/</i> 'drink'	<i>/ag-ino/</i>	<i>/ag-i/</i>	
3prs.F	<i>/agi/</i> 'drink'	<i>/ag-gino/</i>	<i>/ag-u/</i>	
1prp.N	<i>/agi/</i> 'drink'	<i>/ang-oo-mm-o/</i>	<i>/ang-u-mm-o/</i>	
2prp.N	<i>/agi/</i> 'drink'	<i>/ag-in-oonni/</i>	<i>/ag-ini/</i>	
3prp.N	<i>/agi/</i> 'drink'	<i>/ag-ino</i> or <i>ag-gino/</i>	<i>/ag-i</i> or <i>ag-u/</i>	

To show past or present continuous tense the language requires two different words. The first part of the word contains the continuous suffix */-anni/* and the second part contains present/past conjugation of the auxiliary verb */no/*.

For example,

Person	Stem	Present continuous tense	past continuous tense
1prs.M	<i>/agi/</i> 'drink'	<i>/ag-anni noommo/</i>	<i>/ag-anni noomonka/</i>
1prs.F	<i>/agi/</i> 'drink'	<i>/ag-anni noomma/</i>	<i>/ag-anni noomanka/</i>
2prs.M	<i>/agi/</i> 'drink'	<i>/ag-anni nootto/</i>	<i>/ag-anni noottonka/</i>
3prs.M	<i>/agi/</i> 'drink'	<i>/ag-ganni no/</i>	

1prp.N	<i>/agi/</i> ‘drink’	<i>/ag-anni hee’nommo/</i>	<i>/ag-anni hee’noomonka/</i>
2prp.N	<i>/agi/</i> ‘drink’	<i>/ag-in-anni heedhinooni/</i>	<i>/ag-in-anni heedhinoonka/</i>
3prp.N	<i>/agi/</i> ‘drink’	<i>/ag-ani no/ag-gani no/</i>	

Most of the time present and future tense represent with the same suffixes in the language.

For example:

Person	stem	Present/future
1prs.M	<i>/abbi/</i> ‘bring’	<i>/abb-ee-mm-o/</i>
1prs.F	<i>/abbi/</i> ‘bring’	<i>/abb-ee-mm-a/</i>
2prs.M	<i>/abbi/</i> ‘bring’	<i>/abb-a-tto/</i>
2prs.F	<i>/abbi/</i> ‘bring’	<i>/abb-a-tta/</i>
3prs.M	<i>/abbi/</i> ‘bring’	<i>/abb-anno/</i>

2.3 Summary

The tabular description of the tense suffixes that exist in the verb will show overall features of the language to the time aspect.

Table 3.3: Tense suffix for all person.

Person	Tense			
	Present perfect	Simple past	Present/future	Continuous tense (present and future)
1prs.M	-oo-	-u	-ee	-anni noommo
1prs.F	-oo-	-u	-ee	-anni noommo
2prs.M	-oo-	-i	-a	anni nootto
2prs.F	-oo-	-i	-a	anni nootta
3prs.M	-ino	-i	-anno	anni no
3prs.F	-itino	-u	-anno	anni no
1prp.N	-oo-	-u	-ee	-anni hee'noommo
2prp.N	-oonni	-i	-anni	anni heedhinoonni
3prp.N	Ino	-i	-anno	-anni no

2.4 Order of the suffixes

The order of suffix within a given word matters the meaning of it. The suffixes of verbs discussed above have a certain pattern of arrangement to convey carrying information within

the word. The suffix of person (pers) marker, tense/aspect, number, and gender follows the following structure.

For the 1st person the order of suffix for person, tense/aspect, and gender marker are not uniform. Number marker ‘-n’ in plural form precedence other markers and takes the following form such that ‘-n’+T/A+pers+gender marker respectively.

For the 2nd person the order of the suffix-marker for tens/aspect, person, and gender takes T/A+pers+Gende pattern respectively. The third-person singular masculine gender has no complexity to identify here are only T/A suffixes but for the 3rd person, singular feminine gender and 3rd person plural suffixes arrangement takes pers+T/A pattern if it is politely.

2.5 Stem Expansion

The verb expansion can be executed from the simple stem in many different ways. The techniques that are responsible for expansion are causative, auto-benefactive, passive, and reduplication (intensive).

2.5.1 Causative

Causative can be single or double.

2.5.1.1 Single causative

The addition of ‘-s’ morpheme to the simple verb stem bears the single causative in the language. Due to the attachment of causative morpheme to intransitive verbs, it became a transitive one. So that is why ‘-s’ is recognized as transitive.

For example,

Verb	Glosses	Single causative form	glosses
<i>/haaw-/</i>	‘slant’	<i>/haw-i-s-/</i>	‘he made things to be slant’
<i>/daw-/</i>	‘bend’	<i>/daw-i-s-u/</i>	‘she/they made to be bend’
<i>/dod-/</i>	‘run’	<i>/dod-i-s-/</i>	‘Cause to run’
<i>/dafur-/</i>	‘tired’	<i>/dafur-i-s-/</i>	‘became tired’

It is also possible to see initial transitive verbs attaching causative suffix to their stem. For example

Person Verb	glosses	causative suffix	glosses
3prs.F	<i>/it-/</i> ‘eat’	<i>/it-i-s-su/</i>	‘she feed’
3prs.M	<i>/tam-/</i> ‘slap’	<i>/tam-i-s-/</i>	‘he slap someone ‘

2.5.1.2 Double causative

Causative as the name indicates that it repeats the process of single causative. The suffix */s-ii-/* indicates that the verb is with double causative. Most of the time double causative is related to transitive verbs and the limited number of intransitive verbs. Sometimes double causative conveys the command to somebody. The number and persons are indicated between complement and causative suffixes. For example

Person Verb	glosses	double causative	glosses
3prs.F	<i>/it-/</i> ‘eat’	<i>/it-i-s-ii-s-u/</i>	‘she made somebody to eat’

2.5.2 Auto-benefactive

When suffix ‘-d’ attached on the simple verb stems that shows the concept for the benefit of oneself. For example, the verb ‘hayšš-‘ ’whash’ attached /-d/ to the stem changed to the /hayšš-i-d/ which means ‘wash oneself’

Verb	glosses	autobenefactive	glosses
<i>/dib- d/</i>	‘cover’	<i>/dipp’-/</i>	‘cover oneself’
<i>/sut- d/</i>	‘hang’	<i>/sutt-/</i>	‘hang for oneself’
<i>/wor- d/</i>	‘put’	<i>/wod-/</i>	‘put for oneself’

2.5.3 Passive

The verbs are the verbs that are not required to specify the actor of the action. In such way

They can express using /-am/ suffix in the language. For example

Verb	glosses	passive	glosses
<i>/fač-/</i>	‘split’	<i>/fač-am-/</i>	‘be splited’
<i>/giir-/</i>	‘burn’	<i>/giir-am-/</i>	‘be burned’
<i>/it-/</i>	‘eat’	<i>/it-ma-/</i>	‘be eaten’

CHAPTER FOUR: Research Design and Methodology

4.1 Introduction

The finite-state transducer is the preferred approach to implement the morphological generator and analyzer for a language with limited language resources and computational systems. The approach by itself required special designing techniques based on deep knowledge of the language and word formation. The followings are basic designing issues to be addressed.

The first one is lexicon/morphotactic design and the second one is the sound alternation design phase. By nature finite-state transducer morphological analyzer is bidirectional. This creates morphological analysis and generator at the same time using regular expression of the approach. The design for the morphological analyzer creates a conducive environment of morphological generator with minor modifications in the implementation process.

4.2 Methodology

The researcher reviewed relevant articles and journals to the study topic to collect inputs and to design research. For implementation reporting of the study, the researcher followed two methods to address the issue related to the study. The first one is experimental that the raw experimental output from a single test to the whole lexicon implementation. And the second part is quantitative that deals with the analysis of the experimental result. The second method represents experimental results in the term of numeric value representation that can be represented by percent.

4.3 Implementation tools and techniques

The researcher selected a Linux-based operating system special Ubuntu 16.04 to run foma finite-state automata/transducer software for the experiment. Foma is a free and open-source finite-state toolkit that has facilities for a variety of natural language processing applications. It can be considered as a programming language that has its compiler to create and use automata and transducers [1]. This experiment is conducted using foma finite-state transducer implementation software. There is optional implementation tool like OpenFST which is weighted for of FST that is very important for speech recognition and optical character identification. The implementation comprises lexicon files and rules that guide the pattern of linguistic word-formation. Notepad for lexicon design and to write language-specific rules that govern root to morpheme integration. Python programming language is used to develop web crawlers and tokenize documents into word.

