



JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATICS
DEPARTMENT OF INFORMATION TECHNOLOGY

DESIGNING AND DEVELOPING STEMMER FOR GE'EZ
LANGUAGE TEXT: A HYBRID APPROACH

BY: SOLOMON NIGATU

THESIS SUBMITTED TO FACULTY OF COMPUTING JIMMA
INSTITUTE OF TECHNOLOGY IN PARTIAL FULFILLMENT FOR
THE DEGREE OF MASTERS OF SCIENCE IN INFORMATION
TECHNOLOGY

NOVEMBER, 2021

JIMMA, ETHIOPIA

JIMMA UNIVERSITY
JIMMA INSTITUTE OF TECHNOLOGY
FACULTY OF COMPUTING AND INFORMATICS
DEPARTMENT OF INFORMATION TECHNOLOGY
DESIGNING AND DEVELOPING STEMMER FOR GE'EZ
LANGUAGE TEXT: A HYBRID APPROACH

BY: SOLOMON NIGATU

THESIS SUBMITTED TO THE FACULTY OF COMPUTING OF
JIMMA INSTITUTE OF TECHNOLOGY IN PARTIAL
FULFILLMENT FOR THE DEGREE OF MASTER OF SCIENCE IN
INFORMATION TECHNOLOGY

ADVISOR: SOLOMON TEFERRA (PhD)

NOVEMBER, 2021

JIMMA, ETHIOPIA

JIMMA UNIVERSITY
JIMMA INSTITUTE OF TECHNOLOGY
FACULTY OF COMPUTING
DEPARTMENT OF INFORMATION TECHNOLOGY



DESIGNING AND DEVELOPING STEMMER FOR GE'EZ
LANGUAGE TEXT: A HYBRID APPROACH

BY

SOLOMON NIGATU

This research entitled as “Design and development of Stemmer for Geez Language text: a hybrid approach” has been read and approved as meeting the preliminary research requirement of faculty of computing in partial fulfillment for the award of the Degree of Masters of Science in Information Technology.

Jimma University, Jimma, Ethiopia

Position	<u>Name</u>	<u>Signature</u>	<u>Date</u>
Advisor	Solomon Teferra (PhD)	 _____	<u>16/11/21</u>
External Examiner	Teklu Urgessa (PhD)	 _____	<u>16/12/21</u>
Internal Examiner	Hailu Beshada (MSc)	_____	_____
Chairperson	Yonas Gido (MSc.)	_____	_____

DECLARATION

I, the signatories, state that this thesis is my original work and has not been presented for a degree in any other university, and that all sources of materials for the thesis have been appropriately acknowledged.

Solomon Nigatu

November, 2021

This thesis has been submitted for examination with my approval as university advisor.



Solomon Teferra(PhD)

November, 2021

DEDICATION

I dedicate this thesis to the almighty God given gift that I have, my family, I love you all. My dedication also goes to the memory of my father; Mergeta Nigatu and my two aunts who passed away during this study.

You will always remain in our hearts.

ACKNOWLEDGEMNT

First and foremost, I would like to thank the Almighty GOD and his mother St. Marry for giving me the strength and capability to complete my thesis work and help me in my day to day life.

My gratefulness goes to my advisors Solomon Teferra (PhD), for their commitment and patience, reading for each and every unit of the thesis, their valued comments, encouragement and guidance from the first to the last level of the research that empowered me to finish the thesis work.

It is an honor for me to give my sincere gratitude to Tesfaye Nigatu (D/N) and Haiyle Eysus (Memhir), who supports me as language expert and consults me till the end. This research is not possible without their expert knowledge on the language. Additionally I would also like to extend my gratefulness towards Mrs. Abebe Belay, who gave me 14.1 %(1866 Geez words) of my dataset.

I would also like to extend my appreciation towards my beloved families, for always being there for me through all phases of my thesis work for their love, encouragement and giving me their invaluable support without you, I would never be where I am today.

Finally, I extend my heartfelt thanks and respect to my lovely friends and all those people who were not mentioned here but their contributions have been inspiring for the completion of this work.

TABLE OF CONTENTS

ACKNOWLEDGEMNT.....	I
TABLE OF CONTENTS.....	II
LIST OF FIGURE.....	VI
LIST OF TABLES.....	VII
LIST OF ALGORITHM.....	IX
LIST OF ABRIVATIONS.....	X
SYBOLS USED.....	X
ABSTRACT.....	XI
CHAPTER ONE.....	1
BACKGROUND OF THE STUDY.....	1
1.1. INTRODUCTION.....	1
1.2. STATEMENT OF THE PROBLEMS.....	4
1.3. OBJECTIVE OF THE STUDY.....	6
1.3.1. GENERAL OBJECTIVE.....	6
1.3.2. SPECIFIC OBJECTIVE.....	6
1.4. RESEARCH METHODOLOGY.....	7
1.4.1. LITERATURE REVIEW.....	7
1.4.2. DATA SOURCES.....	7
1.4.3. THE PROPOSED APPROACH.....	7
1.4.4. IMPLEMENTATION TOOLS.....	8
1.4.5. EVALUATION TECHNIQUES.....	8
1.5. SCOPE OF THE STUDY.....	9
1.6. APPLICATION OF THE PROPOSED STEMMER.....	9
1.7. ORGANIZATION OF THE THESIS.....	10
CHAPTER TWO.....	11
LITERATURE REVIEW AND RELATED WORKS.....	11

2.1.	INTRODUCTION	11
2.2.	CATEGORIZATION OF STEMMING TECHNIQUES	11
2.2.1.	RULE BASED TECHNIQUES	12
2.2.2.	STATISTICAL BASED STEMMING TECHNIQUES	15
2.2.3.	HYBRID STEMMING TECHNIQUES	19
2.2.4.	SUMMARY OF STEMMING TECHNIQUES.....	20
2.3.	EVALUATION TECHNIQUES FOR STEMMING ALGORITHM.....	21
2.3.1.	DIRECT EVALUATION METHODS	21
2.3.2.	INDIRECT EVALUATION METHODS.....	22
2.4.	RELATED WORKS.....	23
2.4.1.	INTRODUCTION	23
2.4.2.	STEMMERS ON FOREIGN LANGUAGES.....	23
2.4.3.	STEMMERS ON ETHIOPIAN LANGUAGES.....	27
CHAPTER THREE		32
GEEZ MORPHOLOGY		32
3.1.	INTRODUCTION	32
3.2.	OVERVIEW OF GEEZ LANGUAGE.....	32
3.2.1.	WRITING SYSTEM OF GEEZ LANGUAGE	33
3.2.2.	NUMERALS IN GEEZ LANGUAGE	34
3.2.3.	GEEZ LANGUAGE PUNCTUATION MARKS.....	36
3.3.	MORPHOLOGY OF GEEZ LANGUAGE.....	36
3.4.	WORD FORMATION OF GEEZ LANGUAGE	36
3.4.1.	INFLECTIONAL AFFIXES OF GEEZ LANGUAGE	37
3.4.1.1.	NOUN INFLECTIONAL AFFIXES	37
3.4.1.2.	PRONOUN/መሪክያን/ IN GEEZ LANGUAGE	40
3.4.1.3.	GENDER MARKERS	41
3.4.1.4.	NUMBER MARKERS	42

3.4.2.	INFLECTIONAL AFFIXES OF GEEZ VERBS	43
3.4.2.1.	INFLECTIONS OF PERFECTIVE VERBS	44
3.4.2.2.	INFLECTIONS OF IMPERFECTIVE VERBS	45
3.4.2.3.	INFLECTIONS OF JUSSIVE/SUBJECTIVE VERBS.....	46
3.4.2.4.	INFLECTIONS OF GERUNIVE VERBS.....	46
3.4.2.5.	INFLECTIONS OF INFINITIVE VERBS.....	47
3.4.3.	ADVERB IN GE'EZ (ተውሳክ ግስ).....	47
3.4.4.	ADJECTIVES INFLECTION IN GE'EZ.....	48
3.4.5.	DERIVATIONAL AFFIXES OF GEEZ WORDS	49
3.4.5.1.	DERIVATION OF NOUN FROM VERBS	49
3.4.5.2.	DERIVATION OF VERB FROM OTHER VERBS.....	49
3.4.5.3.	ADJECTIVE DERIVATION	50
3.4.6.	PLURAL OF PLURAL WORDS OF GEEZ.....	51
3.4.7.	COMPOUNDING WORDS OF GEEZ.....	52
3.4.8.	NEGATION OF GEEZ WORDS	52
3.4.9.	PREPOSITIONS AND CONJUNCTIONS (መስተጻምር ወመስተዋድድ).....	52
3.5.	SUMMARY.....	52
CHAPTER FOUR.....		54
DEVELOPMENT OF HYBRID STEMMER FOR GEEZ LANGUAGE		54
4.1.	INTRODUCTION	54
4.2.	CORPUS PREPARATION.....	54
4.3.	WORD DISTRIBUTION OF THE CORPUS	55
4.4.	COMPONENTS OF THE STEMMER/ PROPOSED STEMMER.....	56
4.4.1.	INTRODUCTION	56
4.4.2.	TOKENIZATION OF GEEZ TEXT	56
4.4.3.	NORMALIZATION OF THE CORPUS.....	57
4.4.4.	COMPILATION OF STOP WORD LIST.....	58

4.4.5.	COMPILATION OF GEEZ AFFIXES.....	61
4.4.5.1.	PROPOSED RULE SET FOR REMOVING AFFEXES	64
4.4.5.2.	COMPILATION OF PREFIXES.....	66
4.4.5.3.	COMPILATION OF SUFFIXES.....	67
4.4.5.4.	HYBRID STEMMER ALGORITHM.....	70
4.5.	SUMMARY.....	72
CHAPTER FIVE		73
EXPERIMEENTAL RESULT AND DISCUSSION		73
5.1.	INTRODUCTION	73
5.2.	TOOLS AND DEVELOPMENT ENVIRONMENT	73
5.3.	USER INTERFACE PROTOTYPE	74
5.4.	IMPLEMENTATION OF THE STEMMER.....	75
5.5.	PROPOSED STEMMER EVALUATION AND RESULTS	79
5.5.1.	EVALUATION OF THE PROPOSED STEMMER.....	79
5.5.2.	RESULTS AND DISCUSSION	84
5.6.	SUMMARY.....	85
CHAPTER SIX.....		87
CONCLUSSION AND RECOMMENDATION.....		87
6.1.	CONCLUSION.....	87
6.2.	CONTRIBUTION.....	89
6.3.	RECOMMENDATION	89
REFERENCE.....		91
APPENDIXES		96
APPENDIX I: SAMPLE GEEZ STOP WORD LIST		96
APPENDIX II: SAMPLE GEEZ TEST SET		98
APPENDIX III: SAMPLE GEEZ PREPOSITION AND CONJUNCTIONS.....		102

LIST OF FIGURE

Figure 1-Natural Language Processing [9]	2
Figure 2-Categorization of stemming techniques [14]	12
Figure 3 Stemming procedures for Dictionary/table look up technique [4]	13
Figure 4 Affix Stripping Procedure	15
Figure 5 process Successor variety Technique [22]	16
Figure 6 Tokenization process	57
Figure 7 Normalization process	58
Figure 8 Stop word removal process	60
Figure 9 Geez Hybrid Stemmer Model (GHSM)	63
Figure 10 User interface prototype of Geez Stemmer	74
Figure 11 screen shot of the output of first version stemmer	80
Figure 12 Screen shoot of the hybrid version stemmer	82

LIST OF TABLES

Table 1 Successor variety example.....	17
Table 2 N-gram example	19
Table 3 Summary of stemming Techniques	20
Table 4 Sample Geez numbers with corresponding Indo-Arabic numbers	35
Table 5 External plural formation of nouns.....	39
Table 6 internal plural formation of nouns	39
Table 7 Geez Pronouns	40
Table 8 Objective pronouns of Geez	41
Table 9 Subjective Pronoun.....	41
Table 10 the three gender markers in Geez	42
Table 11 Number markers in Geez	43
Table 12 Inflections of Perfective verbs	44
Table 13 Inflections of imperfective verbs	45
Table 14 Inflectional formation of subjective verbs.....	46
Table 15 Inflection of gerundive verbs.....	46
Table 16 Sample Geez infinitives.....	47
Table 17 inflection of adjectives.....	48
Table 18 Derivation of nouns form verbs.....	49
Table 19 derivations of geez verbs	50
Table 20 Derived adjectives from nouns and verbs.....	50
Table 21 plural of plural nouns.....	51
Table 22 Percentage prepared corpus from selected sources.....	55
Table 23 Word distribution of Geez sample dataset.....	56
Table 24 Top ten frequent words from the prepared corpus.....	59
Table 25 Sample of Geez stop word list.....	59
Table 26 a samples Geez compiled Prefixes.....	67
Table 27 samples Geez compiled suffixes.....	68
Table 28 Sample of Geez prefixes removal.....	76
Table 29 Sample suffixes removal.....	77
Table 30 Sample Prefixes-Suffixes pair removal	78

Table 31 Sample infixes removal process	78
Table 32 Sample output by the first version of Geez stemmer.....	81
Table 33 Sample output by the hybrid version of Geez stemmer.....	83
Table 34 Evaluation results of First version stemmer	84
Table 35 Evaluation results of hybrid version stemmer	85

LIST OF ALGORITHM

Algorithm 1 Algorithm for Tokenization of Geez text.....	57
Algorithm 2 Geez character normalization.....	58
Algorithm 3 Removing stop words.....	60
Algorithm 4 Geez Affixes stripping algorithm.....	70
Algorithm 5 Geez Hybrid Stemming algorithm	72

LIST OF ABRIVATIONS

ASS: Automatic summarization System
ENA: Ethiopian News Agency
EOTC: Ethiopian Orthodox Thewahido Church
IR: Information Retrieval
MMI: Modified Minimal Mutual Information that used for clustering
MTS: Machine translation System
NLP: Natural language Processing
OOP: object oriented programming
QAS: Questions answering system
Masc.: masculine
Fem.: Feminine
1st p.s.: First person singular
1st p.p: First person plural
2nd p.s.f: Second person singular feminine
2nd p.s.m: Second person singular masculine
2nd p.p.f: Second person plural feminine
2nd p.p.m: Second person singular masculine
3rd p.s.f: Third person singular feminine
3rd p.s.m: Third person singular masculine
3rd p.p.f: Third person plural feminine
3rd p.p.m: Third person plural masculine

SYBOLS USED

[] What is inside is a reference

// what is inside is a translated Geez word or meaning or explanation

ABSTRACT

Stemming is widely used in information retrieval tasks. Many researchers demonstrate that stemming improves the performance of information retrieval systems. Stemming is a technique for reducing inflection and derivation of morphological variations of words to their stem or root form. It's useful for improving retrieval efficiency, particularly for text searches, and for resolving mismatch issues.

The aim of this study was designing and developing a hybrid stemmer for Ge'ez language text. We have used two approaches namely affixes removal and character n-gram technique. The proposed methods can remove prefixes, infixes, suffixes and its combinations. To remove all affixes, rules are compiled individually for each affixes and exceptional and recording rules are also integrated based on the nature of Geez language morphology. Corpus is manually prepared from ready available sources such as text books, magazine and bible. The size of the prepared corpus has 13,221 word tokens. From the prepared corpus, 20% was used for testing the proposed stemmer.

To evaluate the proposed stemmer manual error counting mechanism was used. The proposed stemmers are evaluated in two stages; first the affixes removal version is evaluated on a testing dataset with 2644 word length and secondly the hybrid version is evaluated on the same testing dataset. According to the evaluation results, affixes removal version registered an accuracy of 92.32% with 7.68% error rates and the hybrid version stemmer also recorded an accuracy of 94.5% with 5.5% error rates. The hybrid version stemmer increased by 2.18% accuracy. Over stemming and under stemming errors are observed on either of the affixes removal and hybrid version stemmer. As a result, 4.5% and 2.2% over stemming and 3.18% and 3.3% under stemming errors are shown respectively on the proposed stemmer. Generally our proposed hybrid stemmer out performed better by 12.26% and 8.28% accuracy with reducing 12.08% and 7.28% error rates than the previous rule based and longest match stemmers respectively. This is due to incorporating exceptional and recording rules based on the detailed study of the language. Finally we found that, our proposed hybrid stemmer was encouraging and using this tool as a pre-processing module for further research may be helpful.

Keywords: Geez Stemmer, Information Retrieval, N-Gram, Hybrid Stemmer, Natural Language Processing, Conflation.

CHAPTER ONE

BACKGROUND OF THE STUDY

1.1. INTRODUCTION

With the huge amount of digital data available in multiple languages, it has become important to develop various language-processing tools that could efficiently manage the large document bases. In many Natural Language Processing (NLP) and Information Retrieval (IR) applications, construction of vocabulary of words and language models is an important task [1][2]. However, a large number of morphological variations in the words, especially in morphologically rich languages, pose a great challenge [3].

Natural language processing is a branch of artificial intelligence that deals with analyzing, understanding and generating the languages that humans use naturally in order to interface with computers in both written and spoken contexts using natural human languages instead of computer languages[4][5]. Natural language processing (NLP) is a field of computer science, artificial intelligence (also called machine learning), and linguistics concerned with the interactions between computers and human through natural languages [6][7]. Mainly focus on the process of a computer extracting meaningful information from natural language input and/or producing natural language output and natural language understanding that require extensive knowledge of the outside world and the ability to manipulate it.

Natural Language Processing (NLP) is an area of research and application that explores how computers can use to understand and manipulate natural language text. NLP researchers aim to collect knowledge on how human beings understand and use language so that fitting tools and techniques can be developed to make computer systems understand and manipulate natural languages to perform the preferred tasks [7].

NLP usually involves one or more level of linguistic analysis such as word level, phrase level, sentence level, semantic level analysis, etc. There are processes made when humans produce or comprehend language. It thought that humans normally utilize all of these levels since each level conveys different types of information. Nevertheless, various NLP systems utilize different

levels, or combinations of levels of linguistic analysis, and it served as a difference amongst various NLP applications. Such tasks include Part of Speech (POS) Tagging, Named Entity Recognition (NER), Information Retrieval (IR), Speech Recognition, Machine Translation, Question Answering and etc. [8].

Stemming comes under Natural Language Processing techniques. It is branch of Artificial Intelligence [6]. On the other hand, stemming is a linguistic process in which the various morphological variants of the words are mapped to their base forms.

It is a useful pre-processing technique to handle these variations and it is among the basic text pre-processing approaches used in Language Modeling, Natural Language Processing, and Information Retrieval applications. For examples the word “played, playing, player, and players” will be mapped to their base form “play” with the help of stemming. Stemming is a simple language processing that found to be quite effective in a number of applications. It is the process of mapping inflectional and derivational words to their respective stems. It basic concept of stemming is to reduce different grammatical forms or word forms to its root, stem or base form and it is significant in spell checking, machine translation, natural language processing and information retrieval, parts of speech tagging [9][10].

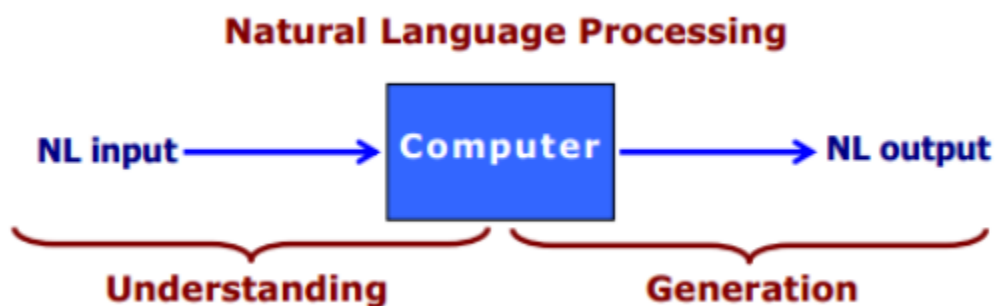


Figure 1-Natural Language Processing [9]

On the other hand, information retrieval aimed to extract all relevant documents for a user query by using index items of natural language text [11]. The text could be unstructured and ambiguous. In order to satisfy users in their searching, it is required to translate user request in to inquiry that can processed by the information retrieval system. Among others, word stemming is an important attribute supported by recent indexing that produces a set of key words relevant to the document[12]. Stemming enables to improve recall by automatic handling of word endings

via reduction of terms to their stems or roots during indexing and searching. Hence, stemming reduces the size of indexing structure and minimizes variants of the same stem or root words in order to have effective searching result [13].

On information retrieval system applications, the construction of vocabulary of word and language models is an important task, but a large number of morphological variations in the words, particularly on morphologically rich languages, pose an unlimited challenge. There are three primary functions for IR: firstly for indexing; process of creating useful index for documents, secondly for search request; has to create query that should retrieve information that is relevant for the user, and lastly for request document matching; deals with comparing the created index with formulated request from the user [14].

The challenge is to meet the need of the user to retrieve data from unstructured data. The representation and organization of information should be in such a way that the user can access information to meet his information need, so text stemming plays a vital role to achieve those needs [5].