4.4 Source of data

The lexicon for Sidaamu Afoo was collected from different sources. These include a website [55]. The data was collected from the website by using a web crawler developed (The code developed attached at appendix II of this document) by the researcher to extract the required information from third-party websites. Sidaama national regional state constitution, student modules of Sidaamu Afoo department of Hawassa University. Constitution and student module documents collected in the form of softcopy contacting respective body in person. The collected document passes through preprocessing steps and pure Sidaamu Afoo text identified at the end of the process. Lastly, the document is tokenized into words by the researcher using the NLTK python library. Tokenized list of words given to the language expertise to differentiate verbs from other parts of speech. The language expert identified

distinct 840 verb root words. These root words are inputs for the morphological generator and analyzer implementation.

4.5 Lexicon design

Lexeme is provided as an input to the system and the system analysis the given word into corresponding grammatical information based on the parts of the input word. The morphological generate follows the reverse order of the above. The lemma is given with appropriate tag sets and the system generate different word form. Different kinds of affixes are considered to be included within the experiment. Based on the nature of the verb they can be regular or irregular. The source of irregularity comes from the nature of language (e.g tonal) as stated in previous chapters.

4.6 Irregular verb

An irregular verb that never follows the known inflection pattern in according to described in the chapter3. These verbs can be characterized by geminate verbs (to provide grammatical functions). Example, different sound representation (to provide grammatical functions), example,

Root	3prs.F	3prs.M	1prp
/Aa/ 'give'	/uyitu/ 'she gives'	/uyi/ 'he gives'	/oommo/ ' we gave '

tone variation (to provide grammatical functions). Examples

4.7 Alteration rule

The orthographic, phonological, and morphophonological change at transducer output string handled. Here in the language most of the time in the infinitive form of a verb that has 'r' sound changed into 'd' sound in inflected verbs. Examples /hari/ 'he has gone' /hadhu/ 'she has gone'. 'w' sound sometimes replace by 'b' /gangawi/ 'he turns around again' /gangabu/ 'she turns around again. 'h' sound in the many words changed into 'k' sound to convey grammatical information (gender, number) at the certain circumstance. eg. /beehi/ 'he divided' and /beeku/ 'she divided'.

4.8 Tags

Tags that represent the grammatical meaning of corresponding morphemes within the given word form include.

- V to represent the root word class is verb
- N to represent the root word class is noun
- adj to represent the root word class is adjective
- 1prs to represent first person singular.
- F to represent feminine gender.
- M to represent masculine gender.
- 2prs to represent second person singular.
- 3prs third person singular.

- 1prp to represent first person plural
- 2prp to represent second person plural
- 3prp to represent third person plural morpheme.
- Pperfect to represent present perfect (/taaga/)
- Spast to represent simple past (/tene mule/)
- pre/futto represent present and future tense (/xa woyim albira/)
- Ctense to represent continuous tense (/harinshote aan nota/)

Lexicon is the result of much continuous class collection with the above tags and root word.

4.9 Continuous class

As stated in chapter 3 about the word-formation structure in the language. The inflection of the words encompasses many suffixes that carries different grammatical information within the given word. Those sequences of suffixes are assigned to the respective classes. The lexical scripting contains numerous continuous classes that make the backbone of analysis. Here is a list of classes considered for the experimentation.

person class

tense class

derived verbs

imperative class

negative prefix

number class

negative suffixes

cause

gender

Person class (PRSN)

Person class contains all suffixes that point specific person at a certain level. Which means 1st, 2nd, and 3rd in a singular and plural manner. Suffixes /-mm-/ for first-person /-tt-/ for second and /-i-/ /-ino/ for third person singular.

/-n-mm/ for first-person plural. The first part of 1prp suffix (-n) can change its position, assimilation, or geminate (last consonant of a root word) based on the nature of the last consonant of the stem/root word. The /-itini/ or /-itinoonni/ for second and /-i-or-ino/ for third person plural respectively.

Tense inflection class (TENIN)

Listed present perfect tense, simple past, and present/future tense suffixes. As described in chapter three (Table 3)

Imperative class (IMP)

Contains the second singular and plural commanding verb suffixes. Such as /-i/ for singular and /-e/ for second person plural.

Number class (NUM)

The class can be described whether geminate last consonant of the word or by person suffixes stated above in-person class.

Causative class (CAUS)

The cause can be single or double. The single causative can be represented by /-s-/ suffix and can be followed by gender suffix. The double suffix can be represented by /-siis-/ suffix.

Derived verbs (DV)

The derivation can be from a noun or adjective part of speech class. The suffix that verbalizes noun and adjective contains (-d, -'r, -', -p', -t').

Prefix

As discussed in chapter 3 about the language there is only one prefix in the language that makes word negative which is (di-/d-) according to the sound assimilation that follows prefix it can be one of the above. But the prefix /di-/ is not only a negative marker but also there is negative marker suffix.

Negative marker suffix class (NMSC)

The negative marker of imperative verbs is different from that of the prefix negative. The negative marker imperative verbs have /-toot/ for the second person singular and /-tinoonte/ or /-inoonte/ for second-person plural based on the nature of the ending consonant of the word. The negative marker of all jussive verbs ends with /-nke/ suffixes.

Gender

The gender marker for the first person masculine indicated by /-o/ and /-a/ for feminine.

Politeness verb suffix (PVS)

The manner of communication between two parties whether polite or aggressive indicated by suffix /-nni/

sample lexical design script

Multichar_Symbols -V -Pperfect -1prs -F -M -3prs -caus -Dcaus -prefix

LEXICON Root	bagi CNTNER;
verb ;	cifi CNTNER;
LEXICON verb	cigilfati CNTNER;
aa:eemma #;	ciiqi CNTNER;
aa:eemmo #;	ciiri CNTNER;
abb CNTNER;	LEXICON CNTNER
aan CNTNER;	TENIN;
adawi CNTNER;	IMP;
adawaani CNTNER;	CAUS;
addaxi CNTNER;	Prefix;
adhi CNTNER;	LEXICON PRSN1p
aduuli CNTNER;	-1prs:mmo #;
afi CNTNER;	LEXICON Prefix
ag CNTNER;	-prefix:di verb;
baari CNTNER;	-prefix:d verb;
baaxi CNTNER;	LEXICON IMP
baci CNTNER;	-V:i #;
badi CNTNER;	-V:e #;

LEXICON CAUS	-V-Pperfect:^oo PRSN1;
-2prs-caus:is IMP;	-V-Pperfect:^noo PRSN1p;
-Dcaus:isiis IMP;	-V-Pperfect:^oo PRSN2;
LEXICON PRSN1	-V-Pperfect:^i PRSN3pper;
-1prs:mm GENDER;	-V-Spast:^u PRSN1;
LEXICON PRSN2	-V-Spast:^nu PRSN1p;
-2prs:tt GENDER;	-V-Spast:^i PRSN2;
LEXICON PRSN2p	-V-Spast:^i PRSN3Sp;
-2prp:anni #;	-V-Spast:^i PRSN3;
LEXICON PRSN3	-V-pre_fut:^ee PRSN1;
-3prp:anno #;	-V-pre_fut:^nee PRSN1p;
LEXICON PRSN3pper	-V-pre_fut:^a PRSN2;
-3prs:no #;	-V-pre_fut:^0 PRSN3;
-3prs:tino #;	-V-pre_fut:^tin PRSN2p;
LEXICON PRSN3Sp	-V-pre_fut:^ PRSN2p;
-3prs:0 #;	-V-pre_fut:^0 PRSN3p;
-3prs:tu #;	LEXICON GENDER
LEXICON TENIN	-F:a #; -M:o #;

4.10 Rules design

The rules stated here govern different orthographic and phonological alterations of the word formation in the language.

4.10.1 Epenthesis Sound

Any word in the language cannot end with a consonant letter. This means the last letter should be from the vowel group. So to fulfill this language-specific rule all root words end with the letter /i/ as default. But it does not mean it will remain for all types of inflection and derivation of the words. There is two basic functionality with regarding /i/ sound while becomes at end of the word and in between two consonants. In the first case, it fulfills the language-specific rule as stated above. In the second case /i/ sound becomes in the middle of two consonants to assure no three consecutive consonants appear in the language. From the brief discussion, there is the possibility of /i/ sound deletion as well as insertion. The insertion part will be designed in the epenthesis part of this design. The nature of the suffix that comes next to the root word determines the fate of the /i/ letter at end of the root word whether to exist as it is or to be removed from the word. If the suffix began with vowel sound the persistence of sound /i/ at its position is too low. But the first letter of the suffix is consonant it will remain as it is.

4.10.2 Nasal metathesis rule

The rule that enters in between three consecutive consonants to avoid language-related rule clash. /i/ the sound that separates three consecutive consonants becoming in the middle. /n/ comes at a different position to indicate the first person plural. The position of the /n/ sound is determined by the nature of the last consonant in the word. /ʔ/ this glottal metathesis sound

may become before the /n/ sound according to the condition. /m/ most of the time 'm' epenthesis sound occurs in the words that end with 'w'.

Rule 1: to insert /i/ between three consecutive consonants.

Define insertion $[..] \rightarrow i \parallel CC-C$;

Rule 2: insertion of /n/ sound to indicate first person plural.

Define ninsert $[..] \rightarrow n \parallel V-C, ?*_Vm, 'Vm$;

4.10.3 Replacement rule

In the language, there is a different type of replacement rules. The first one is gender-related /w/ for masculine will be changed to /b/ to feminine sound for the same tense or aspects. When /r/ sound comes at the end of the word for masculine it will be changed completely to /d/ sound to represent feminine gender. /h/ sound that comes at the end of a word for masculine will be changed to /k/ sound for feminine gender.

Rule 3: /w/ replacement by /b/ for gender and person variation.

$w \rightarrow b, i \rightarrow u \parallel CV_i, _''^''$;

Rule 4: /r/ replacement by /dh/ for gender and person variation.

$r \rightarrow dh, i \rightarrow u \parallel CV_i, _''^''$;

Rule 5: replacement of sound /h/ by sound /k/ for gender and personal variation.

$h \rightarrow k, i \rightarrow u \parallel CV-i, _''^''$;

4.11 Graphical Representation of whole design.

Graphical representation of the overall internal lexicon and rule integration design is represented as follows

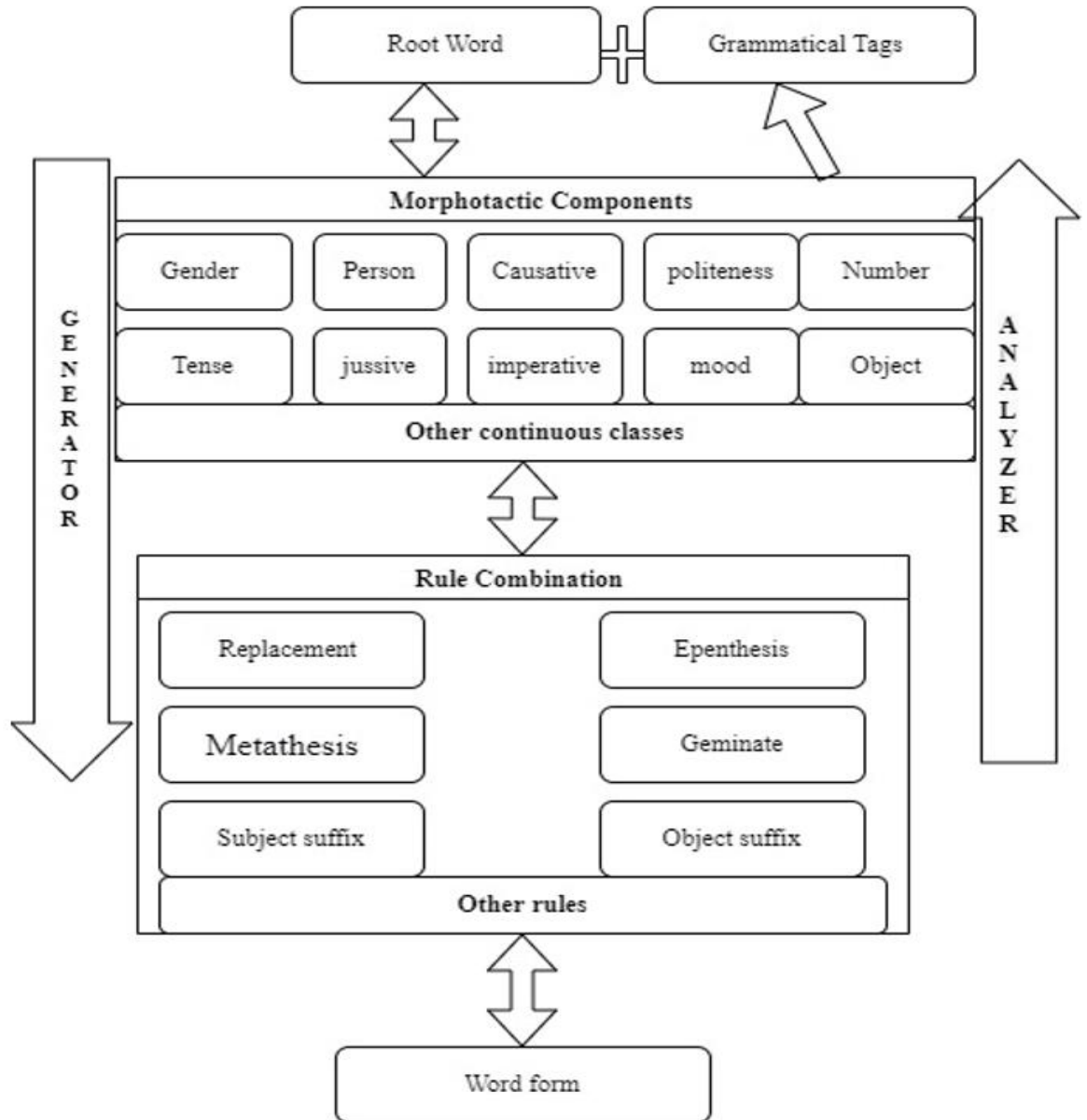


Figure 4.1: Diagrammatic view of Morphological generator and analyzer

4.12 Evaluation of Morphological System

Finite-state transducer morphological generator system efficiency measured against different word forms in the variate corpus [45]. This study evaluates the efficiency of the system by comparing generated words with the words that exist in the collected corpus. In another way, analysis is highly tight to the word generator. Analysis efficiency is measured based on the predefined tag sets and the combined meaning that they bear. The experimental result presented in the term of percental accuracy that obtained from difference in total generated words (Tg) and words generated wrongly (Tgw)

$$\text{Accuracy} = \frac{Tg - Tgw}{Tg} * 100$$

CHAPTER FIVE: Implementation and Discussion

Here is the finite state experimental result presentation and discussion. The experiment was obtained by running a lexicon on the inbuilt functions of foma system. Foma is a lexicon programming language selected to implement a morphological generator and analyzer for the Sidaamu Afoo . It is a natural language processing tool kit that enables the development of NLP applications. The reason that leads to using lexc language is to handle large amounts of lexicon with certain simplicity rather than regular expression in xfst. Experimental results were evaluated against the words that occur in the collected documents from a different source.

Techniquical aspects of foma software usage.

Foma software requires linux based operating system to be installed and configured.

The installation is possible with the terminal using two consicative commands the first one is *sudo apt-get update* and the second *sudo apt-get install foma-bin*. After installation running opening the terminal and by writing *foma* on the terminal it is possible to enter into software. The lexicon design scripted on the notepad and called from foma software being in the terminal using command *read lexc filename.lexc* the lexicon file should be saved with extension (*.lexc*) and *filename* can be any thing that given by the researcher. The rule defining component of the system also scripted on the notepad with file extension (*.foma*). *filename.foma* can call lexicon script from the same folder by the command *read lex filename.lexc in filename.foma* at top or any where it is important in the file. At last the researche can run *filename.foma* file from the folder by using command *sourc filename.foma* from foma software.