Now a days various stemming approaches are available, that can be classified as a language specific based called a rule based approach and a statistical based approach which works based on statistical probability and a hybrid approach by integrating either of rule or statistical based approaches [4] [15]. Lots of works had been done for language like English, Indic, Arabic and etc. [4], [6], [16]. even if for Ethiopian language like Amharic, Tigrigna, Afan Oromo and etc. has been done recently, it needs further investigation[11], [12], [17].

Particularly Ge'ez language is an ancient South Semitic language that originated in Eritrea and the northern region of Ethiopia in the Horn of Africa. It later became the official language of the Kingdom of Aksum and Ethiopian imperial court. Today, Ge'ez remains only as the main language used in the liturgy of the Ethiopian Orthodox Tewahedo Church, the Eritrean Orthodox Tewahedo Church, the Ethiopian Catholic Church, and the Beta Israel Jewish community [18], [19]. Over the past millennia of Ethiopia, the country's literature were mostly used the Geez language and has appropriately recorded on this phenomenon [20].

As far as the researchers knowledge on NLP and IR is concerned, researches made in the area of Geez language are very limited in number; it needs a further investigation as other language like English and other foreign language [1]. In fact, the rule based stemmer was developed for Geez language text, by Abebe [19]. As the author recommended that, in order to increase the accuracy and reduce the error rates of the developed stemmer, it is better to advance the rule and using combination of the rule based approach with statistical approach may intensify the performance of the stemmer. Geez morphology is highly inflected language[21]–[23]; i.e. due to the complexity of morphology of it, trying different approaches may be preferable in order to develop a good stemmer that can conflate or strip all word variants of the language.

One of the main problems involved in information retrieval is variations in word forms. The most common types of variations are spelling errors, alternative spellings, multi-word constructions, transliteration, affixes, and abbreviations. One way to avoid such problems is that using stemming algorithm. Information retrieval systems use stemming to improve the matching algorithms. This study is aiming to design and develop an automatic stemming for Geez language text by using hybrid-stemming techniques.

1.2. STATEMENT OF THE PROBLEMS

One of the first problems related to the use of natural language in information retrieval and Natural Language Processing applications is that of morphological variation of words. Morphological variation of words which refers to the fact that words may occur in inflected forms, or that derivation is used to produce new but related words, or words are combined into compound words [5]. In most cases, morphological variants of words have similar semantic interpretations and can considered as the same for the purpose of IR and NLP applications[13][24]. In addition for IR application; retrieval systems it is difficult for catching information easily and timely from a large body of sources, these reduce the retrieval efficiency and effectiveness [19], [25]. As a result text stemming is the basic building block for retrieval efficiency and effectiveness.

On the other hand, Geez language has become instructional language for Ethiopian Orthodox Tewahedo Church Theological colleges. Accordingly, textbooks, references books of the religion and other historical books of the country compiled by using this language [23]. At this time everything is done with the help of computer, in order to write Geez documents, text editors can

use the stemmer for correcting spelling errors. As result, significant numbers of peoples are able to read and write the language with the presence computers. Offices and educational institutions are now using for teaching and learning purpose. From these, journals, newspapers and books printed in Geez are available on the web. Such opportunities open the bright future to produce more electronic documents in Geez language[19], [11]. Therefore, Information retrieval system that process Geez documents can also use stemmer for indexing.

For different application such as information retrieval, indexing and query formulation on this language, for further natural language processing like part of speech tagger, word sense disambiguation and etc. needs the stemming techniques as a tool. If a well performed stemmer is developed for Geez language; simply the researchers can be easily used a helper tool for further investigation on these area. For example, in order to cope up higher level linguistic analysis for NLP such as parsing, parts of speech tagging, machine translation and etc.; it is difficult to come up better result without using such type of tools.

Hence, in the case of Geez language, finding an effective stemming algorithm seems to be quite difficult, for the time being Geez language has its own specific morphological structure, which is different from other local languages[11], [12], [25], [26],[25], [27]. The main problem found for this study is that, even if there were a lots of research have been conducted for different natural language such as English, Indian and Arabic; due to the difference of morphological nature of the language, it cannot be handled or applied to our local language particularly Geez [4], [6], [16].

Even though there were different studies of stemming natural language text recently for Ethiopian language like Amharic, Tigrigna, Afan Oromo and etc. [11], [12], [17]; it cannot applied on Geez language text. Particularly a study conducted by Abebe [19] was tried to proposed a rule based approach that could stem Geez texts; but this study was faced by many challenges. Firstly the stemmer could not stem all affixes of Geez text i.e. due to limited list of prefixes and suffixes are considered; most of the time all word forms was not conflated correctly. As an example the words ‘ወታደራዊ’ and ‘ኢተወሰኑ’ have a prefixes ‘ወታ’ and ‘ኢተ’ and the word ‘አዲሲያውያን’ has a suffix ‘ሲያውያን’ respectively; all of these words were not conflated by the rule based stemmer developed by[19]. In addition to that the stemmer couldn’t consider infix removal process.

Secondly it cannot remove the possible stop words that are non-content bearing words for Geez text; for example the words በዘ, እንቢይኒ, እስኩ, የጊ and እምዶእዜሰ etc. are a stop word but the stemmer couldn't identified it as a stop word. Thirdly, the designed rules weren't considered exceptional rule that could be applied on some exceptional cases.

On the other hand, an enhanced version of Geez stemmer was conducted by [28] in which longest match approach were used to this end. The proposed system was tested on a data set of 2000 word-lists. According to the evaluation performed on the prototype, the accuracy registered was found 87.22% with the total error rate of 12.78%. Even if the result found was encouraging and shows some enhancement, different tries are needed in order to come up with a good stemming application for the language by reducing the errors found and increasing the accuracy. As a result, it is mandatory to enhance the previously proposed rule based and longest match stemmer by reducing or fulfilling the above identified gaps. Therefore, this study is initiated or motivated to design and develop the potential application of hybrid algorithm (rule based and N-grams) for stemming (conflating) words in Geez language text.

1.3. OBJECTIVE OF THE STUDY

1.3.1. GENERAL OBJECTIVE

The general objective of this thesis work is to design and develop hybrid stemmer for geez language text.

1.3.2. SPECIFIC OBJECTIVE

To achieve the general objective, the following specific objectives will performed

- Review different literature that has been done on the area of stemming approach and adopt the best one that is appropriate for Geez language.
- Study the morphology or word formations of the language.
- Prepare a corpus for Geez language text and develop a rule for Geez language text.
- To integrate the rules based stemmer with the statistical approach.
- Evaluate the performance the proposed stemmer based on the experiment.
- To come up with the conclusion based on the result of the experiment and provide recommendation for future enhancement.

1.4. RESEARCH METHODOLOGY

Research methodology is a way of solving problems systematically. In this study, experimental quantitative research method was selected. The reason for selecting this methodology is that, experimental approaches involve identification of the potential methods of stemming and implementing and testing are made iteratively. Generally this study followed an experimental quantitative method, in the sense of building algorithms and testing them until the needed level of performance is achieved. In order to achieve the objectives of this research, the following methods and techniques were employed.

1.4.1. LITERATURE REVIEW

Extensive literature review done to get more insight into the concept of information retrieval in general and different stemming in particular. Various works of literature and related works that have been done in the area of stemming are reviewed and discussed to understand the state-of-art. Additionally reviews of literatures are conducted to get familiarity to the basic Geez language text features in relation to information retrieval.

1.4.2. DATA SOURCES

To achieve the proposed objectives, first we have prepared a corpus of Geez language text from various documents in order to get the variety of the word forms that can incorporate the effectiveness and efficiency of the developed stemmer. A good-sized text can show a reasonable language morphological behavior. Selection of text is, therefore, an important component in developing a stemmer.

For the purpose of this study, the texts are collected from historical books in which written in Geez, various liturgical books and other sources were used. In addition to that, two professionals educated in Geez language also consulted for preparing standardized morphologically distributed corpus of the Language.

1.4.3. THE PROPOSED APPROACH

The proposed approach for this study is a hybrid approach stemmer for Geez text. This approach integrates two individual stemmers to work together for getting advantages from it. The first component is rule based stemmer approach and the second component is that statistical approach.

The rule based approach can handle transform the variant word forms of a language into their stems or base forms by using certain pre-defined language-specific rules. It incorporates manually handcrafted rule sets that can remove affixes of the language. In order to remove the affixes, exceptional and recording rules are used for further enhancement of the rule based approach. The reason and the major advantages of using rule-based components are, due to ease of use; that means the language-specific rules, once created were applied to any corpus without any additional processing.

In addition to rule based, statistical approach mainly used unsupervised or semi-supervised training to learn stemming rules from a corpus of a given language. The major advantage of statistical approach is that it can be applied to a under resourced language with very little effort provided. that can satisfies the basic assumptions of the stemmer (like variant words should be formed by adding affixes only) and it is good substitutes to language-specific stemmers, especially for languages where linguistic resources are incomplete [1], [29]–[33].

Generally, this study is going to design rule sets and develop stemmer a prototype for Geez language text using hybrid approach or technique by developing suitable algorithm to the language and applying character n-gram techniques. To come up with a good stemmer we designed the possible hand crafted rule sets and develop an algorithm those are applied on the proposed hybrid approach. Finally the developed prototype was used to testing and evaluating the performance of the designed algorithms.

1.4.4. IMPLEMENTATION TOOLS

For implementation purpose, we have to use the java programming language and the IntelliJ Idea Community edition 2021.3 to write the code. The reason for selecting this language is that, java-programming language has a facility to deal with natural language text processing.

1.4.5. EVALUATION TECHNIQUES

For resourced language like English, there is standard or baseline for evaluating a new algorithm or technique. Based on these baselines we can evaluate the performance, whether the developed algorithm is well or not. For the purpose of this study, to evaluate the performance of the developed stemmer, we have used error-counting mechanism. The reason for selecting this mechanism is that, there is not available standard evaluation metrics prepared yet for under

resourced language but not only the Geez for other local language like Amharic. In addition, the result will evaluate in quantitative measures such as percentage of correctly stemmed words and the error rate counting were employed to evaluate the accuracy of the newly proposed hybrid stemmer.

1.5. SCOPE OF THE STUDY

In order to develop stemmer applications in different languages there were different alternatives like lookup table, affixes removals/rule based, statistical and hybrid-stemming. The scope of the proposed research is covered only applying a hybrid approach to develop the stemmer for Ge'ez language. There are also different types of statistical and rule based approaches of stemming techniques, from those techniques, we have selected affixes removal technique and adopt the statistical approach particularly n-gram technique for the purpose of this study.

The reason for selecting the statistical approach is that, it is commonly used on various languages and preferable for under-resourced language[30], [32], [34]–[36]. Hybrid approach will used for enhancing the efficiency and effectiveness of the Stemming algorithm. Ge'ez words that cannot handle by affixes removal technique are covered by n-gram technique.

In addition to that, this study covers only word level analysis for Ge'ez language text. It does not cover the higher level linguistic analysis such as phrase level, sentence level, semantic level, etc. the reason focusing only on word level analysis is that, the main aim of stemming applications are just reducing morphologically variant word forms and mapping to their stem or base forms[24], [36].

1.6. APPLICATION OF THE PROPOSED STEMMER

In an IR system with queries and index stemmed, the user needed no special knowledge of the form of the subject terms to expand the query. Query expansion with stemming results in a much cleaner vocabulary list than without, and this is a main strength of using a stemming process. Text Stemming is widely used as a part of the text pre-processing step in Information Retrieval and Natural Language Processing systems. Stemming employed on text pre-processing stage to solve the problem of vocabulary mismatch and reduction in the dimensionality of representation set or training data.

After developing the stemmer for Geez, further studies related to Geez text processing can be used as input and it may have various applications for Natural language processing and Information retrieval systems and used as an input for the following:

- Part-of-Speech Tagging Systems(POS)
- Document Classification and Clustering(DCC)
- Machine Translation Systems(MTS)
- Automatic Summarization Systems and Question answering systems (ASS and QAS).
- Text searching, spell checker, speech recognition, word sense disambiguation.

1.7. ORGANIZATION OF THE THESIS

The research work consists of six chapters. This chapter introduces the importance of stemming on IR environment and the need to develop stemming algorithm for conflating variants of a word in Geez language. Statement of the problem and the methodology employed and scope of the study were presented.

The next chapter analyses the works on conflation techniques in general and stemming algorithms in particular. Detailed discussions made on approaches to stemming and types of stemmers. Review also made in this chapter on some stemming algorithms developed for other foreign and local languages.

Geez language morphologies were reviewed in chapter three. The inflectional and derivational morphologies of the language are the main concerns of this chapter. Word formation processes for Geez nouns, adjectives, and verbs will presented in detail in the chapter.

Fourth chapter deals with discussions on the development of the proposed stemming algorithm for Geez text. The compilation of stop word lists and affix (prefix-suffix pair, prefix, and suffix) lists presented in this chapter. The approach employed to develop the stemmer and the reasons for its selection also parts of the discussions.

Chapter five focused on the implementation and the experimental results this study. Lastly chapter six present conclusions deduced from the findings and recommendations for future research.

CHAPTER TWO

LITERATURE REVIEW AND RELATED WORKS

2.1. INTRODUCTION

Firstly the problems related to the use of natural language in information retrieval and Natural Language Processing applications is that of morphological variation of words. Morphological variation of words refers to the fact that words may occur in inflected forms, or that derivation is used to produce new but related words, or words are combined into compound words. In most cases, morphological variants of words have similar semantic interpretations and can be considered as the same for the purpose of IR and NLP applications [6]. In Natural language, stemming is a technique that is used to conflate or reduce morphological variants of words to a single term (stem/root), by stripping the root of its derivational and inflectional affixes [3].

On this study, to achieve the main objective we have to review various Geez language documents and Stemming techniques that are helpful for conflation of the word. There are a number of stemming algorithms developed for different languages like English language and various semantic language like Amharic[11], [37].

2.2. CATEGORIZATION OF STEMMING TECHNIQUES

The stemming process has a rich literature, and a number of stemmers of varying flavours were developed over the last decades. Stemming methods may range from simple approaches like the removal of plurals and present and past participles to complex approaches that remove a variety of suffixes and include a lexicon [38]. According to [14] the current stemming algorithms belong to one of three categories which were; Rule Based, Statistical, or Hybrid. Each of these categories finds the stems of the variant words in their own typical way.

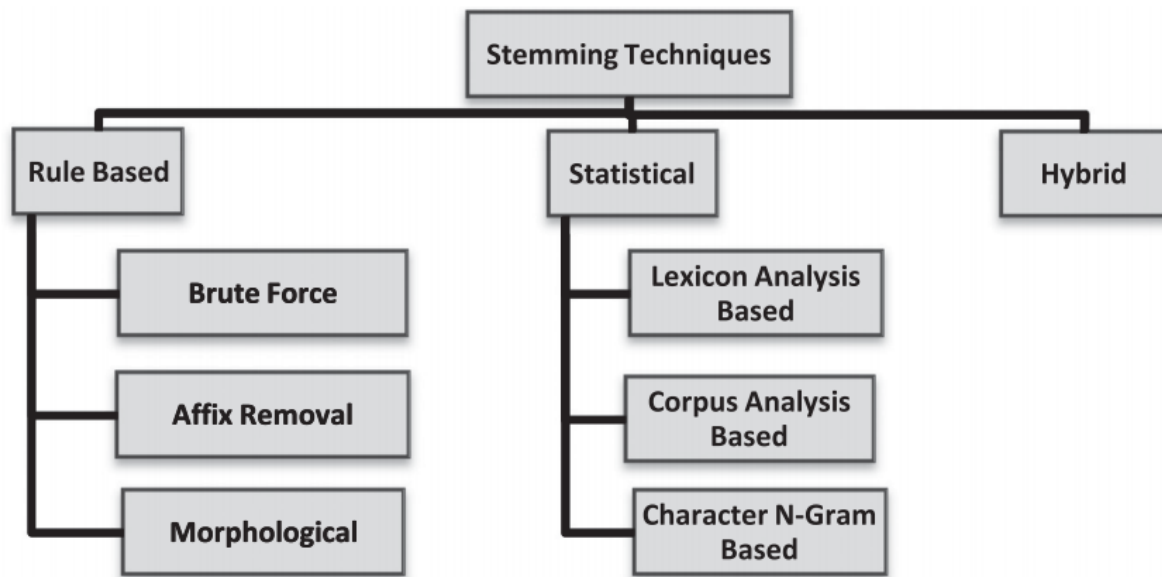


Figure 2-Categorization of stemming techniques [14]

According to [39] and [1] stemming techniques classified as truncated, statistical and mixed. On the other hand different researcher classified it on different ways [39], [12], [7]. On the following sub section we tried to discuss the common classification.

2.2.1. RULE BASED TECHNIQUES

Rule-based stemmers transform the variant word forms into their stems or base forms by using certain pre-defined language-specific rules [19], [25], [27]. The creation of language-specific rules requires expertise in language or at least a native speaker of the particular language. Moreover, rule-based stemmers sometimes employ additional linguistic resources like dictionaries to conflate morphologically related words. The major advantage of rule-based stemmers is due to their ease of use and the language-specific rules created once and applied to any corpus without any additional processing[1], [4].

However, for languages where the resources are poor, these stemmers are not preferred. These stemmers tend to be better in the way of applying complex morphological rules of the language than statistical stemmers [40]. According to [2], [41], [42] there are various rule-based stemmers in which applied on different language from these stemmer techniques, we have discussed some of it as the following categories:

2.2.1.1. BRUTE-FORCE/DICTIONARY TECHNIQUES

In this method collection of word and their conforming stems can be warehoused in a dictionary or table. The stemming process is done by looking up the dictionary or the table. For the purpose of speed of looking up of table, it uses the Hash table or B-tree. In other way Brute-force stemmers make use of a lookup table to return the stem of the word [1], [12]. This lookup table maintains relations between the variant words and their root forms. The table checked to find the matching inflection and the associated stem will be returned.

These stemming techniques also called table lookup or dictionary-based techniques. One notable advantage of these stemmers is that they can handle the inflected word forms of a language that do not obey the language-specific rules appropriately [39]. For instance, suffix removal algorithms can stem the word “eating” to “eat” but it cannot stem the alternate inflection “ate”. The following figure (figure 3) shows the general procedure of dictionary look up procedure [4].

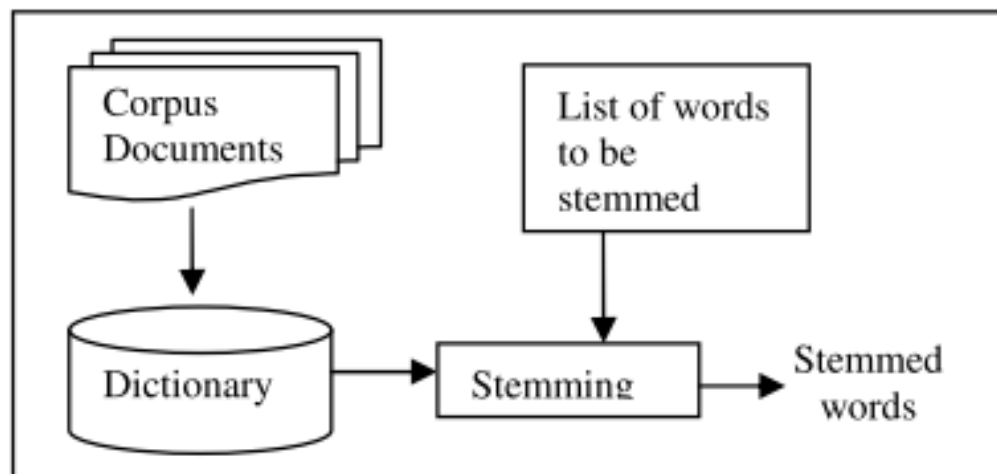


Figure 3 Stemming procedures for Dictionary/table look up technique [4]

The major limitation of these algorithms is that, all the variant words cannot manually collected and/or recorded in a lookup table. Therefore, it cannot stem the words that are not present in the table. Moreover, it consumes a lot of space to store the list of relations [1].

2.2.1.2. AFFIX REMOVAL TECHNIQUE

Affix refers to prefixes, infixes, suffixes or combination prefix-suffixes of words. Therefore, as the name suggests, these techniques remove the suffix and/or prefix from the variant word forms

[7], [43]. The stemmers in this category make use of a suffix/prefix list along with certain context-sensitive rules to obtain the stem. Most of the works were done on suffix removal as compared to prefixes. The affix removals based on rules are either done based on longest match basis or in iterative manner. For example in English, the following inflectional words were stripped the suffixes into the stem “connect”.

Connection		-ion	
Connections		-ions	
Connective	====>	connect + -ive	====> connect (stem)
Connected		-ed	
Connecting		-ing	

According to [44] the necessity of are; firstly an affix stripping algorithm does not require a dictionary. Secondly, the algorithm is very fast. Thirdly, since it does not require any supporting data the algorithm can be run on any device and lastly, there is lack of quality corpus to train statistical algorithms it cover come such problem.

On the other hand, major weakness of these stemmers is that the stems produced after removal of suffixes are not real words of the language [1]. These truncated word forms are poor for human interfaces and present difficulties in certain applications. The problem depends on how the transformation being used [1]. For example, if we use the stems to create clusters of words, then the failure to identify a word is not necessarily harmful but in applications like word-sense disambiguation, these stems cannot used, as it is not possible to resolve the meaning of the word without knowing the word we are dealing with. Moreover, affix removal algorithms sometimes produce aggressive confluations [44]. For example, the words “general,” “generic,” “generous” stemmed to the same root “gener” by the suffix stripping process. Figure 4 shows the general affix stripping procedure [4].