5.1 Word generation by doubling the last consonant

The last consonant geminate to provide grammatical information regarding the first-person plural. There are four (r → rr, l → ll, m → mm , n → nn) consonants that geminate to convey the grammatical information

Table 5.1 Grammatical geminate consonant 1prp word generation

Root word ending in	Percent share from total words	Possible replacements or 'n' follow	Geminate	Correctly generated word forms(%)	Wrong word-form generated(%)
'r'	15.6%	'(28%) dh(0.75%) n(9.1%),	Yes(62.12%)	100%	0
'l'	11.9%	N(13%)	Yes(87%)	100%	0
'm'	6.1%	N(29.41%)	Yes(70.6%)	100%	0
'n'	3%	N(20%)	Yes(80%)	100%	0

From the collected words, 15.6% of words end in 'r', 11.9% of words end in 'l', 6.1% of words end in 'm', 3% of words end in 'n' respectively. The phonological alternation of all above consonant sounds is determined by the preceding sound of the letters. The possible modification for the root words to accept a variety of grammatical morphemes that end with these sounds to convey tense information regarding the first-person plural while generating different word forms.

The nasal metathesis sound 'n' before the last consonant takes part in 32.6 %. The nasal metathesis sound after the last consonant takes a share of 26.2%. The last number conversion to glottal sound (') and followed by nasal metathesis sound 'n' take share 6.5 % and the last

Table 5.2: Nasal metathesis ‘n’ sound insertion for grammatical information

	Number of root words	Correctly generated	Wrongly generated	Success in percent
Root word take ‘n’ before last consonant	274	822	0	100%
Root word take ‘n’ after last consonant	220	660	0	100%
Root word take ‘n’ after last consonant replaced by glottal sound (‘)	55	150	15	91%
Root word take ‘m’ after last consonant replaced by /m-b/ in circumstance	64	159	33	83%

The factor that emanates error in glottal sound replacement in the word-formation process is that the compiler considers glottal (‘) symbols as operators. In many cases, the glottal symbol is represented by the letter ‘z’ but this rule may not be functional in some positions. According to the rule, ‘z’ replaces only the last glottal sound to bear the grammatical information of the word. When glottal consonants come in two or more different positions within a given word it hinders the functionality of the last one which bears the grammatical information of the word. The lexical operation related to the glottal sound affects the other part of the word for unnecessary implication. The reason there is a variable length of words that can take glottal sound at different positions. So the exact glottal sound position

determination is a challenging task. It can occur anywhere because of the rule which is defined to replace the symbol malfunction in some cases. Eg.

Root word Replacement (z) glottal insertion nasal metathesis(N)lprp

Su'nisi suznisi suznisizn suznnisizneemo

When after replacement formed word should be /su'nisi'neemo/ but the actual output of the system is /su'nisi'neemmo/ which is wrong. According to the linguistic rule three consonants cannot occur together consecutively and there is no meaning regarding the formed word.

The cause of error regarding the last consonant replacement to /b/ before insertion of nasal metathesis sound 'm' is that the replacement is optional. For that matter, complete replacement hides the other word-formation process. Most words that end with 'w' consonant have similar behavior that accepts /b/ replacement in addition to 'm' as nasal metathesis sound for the first person plural. By rewriting rules for the case most of the problem is solved but in some cases, there is a failure (Table 5).

5.3 Imperative verb result

The base form of the root word can be an imperative verb form or simple past based on the nature of the language. The tone of the speaker provides additional information. Whether the word is imperative or simple past for the second person singular. In the experimentation, all words perfectly fit this category. Unlike other forms of verbs, imperative verbs have negative marker suffixes as stated in the above chapters. Four different negative marker imperative suffixes require certain conditions to be attached to the root word as stated in chapter four. The experiment shows that:

Table 5.3: Imperative negative marker suffix

Person	Suffix	Root word number	Correctly generated	Wrongly generated
Second-person plural	Tinoonte	481	462	19
	Inoonte	359	338	21
Second-person singular	Tooti	481	462	19
	Ooti	359	338	21

The attachment rule of the above suffixes depends on the nature of the last two consecutive letters. When the last two consecutive letters are consonant without any vowel sound in between the /tinoote/ for second person plural and /toot/ for the second person singular follows the root word. When the last consonant of the root word is the follower of vowel sound the suffix that takes place is /inoonte/ for second personal plural and /oot/ for the second person singular.

5.4 Jussive verb result

There are two forms of jussive generation. The first part is the positive suffix addition according to the person and the second is generated by the negative marker suffix.

Table 5.4: positive jussive attachment generation

Person	suffix	Correct word form	Wrong word form
1prs.N	-o	840	0
3prs.M	-o-na	840	0
3prs.F	-t-o-na	840	0
1prp.N	-i-n-o	840	0

In positive jussive suffix attachment, there is not an ambiguous situation. Because of that, all rules work fine for all person. Sample screenshots.

5.5 Replacement

Replacement rules are partial operation which means there is an inflectional process without replacement for the second person singular masculine. The words which end in ‘ w ’ changed into ‘ b ’ and the words which end in ‘ h ’ replaced by the ‘ k ’ and the words which end in ‘ r ’ may change into ‘ dh ’ or glottal sound ‘ ’ ’. The total share of words within collected words that entertain replacement rules are 15.8%.

Table 5.5: Last consonant replacement and experimental result presentation

Number of words ending with	Total collected words	Number of words entertain replacement	Correctly generated words	Wrongly generated
W -> b	58	56	1,288 (50%)	1,288 (50%)

H -> k	14	14	232 (36%)	412 (64%)
R -> dh	35	35	460 (29%)	1,150 (71%)
R -> ‘	28	28	368 (29%)	920 (71%)

The cause of the problem is here, the first one is a partial replacement for the last consonants and there is no exact demarcation to determine when the replacement rule operates properly. The reason why there is an inflectional word-formation process without consonant replacement and at the same time the replacement rules required for some conditions. Glottals sounds are not recognized as one of Sidaamu Afoo consonant letters. This matter fact compiler detects not after glottal sound consonants and creat error word form. Even there was the replacement for glottal sound by rarely used consonant letter in the language like ‘v’ and the back replacement of glottal sound ‘’ at different position of word generates unnecessary word structure which has no meaning in the language.

5.6 Morphological Analyzer

Lexicon morphological analyzer output screen describes that how the morphological analyzer works using inbuilt functions.

```

kitaw@kitaw-Vostro-15-3568: ~/Desktop/Thesis_folder
File Edit View Search Terminal Help
2.0 kB. 22 states, 27 arcs, 16 paths.
foma[6]: pairs
saali-V-2prp-Ntr-impertv      saali^e
saali-V-2prs-Ntr-impertv     saali
saali-V-2prs-Int             saali
saali-V-3prs-M-Sprf         saali
rumi-V-2prp-Ntr-impertv     rumi^e
rumi-V-2prs-Ntr-impertv     rumi
rumi-V-2prs-Int             rumi
rumi-V-3prs-M-Sprf         rumi
wosani-V-2prp-Ntr-impertv   wosani^e
wosani-V-2prs-Ntr-impertv   wosani
wosani-V-2prs-Int           wosani
wosani-V-3prs-M-Sprf       wosani
xari-V-2prp-Ntr-impertv     xari^e
xari-V-2prs-Ntr-impertv     xari
xari-V-2prs-Int             xari
xari-V-3prs-M-Sprf         xari
foma[6]: up
apply up> rumi
rumi-V-2prs-Ntr-impertv
rumi-V-2prs-Int
rumi-V-3prs-M-Sprf
apply up>

```

Figure 5.2: experimental output of the morphological analyzer

The figure 5.2 above shows that for a given word */rumi/* which means in English ‘curse’ has grammatical meaning for three different conditions. The first analysis shows that the word means imperative which means *giving a command to the second person* to do something. The second one is the *asking second person* whether third person make any curse to any body. The third analysis shows that the word talks about *third person singular simple past* of word */rumi/*.