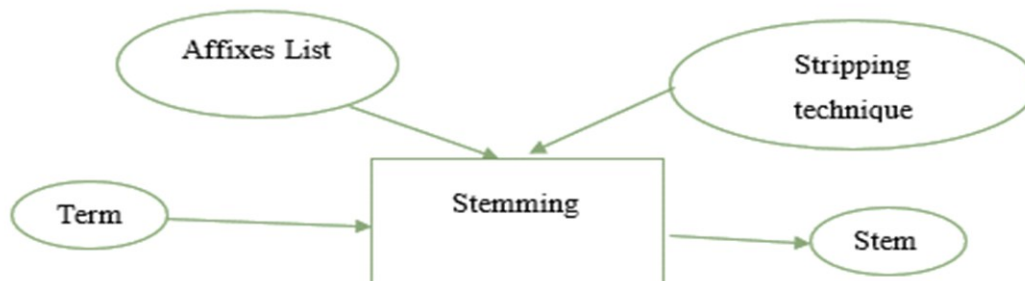


Figure 4 Affix Stripping Procedure

2.2.1.3. MORPHOLOGICAL STAMMERING TECHNIQUE

These stemmers involve inflectional and derivational morphological analysis to perform stemming. They require large language-specific lexicons containing word groups organized by syntactic and semantic variations [19], [5]. Inflectional analysis can detect changes in word forms due to gender, tense, mood, case, number, person, or voice. Whereas derivational analysis can detect changes in part of speech (POS) and can reduce surface forms to the forms from which it derives. For instance, “advancement” is stemmed to “advance” but “department” cannot stem to “depart” as both forms have different semantics.

The advantages of morphological stemmers are that, it produce morphologically correct roots and can handle various exceptional cases. These stemmers handle roots that are out-of-vocabulary by making use of rules as well as a lexicon [19]. The algorithm first finds the root in the lexicon but if the root is not found, and the suffix is productive enough and the word is transformed.

2.2.2. STATISTICAL BASED STEMMING TECHNIQUES

Statistical stemmers use unsupervised or semi-supervised training to learn stemming rules from a corpus of a given language. They group morphologically related words using the ambient corpus, thereby obviating the need for language experts or any additional linguistic resource. For that reason, these stemmers also called language independent or corpus-based stemmers [14], [38]. The major advantage of corpus-based stemmers is that these stemmers can applied to a new language with very little effort provided the language satisfies the basic assumptions of the stemmer (as variant words should be formed by adding affixes only).

Moreover, statistical stemmers can find fewer frequent cases while processing a large corpus of the language. A number of studies [45], [35], [46] have shown that statistical stemmers are good substitutes to language-specific stemmers, especially for languages where linguistic resources are incomplete. There are different techniques included in to statistical stemming approach; for the purpose of this study various techniques such as Successor variety, Lexical analyses based, Corpus analyses based, and character n-gram-based statistical stemming methods were proposed in the literature.

2.2.2.1. SUCCESSOR VARIETY TECHNIQUE

Successor variety techniques are based on the structural linguistics which determines the word and morpheme boundaries based on distribution of phonemes. Successor variety of a string is the number of characters that follow it in words in some body of text [47]. It determine word and morpheme boundaries based on the distribution of phonemes in a large body of utterances and the successor variety of a string is the number of different characters that follow it in words in some body of text [11]. The successor variety of substrings of a term will decrease as more characters are added until a segment boundary is reached. Cut off method, peak and plateau method, entropy and complete methods are the common method used for this technique for determining the cut off and the boundary. Figure 5 shows the general process of successor variety technique [25].

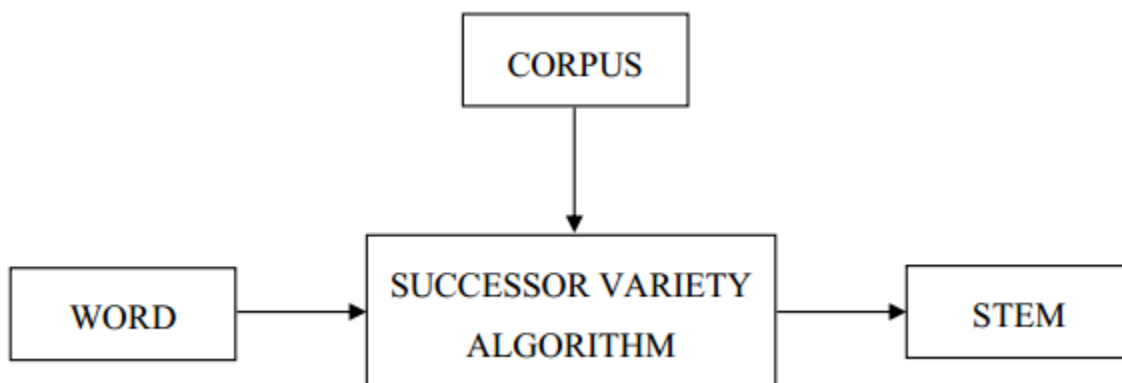


Figure 5 process Successor variety Technique [22]

The stemming process has mainly three parts in which, is that determine the successor varieties for a word, then segment the word using one of the methods stated above and finally, select one of the segments as the stem. In addition to that, it has two main criteria to evaluate various

segmentation methods. The first one is the number of correct segment cuts divided by the total number of cuts and the second one is the number of correct segment cuts divided by the total number of true boundaries.

The successor variety of the word “READABLE” is shown in table below. The successor variety of a string is the number of different characters that follows words in a text set. For example a text set containing the following words “ABLE, APE, BEATABLE, FIXABLE, READ, READABLE, READING, READS, RED, ROPE, RIPE”.

Prefix	Successor Variety	Letters
R	3	E,I,O
RE	2	A,D
REA	1	D
READ	3	A,I,S
READA	1	B
READAB	1	L
READABL	1	E
READABLE	1	(Blank)

Table 1 Successor variety example

When there are a large text set, the successor variety of sub strings of a stem will diminution as more characters are added until a segment boundary is reached. If the successor variety substring is very low, probably it will be a stem.

2.2.2.2. LEXICON ANALYSIS-BASED TECHNIQUE

These stemmers analyse a set of words obtained from the corpus to group the lexicographically related words. They find probable stems and suffixes using various methods like computing string distances, the frequency of substrings, and so on [38]. Additionally, it is also applied potential suffix information to discover suffixes from the lexicon. The algorithm is tested on six Asian languages, namely Hindi, Marathi, Gujarati, Bengali, Tamil, and Odia, and performed well in all the languages [9], [38], [47].

2.2.2.3. CORPUS ANALYSIS-BASED TECHNIQUE

These stemmers group morphologically related words by analysing their co-occurrence or context in the corpus. These are based on the fact that; words that co-occur in the corpus; are a better representative to be merged than words that do not co-occur. As compared to lexicon analysis-based stemmers, they require relatively large corpus to obtain more reliable co-occurrence information [38].

More over [38] proposed an unsupervised stemming method that uses both lexical and semantic information from the corpus. The stemmer works in two steps. In the first step, modified Minimal Mutual Information (MMI) clustering is used to group words that are lexically and semantically related (having the same semantics and share a common prefix). These clusters are used as training data for a maximum entropy classifier that encodes context-specific stemming rules into features. The authors tested the proposed method in three different types of experiments: Inflection Removal, Language Modeling, and Information Retrieval. The stemmer performed well in all three experiments and is hence used as a multi-purpose tool.

2.2.2.4. CHARACTER N-GRAM BASED TECHNIQUE

These stemmers learn the stemming rules through frequency or probability of n-grams obtained from the words of the corpus. They can handle morphological variations in alphabetic languages. As compared to other methods in this category, n-gram-based stemmers can handle not only inflectional and derivational morphology but also compounding of words or spelling exceptions [3], [33], [48], [49].

According to [47] an n-gram is a set of n consecutive characters extracted from a word. The main idea behind this approach is that, similar words will have a high proportion of n-grams in common. Typical values for n are 2 or 3, these corresponding to the use of di-grams or trigrams, respectively. For example the word “productive” and “production” have the following di-grams.

Productive => pr ro od du uc ct ti iv ve

Production => pr ro od du uc ct ti io on

From this example each word has 9 (nine) unique di-gram and they share 7 (seven) unique di-grams: pr ro od du uc ct ti. To calculate the similarity of the two words, we can use Dice's coefficient. Let say, A and B are the numbers of unique di-grams in the first and the second words respectively and C is the number of unique di-grams shared by A and B is given by:

$$S = \frac{2C}{A+B} S = \frac{(2*7)}{(9+9)} = 0.78$$

Then the similarity measures are determined for all pairs of terms in the database, forming a similarity matrix. Once such a similarity matrix is available, terms are clustered as a group using a single link clustering method or other method. From the values of Dice's coefficient, we can extract the first seven unique di-gram as a stem.

The following table (table 2) shows Bi-gram and Tri-gram of the two words "Correction" and "Corrective" that were taken from [2].

WORD	DI GRAMS	TRI GRAMS
Correction	*C,CO,OR,RR,RE,EC,CT, TI,IO,O N,N*	**C,*CO,COR,ORR,RRE,REC,ECT,CTI,TI O,ION,ON*,N**
Corrective	*C,CO,OR,RR,RE,EC,CT, TI,IV,V E,E*	**C,*CO,COR,ORR,RRE,REC,ECT,CTI,TI V,IVE,VE*,E**
A	11	12
B	11	12
C	8	8
Dice- Coeff.	0.727	0.667

Table 2 N-gram example

From the above Dice's coefficient was computed as $(2 * 8) / (11 + 11) = 0.727$ and $(2*8) / (12+12) = 0.667$ for Di-gram and Tri-grams respectively. Likewise the similarity is computed for all the word pairs and they clustered as the groups. The value of the Dice coefficient gives us the hint that, the stem for these pairs of words lies in the first 8 unique di-grams.

2.2.3. HYBRID STEMMING TECHNIQUES

Hybrid stemmers combine several approaches to perform stemming. The combination of approaches generally helps in increasing the efficiency of the stemmer [12]. Hybrid stemmers will be formed by the combination of different stemming methods such as combining various rule-based approaches and/or combining a rule-based approach with statistical methods. For

instance, the efficiency of a suffix stripping algorithm; can be increased with table lookups for unusual word forms (like run/ran) and/or singular/plural forms [50].

Similarly, the classes generated by using the rule-based stemmers can be further refined using co-occurrence or other corpus-specific information. It helps in solving the problem of aggressive conflation in rule-based stemmers. A variety of hybrid stemmers, for different languages have been developed [50], [12], [35].

2.2.4. SUMMARY OF STEMMING TECHNIQUES

Generally researchers have proposed various stemming techniques, but those techniques can be broadly categorized either of manually automatic methods [1], [47], [51]. Those stemming techniques may have advantages and limitations. To overcome the limitation of individual techniques researchers proposed the hybrid approach. The following tables (table 3) show that, the advantage and limitation of the common stemming techniques.

Stemming type	Advantage	Disadvantage/limitation
Affixes Removal	Very fast and no need of storage does not require any supporting data the algorithm can be run on any device does not require a dictionary	given the poor performance when dealing with exceptional relations need to have extensive language expertise to make them
Dictionary Based	Produce true stem/ accurate result	It is domain dependent The storage overhead Need to extensively work on a language
N-gram based	It is language independent	Requires a significant amount of memory and storage for creating and storing the n-grams and indexes.
Corpus analysis based	Avoid making conflations that are not appropriate for a given corpus. over/under stemming drawbacks are resolved	Need to develop the statistical measure for every corpus separately. the processing time increases
Morphological analysis based	Produce morphologically correct root Can handle various exceptional cases Handle roots that are out-of-vocabulary	It is language dependent

Table 3 Summary of stemming Techniques

2.3. EVALUATION TECHNIQUES FOR STEMMING ALGORITHM

The evaluation of stemmers has always been a debating affair. Different research groups have proposed a number of evaluation metrics to measure the effectiveness and/or error rates of the stemmer. According to [14], evaluation techniques can be broadly classified as direct or indirect evaluation methods. The various direct and indirect methods of evaluation proposed on some literature described as follows.

2.3.1. DIRECT EVALUATION METHODS

Direct evaluation methods measure the performance of the stemmer directly on a collection of testing words independent of any application [14]. These methods measure the stemmer performance in terms of error rates, correctly stemmed words (accuracy), statistical methods, and etc. These methods require the collection of test words of the language, which involves a lot of manual work. By the help of this method, we can measure the performance of the newly proposed stemmer with taking into consideration of error rates (under and over stemmed errors) and accuracy.

Under-Stemming Errors: as the name suggests that, it is the case when the stemmer strips the words below the expected level. In these types of errors, the words that have the same stem may not conflate together or related terms may not have same stems. For example, “dentistry” and “dentist” may not stemmed to the same root by the Porter algorithm [4], [2], [4]. If the stemmers have high number of under-stemming errors, the overall performance of the stemmer is decreased. Procedures like partial matching used in stemming algorithms helps to decreasing these errors by conflating the stems if which are morphologically similar with some defined cut-off. These procedures in some cases it may produce more errors but still it is useful and give good results.

Over-Stemming Errors: these errors occurred when the stemmer removes more terms from the given word form, thereby truncating parts that belong to the stem of the word. In these errors, two words having different morphological roots may conflated together to the same stems; for example, “illegal” and “illegible” are both stemmed to “illeg” and are grouped together. Over-stemming errors also decrease the performance of stemmers as two words that are not related but have the same stem might wrongly detected [2], [4].

These errors can be reduced by imposing constraints like minimum stem length of the resultant stem. As Paice suggestion a metric, named as Over-Stemming Index (OI), that used for measuring the errors, which is defined as $OI = 1 - \text{Distinctness index}$; where the distinctness index is the ratio of word pairs that are not conflated together to the total number of word pairs.

2.3.2. INDIRECT EVALUATION METHODS

Indirect evaluation methods are another categories of evaluation techniques that used to measure the performance of the stemmers by using them as a pre-processor of a specific application like Information Retrieval(IR) system, Text Classification(TC), and so on [14]. The major advantages of these methods are, that do not require tedious manual labour as it make uses of various automated tools to measure the performance on the newly proposed stemmer. Nevertheless, it require various resources such as document collections, query sets, and are quite sensitive to the type of collection and queries used during testing process. For Information Retrieval tasks precision, recall and F-score are used for measuring the stemmer of performance directly [14].

Precision measures the number of relevant documents retrieved out of the total documents retrieved. Whereas recall measures the total number of documents retrieved that are relevant to the total number of relevant documents with respect to the query. The weighted mean of precision and recall is termed the F-Score and it is widely used for testing the retrieval accuracy, as it considers both Recall and Precision. The mean of Precision and Recall values at various ranks in a ranked list of documents is termed Average Precision and quite frequently used in evaluating the retrieval accuracy of an IR system [14], [13].

Stemming improves precision as well as recall. This is because of the impact on Term Frequency-Inverse (TF), Document Frequency (TF-IDF) weighting. We can get a different frequency by grouping variant word forms i.e. documents that are more relevant are promoted at superior ranks [13], [40]. On the next section, the researcher tried to asses or reviews some recent related works in the field of stemming.

2.4. RELATED WORKS

2.4.1. INTRODUCTION

In the field of Natural Language processing and Information Retrieval system; numerous research have being done and researchers invest their time to build good IR system and NLP applications. Stemming approaches are the under laying areas of such applications to process and handles natural languages. On this sub section, we have tried to review researches that were done particularly on stemming techniques for some of the local and foreign languages.

Much of research work on stemmer have been done on English, Indian, and Arabic languages [1], [7], [41], [50], [52]–[54]. In contrast, for Ethiopian language there were little research have being done, particularly; there were little research tried on some language like Amharic, Afan Oromo, Tigrigna, Afaraf, Siltie, Geez and Awing [11], [12], [17]–[19], [25], [26], [37]. Due to these reason it's amenable to do more and more investigation on Ethiopian language to support IR system. On the following section we have tried to discuss about some of those research work.

2.4.2. STEMMERS ON FOREIGN LANGUAGES

From the past many years, unending attempts have been made to build efficient stemming algorithms. There have been a lot of research works introducing some new theories and implementations of stemmer. But the product produced didn't map the desired expectation [32]. This leads to further investigation for stemming different language.

In the meadow of stemming, Lovins stemmer was the first published work by Lovins on (1968) [55] . It was a single-pass stemmer that works in two steps. First, it removes the suffixes by performing a lookup on a list of 294 suffixes each associated with 1 of 29 context-sensitive conditions. The suffixes in the list are arranged according to their lengths. In order to stem the word, the suffix list is enquired on the basis of the longest-match principle.

If the suffix with a satisfying condition is found, then it is removed from the word. For instance, in order to stem the word “rationally”, the first suffix that matches in the list is “ationally” with condition “minimum stem length of three”. This suffix is discarded as the stem will be of length less than three. The next suffix in the list “ionally” with no constraint on stem length is selected, and the root “rat” is returned. In the next step, the stem is recoded by using another list containing 35 transformation rules to convert the roots into valid English words. Finally, the

words whose roots are moderately close but not essentially same are grouped together using the partial matching method. The Lovins stemmer is simple and fast, but it missed many common suffixes.

The Porter stemmer (1980) [56], is the first written, most popular and widely used English rule-based stemmer. Porter defined English words as a sequence of vowels and consonants, that is, $[C] [VC]^m [V]$, where V and C denotes one or more vowels and consonants, respectively, and m is the measure of the word. The Porter algorithm defines 60 rules that are applied to the word to be stemmed in five steps.

Each rule of the algorithm is of the form (condition) (suffix) \rightarrow (resultant suffix). The rule specifies the indispensable condition in which it is to be applied and how the word is altered to obtain the stem. As an example, rule $(m>0) \text{NESS} \rightarrow \phi$ denotes that if a word has ending NESS and the measure of the resultant stem is greater than zero, then it remove the ending. So, according to this rule, the word “goodness” is stemmed to “good”. The Porter stemmer is efficient with regard to readability and complexity, but errors like over-stemming (probe/probable) are well known. An improvement to the Porter algorithm, called the Porter2 algorithm, has also been developed [56].

Another research was done by the F. Ahmed et al. on (2009) [3], to evaluate n-gram compilation approach for Arabic text. They proposed a language independent approach based on unsupervised method that enhance pure n-gram model. It can group related words based on various string similarity measure while restricting the search to specific location of the target word by taking into consider the order of n-gram. The stemmer produce a best result and reduce ambiguity rather than the pure n-gram additionally the present an adaptive user interface for “Arasearch” that helps as meta search for current meta search engine.

In order to assess the new approach by comparing with pure n-gram, they select bi-gram and tri-gram for eliminating the problem of short words. The previous work demonstrated that, $n=3$ or 4 was well suited for Arabic Information retrieval (AIR), this constraint leads to a problem of handling short words. In contrast the suggested approach can handle the short words by using bi-gram ($n=2$) and reverse n-gram for avoiding ambiguity. Final they have got revised bi-gram is better than the pure n-gram, and they have recommended that, an n-gram model was preferable for highly inflected language.

Hybrid stemmer for Gujarati language was proposed by (2010) in which collected the linguistic knowledge in the form of hand crafted suffix list for improving the quality of the stem and suffixes during the learning phase. The proposed approach is based on Goldsmith's (2001) methods by taking all spit method. They have used EMILLE corpus for training phase in order to learn the probable stem and suffix. For evaluating the performance, they have performed various experiment and used 5-fold cross validation. The experimentation was performed with and without handcrafted suffix list. As the experiment showed that, with handcrafted suffix list have the better result. As the researcher conclusion, the proposed system has an accuracy of 67.86 %. This stemmer can handle only inflectional endings, i.e. could not handle derivational ending.

Kumar D. and Rana P. (2010) [57] develop a stemmer for Punjabi language by the help of using brute force and suffix stripping techniques. The proposed system uses mainly dictionary look up and suffix stripping as additional methods. The approach has two pointers (input and matching pointer) and three constituents in which input, output and process. In order to evaluate the performance of the proposed techniques the authors have used the parameter like correctly stemmed word, effectiveness and performance of the stemmer. The system was normally a beginner's version for the language and does not require processing of the text before stemming the word. Even though, the stemmer was good it have some problem due to errors of suffix stripping.

Continuously Gupta V. and Lehal G. (2011) [41], was proposed a new stemmer for Punjabi noun and proper name. The authors have generated various rules with nineteen steps and as they have evaluated the stemmer, the output of Punjabi language for nouns and proper names has been done over 50 Punjabi documents of Punjabi news corpus of 11.29 million words. The efficiency of the stemmer was recorded 87.37% which is tested on over fifty (50) news documents. It have some errors due to the violation of the rules, dictionary error or syntax mistakes. Even if it has some errors this stemmer was successfully used in Punjabi language text summarization.

An improved Arabic Light Stemmer was one of the best Arabic Stemming Algorithm in which proposed by Elrajubi, Osama Mohamed in (2013), [58]. It was designed to conflate Arabic Word that out-performed the other light stemmers. The proposed approach has eight steps for generate the stem of the given word. But the rule causes changing of the meaning of some words, as a

result the author applied some other rules to correct these words using. This algorithm was implemented and compared the results with the light10 stemmer.