5.7 Morphological Generator

The morphological generator here includes all forms of suffixes discussed in chapters three and four. As seen from the screen single root word may generate a maximum of 92 distinct words.

```

kitaw@kitaw-Vostro-15-3568: ~/Desktop/Thesis_folder
File Edit View Search Terminal Help
iilli-V-2prs-Ntr-impertv iilli
iilli-V-2prs-Int iilli
iilli-V-Spast-3prs-M iilli
iilli-V-Spast-3prp iilli
iilli-V-3prs-M-Sprf iilli
iilli-V-Pperfect-3pr-F iillitino
iilli-V-Pperfect-3prp iillitino
iilli-V-pre_fut-2prp-imprfct iillitinanni
iilli-V-2prp-Ntr-impertv iillie
iilli-V-3prs-F-jsv iillitona
iilli-V-1prp-Ntr-jsv iillino
iilli-V-2prp-polite-Int iillinni
iilli-V-2prs-Dcaus-M iillisiisi
iilli-V-2prs-Dcaus-M-obj-M iillisiisisi
iilli-V-2prs-Dcaus-M-obj-F iillisiisise
iilli-V-2prs-Dcaus-F iillisiisu
iilli-V-2prs-Dcaus-F-obj-M iillisiisusi
iilli-V-2prs-Dcaus-F-obj-F iillisiisuse
iilli-V-pre_fut-1prp iillineemmo
iilli-V-Spast-1prp iillinummo
iilli-V-Pperfect-1prp iillinoommo
iilli-V-Spast-3prs-F iill
iilli-V-Spast-3prp iill
iilli-V-pre_fut-1prs-M iilleemmo
iilli-V-pre_fut-1prs-F iilleemma
iilli-V-Spast-1prs-M iillummo
iilli-V-Spast-1prs-F iillumma
iilli-V-Pperfect-1prs-M iilloommo
iilli-V-Pperfect-1prs-F iilloomma
iilli-V-Pperfect-2prs-M iillootto
iilli-V-Pperfect-2prs-F iillootta
iilli-V-Pperfect-3prp iillino
iilli-V-Pperfect-3prs-M iillino
iilli-V-pre_fut-2prs-M iillatto
iilli-V-pre_fut-2prs-F iillatta
iilli-V-3prs-M-jsv iillona
iilli-V-1prs-Ntr-jsv iillo
iilli-V-2prp-jsv-NgtM iilloonke
iilli-V-2prs-F-caus-obj-F iilluse
iilli-V-2prs-F-caus-obj-M iillusi
iilli-V-2prs-M-caus-obj-F iillise
iilli-V-2prs-M-caus-obj-M iillisi
iilli-V-pre_fut-2prp-imprfct iillanni
iilli-V-pre_fut-3prp iillanno
iilli-2prp-NgtM-impertv iill
iilli-2prs-NgtM-impertv iill
iibi-V-2prs-Ntr-impertv iibi
iibi-V-2prs-Int iibi
iibi-V-Spast-3prs-M iibi
iibi-V-Spast-3prp iibi
iibi-V-Spast-3prs-F iibitu

```

Figure 5.3: Morphological generated word forms from the single root word

The root word /iilli/ ‘arrive’ is generated into distinct word forms. This is the examples word for the rest of the words

5.8 Overall evaluation

The system generated 74,934 words from 840 root words. The correctness of generated word is evaluated against the collected words from a different source as stated in the methodology section. The evaluation technique demands word clustering based on the similarity in ending and phonological alternation. The cluster contains a minimum ten-word to thirty words based on the similarity of the above criteria. The words generated but the form of it doesn't exist in the document were evaluated by the linguistic expertise. The analysis is determined by the form of words generated by the system correctly. From total generated of which 3,898 are an incorrect form of a word due to some reasons stated in (Table 8, Table 6, Table 5,)

CHAPTER SIX: Conclusion and Future work

6 Conclusion

Sidaamu Afoo is one of the under-resourced languages and with very small morphological studies. The first attempt of the study was based on a machine learning approach that

requires a large amount of corpus. Here is a controversial notion regarding the previous study, which requires tremendous corpus for experimentation of the morphological for this under-resourced language. On the other side, newly formed sidaama national states require the advancement of the language that benefits this state. Advancement in the morphological generator and analyzed development is very crucial right now further utilization of the language for the many high-level NLP application development. To achieve the development of a morphological generator and analyzer for the language, this study was conducted by using a finite state transducer design and implementation approach. Even if some challenges were faced in the middle of the study the researcher got good achievements. The challenges were a spelling error that comes from a different data source which was corrected by the linguistic expertise and a lack of exact demarcation of replacement rule which was not stated clearly in the linguistic documents. Another challenge was the twenty-ninth glottal consonant of Sidaamu Afoo (') was never recognized by the compiler in the computation and the researcher tried to replace it with a less frequently used consonant (z) and achieved some results for the words with only one glottal consonant. But for more than one glottal sound occurrence in the given word during computation and replacement of glottal sound (') creates abnormality of the word-formation.

A finite-state transducer demands a language-specific rule definition. The rule stated here is to reshape the morphotactic of the lexical design of the language. Lack of predefined rules and morphotactic arrangement of words in the organized documents about the Sidaamu Afoo language leads the research to take much time of the study in understanding the things in detail from linguistic expertise. The tasks were grouping different words to relevant categories that have similar behavior and identify their nature. Most of the tasks were constructed from scratch to meet the experimental objective of the study. The research

defines 27 unique continuation classes for language design. Each class contains one up to four suffixes for the new word construction process. There were 840 root words from which 74,934 words were constructed/formed using those classes and 41 distinct rules defined. Out of the total generated 74,934 words, 3,898 words were incorrect. This shows that 5.2% of words are wrong. The experiment shows that 94.8% of the words generated are correct. And all correctly generated words are analyzed perfectly without any challenge.

6.1 Future work

The future works that are partial or not included in this study, but maybe need to be accommodated in the consecutive studies.

- This Study focused on word generation and analysis of Sidaamu Afoo verbs. But different word classes require study and implementation as did for the verbs.
- Compound words are not included in this study even for the verbs and they can be improved in the future.
- Words with many glottal sounds and optional consonant replacement rules require further investigation.

REFERENCES

- [1] T. Teklehaimanot, "Ethiopian Treasures,," ethiopiantreasures, [Online]. Available: <http://www.ethiopiantreasures.co.uk/pages/language.htm>. [Accessed 18 October 2020].
- [2] Joshua Project, "Joshua Project,," Joshua Project, [Online]. Available: <https://joshuaproject.net/languages/sid..> [Accessed 9 November 2020].
- [3] K. a. z. u. h. i. r. o. K a w a c h i, "A grammar of Sidaama, a Cushitic language Ethiopia," State University of New York at Buffalo, 2007.
- [4] A. Kemal, "Speaker dependent speech recognition for sidaama language," Addis Ababa University respository, 2010.
- [5] D. L. Bolinger, "On Defining the Morpheme," 1948.
- [6] P. M. D. K. Malladi, "in Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically Rich Languages," Seattle, Washington, USA, 2013.
- [7] V. Dhanalakshmi, M. Anandkumar, R. U. Rekha, C. Arunkumar and S. K. P, "Morphological Analyzer for Agglutinative Languages Using Machine Learning Approaches," in in Computational Engineering and Networking, Coimbatore India, 2009..
- [8] V. Olga and Nagel, "Morphological and Evaluation Effect within Russian Syncretic," in in THE XXVI ANNUAL INTERNATIONAL ACADEMIC CONFERENCE, LANGUAGE AND, Tomsk Russia, 2015.
- [9] M. Gasser, "HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya," Indiana University Bloomington, 2011.
- [10] A. WOLDEMARIAM, "Developing Morphological analyzer for Affan Oromo text," Addis Abeba University Addis Abeba, 2005.
- [11] M. D. Mosa, "Morphological Analyzer for Afaan Oromoo Using Machine Learning," Addis Abeba university thesis, 2020.
- [12] T. BAYU, "AUTOMATIC MORPHOLOGICAL ANALYZER FOR AMHARIC AN EXPERIMENT EMPLOYING UNSUPERVISED LEARNING AND AUTOSEGMENTAL ANALYSIS APPROACHES," Addis Abeba university , 2002.
- [13] D. Legese, "Morphological Analyzer for Sidaamu Afoo," Addis Abeba University, 2020.
- [14] Z. Jakub and D. Walter, "Memory-Based Learning: Using Similarity for Smoothing," in Computational Linguistics, Tilburg the Netherlands, 2016.

- [15] K. Shaalan, "Rule-based Approach in Arabic Natural Language Processing," *International Journal on Information and Communication Technologies*, vol. 3, 2010.
- [16] I. A. A.-S. a. I. A. I. A. Al-Kharashi, Al-Sughaiyer, I. A and Al-Kharashi, "Arabic Morphological Analysis Techniques: A Comprehensive Survey," in Wiley, Riyadh, Saudi Arabia, 2003.
- [17] L. Judith, Klavans, S. Martin and Chodorow, "Using a morphological analyzer to teach theoretical morphology," Springer, vol. 25, pp. pp. 281-187, 1991.
- [18] Y. Abate, "Morphological Analysis of Ge'ez Verbs Using Memory Based Learning," Addis Abeba University , 2014.
- [19] G. E. Booij, "The Grammar of Words: An Introduction to Linguistic Morphology," New York: Oxford: Oxford University Press, 2007.
- [20] U. o. Sheffield, "all-about-linguistics," University of Sheffield, [Online]. Available: <https://all-about-linguistics.group.shef.ac.uk/branches-of-linguistics/morphology/what-is-morphology/>. [Accessed 9 December 2020].
- [21] F. a. U. d. Nemo), "Morphemes and Lexemes versus “Morphemes or Lexemes ? in Morphology and Linguistic Typology," in Proceedings of the 4th Mediteranean Morphology Meeting (MMM4), Catania , 2003.
- [22] D. B. Weldegiorgis, "DESIGN AND IMPLEMENTATION OF AUTOMATIC MORPHOLOGICAL ANALYZER FOR GE'EZ VERBS," Addis Abeba University , 2010.
- [23] S. S, "MORPHOLOGICAL ANALYZER FOR MALAYALAM VERBS," AMRITA VISHWA VIDYAPEETHAM, 2008.
- [24] L. Bauer, "Introducing linguistic morphology, 2 ed," Edinburgh: Edinburgh University press, pp. pp. 140-142, 2003.
- [25] A. M. Zwicky, "Clitics and Particles," *Linguistic Society of America*, vol. 61, pp. 283-305, 1985 .
- [26] M. K. P. Arkadiev, "MORPHOLOGICAL THEORY AND TYPOLOGY," Leiden University, 2016.
- [27] Z. Byoung- Tak and K. Yung- Taek, "Morphological Analysis and Synthesis by Automated Discovery and Acquisition of Linguistic Rules," *Seoul National University*, vol. 2, 1990.
- [28] G. K. a. I. Cicekli, "Rule Based Morphological Analyzer of Kazakh Language," Hacettepe University, 2014.
- [29] S. Lushanthan, A. R. Weerasinghe and D. L. Herath, "Morphological Analyzer and Generator for Tamil Language," in *In International Conference on Advances in ICT for Emerging Regions ICTer2014*, Colombo, 2014.

- [30] K. R. B. a. L. Karttunen, "Finite-State Non-Concatenative Morphotactics," in In Proceedings of the Fifth Workshop of the ACL Special Interest Group in Computational Phonology, Luxembourg, 2000.
- [31] L. P. a. S. Bosch, "Finite-State Computational Morphology: An Analyzer Prototype for Zulu," Springer, vol. 18 , pp. pp. 195-216, 2004.
- [32] A. K. M, "Morphology based prototype statistical machine translation system for English to Tamil language , A Thesis Submitted for the Degree of Doctor of Philosophy in the School of Engineering," Amrita School Of Engineering Amrita Vishwa, 2013 .
- [33] N. H. O. R. R. Eskander, "Automatic Extraction of Morphological Lexicons from Morphologically Annotated Corpora," Columbia University, 2013.
- [34] F. Katshemererwe, "Computational morphology and Bantu language learning," University of Groningen, pp. pp. 15-16, 2013.
- [35] britannica.com, "britannica," britannica.com, [Online]. Available: <https://www.britannica.com/science/linguistics..> [Accessed 13 1 2021].
- [36] oreilly.com, "mastering-natural-language/9781783989041/ch03s05.htm," O'Reilly Media , [Online]. Available: <https://www.oreilly.com/library/view/mastering-natural-language/9781783989041/ch03s05.html>. [Accessed 30 December 2020].
- [37] A. Ranta, "Computational Morphology: Introduction," Malta European Masters Course, 2011.
- [38] B. Alan, S. G. Pulman, G. D. Ritchie and G. Russell, "Computational Morphology of English," Technical report Published by the university of Cambridge computer labratory are freely available via internet, Cambridge, 1989.
- [39] H. Uiibo, "On Using the Two-level Model as the Basis of Morphological Analysis and Synthesis of Estonian," in In Proceedings of NODALIDA, Tartu, 1999.
- [40] D. Kumar, M. Singh and S. Shukla, "FST Based Morphological Analyzer for Hindi Language," in In Department of Computer Science & Engineering, JSS Academy of Technical Education Noida, Noida, 2012.
- [41] M. P. C. Mahlow, "Systems and Frameworks for Computational Morphology," in In Third International Workshop SFCM 2013 Berlin Proceedings, Berlin , 2013.
- [42] A. Saba and G. Dafydd, "Finite State Morphology of Amharic," Researchgate, 2014.
- [43] A. Mesfin and A. Yaregal, "Development of Amharic Morphological Analyzer Using Memory-Based Learning. In: Przepiórkowski A.," in Advances in Natural Language Processing. NLP 2014. Lecture Notes in Computer Science, Cham, 2014.

- [44] T. Abebe, J. N. Washington, M. Gasser and B. Yimam, "A Finite-State Morphological Analyzer for Wolaytta," in In International Conference on Information and Communication Technology for Develoment for Africa, Addis abeba, 2018.
- [45] A. G. M. Bhuvaneshwari C Melinamath, "A Morphological Generator for Kannada Based on Finite State Transducers," IEEE, 2011 .
- [46] K. R. Beesley, "Arabic Finite-State Morphological Analysis and Generation," Rank Xerox Research Centre, 1996.
- [47] I. Said, Y. Abdellah, B. Mostafa and A. Driss, "Arabic Morphological Analysis Based On Graphs And Correspondence tables Between Affixes AndRoot," in Mohammed V University, Rabat, Morocco, 2018.
- [48] I. A. A.-S. a. I. A. I. A. Al-Kharashi, Al-Sughaiyer, I. A and Al-Kharashi, "Arabic Morphological Analysis Techniques:A Comprehensive Survey," Wiley, 2003.
- [49] Y. H. M.-W. K. G. Yand, "A KOREAN MORPHOLOGICAL ANALYZER FOR SPEECH TRANSLATION SYSTEM," Dongguk Univ, 1996.
- [50] H. G. P. B. M. Bapat, "A Paradigm-Based Finite State Morphological Analyzer for Marath," in In Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP).
- [51] X. A. a. K. S. INAKI ALEGRIA, "Automatic Morphological Analysis of Basque," Informatika Fakultatea, Basque Country University, vol. 11, 1996.
- [52] G. S. L. Vishal Goyal, "Hindi Morphological Analyzer and Generator," IEEE, 2008 .
- [53] G. A. D. a. R. Meyer, "LISSAN JOURNAL OF AFRICAN LANGUAGES & LINGUISTICS," ADDIS ABABA UNIVERSITY ETHIOPIAN LANGUAGES RESEARCH CENTRE, vol. XX , 2006.
- [54] M. W.Gorgisi, teaching manual, Hawassa, 2003.
- [55] W. T. B. a. T. S. o. Pennsylvania, "/sid/," Jehovah's Witnesses, [Online]. Available: <https://www.jw.org/sid/>. [Accessed 20 1 2020].
- [56] "sida," 1 04 2020. [Online]. Available: <https://www.jw.org/sid/>.
- [57] K. M. Yri, "Orthography and Phonology in Sidaamu Afoo (Sidamo)," Institute of Ethiopian Studies, vol. 37, pp. 41-56, 2004 .