For implementation purpose, he used four news articles written in Arabic language were chosen from Aljazeera website channel on the Internet (<http://www.aljazeera.net>). The word count of these articles was 2791 words. After employing these words to the stemmer, the accuracy rate of the Light10 and proposed stemmer were 66% and 88.25 % respectively. Therefore, the proposed stemmer is better than Light10 stemmer. Even though the proposed stemmer improved the accuracy rate of the system, it does not provide the correct stem for a large number of words (328 out of 2791 from the test data).

On the other hand for Hindi language, there was a lot of works done so far. Mishra U. and Prakash C. (2012) [6], proposed an effective stemmer for Hindi language called MULIK that are purely based on Devangari script and works on a hybrid approach particularly a combination of dictionary lookup and suffix removal techniques. For the case of evaluating the proposed stemmer, they have used accuracy of stemmed word, effectiveness and performance of stemmer. For accuracy purpose, they have considered a look up database of 15,000 words. The system will works at an abnormal condition occurred, even if the inputted word does not exist in the look up. As a final point the proposed stemmer showed an accuracy of 91.58 % and reduced under and over stemming error.

Mohammed N. Al-Kabi, (2013) [16] proposed a new Khoja stemmer that uses various patterns and flaws. This stemmer is well-thought-out by a number of researchers as a standard stemmer for Modern Standard Arabic (MSA), which was a typical analysis of pattern framework. The systems identify the flaws leads to identification of missing Patterns not used by Khoja stemmer. As a result the augmentation to Khoja stemmer is restricted to adding missing patterns this leads to a round five percent improvement to the accuracy of khoja stemmer. From the experiment the accuracy was registered to 90.93 % by using more than 600 Arabic words with their correct three lateral verbs.

Paul A. et al (2014) [7] introduced a system described an affix stripping technique for finding out the stems from context free text in Nepali language using lexical lookup based and rule based approach. The system starts by introducing different types of lexicon, the basic unit of Nepali

stemmer, and few rules to identify the word in the lexicon and by integrating them. They develop extensible architecture for stemmer development system that handled data related to samples of economics, health and politics in Nepali language, which are based on Devanagari Script. Their system showed some improvement in the performance over simple rule based system.

Mahmud, Redowan proposed (2014) [52] a rule-based algorithm that eliminates inflections stepwise without continuously searching for the desired root in the dictionary. The stems can be computed algorithmically cutting down the inflections step by step. The algorithm is independent of inflected word lengths, they used two separate stemming algorithms i.e. one for the verbal inflected words and another for noun inflected words with integration of hierarchical approach for stripping suffixes from the inflected words.

Al-Omari A. and Abuata B. (2015), [59] Proposed an Arabic light stemmer (ARS) in which they design and implement a new Arabic light stemmer (ARS) which is not based on Arabic root patterns. Instead, it depends on well-defined mathematical rules and several relations between letters. They have compared the proposed stemmer effectiveness against two other light stemmers. As the result showed that, ARS out weight its performance even if few wrong stems found when applied on a set of 6,225 Arabic words.

2.4.3. STEMMERS ON ETHIOPIAN LANGUAGES

Unlike English and other western languages, Ethiopian languages are less researched languages in the areas of information retrieval and natural languages processing applications. Recently there are some researches done in the areas of IR and NLP for Ethiopian languages like Amharic [11], [37], Tigrigna [17], Silt'e [27], Awnigi [25] and Afan Oromo [12] and Geez [18], [19] were some of the reports done[60]. We have discussed each of them as follows.

Bethlehem M. (2002) [37], proposed an automatic indexing for Amharic language text by using N-gram based approach. This approach computes similarity and cluster similar words into group and represents the groups by one stem or root term. She developed the system by assigning bi-gram and tri-gram particularly. To compare the results of the N-gram approach she used word based indexing. The researcher tests the system by applying 100 documents with 24 queries. As the experiment showed that, the word based indexing was better than n-gram based retrieval. However accomplish n-gram based approach with bi/tri-gram still perform comparable results.

Another research was conducted by Mezemir G. (2009) [11] for Amharic language text. He have developed an automatic stemmer algorithm using successor variety approach and for the purpose of training and testing this methods he have prepared a corpus of 6270 words obtained from the Ethiopian News Agency (ENA) and Walta Information Center (WIC).

The algorithm was implemented based on entropy and complete, peak and plateau method. From the experimentation result showed that, the successor variety algorithm with the peak and plateau method had a better performance than successor variety algorithm with the entropy and complete method. The performance of the proposed approach were performed an accuracy of 71.8 % for peak and plateau method, while the entropy and complete methods performed 63.95 % and 57.99% level of accuracy.

Debela Tesfaye (2010), [12] develop a hybrid stemmer for Afan Oromo language text. The algorithm follows the known Porter algorithm for the English language and it is developed according to the grammatical rules of Afan Oromo language. Particularly he adopt some concepts like measure, arranging the rule into cluster and analyzing word formation based on the nature of their endings. The rules have seven clusters, each of them represents a particular class of affixes and the rules class was ordered and mutually exclusive.

Two version of algorithm were developed, the first algorithm was purely rules based and the second algorithm was statistics (n-gram). The author first checked the rule based algorithm and tried to integrate with n-gram. To evaluate the performance of the proposed stemmer, error counting technique was employed. For testing purpose 198 sentences with a total of 2458 words collected from various sources and the result registered was 95.73% correct and shows an enhancement from previously designed rule based approach.

Yonas Fisseha [17] investigated that, a rule based stemmer for Tigrigna text on (2011). This stemmer was created from small rule-sets by affixes removal techniques particularly, inflectional and derivational affixes. In order to make the rule the researcher has taken in to consideration various exceptional issues. He has developed ten rule set for prefix stripping and seven rule set for stripping suffix. The proposed stemmer was evaluated and tested based on counting of actual under stemming and over stemming errors using a total of 5437 word variants derived from two datasets. As his experiment showed that, the average accuracy registered was 86.1% and the error rate was 13.9%

On the other hand Muzeyn Kedir (2012), [27] designing a rule based stemming algorithm for silt'e language. The proposed stemmer was used an iterative approach, context sensitive and recoding rules to remove prefix, suffix and reduplication of letters from silt'e language text. Stemmer was applied firstly prefix, secondly suffix and finally letter reduplication were examined. To test the proposed stemmer he used 1486 words, which were selected randomly from the sample texts. The result of the experiment shows that, the designed stemmer achieved an accuracy of 85.71%, and brings a dictionary reduction of 34.99% for stem words. The proposed stemmer conflates only derivational and inflectional words; it could not handle irregular and compounding forms of the word.

Lastly, Abebe (2010) [19] develop a rule based stemmer for Geez language. Affix removal and morphological analysis techniques were used for developing the proposed stemmer. As he was clarified the language, it is morphologically rich and complex. The main word formation process of Geez is affixation like prefixes, suffixes, infixes, circumfixes and concatenation of affixes. The stemmer has generally three actions in which the first two actions were applied on the affixes removal phase and third one was applied on the morphological analysis phase with its respective condition. The conditions were used to check the rule and applying the required action.

In order to evaluate the proposed stemmer, manual error counting mechanism was employed. Through the experiment they have seen three types of errors namely under stemming, over stemming from affix removal techniques and some structural problems form morphological analysis technique also. The accuracy performed by the system were 82.42%; even if the performance was good, he recommended to improve the performance of the stemmer by adding additional rules and applying another approach.

Generally, one of the short coming of the stemming research conducted before, whether for foreign language's like English, Indian, Arabic etc. [4], [6], [16], and local language like Amharic, Tigrigna, Afan Oromo etc. [11], [12], [17]. It couldn't applied on the rules for Geez language text. This is because, the morphological complexity and the pattern of word formation process of Geez is totally different from these stated languages.

Additionally, one of the stemmer developed by Abebe [19] for Geez language has been faced by many challenges. The first problem is that, the compilation of affixes list is very limited affixes;

hence, the designed stemmer can handle only 22 prefixes and 32 suffixes lists with taking into consideration of one and two radicals only. For example, ዘኢይ-, ዘይት-, ለአስተ- ወዘኢትት- and -ኩክን, -ያቶን, -ያተሆሙ, -ከናሆሙ etc. are a prefixes and suffixes that did not striped by the rule based stemmer respectively. As a result, it does not have the complete list of affix, i.e. it cannot conflate even common possible affixes of the language. This leads to produce improper stem as a final result.

The second problems of this stemming is that, the compilation of stop words excluded the common and frequently occurring non-content bearing words that are found on Geez text collections. As an example the word ማእከለ/between/, ላእለ/with---on/, እስፍንቱ/how many/, ድንገር/behind/, አዲ/or/ and ለምንት/why/ are non-content bearing words; that should be included as a stop-word list but it did not considered by this stemmer.

Another problem is that, the stemmer simply strips any end of a word that matches one of the affixes in a list without any detailed condition have been examined. It consider only length of the term to be stemmed should be more than three as a pre-conditions. For example, based on this stemmer the length of the term ‘ተንበልከዎሙ’ is more than three; as a result the letter ‘ተ’ will be removed by the stemmer and will get ‘ንበልከዎሙ’ as an output. Then also it will remove the last suffixes ‘ዎሙ’ because it satisfies the the condition, and the final result will be ‘ንበልከ’ which is not popper stem for the word ‘ተንበልከዎሙ’. The correct stem of the given word/term is ‘ተንበለ’ in which the long possible suffix ‘-ከዎሙ’ is removed and the last letter ‘ል’ should be changed to ‘ለ’ by considering some recording and exceptional rules that will be examined before and/after affixation process. These conditions show us, exceptions and recording rules for the stemmer should be incorporated rather than considering only the length of the stemmed words.

On the other hand, structural problem was facing this rule based stemmer and the error rates were high. This requires the need for the detailed knowledge of the language to come up with the good stemmer by seeing different exceptional case.

Recently Afeworki et al. (2019), [28] conducted a study on Geez language by using longest match approach. The stemmer can handle irregular words and it removes affixes with considering some exceptional cases. For evaluation purposed stemmer a test data set of 2000 words were applied on the proposed prototype and finally the performance of the stemmer with

respect to accuracy were registered as 87.22%. According to the result found, this stemmer outperformed by 4.8% accuracy with reducing an error rates of 4.8%.

Even if the result found was encouraging and shows some enhancement than the research conducted by [28], it was challenged by error rates of 12.78%. This leads to degrade the performance of the stemmer and it was the shortcoming of the study. This shows us trying different methods and approaches with the detailed study of the morphology of the language are needed in order to come up with a good stemming application for the language by reducing the errors found and increasing the accuracy level of stemmer for this language.

As far as the knowledge of the researcher goes, there is no previously conducted research in Geez language by applying hybrid techniques. Therefore, the researcher has an interest to apply a hybrid approach for Geez text and test the performance of its result by trying to overcome the limitation of the previously rule based stemmer studied by Abebe [19] and Afeworki et al. [28]. In order to apply the newly proposed hybrid approach, further predefined rule that didn't covered by the previous researcher were included by conducting detailed study of Geez morphology and statistical character n-gram approach is selected. Character n-gram technique can handle some inconvenience that cannot handle by the designed rule sets.

CHAPTER THREE

GEEZ MORPHOLOGY

3.1. INTRODUCTION

Geez language is rich in vocabulary and it has the characteristics of carrying different messages with a single word alone. Geez is widely used written language historical Ethiopia and EOTC. Various documents like art works, governmental documents, and religious scripts were widely available in the church and governmental possession are inherited to different users. Developing a stemmer for a language requires a study and modeling of the language phoneme in terms of word formation. As a result, on this chapter we have tried to discuss the general overview of the Geez language in details.

3.2. OVERVIEW OF GEEZ LANGUAGE

According to [61] Geez (ግዕዝ) is an ancient South Semitic language that originated in Eritrea and the Northern region of Ethiopia in the horn of Africa. It later became the official language of the kingdom of Aksum and the Ethiopian imperial court. Today Geez remains only as a main language used in the liturgy of the Ethiopian Orthodox Tewahido Church, Eritrean Orthodox Tewahido Church, the Ethiopian Catholic Church and the Beta Israel Jewish community.

Tigrigna and Tigre are closely related to this language with at least four different configurations proposed. Some linguists do not believe that Ge'ez constitutes the common ancestor of modern Ethiopian languages, but that Ge'ez became a separate language early on from some hypothetical, completely unattested language and can thus be seen as an extinct sister language of Tigre and Tigrinya. The foremost Ethiopian experts such as Amsalu Aklilu point to the vast proportion of inherited nouns that are unchanged and even spelled identically in both Ge'ez and Amharic (and to a lesser degree, Tigrinya).

According to [61] the study of languages forms the foundation of any study of ancient societies. A study of the Ge'ez writing systems is essential to understanding the history of Ethiopia and the evolution and modern usage of the Roman alphabet. This is not to say, by any means, that Ge'ez is merely a "bridging" system that serves only to connect ancient pictograms to the modern western alphabet, though that relationship may be unjustly implied in a Western study

concerning roman letterforms in comparison with the ancient language whose evaluation stopped where roman letter forms began is a very easy trap to fall into, especially in a distinctly Eurocentric society. This implies incorrectly that Geez is an outdated system that stopped being use full as Roman letterforms to the (Western) world stage.

By the 9th or 10th centuries ancient Geez ceased to exist as a spoken language in Ethiopia followed a century or to after, by the death of Latin in Europe after the thirteenth centuries as the remains of Latin were making metamorphoses into the romance languages, spoken Geez also split in to many closely related tongues, mainly Tigrigna in the north and Amharic in the south. However written Geez was kept firmly in use purely for sacred and scholarly endeavors, from the thirteenth through the seventeenth centuries, known as the classical period of Ethiopian literatures.

3.2.1. WRITING SYSTEM OF GEEZ LANGUAGE

Ge'ez is written with Ethiopic or the Geez abugida as script that was originally developed specifically for this language. In languages that use it such as Amharic and Tigrinya, the script is called Fidel, which means script or alphabet. It read from left to right and the script has been adopted to write other language in which the language is Semitic. The widely used one is Tigrinya in Eritrea and Ethiopia and Amharic in Ethiopia. It also used for Sebatbeit, Me'en, Agew and other languages of Ethiopia [61].

In Eritrea it used for Tigre, and Bilen, a Cushitic language. For other language in the horn of Africa like Oromo, used to be written using Geez but have switched to Latin based alphabets. The only language in Ethiopia which has its Owen alphabet is Geez language. Other languages like Amharic and Tigrigna adopts these alphabets fully from Geez [18], [19].

The alphabet/Fidel/ of a language represents its sound and it can be studied by dividing them into simple sound and complex sound. The simple sound represented with 182 (one hundred eighty two) alphabets; 7 (seven) of them represent vowel sounds and the remaining 175 (one hundred seventy five) sound represent consonants. Generally Geez language have 26 (twenty six) syllographs and alphabet, all consonants and each with six more derivation. On the other hand

the complex sound are represented with 20 (twenty) letters and the alphabet four in number [21], [22], [62].

Geez is fairly massive in size with its 182 syllographs as compared to ancient Romans 21. However in order to make a fair comparison it must be said that there are essentially 26 main syllographs , all consonants in Geez while the rest are essentially those with additional strokes and modifications added on to the main forms to indicate a vowel sound associated with it or to make aural adjustment in the basic consonant sound . It must be acknowledged also that, there are not upper and lower case distinctions in Geez as had evolved in the Roman alphabet by the seventh century. There are not ligatures or other symbol modifiers (as seen in “G” and “g”) as well as very little punctuation. So to be more accurate in comparison the uppercase “A”, lower case “a” and accented letters “a” in the Roman alphabet would have to certain punctuation rules associated with them (‘s). Even on the curve Geez is significantly larger in size. It should be recognized though as also being large in scope.

The basic columns are labelled as ግእዝ (1st -order), ካእብ (2nd order), ሳልስ (3rd-order), ራብዕ (4th-order), ሐምስ (5th-order), ሳድስ (6th-order), and ሳብዕ (7th-order) in each of the alphabets. The other columns more than the seventh-order (ሳብዕ) are ግእዝ (first-order), ካእብ (second-order), ሳልስ (third order), ራብዕ (fourth-order), and ሐምስ (fifthorder) families. Simple-sounds are represented with 182 alphabets. These, seven of them represent vowel sounds: አ, ኡ, ኢ, ኣ, ኤ, ኦ and ኦ. Whereas, the remaining represent consonant sounds [20], [22].

In conclusion, the Ge’ez writing system is one of the oldest working systems in the world. This African writing system has remained unchanged for 2000 years, attesting to its adaptability and innovative method of organizing sounds. It serves not only as a system of grammar, but as an insight into the ancient world of Africa, its philosophies, belief systems, and exceptionally advanced early societies [20].

3.2.2. NUMERALS IN GEEZ LANGUAGE

As other language Geez have its own numbering style. Amharic language adopts the numbering style of this language in addition to Indo-Arabic numbers like 1, 2, 3 etc. Ethiopian yearly calendars widely used Geez numbers for celebrating national ceremony. More over EOTC uses the numbers for yearly celebrations of monthly, yearly and other special ceremony of the church.

The following table shows us some of Geez numbers associated with corresponding Indo-Arabic numbers to clarify how numbers are used.

Geez Numbers	Geez Numbers with letters	Indo-Arabic numbers	Geez Numbers	Geez Numbers with letters	Indo-Arabic numbers
—	አልቦ	0	፳	እስራ	20
፩	አሐዲ	1	፴	ሠላሳ	30
፪	ክልኤቱ	2	፵	አርብዓ	40
፫	ሠለስቱ	3	፶	ሃምሳ	50
፬	አርባዕቱ	4	፷	ስድሳ	60
፭	ሐምስቱ	5	፸	ሰብዓ	70
፮	ስድስቱ	6	፹	ሰማንያ	80
፯	ስብዓቱ	7	፺	ተሰዓ	90
፰	ስመንቱ	8	፻	ምዕት	100
፱	ተሰዓቱ	9	፲፪	አሠርቱ ምዕት	1000
፲	አሠርቱ	10	፻፻	እልፍ	10,000
፲፩	አሠርቱ ወአሐዲ	11	፲፻፲	አሠርቱ እልፍ	100,000
፲፪	አሠርቱ ወክልኤቱ	12	፻፻፲	አእላፋት	1,000,000
፲፫	አሠርቱ ወሠለስቱ	13	፲፻፻፲	ትእልፊት	10,000,000
፲፬	አሠርቱ ወአርባዕቱ	14	፻፻፻፲	ትልፊታት	100,000,000
፲፭	አሠርቱ ወሐምስቱ	15	፲፻፻፻፲	ምእልፊት	1,000,000,000

Source:[21]

Table 4 Sample Geez numbers with corresponding Indo-Arabic numbers

The above table demonstrates a sample of Geez numerals with corresponding alphabetic representations and equivalent Indo-Arabic numbers.

3.2.3. GEEZ LANGUAGE PUNCTUATION MARKS

There are around 16 punctuation marks existed in the language with their role of writing to separate sentences and their elements meaning clarification. However, only a few of them are commonly used for writing purpose of Ge'ez sentences such as, section mark(*), word separator(:), full stop/ period (#) comma (፤), colon (፥), semicolon (፦), preface colon (:-), question mark (፤), paragraph separator (፦፦), and some of them are no longer used [40]. For example the word separator (:) is no longer used nowadays literature; it is replaced by white space; Instead of writing ሰንበት:ተዐቢ:እምሰሉ:ዕለት:ወሰብእ:ይከብር:እምሰሉ:ፍጥረት we can write simply as ሰንበት ተዐቢ እምሰሉ ዕለት ወሰብእ ይከብር እምሰሉ ፍጥረት by replacing two dots with white spaces.

3.3. MORPHOLOGY OF GEEZ LANGUAGE

Morphology is a branch of linguistics in which that studies and describe about how words are formed. Mainly it covenants with the internal structure of a word in the natural languages. On the other hand computational morphology deals with developing theories and techniques for computational analysis and synthesis of word forms.

Morpheme is the minimal linguistic units of a language in which that carry meaning and cannot be further decomposed in to meaning full units. Geez morpheme can be divided in to free and bounded. Free morpheme is a morpheme that can stand as a word alone and bounded morpheme cannot found or occur on its own as a word.

In the following subsection the researcher describes in details about Geez morphology especially how different word classes are formed and individual words are inflected and derived to form word variants.

3.4. WORD FORMATION OF GEEZ LANGUAGE

The Geez word variation is done by mainly inflectional and derivational affixes. On the following sub sections we have to discuss about derivational and inflectional affixes of the languages. Geez morphology is formed from affixation and derivations of a given words. It has a concatenated morphology like prefixes, suffixes and prefix-suffix pairs, non-concatenated morphology like infixes and compounding morphology like joining of two or more base form words to form new word forms. Example the word ወሰብእ/wesebe/ is a word with a prefix ወ and a noun ሰብእ/human/; ሰአልከሙ/you bagged/ is a word with a verb ሰአል/bagged/ and a suffix ከሙ and

ወይዘንወክ/he told to you/ is a word with prefix-suffix pair ወይ...ክ and the verb ዘነወ on the other hand the word ቤት ዘንጉሥ/ king's house/ is a compound word ቤት/house/ and ዘንጉሥ/king's/ i.e., ዘ indicate possession.