APPENDIX I: MORPHOLOGICAL ALTERNATION RULE:

THAT GOVERNS MORPHOSYNTACTIC

Vowels

```
define V [ a | e | i | o | u | a a | e e | i i | o o | u u ];
```

consonant in Sidaamu Afoo

```
define cns [ c | n | j | d | b | f | k | h | l | z | s | v | x | g | m | p | q | r | t | w | y ];
```

substrings that determine the nature of suffixes attached to the root word in Sidaamu Afoo

.

```
define Wv [ b a a | t i | b u | r i | y i | i i | r c i | g o o | f i | j i | h a a | h a | h e e | x i | k i | s o |  
d i | q i | s i ];
```

```
define wend [ c i | d i | d i i | f i | g i | h i | j i | k i | q i | n i | m i | v i | s i | t i | x i | w i ];
```

#READ LEXICON FILE

```
read lexc Sida.lexc
```

```
define Lexicon;
```

#Rule #1: #geminate last consonant to Convery first person plural.

```
define ConsonantDoubling l -> l l , r -> r r , m -> m m , n -> n n || V _ i "+" [ u m m o | e e  
m m o | o o m m o ] ,,
```

```
n -> 0 || "+" _ ,,
```

```
i -> 0 || _ "+" ;
```

#Rule #2: 'sh' replacement by single consonant V

```
define replacesh s h -> v ;
```

#Rule #3: N Insertion after to indicate first person plural: for words containing glottal sound at the end

```
define replacez ' -> z || _ n i , _ m i ;
```

#Rule #4 Nasal metathesis sound 'n' and replacement sound '' that replace ri,ei ending, and Nasal metathesis sound 'm' and 'b' replacement of 'w'

```
define NInsertion r i -> ' n || Wv _ "+" ,,
```

```
i -> ' n || e _ "+" ,,
```

```
w i -> m b i || V _ "+" ,,
```

```
[.] -> n || b i _ "+" , l l i _ "+" , cns wend _ "+" , cns d i i _ "+" , lodi _ "+" , modi  
_ "+" , y i _ "+" ;
```

#Rule #5: Nasal metathesis sound insertion before any consonant that fulfills conditions

```

define Ninserbwtw [..] -> n || V _ wend "+" ;
#Rule #6: V replacement to its original consonant sh
define replacehs v -> s h ;
#Rule #7: glottal letter (apostrophe) replacement to its original position
define zreplace z -> ' || _ n i, _m i ;
#Rule #8: simple past representation for thrid person feminine using 'u' depending on the
condition
define Addu [..] -> u || wend _ "-" ,,
    i -> 0 || cns _ "^" ,,
    t u -> 0 || "^" "-" _ ;
define prei [ l | r | m | n ] ;
#Rule #9: simple past representation for thrid person feminine using 'tu' depending on the
condition
define Addtu [..] -> t u || b i _ "-" , n i _ "-" , cns wend _ "-" ,,
    i -> 0 || prei _ "^" ;
define woot [ c i | d i | f i | g i | j i | k i | q i | s i | t i | x i | y i ] ;
#Rule #10: imperative negative marker suffix (-ooti)addition to the root word for the second
person singular.
define Addoot i -> 0 || _ "*" , [..] -> o o t i || V woot _ "*" ,,
    o o t i -> 0 || "*" _ ;
#Rule #11: imperative nagative marker suffix (tooti)addition to the root word for second
peson singular.
define Addtoot [..] -> t o o t i || cns woot _ "*" ,,
    t o o t i -> 0 || "*" _ ;
#Rule #12: imperative nagative marker suffix (inoonte)addition to the root word for second
peson plural.
define Addinoonte [..] -> i n o o n t e || V woot _ ")" ,,
    i n o o n t e -> 0 || ")" _ ;
#Rule #13: imperative nagative marker suffix (tinoonte)addition to the root word for second
peson plural.3
define Addtinoonte [..] -> t i n o o n t e || cns woot _ ")" ,,
    t i n o o n t e -> 0 || ")" _ ;
#Rule #14: present perfect suffix addition to third-person using "ino" suffix

```

```

define Addino [..] -> i n o || _ "(" ,,
    V -> 0 || cns _ "(" ,,
    i n o -> 0 || "(" _ ;

#Rule #15: present perfect suffix addition to third person using "tino" suffix
define Addtino [..] -> t i n o || b i _ "^" "(" , n i _ "^" "(" , cns wend _ "^" "(" ,,
    i n o -> 0 || "^" _ "(" ,,
    t i n o -> 0 || "^" "(" _ ;

#Rule #16 Epenthesis sound deletion:- to let next vowel to be functional(i), to make previous
letter obstruent.
define IDeletion i -> 0 || _ "^" [ o o | e e | u u | i | a | u | o ] ,, #
    i -> 0 || V l _ "^" t ; #

#Rule #17 prefix insertion at the beginning of the word
define prfxDeletion di -> 0 || _ "-" & "^" [ tooti | ooti | tinoonte | inoonte ] ;

#Rule #18 #replacement some consonants: b->w w->b h->k
define Replacementcons w ( -> ) b , h ( -> ) k , r ( -> ) d h , r ( -> ) ' || V _ i "^" [ t u | o o t i | i
n o | i n a n n i ] ,, #
    i -> 0 || _ "^" [ o o m m o | e e | u u | i | a | u | o ] ;

# Cleanup: remove morpheme boundaries
define Cleanup "^" -> 0;
define Cleanupp "+" -> 0;
define Cleanuph "-" -> 0;
define Cleanubrck "(" -> 0;
define Cleanasr "*" -> 0;
define Cleanarbck ")" -> 0;
read lexc Sida.lexc
define Lexicon
define Grammar Lexicon .o.
    replacesh .o.
    replacez .o.
    NInsertion .o.
    ConsonantDoubling .o.
    Ninserbtw .o.

```

rerpresntr	.0.
replacehs	.0.
Addoot	.0.
Addtoot	.0.
Addinoonte	.0.
Addtinoonte	.0.
Addtu	.0.
Addu	.0.
Addino	.0.
Addtino	.0.
prfxDeletion	.0.
Replacementcons	.0.
IDeletion	.0.
zreplace	.0.
Cleanupp	.0.
Cleanubrck	.0.
Cleanuph	.0.
Cleanasr	.0.
Cleanarbck	.0.
Cleanup;	

regex Grammar;

APPENDIX II: WEB CRAWLING CODE

```
import requests #to request web page library
from html.parser import HTMLParser
import re
import time
import codecs
import os #access file path
from bs4 import BeautifulSoup # to parse document
x=1
count=0
link_lists=[]
data1=""
bsurl="https://www.jw.org"#"https://www.jw.org"
url=input('Base URL_Address: ')
send_request(url)

def crawl(pageObject, x): #collects only string file from the page contet

global data1
    data1=""
    for tt in pageObject.stripped_strings:
        data1+=tt + '\n'
    create_data_files('jworg','docGU_'+format(x), data1)
def create_data_files(project_name, filetype, data1): # create file name and call write finction
    fileName = os.path.join(project_name , filetype +'.txt')
    if not os.path.isfile(fileName):
        write_file(fileName, data1)
```

```

def write_file(path, data2):# writ text file to txt file
    with codecs.open(path, 'w','utf8') as f:
        f.write(data2)
def link_finder(pageObject): # collects links from each page
    global count, x
    global link_lists
    links=pageObject.find_all('a')
    for link in links:
        linker=link.get('href')
        if count<len(links):
            if linker!=None and '/sid/' in linker :
                # if linker!=None and 'Sidamigna/' in linker :# for anothe website scaping
codehttp://www.gudumaale.com/SidaamuAfo.html
                if linker.startswith('http'):
                    check_repeat(link_lists,linker)
                else:
                    linker=bsurl+linker
                    check_repeat(link_lists,linker)
    if count <=509:
        time.sleep(10)
        send_request(link_lists[count])
def send_request(url): # send request to web server and get retern from the server
    global x, count
    page= requests.get(url) # request server page which is displayed on given url
    pageObject =BeautifulSoup(page.content,'html.parser')
    print(page.headers)
    crawl(pageObject,x)
    x+=1
    count+=1

```

```
link_finder(pageObject)
def check_repeat(abe,key):# check wether the url address repeated in the list variable
    i=0
    for ii in abe:
        if i<len(abe):
            if ii==key:
                break
            elif ii!=key:
                i+=1
                continue
    else:
        abe.append(key)
```

APPENDIX III: SIDAAMU AFOO ROOT WORD LIST OF VERBS

USED FOR EXPERIMENTATION.