But Geez does not have clitics such as “s” as English language to show possession [18],[20]. On Geez language the formation of words can be used affixation, compounding, duplication or reduplication and different vowel patterns like other Semitic language such as Amharic, Tigrigna and Tigre. For example በበይናቲሆሙ/with together/ is a word that is formed by duplicating the latter በ.

Different languages grammars have its own word classes based on its nature. Like that; Geez Language grammar has six word classes or part of speech; these are noun /ስም/, pronouns /ስመ ተውላጥ /, adjective /ቅፅል/, verb /ግስ/, adverb /ተውላክ ግስ/ and prepositions and conjunctions /አገባብ/ [20], [22]. For example the word አቡነ/our father/, ሰማይ/sky/, and መልአክ /Angele/ are a nouns; the word ውእቱ/he/, ይእቲ /she/ and etc. are a pronouns, the word ሐረ /went/, መጽአ /came/ are a verb and also the word ነዊን /tall/, ቀይሕ /red/ and ሐፂር /short/ are an adjectives; እፎ /how/ is an adverb and the word ምስለ/with/ is a preposition.

3.4.1. INFLECTIONAL AFFIXES OF GEEZ LANGUAGE

Inflectional affixes describe about word stems are united with grammatical indications for things such as gender, person, number, tense and cases. To agree with the subject of the language Geez, noun and verbs can be marked for these different grammatical markers due to its richness in morphological character.

3.4.1.1. NOUN INFLECTIONAL AFFIXES

From the ground the term nouns are name of peoples, places, things and abstract ideas in which that tell us what we are telling about for example, Ethiopia /ኢትዮጵያ/ and Solomon/ሰሎሞን/ are nouns. According to [19], Ge'ez noun can be inflected into number, gender and case in which they have their own phonetic structures. The phonetic structures fundamentally comprises of numerous character arrangements.

According to Dillman [63] the formation on Nouns are passes through three stages in which the Nominal stem is formed from the root, the stem that differentiated by number and gender and the word those elaborated assume special forms, or cases according to the special relations upon

which they enter in the sentences. Geez noun is very rich in morphological character; it can be pluralized in to two ways. These ways are by using internal plural marker and external plural marker. A number marker such as prefixes, suffixes, infixes and their combination creates plural nouns. Number usually represent for noun adjectives and verb conjugations. There are different ways for variant word formation of the nouns. According [21], [22], [62], the common Geez alphabet that used to form variant words are ኢ/a/, ን/n/, ት/t/, ሙ/mu/, ይ/y/, ወ/we/, and ለ/l/. These alphabets may change its orders based on the nature of the nouns that will be attached with.

To use external plural marker, we can use the following rules:

- A nouns that are inflected by an alphabet ‘ወ/wu/’ may change the last alphabet form 6th order to 1st order and the suffix added to the ending positions.
- A nouns that ends with 5th order letter; it can be pluralized by adding a suffix ያት /yat/ at the end positions.
- A nouns that ends with 4th order letter; it can be pluralized by adding a suffix ት /t/ at the end positions.
- A nouns that ends with 7th order letter; it can be pluralized by adding a suffix ዋት /wat/ the end positions.
- A noun that ends with 3rd order letter; it can be pluralized by adding a suffix ያን /yan/ to masculine and ያት /yat/ to feminine at the end positions.
- A nouns that ends with 6th order letter; it will changes the end radical to fourth radical and adds ት /t/ or ን/n/ at the end positions.
- A nouns that ends with “ት /t/”; the last letter will be removed and can be inflected by the alphabet ይ/y/ at the end positions.
- A nouns that have two radicals can be pluralised by preceding the alphabet ኢ-/a- and attaching or inserting additional letter -ዋ-/wa/ at the middle positions.
- A nouns that can be inflected by ሙ/mu/ and ለ/l/; some change can be made at the internal alphabet and also adding the suffix ሙ/mu/ and ለ/l/ to the end positions.
- A nouns that start with the alphabet ኢ/a/; the second radicals the last alphabet of the word in to a sixth order and may add “ት/t/” at the end. Sometimes because of the presence of the gutturals the second order may be changed in to the fourth order.

- There are also exceptional nouns that don't follow these rules. Such nouns can be pluralized by making change on the internal part of it without adding any additional affixes. Generally internal plural markers are summarized on the next tables (see table 5 and table 6).

Singular noun	Meaning	Plural forms	Meaning	Prefix and Suffixes
ደመና	Cloud	ደመናት	Clouds	-ት
ሐዋርያ	Apostle	ሐዋርያት	Apostles	-ት
ጽጌ	Flower	ጽጌያት	Flowers	-ያት
ምሳሌ	Example	ምሳሌያት	Examples	-ያት
መርኅዋት	Key	መርኅዋት	Keys	-ዋት
ማይ	Water	ማያት	Waters	-ት
ጸም	Fasting	አጽዋም	fastings	አ-
ገም	Tree	አዕዋም	trees	አ-
መከሊት	Money	መከሊይ	Moneys	-ይ
ሱራፊ	Angele	ሱራፊል	Angeles	-ል
ነዳይ	Poor	ነዳያን	poor's	-ን
ዐይን		አዕይንት		አ-, -ት
እድ	Hand	አእዳው	Hands	አ-, -ው

Table 5 External plural formation of nouns

From the above table we can understand that, most of Geez nouns can be pluralised by following the aforementioned rules. For the purpose of making agreements with numbers the noun can be inflected by using affixations.

On the other hand an internal plural marker also creates plural nouns in which may not follow similar rules like external plural markers. In Geez language; such nouns are few in number. We can see how singular nouns are changed to its plural form on the following table (see table 6).

Singular noun	Meaning	Plural forms	Meaning	Infixes
ድንግል	Virgin	ደናግል	Virgins	ድ is changed to ደ and ን changed to ና
ደብተራ		ደባትር		ብ is changed to ባ and ራ changed to ር
ወልድ	Son	ውሉድ	sons	The first two radicals are changed
መዘራኝት		መዛርኝት		The middle two radicals are changed
መቅደስ		መቃድስ		The middle two radicals are changed
መልታህ		መላትህ		The middle two radicals are changed

Table 6 internal plural formation of nouns

As a result we can understand from the above table (see table 6) internal plural markers change the singular forms to plural by making some changes internally and adding some additional words from the start, the middle and the end position of the given word respectively.

3.4.1.2. PRONOUN/መራሕያን/ IN GEEZ LANGUAGE

A pronoun is a word in which that takes the place of a noun. There are different types of pronoun in English. For example personal/subjective and objective/, reflexive, demonstrative, interrogative and indefinite pronouns are the main kind of pronoun. Like this Geez language has different kinds of pronoun as English.

Unlike English which has six pronouns, Geez language has 10 /ten/ pronouns that used to for representing the noun, object and adjective. For example ውእቲ ሐረ ኅበ ቤተ-ትምህርት / he went to school/ the underlined word ውእቲ is a pronoun that represents the noun. Generally the following table shows these ten pronouns with its corresponding English pronoun.

Category	Gender	Singular	Plural
3 rd person	Masc.	ውእቲ/he/	ውእቶሙ/they/
	Fem.	ይእቲ/she/	ውእቶን/they/
2 nd person	Masc.	አንተ/you/	አንትሙ/you/
	Fem.	አንቲ/you/	አንትን/you/
1 st person	የወል/for both/	አነ/I/	ንሕነ/we/

Table 7 Geez Pronouns

According to [21], Geez pronouns used as a subject and an object by representing the subject and object of geez sentences as follows:

A. Objective Pronouns (ተሳሐቢያዊ ተውላጠ ስም)

The object of a verb receives the actions of the verb due to this, the personal pronouns me, you, him, her, it, us, and them can all be used as the object of the verb. For example መኑ ይጻውእ ኪያክ /who calls you/ and እግዚአብሔር ፈጠረ ኪያክሙ /God create you /; the words ኪያክ and ኪያክሙ indicates that an object pronouns respectively. The following table shows us the objective pronouns with singular and plural forms.

Forms	Objective Pronouns	Meaning
Singular form	ከ,የየ	/እኔን /Me
	ከ,የከ	/አንተን /You
	ከ,የከ	/አንቺን /You
	ከ,የሁ	/እሱን / Him
	ከ,የሃ	/እሷን /Her
Plural form	ከ,የነ	/እኛን /Us
	ከ,የከሙ	/እናንተን/You
	ከ,የከን	/እናንተን /ለሴቶች/ You
	ከ,የሆሙ	/እነርሱን /them/
	ከ,የሆን	/እነርሱን /ለሴቶች/them

Table 8 Objective pronouns of Geez

B. Subjective Pronouns /ባለቤት ተውላጠ ስም/

In English language the subject of a verb does the actions of the verb so the personal pronoun we, I, he/she, it and they can be used as the subject of the verb. On the other hand Geez have its own corresponding subjective pronouns. For example እግዚአብሔር ለሊሁ ፈጠረ ዓለም /God create the world himself/ and ለሊከ አውሰብከ / married yourself/ from this two simple sentences the underlined words ለሊሁ and ለሊከ are a subjective pronouns.

Forms	Subjective Pronoun	Meaning
Singular form	ለሊሁ	Himself
	ለሊሃ	Herself
	ለሊከ	Yourself
	ለሊከ	Yourself
	ለሊየ/ለልየ	Myself
Plural form	ለሊሆሙ	Themselves
	ለሊሆን	Themselves
	ለሊክሙ	Yourselves
	ለሊክን	Yourselves
	ለሊነ	Ourselves

Table 9 Subjective Pronoun

3.4.1.3. GENDER MARKERS

On Geez language there are three types of gender; these are masculine, feminine and neutral, 2nd and 3rd person singular/plural indicates masculine and feminine, and 1st person singular and plural indicate neutral that means, it does not indicate feminine or masculine due to this reason; both masculine and feminine used it. The following table illustrate genders in Geez.

Category		Gender	nouns	Suffixes
3 rd person	Singular	Masc.	ዜናሁ	-ሁ
		Fem.	ዜናሃ	-ሃ
	Plural	Masc.	ዜሆሙ	-ሆሙ
		Fem.	ዜናሆን	-ሆን
2 nd person	Singular	Masc.	ዜናከ	-ከ
		Fem.	ዜናኪ	-ኪ
	Plural	Masc.	ዜናከሙ	-ከሙ
		Fem.	ዜናከን	-ከን
1 st person	የወል/for both/		ዜናየ	-የ
			ዜናነ	-ነ

Table 10 the three gender markers in Geez

Form the above table we can understand that; the suffix *-ሁ*, *-ሃ*, *-ሆሙ*, *-ሆን*, *-ከ*, *-ኪ*, *-ከን*, *-የ* and *-ነ* are the gender markers. These listed suffixes are used not only as the gender marker; it may use as the number and possessions markers.

3.4.1.4. NUMBER MARKERS

According to [20], [21], [22], Geez has singular and plural numbers. The number markers are mostly present in nouns, adjectives, and verb conjugation. It can be indicated by affixes just indicating the gender as well. The number markers in other nouns are complex with the exception of verb conjugation. We can see the following table as an example (see table 11).

Singular noun	Plural noun	Number marker Affixes
ካጋዲ/merchant/	ካጋዲያን/merchants/	-ያን
ንጉሥ/king/	ንጉሥታት/kings/	-ት
አብ/father/	አበጫ/fathers/	-ጫ
እም/mother/	እመጫ/mothers/	-ጫ
ሐዋሪ/apostle/	ሐዋርያት/apostles/	-ያት
ሱራፊ/Angel/	ሱራፊል/angels/	-ል
መምህር/Teacher/	መምህራን/ት/Teachers/	-ን/ት

Table 11 Number markers in Geez

The above table demonstrate that, a suffixes that like -ያን,-ት, -ጫ, -ል, -ን and etc. can attached to the for the purpose of changing a noun from its singular form to the plural formats.

3.4.2. INFLECTIONAL AFFIXES OF GEEZ VERBS

In this sub section the study presents inflectional affixes of Geez verbs. For the compilations of Geez verbs inflections the researcher used different sources like [20]–[23], [62], [64]. In linguistics the term verbs are a word class in which it describes the actions and tells what peoples and/or things are doing. For example in English the word looking, eating, walking is an action verb which tells us what action is going on. Likewise Geez verb tells what peoples and/or things are performing. For example the verb በሊዕል /I having eaten/ indicate that what I am doing at this moment.

Geez verbs are highly rich in morphology and it has the ability to form different word classes such as noun, adjective and adverbs as a result verbal noun, adjectives, and or other verbs (infinitives and jussive) can be formed from Geez verbs. For the purpose of making agreements with numbers, genders and tense, affixes namely prefixes, suffixes, infix and prefix- suffix pairs are attached to the verbs.

Geez verb are may contains two, three and four radical root consonants called bi-lateral, tri-lateral and quadri-lateral respectively. But the common type is three radical types. The number of radical that one verb may contains, couldn't be less than two and exceeded than seven [18], [19], [20].

Geez verbs can be categorized based on different criteria's; the main categories are transitive and intransitive verbs. Transitive verbs are a verb in which it has an object that receives the action that is performed by the subject. For example 'Alice eats a banana for breakfast'; Alice is a subject, a banana is an object. Like this in Geez 'ሐራሲ ሓረሰ መርዔቶ' the word ሐራሲ indicates the subject and the word መርዔቶ describes the object of the verb.

On the other hand intransitive verbs are other categories of verbs that do not have an object in which affected by the action of the verbs. For example 'I will go to the market today' has not an object that receives the action. At the same time the sentence እነ ኢሐውር ዮም ኅበ ምስየጥ has not an object that is affected by the action performed by the subject እነ. Other criteria of Geez verbs are it can be categorized based perfection and imperfection which are the core verbs in Geez; perfection expresses completed action but imperfection express present, continuous, and future. The ending of all perfect verbs are first order and the ending of all imperfect verbs are sixth order under the pronoun ውእቱ/wuetu/ means a pronoun 'he'. In this sub section we have discussed verb categories and their inflection.

3.4.2.1. INFLECTIONS OF PERFECTIVE VERBS

Perfective verbs are a verb that describes finished or completed actions. In Geez the perfective forms of the verbs are the basis of the other verbs. For example ውእቱ ሰከበ ላዕለ ዐራት/he slept on the bed/ and እነ ተመሀርኩ ትመሀርተ ግእዝ /I learnt Geez language/; from these simple sentences the verbs 'ሰከበ ' /slept/ and ተመሀርኩ /learnt / expresses completed action on the past. The following table (table 12) clarifies to us how Geez perfective verbs are formed and inflected into varies forms.

Pronoun and gender	Verbs	Meaning	Suffix
ውእቱ/3 rd p.s.m/	ሰከበ/sekebe/	He slept	-
ደእቲ /3 rd p.s.f/	ሰከበት/sekebet/	She slept	-ት
ውእቶሙ /3 rd p.p.m/	ሰከቡ/sekebu/	They slept	-
ውእቶን/3 rd p.p.f/	ሰከቧ/sekeba/	They slept	-
አንተ /2 nd p.s.m/	ሰከብክ/sekebke/	You slept	-ክ
አንቲ /2 nd p.s.f/	ሰከብክ/sekebki/	You slept	-ክ
አንትሙ /2 nd p.p.m/	ሰከብኩ/sekebkmu/	You slept	-ኩሙ
አንትን /2 nd p.p.f/	ሰከብክ/sekebkn/	You slept	-ክን
እነ/1 st P.s./	ሰከብኩ/sekebku/	I slept	-ኩ
ንሕነ/1 st P.p/	ሰከብነ/sekebne/	We slept	-ነ

Table 12 Inflections of Perfective verbs

As the above tables shows that we can understand that perfective verbs can change the radicals (from the end letters of the verb the given verb, it changes from 1st radical to 2nd and 4th radicals for 3rd p.p.m and 3rd p.p.f respectively) of the letter and added same suffixes like ት, ከ, ኪ, ከሙ, ከን, ከ and ነ to agree with the person and genders. Generally all Geez perfective verbs are always inflected by these suffixes. In addition to suffixes it may use prefix like ተ/te-, አ/a-, and አስተ/aste- to further inflect in the form of five pillars of Geez. For example መከረ /he gave advise/ can be inflected as አምከረ /he gave advise someone by/, ተመከረ /he got advised/, አስተማከር /gave advise with other/ and ተማከረ/got advise with other/.

3.4.2.2. INFLECTIONS OF IMPERFECTIVE VERBS

Unlike perfective verbs, imperfective Geez verbs described non past action and it always uses prefixes, and suffixes for making agreement with the gender, numbers and persons respectively. For example አነ እሔውጽ እኑየ ‘I kill visit my brother’ from this sentence, the verb እሔውጽ /will visit/ indicates that the action is not completed. Table13, Illustrates how imperfective Geez verbs are inflected to various forms.

Person, number and gender	Verbs	Meaning	prefix
ውእቱ/3 rd p.s.m/	ይሔውጽ/yhewuts/	He will visit	ይ-
ይእቲ /3 rd p.s.f/	ትሔውጽ/thewuts	She will visit	ት-
ውእቶሙ /3 rd p.p.m/	ይሔውጹ/thewutsu/	They will visit	ይ-
ውእቶን/3 rd p.p.f/	ትሔውጹ/thewutsa/	They will visit	ት-
አንተ /2 nd p.s.m/	ትሔውጽ/thewuts/	You will visit	ት-
አንቲ /2 nd p.s.f/	ትሔውጹ/tehwutsi/	You will visit	ት-
አንትሙ /2 nd p.p.m/	ትሔውጹ/tehwutsu/	You will visit	ት-
አንትን /2 nd p.p.f/	ትሔውጹ/thewutha/	You will visit	ት-
አነ/1 st P.s.m/	እሔውጽ/ehewuts/	I will visit	እ-
ንሕነ/1 st P.p.m/	ንሔውጽ/nhewuts/	We will visit	ን-

Table 13 Inflections of imperfective verbs

As the above tables shows that we can understand that imperfective verbs can change the internal with and ending radicals of the letter and preceded by some prefixes like ይ-, ት-, እ-, and ን- to agree with the persons, numbers and genders. Generally all Geez imperfective verbs (including present and future have the same patterns) are always inflected by the prefixes ተ/te-, አ/a-, የ/ye- and ነ/ne/ and its variations.

3.4.2.3. INFLECTIONS OF JUSSIVE/SUBJECTIVE VERBS

Geez subjective/jussive describes the behaviours in which the actions are depend up on a preceding verbs volition or conjugation. For this language jussive and subject verbs have similar forms due to this reason; the affixes that are used for formation of jussive or subjective verbs are ይ-/y-/ , ት-/t-.i/ , ት-/t-/ , ን-/n-/ , ት-/t-...u/ , ይ-/y-...u/ , and ይ-/y-...a/. The only difference of subjective and jussive is that; on second person pronouns, i.e. jussive has not employed prefixes. In addition to that command verbs follow similar forms with subjective. Table 14 describes inflection of subjective verb ሐወጸ /he visited/.

Person, number and gender	Subjective Verb	Meaning	prefix
ውእቱ/3 rd p.s.m/	ይሐወጸ/yhewuts/	let him visit	ይ-
ይእቲ /3 rd p.s.f/	ትሐወጸ/thewuts	let her visit	ት-
ውእቶሙ /3 rd p.p.m/	ይሐወጹ/thewutsu/	let them visit	ይ-
ውእቶን/3 rd p.p.f/	ትሐወጹ/thewutsa/	let them visit	ት-
አንተ /2 nd p.s.m/	ትሐወጸ/thewuts/	let you visit	ት-
አንቲ /2 nd p.s.f/	ትሐወጹ/tehwutsi/	let you visit	ት-
አንተሙ /2 nd p.p.m/	ትሐወጹ/tehwutsu/	let you visit	ት-
አንትን /2 nd p.p.f/	ትሐወጹ/thewutha/	let you visit	ት-
አነ/1 st P.s/	እሐወጸ/ehewuts/	let me visit	እ-
ንሕነ/1 st P.p/	ንሐወጸ/nhewuts/	let us visit	ን-

Table 14 Inflectional formation of subjective verbs

3.4.2.4. INFLECTIONS OF GERUNIVE VERBS

In Geez language gerundives verbs cannot close sentence, as a result it needs other verbs in order to close the sentence. Gerundive verbs shows us an action is being done or not and it defines the occurrence of things. Table 15 illustrates the gerundive forms of the verb kedese.

Person, number and gender	gerundive Verb	Meaning	suffix
ውእቱ/3 rd p.s.m/	ቀዲሶ	He having praise	-
ይእቲ /3 rd p.s.f/	ቀዲሳ	She having praise	-
ውእቶሙ /3 rd p.p.m/	ቀዲሶሙ	They having praise	-ሙ
ውእቶን/3 rd p.p.f/	ቀዲሶሙ	They having praise	-ን
አንተ /2 nd p.s.m/	ቀዲሶከ	You having praise	-ከ
አንቲ /2 nd p.s.f/	ቀዲሶኪ	You having praise	-ኪ
አንተሙ /2 nd p.p.m/	ቀዲሶከሙ	You having praise	-ከሙ
አንትን /2 nd p.p.f/	ቀዲሶከን	You having praise	-ከን
አነ/1 st P.s.m/	ቀዲሶየ	I having praise	-የ
ንሕነ/1 st P.p.m/	ቀዲሶነ	We having praise	-ነ

Table 15 Inflection of gerundive verbs

Form the above table we can understand that the suffixes *-ሙ*, *-ን*, *-ከ*, *-ኪ*, *-ከሙ*, *-ከን*, *-የ* and *-ኅ*, are attached to the verbs in order to make it gerund forms and the second radical of the word is changed from first order to third order. For the 3rd person singular forms of both masculine and feminine, the change is made on the last radicals from first to 7th and 4th radicals respectively.

3.4.2.5. INFLECTIONS OF INFINITIVE VERBS

Infinitive verbs in Geez are inflected by the suffixation process that means to make an infinitive verb simply adding the suffix like *-ት/-ቲ* and making some changes on the last radicals on the given verbs. These verbs are always formed from the main verbs. Tables 16 clarify this as follows:

Infinitives	Meaning	Suffix
ሐዲጽ/ሐዊጾት	To decrease	-ት
ቀዲሶ/ቀድሶት	To praise	-ት
መጥዎ/መጥዎት	To give	-ት
ክሄል/ክሄሎት	To possible	-ት
ገቢር/ገቢሮት	To work	-ት
ነቢር/ነቢሮት	To seat	-ት
ሰሚር/ሰሚሮት	To like	-ት
ቀቲል/ቀቲሎት	To kill	-ት

Table 16 Sample Geez infinitives

As the above table shows the last radicals of the main verb is changed from 1st to 6th and 7th radicals and the second radicals from the left are changed to 3rd radicals. Finally the suffix is attached to the last positions. For example ነቢር/ነቢሮት /to seat/ is formed from the main verb ነበረ/he sat/.

3.4.3. ADVERB IN GE'EZ (ተውሳክ ግስ)

In English language adverbs are used for giving additional information about the verb, for example tomorrow, yesterday, today, always etc. Likewise in Geez language the word ‘ትማለም /yesterday/, ጌሴም/tomorrow/ and ዘልፈ. and ወትረ/always/ and የም/today/’ etc. are an adverbs.

According to [20] in geez language adverbs are used to express additional information about verbs or it gives descriptive information in which the action is performed, such as when, what, where how and etc. There are different kinds of adverbs in Geez language, these are: adverb of manner, adverb of frequency, adverb of place, adverb of time, adverb of degree, adverb of certainty, interrogative and relative.

For example adverbs of time tells us when the action is performed like የግም/today/, ጌሁግም/tomorrow/ ጎሳግም/yesterday/and በጽባ/at morning/. Whereas adverb of place express where the action is performed ዝየ/here/, አፍክ/outside/, ውስጠ/inside/. Generally adverbs in Geez always modify verbs and in any sentences; it comes before and/or after the verbs that are to be modified. Sample adverbs of Geez texts are compiled on appendix.

3.4.4. ADJECTIVES INFLECTION IN GE'EZ

The role of adjectives in any languages are describes or clarifying a noun that means it gives additional information for the nouns. It express physical and other qualities (like large, friendly) and the writer's opinion or attitude (like excellent, beautiful). The adjective residential classifies the area; tell us what type of area it is. It also express other meanings such as origin (an Ethiopian writer), place (an Ethiopian water fall) frequency (a weekly newspaper), degree (a complete failure), necessity (an essential safeguard) and degree of certainty (the probable result).

In geez language, adjectives give more information about people, places and things. According to[22], adjectives tells the size, color, quality, origin, and behaviors of nouns. Adjectives can be categorized in differ ways; some of it are adverbs of place, quality, commands and numerals. Adjectives in Geez are inflected in to various formants by using affixations like other word categories. For example 'ነዊን ወቀይሕ ኢትዮጵያዊ ብእሲ መጽአ' /the tall and red Ethiopian person came/. From this simple sentence the underlined three words are an adjective that gives information about the person. The following table illustrate about the Geez addjectives and its affixes.

Adjectives	Meaning	Suffix
ኢትዮጵያዊ (singular masculine)	Ethiopian	-ዊ
ኢትዮጵያዊት (singular feminine)	Ethiopian	-ዊት
ኢትዮጵያውያን (plural masculine)	Ethiopians	-ውያን
ኢትዮጵያውያት (plural masculine)	Ethiopians	-ውያት
አይሁዳዊ(singular masculine)	Jewish	-ዊ
ከቡራን (plural masculine)	Respectful	-ን
ከቡራት(plural feminine)	Respectful	-ት
አእማርያን (plural masculine)	Knowledgeable	-ያን
አእማርያት (plural feminine)	Knowledgeable	-ያት

Table 17 inflection of adjectives

As the above table illustrates that, adjectives are inflected by number and genders to make an agreement with the nouns. The first fives words are an adjective that are formed from the noun

called ኢትዮጵያ/Ethiopia/ and አይሁ/Jewish/ and the rest four words are formed from a verbs. For instances, the suffixes -ዊ,-ውያን, -ን used for masculine and -ያን and -ዊት, -ውያት, -ት, -ያት used for feminine are attached to the given words to agree with numbers.

3.4.5. DERIVATIONAL AFFIXES OF GEEZ WORDS

On this sub sections the researchers try to compile some derivational morphology of Geez languages as much as possible. Derivational affixes play vital roles to form various word classes for the language. The affixation process creates new words from existing words. Geez verbs mostly can derive others word class such as; noun, adjectives, and adverbs.

3.4.5.1. DERIVATION OF NOUN FROM VERBS

According to [50], Geez language verbal noun can be derived from verbs which have not more than three radicals. To form the noun the first radicals from the left and its follower are changed to six orders and the third alphabet will be changed to first order and finally adding a suffix “t” at the ending positions. For example the nouns ስብከት/an act of teaching/ and ቅትለት/an act of killing/ are derived from the verb /he thought/and /he killed/ respectively. The following table shows us how Geez verbal nouns are formed from verbs.

Verbs	Verbal nouns	prefix	suffix
ሰበከ /he thought/	ስብከት/an act of teaching/	-	-ት
ቀተለ/ he killed/	ቅትለት /an act of killing /	-	-ት
ቀደሰ/ he praised/	መቅደስ/ house of praising/	መ-	-
ምዕደ/ he advised/	ምዕዳን/ advice/	-	-ን
ነበረ/ he sat/	መንበር/ chair/	መ-	-
መሀረ/ he taught/	መምህርት/ teacher/	ት-	-ት

Table 18 Derivation of nouns from verbs

As we can see from the above table, the affixation process varies in different forms depending on the nature of the verb. So some nouns that are derived from Geez verbs have prefixes like መ-, ት- and the suffixes that are attached to the ending are -ት and -ን. Furthermore, in some derived nouns the internal radicals of the word are added in addition to affixations.

3.4.5.2. DERIVATION OF VERB FROM OTHER VERBS

Geez verbs have an ability to create new words that are not similar to the original verb form. According to [46] and [22], a Geez verb can appear in all or some of the following possible verb derived forms using; prefixes (simple past), causative (prefix “አ”), passive reflexive (prefix “ተ”),

if not preceded by a subject prefix otherwise “ተ”) and causative passive (prefix “አስተ”). The following table describes how geez verbs are passive reflexive, causative passive, causative, and it affixations by using the verb መከረ /he advised/ as an example.

መራሐ.	Perfective	Causative	Causative- reciprocal	reflexive	Reciprocal
ውእቱ/3 rd p.s.m/	መከረ	አምከረ	አስተማከረ	ተመከረ	ተማከረ
ይእቲ/3 rd p.s.f/	መከረት	አምከረት	አስተማከረት	ተመከረት	ተማከረት
ውእቶሙ/3 rd p.p.m/	መከሩ	አምከሩ	አስተማከሩ	ተመከሩ	ተማከሩ
ውእቶን/3 rd p.p.f/	መከራ	አምከራ	አስተማከራ	ተመከራ	ተማከራ
አንተ/2 nd p.s.m/	መከርከ	አምከርከ	አስተማከርከ	ተመከርከ	ተማከርከ
አንቲ/2 nd p.s.f/	መከርኪ	አምከርኪ	አስተማከርኪ	ተመከርኪ	ተማከርኪ
አንትሙ/2 nd p.p.m/	መከርከሙ	አምከርከሙ	አስተማከርከሙ	ተመከርከሙ	ተመከርከሙ
አንትን/2 nd p.p.f/	መከርከን	አምከርከን	አስተማከርከን	ተመከርከን	ተማከርከን
አነ/1 st P.s.m/	መከርኩ	አምከርኩ	አስተማከርኩ	ተመከርኩ	ተማከርኩ
ንሕነ/1 st P.p.m/	መከርነ	አምከርነ	አስተማከርነ	ተመከርነ	ተማከርነ

Table 19 derivations of geez verbs

Form the above tables we can see that, affixations are used to form various forms of a verbs from the given verbs. The prefixes አ-, አስተ- and ተ- are used for causative, causative-reciprocal, reflexives and reciprocal. Whereas the suffixes -ት, -ከ, -ኪ, -ከሙ, -ከን, -ኩ and -ነ are used for making agreements with the nouns, genders and numbers.

3.4.5.3. ADJECTIVE DERIVATION

Like derived nouns and verbs adjectives can be derived from Geez verbs. Additionally adjectives can be derived from nouns [22]. The following table illustrates how derived adjectives are formed from noun and verbs.

Verbs and nouns	Derived adjectives	suffix
መሀረ/he taught/	መሀርት/Teachers/	-ት
ጸሐፊ/he wrote/	ጸሐፍት/authors/	-ት
ዘመረ/he sang/	ዘመርት/singers/	-ት
ወንጌል/bible/	ወንጌላዊ/ት/ይ/biblical/	-ዊ/ት/ይ
ሀገር/country/	ሀገራዊ/ት/ይ/coutrys’/	-ዊ/ት/ይ
ሰማይ/sky/	ሰማያዊ/ት/from the sky/	-ዊ/ት

Table 20 Derived adjectives from nouns and verbs

As we can understand from the above table; the first four lines is a verbs with its corresponding derived adjectives and the last three lines show us the noun with its corresponding derived

adjectives. In order to form derived adjectives suffixations are attached to the verb and the noun. The suffix *-ት* is employed to the verb and the suffixes *-ዊ*, *-ዊት*, *-ይ* and *-ይት* are employed to the noun forms. The suffixes *-ዊ* and *-ይ* indicates masculine and the suffix *-ዊት* and *-ይት* indicate that the feminine.

3.4.6. PLURAL OF PLURAL WORDS OF GEEZ

According to [22], plural of plural nouns are common in Geez language phenomenon, so in addition to making plural noun from its singular form; making plural of plural nouns for its plural form are well known. In order to make plural of plural from its plural forms, the suffix *-ት* can be added or attached to the ending positions of the plural nouns and the last alphabets will be changed from 6th order to 4th order. For example /kings/ will be pluralized to /kings/. Adding prefix-suffix is another way ways of constructing double plural noun. The following table demonstrate that, how plural of plurals are formed from its plural forms.

Singular nouns	Plural nouns	Plural of plural	suffix
ደብር/mountain/	አደብር /mountains/	አደብራት /mountains/	-ት
ልብስ/clouse/	አልብስ /clouses/	አልብሳት /clouses/	-ት
ፈለግ/river/	አፍላግ /revers/	አፍላጋት /revers/	-ት
ጾታ/bird/	አዕዋፍ/birds/	አዕዋፋት /birds/	-ት
ከንፍ/wing/	አከናፍ /wings/	አከናፋት /wings/	-ት

Source:[21]

Table 21 plural of plural nouns

As the above table shows that, the plural forms of the nouns are inflected by preceding the prefix *አ-* and changing the first letter of the nouns from 1st order to 6th order the left side. Plural of plurals are made by adding the suffix *-ት* at the end positions and the last alphabets of the nouns are changed from 6th order to 4th order in addition to the preceding the prefix *አ-*.

According to [21] and [23], the earliest literatures of Geez uses plural nouns and plural of plurals differently; a plural form used for making agreements with numbers that indicate the quantity of the nouns specifically two whereas plurals of plural used for making agreements with numbers that indicate the quantity of the nouns particularly more than two correspondingly.

3.4.7. COMPOUNDING WORDS OF GEEZ

As the name indicate that compounding means creating new word forms by combining two independent word forms. Geez language texts are used such mechanism to form another words by combining two different words. For example the word ቤት/house/ +ንጉሥ /king/ can be combined together to form a noun ቤተ-መንግሥት /kings house/ and ዲበ /on/ + ባሕር/river/ can be combined together to form a noun ዲበባሕር /on the river/ respectively.

3.4.8. NEGATION OF GEEZ WORDS

In Geez language the negation markers can be attached with the verbs. According to[20], the common negation markers are አኮ/ako/, ሐሰ/hase/, እንዳኢ and ኢ/e/ in which have an English corresponding meaning “not”. The alphabet ‘ኢ.’ is always attached to perfective verbs from the beginning positions and the word አኮ and ሐሰ can stand alone by preceding the nouns.

3.4.9. PREPOSITIONS AND CONJUNCTIONS (መስተጻምር ወመስተዋደድ)

Like other languages, Geez grammar has its own prepositions and conjunctions. Prepositions are words that show a connection between other words whereas conjunction in Geez used to link sentences in order to join two sentences to make them one, phrases, and clauses. As discussed by [22], [23] Prepositions can be categorized in to three groups. These are “አቢይ አገባብ (conjunctions), ደቂቅ አገባብ and ንዑስ አገባብ”. Some of the prepositions that are grouped under the first categories are እስመ/since/, አምጣነ/because/, አኮኑ/for/, እመ/although/, እስከ/till/ and አመ/ጊዜ/ሰበ/in time/ ወ/and/, አምሳለ/like/ and so on.

On the other hand ውስተ /in/, ምስለ/with/, ዓዲ/or/, እንከ/after/ are grouped under second categories whereas the prepositions like ላዕለ/on/, መልዕልተ/above/, ታሕተ/under/, ዲበ/on/, ውስተ/from/, መንገለ/to/, ኅበ/to/, ማእከለ/between/ are categorized under the third one. According to [62] and[64], the third categories prepositions are widely available in written and verbal forms of Geez language. But most of the time these prepositions are spoken with the noun that comes after it as one word. Samples of prepositions and conjunctions for Geez text are compiled on Appendix III.

3.5. SUMMARY

In this chapter the researchers discussed about the various concepts of the Geez language morphology, particularly word formation process by dividing the formation process into its word classes. Geez language has complex morphology, that why different researchers invest their time

and efforts now day. Geez morphologies are used inflectional and derivational process to form its word variants. In today grammars of any language, it must have at least five word categories namely nouns, verbs, adjectives, adverbs and prepositions. Geez have very rich in those word categories' and it have complex word formations process.

CHAPTER FOUR

DEVELOPMENT OF HYBRID STEMMER FOR GEEZ LANGUAGE

4.1. INTRODUCTION

On the previous chapter the morphology parts of Geez language have been reviewed in detail. As we showed that the main word formation process in the language is done through affixation. Semitic languages like Arabic, Amharic and Geez; the word formation process are not only by affixation (prefixes and suffixes) as English language.

The complex morphological structure of the Geez language, results in a very large number of variants of word forms in the language. In order to design an IR system and other NLP system for the language, it needs a tool for deducing those variant word forms in to one form to improve the performance of the system. This can be achieved by a conflation technique, which is stemming. The main purpose of a stemmer is reducing different variants of words in to their standard form called stem (root). On this chapter the researcher present the design and development of stemmer for Geez language text by using hybrid approach. This chapter deals about the development process of the stemmer, the compilation of stop word and proposing the affixes removal algorithm in details.

4.2. CORPUS PREPARATION

For this study the researcher prepared a sample text for the development of stemmer and evaluating the proposed stemmer. For resourced language like English, there are a lot of standard corpuses for evaluating a newly proposed algorithm. For Geez language there are no any standardized and publicly available corpora like Treebank and propbank for English[32], [65].

Unlike that for Ethiopian language there are no such collections of standard corpora. However preparing balanced corpus is essential for processing natural language tasks and IR system such as stemmer.

For this particular study, we have used our own manually prepared collection from ready available sources radomly. Consequently, the corpus used for the research was compiled from different sources like Geez text book, newspaper and teaching materials of the language so that,

the corpus consists of different variant of words and be representative interms of the number of words. The corpus consists of 8, 662 word types with the total of 13,221 word tokens. From the corpus 14% (1866) words were prepared by the previous researcher [19], whereas 86% (11,355 words) of the corpus were prepared and compiled by the researcher by getting consulted by two language expert. The following table shows the total number of words and it percentage of prepared corpus from each sources[20]–[23], [62], [64], [66], [67].

Name of sources	Total words/tokens	Total words in %
Bible(Plasm)[68]	1,167	8.8%
Newspaper and Megazine [67]	3,108	23.5%
Text books[21], [22], [64], [66]	5,366	40.6%
From Abebe [19]	1,866	14.1%
Others [62], [66]	1,714	13.0%

Table 22 Percentage prepared corpus from selected sources.

In order to split the corpus in to training and testing sets; we have used 80% by 20% mechanisms. Acording to [1], [32], we can use 80% by 20% mechanism for the purpose of splitting prepared courpus into training and testing sets. Due to this reason the researchers try to prepare the training set from the above prepared corpus based on 80% by 20% mechanism. The reason for selecting this mechanism is that, the size of our prepared corpus was sufficient. As a result 10,577(80%) out of the total of 13,221 word tokens ware used as the training set. Whereas the test set is used for testing the newly proposed prototype whether it is successful or not. Therefore; the performance of the proposed stemmer is evaluated by using test data-set in which 2,644(20%) words were used for evaluation purpose out of the total word tokens.

4.3. WORD DISTRIBUTION OF THE CORPUS

Word distribution in sample text documents of a language helps to study language’s behavior and this distribution can be shown using word-ratio (number of distinct words to total number of words). This helps to show how many words are morphologically distributed within a document. As the following table shows that number of words with frequency one are tripled in present than that of number of words with frequency more than one. This shows us the datasets are morphologically distributed with in documents.

Name of text	Total words/tokens	Distinct words	Word ratio in %	%of distinct words with freq. 1	% of distinct words with freq. more than 1
GeezCorpus	13,221	8,662	65.52	77.68	22.32

Table 23 Word distribution of Geez sample dataset

As the above table demonstrate, the percentage of distinct words with frequency one is higher than 1/3rd (one third) of the percentage of distinct words with frequency more than one.

4.4. COMPONENTS OF THE STEMMER/ PROPOSED STEMMER

4.4.1. INTRODUCTION

In this section the researchers discussed in details about the pre-processing of Geez text which is help-full to clean and make the corpus ready for additional processing. The roles of common pre-processing jobs are regularising and configuring input text documents, i.e. the later tasks can be computed without difficulty. In addition to this, proposed algorithm that used to conflate Geez texts are described to show how each and every components are designed and implemented.

4.4.2. TOKENIZATION OF GEEZ TEXT

In natural language processing the term, tokenization is the course of splitting character streams in to tokens. This includes dividing sentences, phrases and paragraphs into a sort of tokens. To this end, the identifications of words are different from one language to other language and most languages use white spaces to separate words which depend on language features. Geez language has different delimiters to bound words in the text in addition to white spaces.

For this study, tokenization was used for splitting the input Geez documents or sentences in to tokens by removing certain characters such as punctuation marks. A consecutive sequence of valid characters was recognized as a word in the tokenization process. For this study tokenization process is responsible for splitting the given input text by using punctuation marks (these are discussed on section-3.2.3 of the previous chapter in detailed) like period (#) etc. and white space characters. Additionally it also replaces more than one white paces character into one white paces character, if any. The following figures describe how Geez text/words are tokenized, see figure 6.



Figure 6 Tokenization process

From the above figure tokenization process takes place by accepting geez text documents or words to produce list/bag of tokenized words for further processing. For example, the two sentences “አማመክ፡ ሰሚዕየ ቦ።” /I heard you were ill/and “ኅሁሰ እፎ፡ ሀሎክ።” /how are you now? / can be tokenized by white space and Geez punctuation marks: and ።. The following algorithm is a simple algorithm that is used for tokenizing Geez text document or corpus. The output of this algorithm is used as an input for normalization component.

Input: Geez text document/one or more words

start

create a string container and replace more than one white space with one white space

For

Read the content of the file and split it in to string by Geez punctuation marks then

put the word in to the container

until end of file

end of for

stop

Output: List of tokenized Geez words

Algorithm 1 Algorithm for Tokenization of Geez text

Algorithm 1 shows us, only Geez words/text is passed to this algorithm and produce list of words which are provided to next component, i.e. normalization component that used as an input.

4.4.3. NORMALIZATION OF THE CORPUS

All control character number and special character are removing from the text before the data is processed. So normalization will do these tasks. For example the number 2013(two thousand thirteen) is written as “ጳጼ፻፲፫” or “ጳጼ፻፲፫” or “እስራ ምዕት አሠርቱ ወሠለስቱ” in Geez. For this study, normalization was used for removing the special character; number and the representation of numbers in the form of alphabets, from tokenized Geez word lists/tokens. A consecutive sequence of valid characters without special character and numbers and/or numbers in its alphabet form; were recognized as regularized word in the normalization process. The following figures describe how tokenized Geez text/words are normalized, see figure 7



Figure 7 Normalization process

to produce list/bag of normalized words for further processing. The following algorithm is a simple algorithm that is used for normalizing Geez text document. The output of this algorithm is used as an input for stop-word removal component for further processing.