aa	ammani	batiri	boroodi
aani	amaxxi	battawi	boroori
aarawi	araari	baxi	borreessi
abbali	arreedii	bedeedi	bowi
abbi	ariifati	beewi	buki
adami	arrasi	behi	bulbuli
adawaani	arraawi	beleqi	buli
adawi	assi	bexi	bulli
addaxi	atooti	bici	buluuli
adhi	ayirrisi	biifi	buqqisi
aduuli	baari	bikki	huri
adaari	baasi	bilbili	burqi
adami	baaxxi	bilici	buruuxi
adumi	babbi	bililiisi	bushi
afi	bachuuri	billaali	bushshi
agari	baci	biqi	buti
agawi	badi	boci	buuri
agi	bagi	bodiri	buusi
aguri	bai	bokkeesi	buuwi
aji	balaxi	boohari	buuxi
akeeki	balli	booli	buwi
akkali	baqqali	booni	buxi
albisi	baragi	boorasi	caabbishi
allaali	bararaaqi	boori	caaccawi
allawati	barkiri	boowiri	caaqqi
amaali	barri	borati	cabbi
amadi	bashili	borawi	caci

caciissi	corqi	dasi	dugi
cali	coyiri	dawari	duhi
canci	cu'mi	dawi	duli
canfi	cuami	dayi	dumi
cawati	cubbisi	deei	duni
cee'mi	cuuki	deesi	duti
cefeleeqi	cufi	dei	duu'mi
cei	cululluqi	dhami	duuri
cibbi	cuqqi	dhawi	duuwi
cifi	da'ati	dhiwi	duwi
cigilfati	daadi	dhoohi	edderami
cigilifati	daafi	diigi	eegami
ciiqi	daafuri	diiri	eemari
ciiri	daahi	dikkiri	eessi
cili	daandii	dimbi	efesi
ciliqisi	daansi	dini	effiri
cinci	daasi	diri	egemi
cingi	daassi	diriiri	egenni
ciri	daddali	diwi	ei
ciriiqi	daddilli	dodi	eli
ciriqi	dagaagi	dogi	ereeri
citi	dalagi	dogoorri	ergiri
ciwii	dammaqi	doli	eri
codhi	dammi	dollisi	ewi
coi	danqi	doodhi	faaci
coli	dansi	doogisi	faarsi
coni	daraari	doofari	faasiqi
coomi	darami	doyi	faci
coommi	dari	du'mi	fafi
cori	darishi	duduwi	fajji

fali	fugudi	gatamari	gubbi
fani	fuli	gati	gudi
faradi	fullahi	gawajji	gufiri
farciri	fuqqi	gawali	gugi
farrami	furgugiisi	geedhi	gulufi
farriri	furi	geegiissi	guluphi
feceleeqi	furmi	geewi	gumaati
feeffati	fuugi	gei	gumuli
feessi	fuuri	gereeri	gurdi
fenjeejji	fushshi	geeraari	guri
ficiriiqi	fuuti	giddeessi	guti
fiigi	ga'mi	giddi	guuqi
fiili	gaabbi	giiri	guuri
fiinci	gaaci	giwi	guwi
fiiqi	gaadi	go'li	guxxi
fili	gaa'li	godo'li	ha'li
filikisi	gaamaari	godowi	haanjiri
finci	gaami	golodi	haaqiiri
finqili	gaarami	gombisi	haari
fishshi	gafi	gomichi	haariimi
fixxi	gaggawi	gongo'mi	haawi
fokki	galati	goofi	hafuuri
fonqoli	galchi	goori	hagi
fooci	galfati	goowi	hagiidhi
foogali	gali	gori	hajaji
fooli	gambi	gorri	hajami
foorari	gammati	goshooshi	hala'li
foori	gani	gowi	halami
fotoqi	gari	gowwii	halchi
fugi	gashshi	goxi	hanaaqi

hanafi	hiiqqi	huuci	jeeji
hanqafi	hiissi	huunci	jifi
hanqati	himani	huuwi	jiili
hanqi	himimisi	huwati	jili
harfaawi	hinaasi	huxiri	joogi
harfati	hiri	huxxi	joorjori
hargani	hirisiisi	iddi	ka'li
hari	hixamanyi	iibi	kaadi
harunsi	hobbahi	iilli	kaajji
hasaawi	hodhi	ikki	kajjeeli
hashaashi	hordofi	ili	kaasi
hasi	hogi	ililisi	kadi
hawi	hogoofi	illaalisi	kaffali
hawuluki	songi	illisi	kai
hawuuri	hoogi	inji	kakkachi
hayishshi	hooli	iraawi	kakkali
hayyeessi	hordofi	irbi	kalaqi
heeri	horri	irkiri	kali
heewisi	hoshoosh	iti	kalli
hegersi	hosi	ittani	kaphi
hembeeli	hudeeli	iyyahi	kari
hettisi	hudiri	iwiiwi	karsi
hexxi	hufi	jaawaati	kashili
hiddani		jaanjuri	kassasi
hidhi	huluulli	jaawi	kawali
hifati	huluuqi	jaji	kaxxi
higi	huni	jajjari	kee'mi
hiikki	hurgufi	jallaagi	keeni
hiinci	huri	jamari	keerawi
hiinxiri	hururami	jeefi	keeshshi

ki'i	laaxi	maasiri	meecci
kibbi	laci	maaxaari	melegi
kiifi	laddi	maaxi	memiisi
kiili	laggi	macciishsh i	mereedhi
kiiri	lai	madaari	miici
kiisi	lalawi	madiri	miildagi
killi	laqqi	magadi	miliqi
kimi	lassi	magansi	millisi
kirbi	lati	makali	mini
kisi	lawi	malaatisi	mishi
kofi	ledi	malawi	misi
koyisi	leelli	mali	mixiri
koofi	lei	malli	mixxi
koori	lipheessi	mamahi	mololi
koqi	lix	mancai	moohi
kori	lolahi	maqqaalli	mooli
kubbi	lolohi	marari	moori
kuli	loosi	mari	moromi
kumbuuli	loowi	marqi	muddi
kuri	lophi	masaali	muudi
kusi	lugi	masali	mugi
kutuwi	luphisi	masi	mulqi
kuukisi	luqai	massagi	muluugi
kuusi	luuli	massi	mundii
laafi	luuxi	maxari	murciri
laalaati	ma'li	maxi	muri
laali	maaeeli	mayi	musiri
laanshaawi	maafi	mayyi	muuxi
laaqi	maalali	meceqqissi	muxi
laashshi	maari		naadi

naafi	qarri	qufi	saga'li
naaxxi	qasi	qullaawi	sai
nabbawi	qawaaxxiri	qunqumi	sammari
nagasi	qaxari	quqquxi	seedi
naqi	qaxxai	qurbaani	seejji
niixi	qeeli	quuri	seekki
noo	qeenfati	qurti	seeri
odeessi	qi'miidi	qussi	seesi
ofolli	qiidi	quxi	seyi
okki	qiishi	raami	shaali
oli	qinii	raari	shaari
ollohi	qishi	rabbisi	shafi
oni	qixxawi	ragiri	shalaqi
oo	qoci	rahi	shali
oofi	qodhi	rai	shallagi
oso'li	qolchi	rarraai	shami
owaati	qoli	reewi	shaqqi
qa'mi	qoldi	reyi	sharrami
qaafi	qoodi	riqiwi	shawali
qaagi	qoomi	roori	sheekki
qaci	qoori	rosi	sheeli
qacifati	qoosi	rowi	sheerumi
qaddai	qori	royi	sherki
qafadi	qorophi	rumi	sheshi
qajeeli	qorri	rumuxi	sheti
qali	qoromodi	ruukki	shii
qalxi	qorqori	saali	shiili
qama7si	qoti	saami	shiishi
qansi	qoqqowi	sabbaki	shiqi
qaraxi	qoxxowi	safari	shishiqqi

sholi	sisiwi	sufi	tari
sholli	sissilli	sunqi	tee'li
shoomi	somi	suri	teesi
shooshaqi		suti	tidhi
shori	soodi	suufi	timi
shorri	soofi	suuqi	tiri
shoshshog gi	soorri	suuri	tiyiri
shoti	soori	suwi	totali
siikisi	soqqi	taafisi	towaati
sirbi	sori	taali	tugi
siri	soyi	tai	tultawi
siriqi	su'mi	tami	
	su'nii	tareessi	
tunsi	waahi	wodanchi	xa'ni
tuqi	waajji	wodhi	xaadi
tuqqi	waami	wodi	xaami
tumi	waari	wolaphi	xaari
uddiri	waaxi	wolqisi	xafii
uduli	waci	wonqiri	xagari
ufi	warxi	wonshi	xagisi
ufuufi	wi'li	worbi	xalali
uggi	wigidiidi	wori	xalli
uli	wiini	wosani	xari
umi	wiisi	woshi	xawi
uri	wiliili	woshshaad i	xei
usuri	wiri	wosinsi	xiinxalli
uuri	wirisi	woxxi	xiiwi
uwi	wishi	woyi	xinqi
waaci	wixi	woyi	xinti
waadi	wo'naali	xa'mi	xissi

xiwi

xoi

xoni

xoqoli

xooqi

xorri

xoxiisi

xunsi

xuri

xuruuri

xuuri

xuuxi

yaadi

yaali

yari

yaasi

yekkeeri

yii

yori