Input: tokenized Geez word list

```

start
  read tokenized geez word list line by line
  for word in tokenized word list do
    replace special characters, Geez numbers and Geez numbers in its alphabet
    form with white space and store the remaining word list
  end for
stop
  
```

Output: normalized geez word list

Algorithm 2 Geez character normalization

As algorithm 2 shows us, tokenized Geez word lists are passed to this algorithm and the final results of this component are normalized word-lists which are provided to next component, i.e. stop word removal component that used as an input.

4.4.4. COMPILATION OF STOP WORD LIST

In natural languages stop words are a non-content bearing word which contains prepositions, conjunction, articles, particles and etc. Stop word can be compiled from the corpus text by using rank frequency distribution and by preparing stop word list dictionary. The dictionary is checked for removing the attempt word whether it is stop word or not. The rank frequency distributions of the words, especially to a language which have high morphological complex words are not suitable to control stop words. But the frequency may leads in the selection of stop words [19]. The following table shows us the top ten frequent words from the prepared corpus.

Words	Meaning	Frequency
ውእቱ	is, was	73
እፎ	How	49
እስመ	As	44
ኦ	oh!	44
ውስተ	To	42
አንቲ	You	37
ኅበ	To	36
ሰላም	Peace	34
እግዚአብሔር	God	34

Table 24 Top ten frequent words from the prepared corpus

The above table (table 23) showed that, the rank frequency of the words in the corpus leads to see the stop words but the words ሰላም/peace/ and እግዚአብሔር/God/ are not a stop word for Geez.

According to [22],[19] and [21], the common stop word of Geez language include articles, infinitives, verb to be, demonstrative adjective, pronouns and prepositions. For the sake of this study we have prepared a stop word list dictionary to catch and remove stop words from the sample datasets that contains a total of 756 stop-words. The following are some of the stop word list, the complete list of the stop word were presented on APPENDIX-I.

Type	Stop word	Meaning
Preposition	እም	From
	እምነ	From
	ጊዜ	
	ማእከለ	Between
	ዋስጤ	Within
	ውስተ	In
	መትህተ	‘to---down’
	ታህተ	Down
	ላእለ	with---on
Demonstrative adj.	ዲቦ	On
	ዝንቱ	‘this’
	ዛቲ	This
	እሉ	These
	እላ	These
	ዝክቱ	That
Verb to be	እልክቱ	Those
	አነ	am, was
	አንተ	are, were
	አንቲ	are, were
	ውእቱ	is, was

Table 25 Sample of Geez stop word list

For Geez language text, the researchers proposed a stop word removal algorithm to remove non-content bearing words for the text. These words are most frequently occurred words on Geez documents. The algorithm tries to remove such words; the output of this algorithm is used as an input for further processing that is shown in figure below.

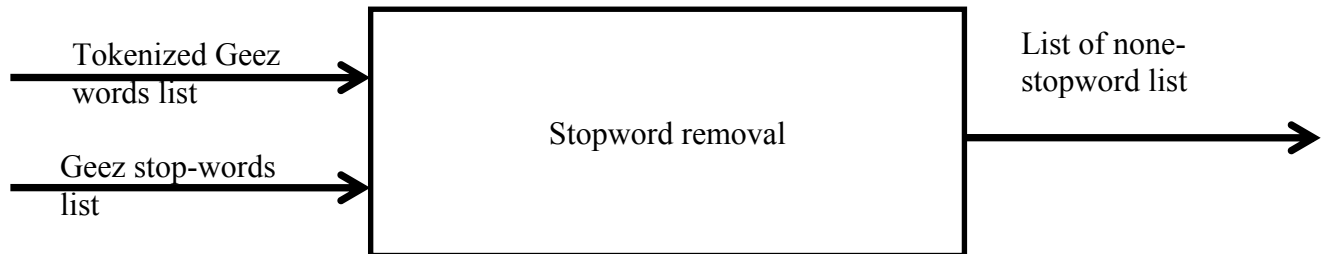


Figure 8 Stop word removal process

Based on the above figure list of compiled stop words are compared to the input text to filter out non-stop word lists. The researcher proposed an algorithm to do this as follows:

Input: List of stop word and List of normalized word.

Start

Read list of stop word and Geez normalized text

for each word in list of normalized word

do

if word length of normalized word equal with list of stop-word length

then

if word is in list of stop-word then

continue

else

add to non-stopword list

end of if

end of if

end of for

stop

Output: bag of non-stop-word list

Algorithm 3 Removing stop words

The above algorithm is responsible for doing two tasks. Firstly it accepts Geez normalized word and list of stop word from dictionary. Secondly it checks each normalized word-list with compiled stop-word lists based on its length. Considering the length of normalized word-lists can minimize the time taken to search a given word in a list of stopwords. Then it checks the

existence of the word in stop word list. Finally the non-existed words in stop-word lists are returned for further process.

4.4.5. COMPILATION OF GEEZ AFFIXES

English language which is less morphologically rich and suffixation is the common building block of its morphology. As a result suffix stripping algorithm/stemmer are fairly well just removing the suffix to come up with best result. In contrast Geez language is one of morphologically rich language; due to this reason the morphology of this language is build up from prefixes, suffixes, prefixes-suffixes and rarely infixes. Without removing all the affixes Geez text; the stemmers cannot be effective and efficient. In this study the following models (GHSM) describes the all over process of the proposed hybrid stemmer. The models have mainly five modules. These modules are discussed as follows:

A. Input Module

This module is responsible for accepting Geez language texts such as sentences, paragraphs and/or list of words in the form of text file. After accepting such text file, it gives to the next module for further processing namely preprocessing module.

B. Preprocessing Module

In order to get valuable and preprocessed data from unstructured plain text, the study assumed necessary preprocessing steps. This module undertakes three processes which are tokenization, normalization and stop words removal. These processes are presented in detailed on the section above. As a result, the output of the module is preprocessed word lists.

C. Data Splitting Module

The word lists which are preprocessed by the preprocessing modules are divided/splited/ as training and testing dataset. The training data (80%) used to train the models and the remaining 20% of the corpus is used for testing the proposed stemmers and making evaluations.

D. Stemmer Module

This module is accountable for accepting preprocessed input data and stemming the input based on the specified rules. It contains two individual components which are, affixes removal components and character n-gram components. Affixes removal components are also further divided into prefixes, suffixes and infixes removal components. The order of affixes removal follows prefixes, suffixes and infixes respectively. Prefixes removal process works with the help of the prefixes list compiled for this purpose in order to remove the prefixes from the input.

On the other hand the suffixes removal process is used for removing the suffixes from the input text by considering the compiled suffixes list. Infixes process also used to remove the infix from the input text. In addition to this, individual processes have its recording rule that used to check the validity of stemmed words after each steps.

After removing the affixes of the given input, the stemmer modules also compute the character N-gram for those input words which are not touched and examined by affixes removal components. The final output of this module is given to the next module.

E. Output Module

This module is responsible for accepting the output of the stemmer module and displaying the final stemmed words or words list to the proposed user interface. Generally the following figure shows that, the generalized over views of the proposed models and its flows with the above five modules. See figure 9 below.

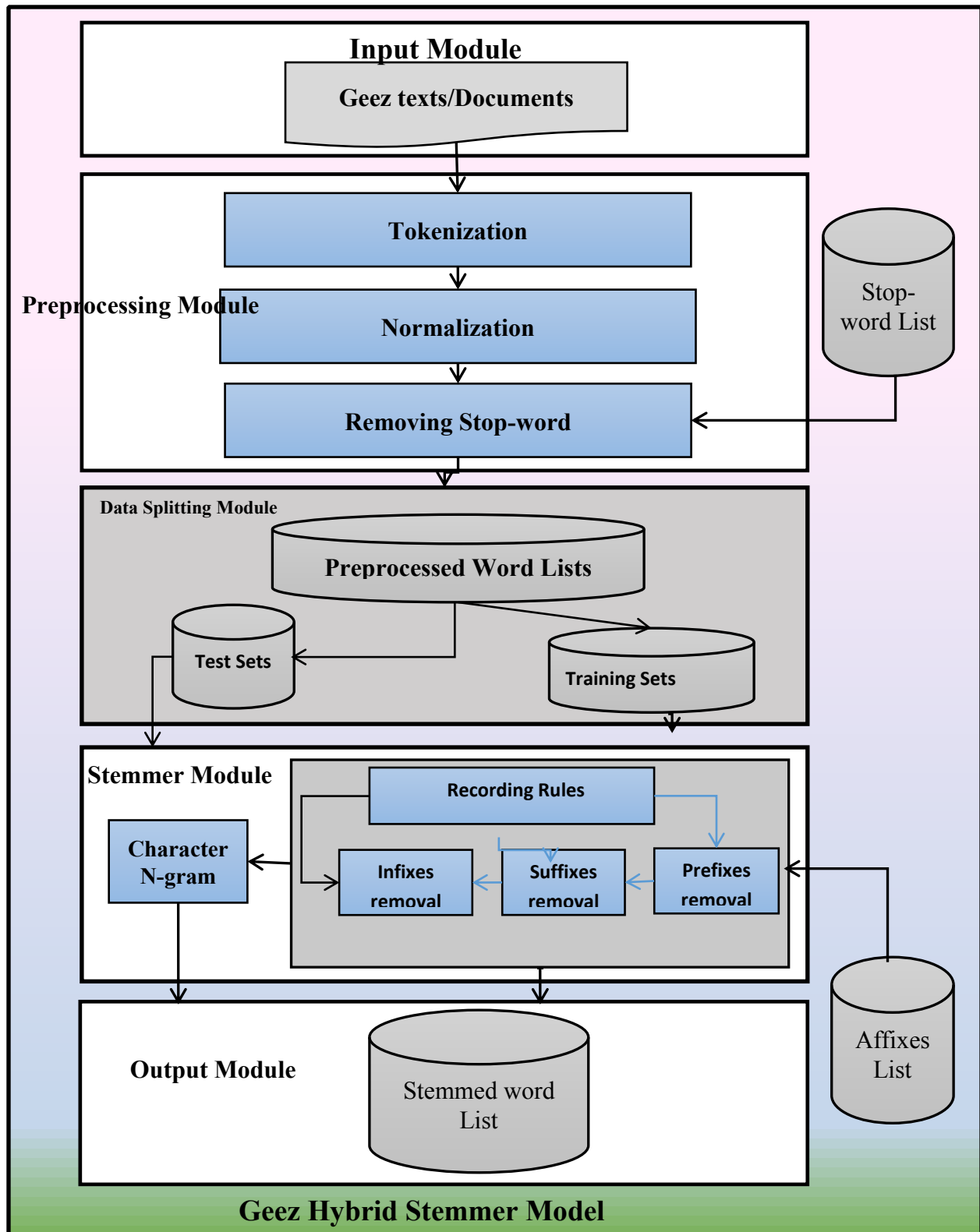


Figure 9 Geez Hybrid Stemmer Model (GHSM)

4.4.5.1. PROPOSED RULE SET FOR REMOVING AFFEXES

The researchers tried to compile a set of rules for catching an affixes individually. In this part we have discussed a set of rules that are used for stripping prefixes and suffixes for Geez text. For stripping all affixes we start from the longest affixes up to single affixes like “ወዘኢይት-/-ክምቀሙ” that have five and four radicals and “ነ-/-ሁ” which have one radicals respectively. In Geez language, the common lengths of the affixes that attached to the stems are, either of three or two. There are a few affixes with length four and five. For the sake of this study, the rules are compiled and presented as follows:

Proposed rules for removing prefixes:

Rule-1: if words start with (ወዘኢይት|ወኢይትት|ዘኢይትት|ወዘኢትት) then

Remove the matching prefix

Rule-2: if words start with

(ለአስተ|ወአስተ|ዘኢይት|ዘናስተ|ዘያስተ|በአስተ|ወዘኢይ|ወኢትት|ወናስተ|ወመስተ|ለዘይት|ወኢይት) then remove the matching prefix

Rule-3: if words start with

(እምዘ|እምነ|በአስ|ዘኢይ|ዘይት|በዘይ|እምዘ|ለዘኢ|መስተ|ወኢት|እምኢ|ዘኢተ|ዘንት|ወኢይ|ኢይት|ለዘያ|ለዘይ|ኢያስ|ወእም|ዘትት|ያስተ|አስተ|ኢትት|ወለዘ|ለዘበ|ለዘአ|ለይት|እምበ|ወበመ|ወትት|ወዘይ|ወዘኢ|በዘአ|ወእት) then: remove the matching prefix

Rule-4: if words start with

(ይት|አን|አስ|እም|ዘተ|ወይ|ወለ|ወየ|ለዘ|ኢይ|ወኢ|ዘይ|በተ|በዘ|ወበ|ወዘ|ዘኢ|ኢተ|ዘያ|ዘን|ወን|ዘየ|ወተ|ዘት|በበ|ለለ|ለዘ|ዘበ|ወወ|እት|ትት|ወይ|ኢየ|ወያ|ወእ|ወታ|ወመ|ለይ|ለአ)

Remove the corresponding prefix

Rule-5: if words start with

(አ|ኡ|ኢ|መ|ሙ|ሚ|ተ|ቱ|ቲ|ታ|ት|አ|እ|የ|ያ|ይ|ነ|ና|ን|ወ|ነ|በ|ዘ|ለ|ም)

Remove the matching prefix

Proposed rules for removing suffixes:

Rule-1: if a word ends with (ውንቲክሙ|ውንቲክን|ውንቲሆሙ|ውንቲሆን) then remove the matching suffix.

Rule-2: if a word ends with

(ከምዎሙ|ከምዎን|ከናሆሙ|ከናሆን|ያኒክሙ|ያኒክን|ያኒሆሙ|ያኒሆን|ያቲሆሙ|ያቲሆን|ውንቲክ|ውንቲሃ|ውንትኪ|ዋቲሃኒ|ዋቲሆሙ|ዋቲሆን) then remove the matching suffix.

Rule-3: if a word ends with

(ውያን|ውያት|ትክን|ትክሙ|ከምዎ|ከምዋ|ከሙኒ|ከሙነ|ኪዮሙ|ኪዮን|ከናሃ|ከናኒ|ኪናነ|ከዎሙ|ከክሙ|ከክን|ናሆሙ|ናክሙ|ያኒክ|ያንክ|ንክሙ|ኒክሙ|ንክን|ኒሆን|ያኒየ|ያኒነ|ያኖሙ|ያኖን|ያንነ|ያትኪ|ያቶን|ቲሆን|ያትያ|ያቲየ|ያትነ|ያንኪ|ተክሙ|ተክን|ውያት|ውያን|ቲሆሙ|ያቲሆ|ትየሰ|ቲክሙ|ቲሆኒ|ሆሙኒ|ኒሃኒ|ናቲሃ|ውንተ|ውንት|ያኒሃ|ያኒከ|ያኒሆ) then remove the matching suffix

Rule-4: if a word ends with

(ተኒ|ያት|ሆሙ|ሆን|ክን|ዋት|ተነ|ተከ|ተኪ|ቶሙ|ቶን|አት|አን|ከሙ|ዊት|ዋን|ያን|ዎሙ|ትየ|ትነ|ኮን|ዎን|ተኒ|ኪዮ|ዮሙ|ዮን|ኪያ|ኪኒ|ከሆ|ከዎ|ከክ|ከኪ|ኪዋ|ናሆ|ናሃ|ናኪ|ትክ|ትኪ|ኒሆ|ያነ|ኒክ|ንየ|ኒየ|ኒነ|ንነ|ያኑ|ኖሙ|ያና|ኖን|ቲሃ|ቲየ|ንክ|ንኪ|ሆኒ|ከሙ|ሆሙ|ሆን|ሙነ|ሙኒ|ቲነ|ከኒ|ታተ|ታት|ናት|ኮሙ|የኑ|ትሰ|ትኒ|የኒ|ዊተ) then remove the matching suffix

Rule-5: if a word ends with (ሁ|ይ|ነ|የ|ዎ|የ|ዋ|ዊ|ከ|ከ|ኪ|ነ|ኒ|ና|ን|ቱ|ታ|ት|ቶ|ሃ|ሁ|ሙ|ሂ) then remove the matching suffix

The above rule sets are compiled for the purpose of removing prefixes, suffixes and also prefix-suffixes from a given word variants aiming to produce the stem. Finally, removing the matching affixes from the give input text always consider some exceptional cases. For considering these exceptions, exceptional rules are compiled as follows:

A. Exceptional rules for removing prefixes

1. It word length=2 don't apply rules and take it as a stem.
2. If the word starts with a letter “ተ”, and followed by “ን” don't remove the prefix.
3. If the word length=3 and starts with a letter “ወ” followed by “ን” don't remove the prefix “ወ”.
4. If the word length=3 and starts with a letter “መ” don't remove the prefix “መ”.
5. If the word starts with a letter “አ” and followed by “ፍ” and “ፋ” don't remove the prefix.

6. If the word starts with a letter “ኢት” and “ኢየ” followed by “የ” and “ኛ” don’t remove the prefix.

B. Exceptional rules for removing suffixes

1. If word length=3 after stripping prefixes don’t apply suffix stripping process and check if any corresponding recording rule match.
2. If the word ends with a letter “ል” preceded by “ኔ” don’t change the letter “ል” to “ለ” at recording stage.
3. If the word length=3 and ends with a letter “ይ”, don’t remove the suffixes and don’t change the letter “ይ” to “የ” at recording stage.
4. If the word length<=3 and ends with a letter “ት” don’t remove the suffixes and don’t change it to “ተ” at the recording stage.
5. If the word length=3 and ends with a letter “ይ”, don’t remove the suffixes and don’t change it to “የ” at recording stage.
6. If the word length=3 and ends with a letter “የ”, don’t remove the suffixes and don’t change it to “ይ” at recording stage.

C. Exceptional rules Infixes

1. If the word contains “ው”, “የ” and its variations at the middle of the word and its length=3 remove these letters and its variation.
2. After removing the infixes term, if the word start with “መ”, “ሠ” and “ቀ” change the give the letter to “ሞ”, “ሠ” and “ቆ” respectively.
3. After removing the infixes term, if the word ends with the 2nd, 3rd, 4th, 5th, 6th and 7th order change of the next letter to 1st order.

4.4.5.2. COMPILATION OF PREFIXES

In Geez language prefixes are the common mechanism to inflected words variants. The most common prefixes for the language are; መ-, ማ-, በ-, ለ-, ወ-, እ-, ም-, የ-, ይ-, ወዘ-, ወይ-, ወለ-, ለዘ-, ዘበ-, ወበ-, ወኢ-, ወወ-, እም-, ወዘበ-, ዘእም-, አስተ- and ናስተ- etc. the following table shows a sample of Geez prefix.

One radical	Two radical	three radical	Four radical	Five radical
ት-	ዘየ-	በአሰ-	ለአሰተ-	ወዘኢይት-
ለ-	ወተ-	ዘኢይ-	ወአሰተ-	ወዘኢትት-
እ-	እት-	ዘይት-	ዘኢይት-	ወዘኢትት-
የ-	በበ	በዘይ-	ዘናሰተ-	ወኢይትት-
ዘ-	ለለ	እምዘ-	ዘያሰተ-	

Table 26 a samples Geez compiled Prefixes

As table 25 shows, the prefix length for Geez is ranging from one radical up to five radical. As a result the minimum length of prefixes is one and the maximum length of prefixes is five. The complete lists of compiled prefixes are on appendix-II.

For the purpose of this study the researchers have first compiled list of prefixes from different sources based on the grammatical function of individual words with in the document collection. After doing this task the researcher try to develop an algorithm that used to remove/ strip prefix from the given input text by comparing with the compiled Geez prefixes list. This component of the proposed algorithm used nonstop word list as an input which are produced from stop-word removal component. Generally 112 lists of prefixes are assembled for removing prefixes from the given input text. See algorithm 4

4.4.5.3. COMPILATION OF SUFFIXES

Before developing the algorithm for striping the suffixes, we have compiled suffixes lists. To this end the same method is used as that of used to compile the list of prefixes is used to develop the list of suffixes. The most common prefixes for the language are -የ, -ኪ, -ካ, -ክሙ, -ክን, -ን, -ሆሙ, -ሁ, -ሃ, -ሙ, -ን, -ያን, -አን, -ያት, -አት, -ት, -ያት, -ዋት, -ውያ, -አም, -ከ, -ከ, -አ, -አ, -ከሙ, -አት, -ይ, -ያ, -ዊ, -ው, -ና, -ም, -ል and etc. most of the suffixes are found in the form of combination to other suffixes.

Table 27 below shows us a sample of suffixes with their combinations.

One radical	Two radical	three radical	Four radical
-ከ	-ከሙ	-ከከሙ	-ያኒከሙ
-ከ	-ሆን	-ከከን	-ከምዎን
-ቶ	-ሁኒ	-ያቶን	-ከናሆሙ
-ካ	-ንከ	-ንከን	-ያኒሆሙ
-ዋ	-ቲሃ	-ናከሙ	-ያቲሆሙ

Table 27 samples Geez compiled suffixes

As table 27 shows, the suffix length for Geez is ranging from one radical up to four radical. As a result the minimum length of suffix is one and the maximum length of prefixes is four. The complete lists of compiled suffix are on appendix-III.

For the drive or determination of this study, the researchers have first compiled list of suffixes like prefix list from different sources based on the grammatical function of individual words with in the document collection. After the achievement of this task the researcher try to develop an algorithm that used to remove/ strip suffixes from the given input text by comparing with the compiled Geez suffixes. This component of the proposed algorithm used nonstop word list as an input which are produced from stop-word removal component. In general 169 lists of suffixes are assembled for removing suffixes from the given input text.

Input: Geez nonstop-word list

Step-1: read normalized geez word-list line by line

 If the word length ≤ 3 then
 go to step-2
 else
 Store the word on temporary variable
 go to step-3

Step-2: check the word with recording rules

 if match found then
 apply recording rule and return the word
 go to step-6
 else
 return the word and go to step-7

Step-3: Read prefix-list and check the word with the prefix lists

 If the word starts with prefixes list then
 remove prefix and return the remaining terms
 and go to step-4
 else
 return the word and go to step-4

Step-4: read suffix-list and check the word with the suffix lists

 If the returned word length > 3 from step-3 then
 Go to step-5
 else
 return the word and
 go to step-5

Step-5: if words ends with suffix list and match exist

 remove the matching suffix and
 check the remaining terms with recording rules
 return the terms and go to step-6
else
 return the word and

go to step-7

Step-6: If word length = 3 and check the infix

if any infix found then

apply recording rule return the terms

go to step-7

else

return the word and

go to step-7

Step-7: If end of file not reached

go to step 1

Else

Return the stemmed words and

End up process

Output: Stemmed word

Algorithm 4 Geez Affixes stripping algorithm

The above algorithm (Algorithm 4) shows us, the elimination processes of affixes from the input of Geez text.

4.4.5.4. HYBRID STEMMER ALGORITHM

It is difficult to come up with a system that conflates variation of words for a language by the help of the rules only especially for morphologically rich and under-resourced language like Geez. The option is that using a combined version of rule based and statistical approach. In this study the researchers tried to develop a hybrid version of the stemmer by using set rules and statistical approach namely n-gram due to this reason. The hybrid version of the proposed stemmer is responsible for conflating variant words that cannot handled by the first version (affixes stripping only) conflating variant words. The following algorithm illustrates that the integrated version of the stemmer algorithm (see algorithm-5).

Input: Geez nonstop-word list

Step-1: read normalized geez word-list line by line

 If the word length ≤ 3 then

 go to step-2

 else

 return the word

 go to step-3

Step-2: check the word with recording rules and

 if match found then

 apply recording rule and return the word

 go to step-6

 else

 return the word and go to step-8

Step-3: Read prefix-list and check the word with the prefix lists

 If the word starts with prefixes list then

 remove prefix and return the remaining terms

 go to step-4

 else

 return the word and go to step-4

Step-4: read suffix-list and check the word with the suffix lists

 If the returned word length > 3 from step-3 then

 Go to step-5

 else

 return the word and

 go to step-6

Step-5: if words ends with suffix list and match exist then

 remove the matching suffix and

 check the remaining terms with recording rules

 return the terms and go to step-6

 else

 return the word and

 go to step-7

Step-6: If word length = 3 check the infix

 if any infix found then

 apply recording rule return the terms

 go to step-8

 else

 return the word and

 go to step-8

Step-7:if there is no matching rule then

 If word length ≥ 5

 compute quad-gram and

 return the quad-gram and

 go to step-8

 else

 compute tri-gram and

```
        return the tri-gram and
        go to step-8
Step-8 If end of file not reached
        go to step 1
Else
    Return the stemmed words and
    End up process
```

Output: Stemmed word

Algorithm 5 Geez Hybrid Stemming algorithm

Algorithm 5 shows that, the input Geez text cannot handle by the affixes stripping rules nor if there weren't any matched affixes list found from the rule set, n-gram is computed or calculated accordingly. According to [20]–[23] and [64], the most common word lengths of Geez without attaching affixes are three and four. In order to compute quad-gram (4-gram), the length of the given word must be greater than or equal to five. Other ways if the length of the word is less than five, most probably the stem of the word lays on the first three terms; that way we are using tri-gram (3-gram) as option for producing a stemmed word form from the given input text.

4.5. SUMMARY

In this chapter, the researchers have discussed in detail about the developments of hybrid stemmers and its main components namely tokenization, normalization, and complication of stop words and affixes. Additionally we tried to discuss all the algorithms that are necessary for the pre-processing of texts and removing affixes are presented. The proposed set rules that are helpful for removing all the affixes are also presented. Finally we have developed a user interface prototype that is responsible for stemming Geez text based on the give input text with the help of the proposed algorithms and rule sets.

Next chapter focused on presenting the experimentation of results and evaluations of the proposed system to come up a final conclusion and recommendation.

CHAPTER FIVE

EXPERIMENTAL RESULT AND DISCUSSION

5.1. INTRODUCTION

On this chapter the researchers tried to discuss about implementations in order to see the all over performance of the proposed system. In general a series of experiments is conducted in order to assess the quality of the stemming application. Additionally, the tools and environments that are used to implement the designed algorithm and the experiment that is conducted to demonstrate the accuracy have been presented. The result of the experiment would be deduced in this part and the system is evaluated by using evaluation method and parameters that are presented in chapter two. Error counting mechanism was used to evaluate the performance of the proposed approach. Which was made by counting correctly stemmed and incorrectly stemmed words. Additionally coFinally discussion could be taken to come up with conclusion and recommendation.

5.2. TOOLS AND DEVELOPMENT ENVIRONMENT

In order to implement the proposed designed algorithm, we have used different tools and development environments. Based on the designed algorithm, the researchers developed a prototype to evaluate the performance of the proposed system. The compiled data from Geez unstructured texts are then stored in the file for further use by researchers. To take the input texts and display the stemmed word of given input text, Geez hybrid stemmer prototype interface was developed using IntelliJ idea community edition for coding environment with Ubuntu operating system.

Notepad++ also, plays an energetic role to develop a decent system by editing and changing unnecessary and invalid words during corpus preparation that does not process automatically by java programs. It helps us to modify and correct the spelling variation and errors. The corpus processed and organized manually in proper manner with linguistic expert to create clear and understandable character features. Other tool that we have used for editing and preparing the documentation part of this study was Microsoft Office 2010 version. Additionally Mendeley Desktop also, used for preparing the referencing and citing purpose.

5.3. USER INTERFACE PROTOTYPE

Based on the designed algorithm from the previous chapter, the researchers developed a prototype to evaluate the performance of the proposed system. The compiled data from Geez unstructured texts are then stored in the file for further use of this study. The prototype is developed by using IntelliJ idea community edition 2021.0 versions. In the following figure, we tried to show the snippet of the user interface prototype.

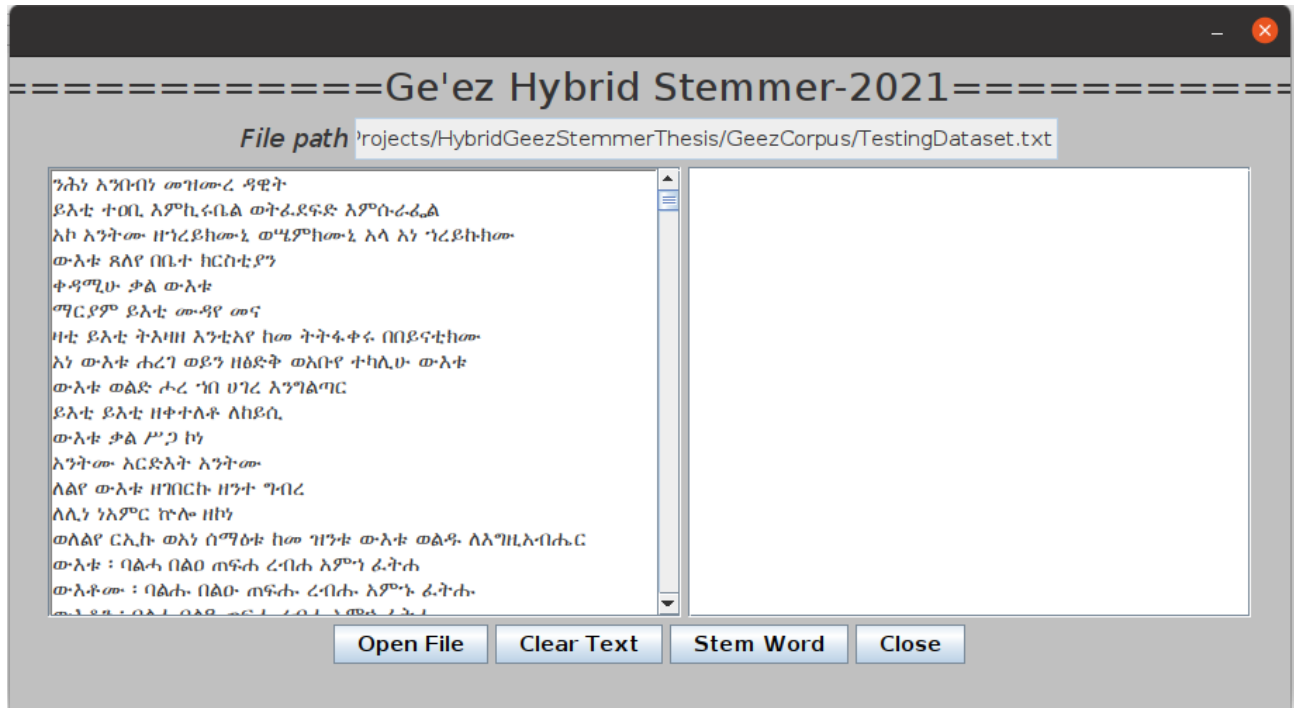


Figure 10 User interface prototype of Geez Stemmer

As shown in figure 10 above, the prototype interface presented information along three different output areas. These components are:

- **The File path area:** that shows the paths of test and stemmed Geez text file from where it is found or opened.
- **The Test File content Area:** An area where the text content associated with the test document is loaded when file is selected by the user with the file selection button called “Open File” located on the bottom of Prototype interface.

- **The Stemmed File content Area:** An area where the text content associated with the stemmed text is loaded when stemmed file is selected by the user with the file selection button located called “Stem Word” on the bottom of prototype interface.

5.4. IMPLEMENTATION OF THE STEMMER

The proposed stemmer is implemented using Java programming as discussed on section 5.2 above. As much as possible the researcher has collected all the stop words and affixes of the language from different sources like [20]–[23], [62]. The algorithms that are proposed, uses rules for removing single and concatenated affixes. By using this set of affixes, all conceivable combinations of the affixes are created and the correct ones are carefully chosen to form the rules.

Before applying the rules that are helpful for conflating all the word variants; the tokenization, normalization and stop word removal were done. We have discussed about it on chapter four under section 4.3.2 up to 4.3.4 respectively. Generally ten rules are configured for removing the affixes of Geez text. The first five rules deal with the prefixes stripping purpose and the next five rules are answerable for removing the suffixes of the text. The designed rules always start from longest or concatenated affixes and continue to single affixes. Even the longest match is considered, the exceptional cases are applied before removing the matching prefixes.

To demonstrate how the proposed algorithm works, the following examples are used. The words $\lambda\phi\tau\Delta$, $\lambda\phi\tau\Delta$, $\tau\phi\tau\Delta$, $\tau\phi\tau\Delta$, $\eta\phi\tau\Delta$, $\rho\phi\tau\Delta$, $\rho\phi\tau\Delta$ are variants of a word with single prefixes attached before the first/the left side of the individual words from its stem $\phi\tau\Delta$. The variations of the words are due to the prefixes λ , τ , η and ρ that are attached to the stem.

The same process is executed for all the other words with prefixes. If a single word contains a possible combination of prefixes with common starting substring, the algorithm only removes the longest conceivable prefix to evade errors in stemming the word. The following table demonstrates that, how variant words are stripped by using affixes removal algorithm.

No.	Word variants	Prefixes	Stemmed Word
1	ወአስተዳለወ	ወ-	አስተዳለወ
2	አስተዳለወ	ወአስ	ተዳለወ
3	አስተዳለወ	ወአስተ-	ዳለወ
4	ዳለወ	-	ደለወ (ዳ changed to ደ)

Table 28 Sample of Geez prefixes removal

As table 27 illustrated, the word variants are coming from the prefixes ‘ወ-’, ‘ወአስ-’, and ‘ወአስተ-’ and after removing the longest prefix ‘ወአስተ-’ the final result is checked by using recording rules (ዳ changed to ደ) i.e. the correct stemmed word form after removing prefixes is ‘ደለወ’.

Likewise the suffixes removal process follows the process of prefixes removal. The difference of the two is that, the type of the substring that is attached from the beginning (prefixes), in the case of prefixes removal and to the ending part of the original/stem words, in the case of suffixes removals. As a result, from the ten rule sets, the last five rule sets are responsible for stripping the suffixes.

To demonstrate how the proposed algorithm works, the following examples are used. The words ደሎትከሙ, ደሎትነ, ደሎትየ, ደሎትከ, ደሎትከን, ደሎቶሙ, ደሎቶን, ደሎታ and ደሎቱ are a variant/inflected words with a suffixes attached after the last/the end part of the individual words from its stem ደሎት. The variations of the words are due to the suffixes /-ከሙ/, /-ነ/, /-የ/, /-ከ/, and /-ከን/, /-ሙ/ and /-ን/ are attached to the inflected words respectively in addition to this ደሎታ and ደሎቱ haven’t external suffixes as a result, the change are made by using recording rules since the word length of the two words are three i.e. the last letter “ታ” and “ቱ” are changed to “ት”.

The same process is executed for all the other words with suffixes removal. If a single word contains a possible combination of suffixes with common starting substring, the algorithm only removes the longest conceivable suffixes to evade errors in stemming the word. The following table demonstrates that, how variant words are stripped by using affixes removal algorithm. See table 29 below.

No	Word variants	suffixes and changes	Stemmed Word
1	አእመርኖ	-ኖ and changes ር to ረ	አእመረ
2	አእመርናሁ	-ናሁ and changes ር to ረ	አእመረ
3	አእመርኖሙ	-ኖሙ and changes ር to ረ	አእመረ
4	አእመርናሆሙ	-ናሆሙ and changes ር to ረ	አእመረ
5	አእመርኖን	-ኖን and changes ር to ረ	አእመረ
6	አእመርናሆን	-ናሆን and changes ር to ረ	አእመረ
7	አእመርና	-ና and changes ር to ረ	አእመረ
7	አእመርናሃ	-ናሃ and changes ር to ረ	አእመረ
9	አእመርናከ	-ናከ and changes ር to ረ	አእመረ
10	አእመርናኪ	-ናኪ and changes ር to ረ	አእመረ
11	አእመርናከሙ	-ናከሙ and changes ር to ረ	አእመረ
12	አእመርናከን	-ከን and changes ር to ረ	አእመረ

Table 29 Sample suffixes removal

As table 29 illustrated that, the word variations are coming from the suffixes ‘-ኖ’, ‘-ናሁ’, ‘-ኖሙ’, ‘-ናሆሙ’, ‘-ኖን’, ‘-ናሆን’, ‘-ና’, ‘-ናሃ’, and ‘-ናከ’, ‘ናኪ’, ‘ናከሙ’, ‘ከን’ and the stemmed word forms after removing the suffixes and making changes at the last letter of the remaining terms is ‘አእመረ’.

On the other hand prefixes-suffixes pairs are conflated by following the same process like prefixes and suffixes. After normalizing Geez text and removing stop-words and the algorithm also strip prefixes-suffixes pairs respectively.

No	Word variants	Prefixes	Suffixes	Stemmed Word
1	ለዘሐወጸነ	ለዘ-	-ነ	ሐወጸ
2	ለዘሐወጸኒ	ለዘ-	-ኒ	ሐወጸ
3	ዘሐወጸነ	ዘ-	-ነ	ሐወጸ
4	ዘሐወጸኒ	ዘ-	-ኒ	ሐወጸ
5	ለዘሐወጸኪ	ለዘ-	-ኪ	ሐወጸ
6	ለዘሐወጸከ	ለዘ-	-ከ	ሐወጸ
7	ወሐወጸከ	ወ-	-ከ	ሐወጸ
7	ዘሐወጸከ	ዘ-	-ከ	ሐወጸ
9	ዘሐወጸኪ	ዘ-	-ኪ	ሐወጸ

Table 30 Sample Prefixes-Suffixes pair removal

As table 30 demonstrated that, the above nine variant words are due to the variations of the prefixes suffixes pairs like ‘ለዘ- and -ነ’, ‘ለዘ- and -ኒ’, ‘ዘ- and -ነ’, ‘ዘ- and -ኒ’, ‘ለዘ- and -ኪ’, ‘ለዘ- and -ከ’, ‘ወ- and -ከ’, ‘ዘ- and -ከ’ and ‘ለዘ- and -ኪ’. The proper stem of the above word after removing the prefixes-suffixes pairs is ‘ሐወጸ’. Most of the words variations of Geez are inflected by prefixes-suffixes pairs.

The same function is executed for all Geez words. If a single word contain a possible combinations of prefixes-suffixes pairs with a common substring, the calculations were expels the possible longest prefixes suffixes pairs in order to remove the correct affixes and to reduce the errors for the purpose of achieving the correct stems.

Furthermore, the infixes removal process follows the same process like prefixes and suffixes removal. The process of removing infixes conducted always after removing other affixes. Table 31 demonstrates about removing infixes removal process.

No.	Words	Prefixes	infixes	suffixes	Same changes on terms	Stemmed Word
1	አብያጸከ	አ-	-ያ-	-ከ	ብ to ቢ and ጸ to ጽ	ቢጽ
2	አብያጸከሙ	አ-	-ያ-	-ከሙ	ብ to ቢ and ጸ to ጽ	ቢጽ
3	አብያጸከን	አ-	-ያ-	-ከን	ብ to ቢ and ጸ to ጽ	ቢጽ
4	አብያጸሁ	አ-	-ያ-	-ሁ	ብ to ቢ and ጸ to ጽ	ቢጽ
5	አብያጸሃ	አ-	-ያ-	-ሃ	ብ to ቢ and ጸ to ጽ	ቢጽ
6	አብያጸነ	አ-	-ያ-	-ነ	ብ to ቢ and ጸ to ጽ	ቢጽ
7	አብያጸሆሙ	አ-	-ያ-	-ሆሙ	ብ to ቢ and ጸ to ጽ	ቢጽ
7	አብያጸሆን	አ-	-ያ-	-ሆን	ብ to ቢ and ጸ to ጽ	ቢጽ
9	አብያጸየ	አ-	-ያ-	-የ	ብ to ቢ and ጸ to ጽ	ቢጽ

Table 31 Sample infixes removal process

As we have illustrated in table 31, the proposed stemmer can remove prefixes, suffixes and infixes and it may apply the recording rules if it is necessary to the final stemmed terms in order to get the correct stem word forms.

5.5. PROPOSED STEMMER EVALUATION AND RESULTS

There are different criteria for judging stemmer as discussed on chapter two of this study, under section 2.3. These criteria are retrieval competency, correctness and compression performance. Over stemming and under stemming are the two issues in which a stemming process can be incorrect. The first one were occurred when the stemming process removes or conflates too much parts of the stemmed word whereas; the second one were occurred when the stemming processes removes or conflates too little parts of stemmed words. Both of them decrease the performance or quality of the stemmer. As a result a good stemmer should increase its performance by reducing the occurrence of over and under stemming problems.

5.5.1. EVALUATION OF THE PROPOSED STEMMER

To evaluate the proposed stemmer, first the test data is prepared by gathering different Geez text documents as we discussed on chapter four under section 4.2. The next step is labeling text documents manually for testing purpose. Evaluation of the stemmer is done with the evaluation parameter that compares the number of words which are stemmed correctly and incorrectly. Normally, the comparison is done with the expert stemmed words. As stated in details about description of the evaluation metrics in chapter two, for the aim of evaluation of the proposed system, error counting approach is used in this report to evaluate the algorithm in terms of the number of accurately conflated results. For analysis, the number of correctly and incorrectly conflated words is counted.

The evaluation takes place in two versions. The first version is evaluating the proposed system in which, the affixes removal only independently employed and the second version were employed by the combination of the affixes removal with the statistical one, namely n-gram. The output of the stemmer was then compared to the expected valid stem. Geez Language experts count the valid and invalid conflated terms. These errors were then classified as under stemmed and over stemmed. When too much of the term is removed, over-stemming, and under-stemming occurs when too little of the term is removed.

Although the compression ratio can be used as a general indicator of the stemming algorithm's effectiveness, other evaluation measures are required to tell specific error patterns. In this case the proposed stemmer is applied to a test set of 2,644 words that are assumed to address a variety of issues as discussed in section 4.2. The literature from which the stemmer's rules were derived was completely different from the test dataset. This was done for the purpose of predicting the stemmer's performance in real-world data. The following figure shows the screen shot of output of the given testing dataset on the first version stemmer.

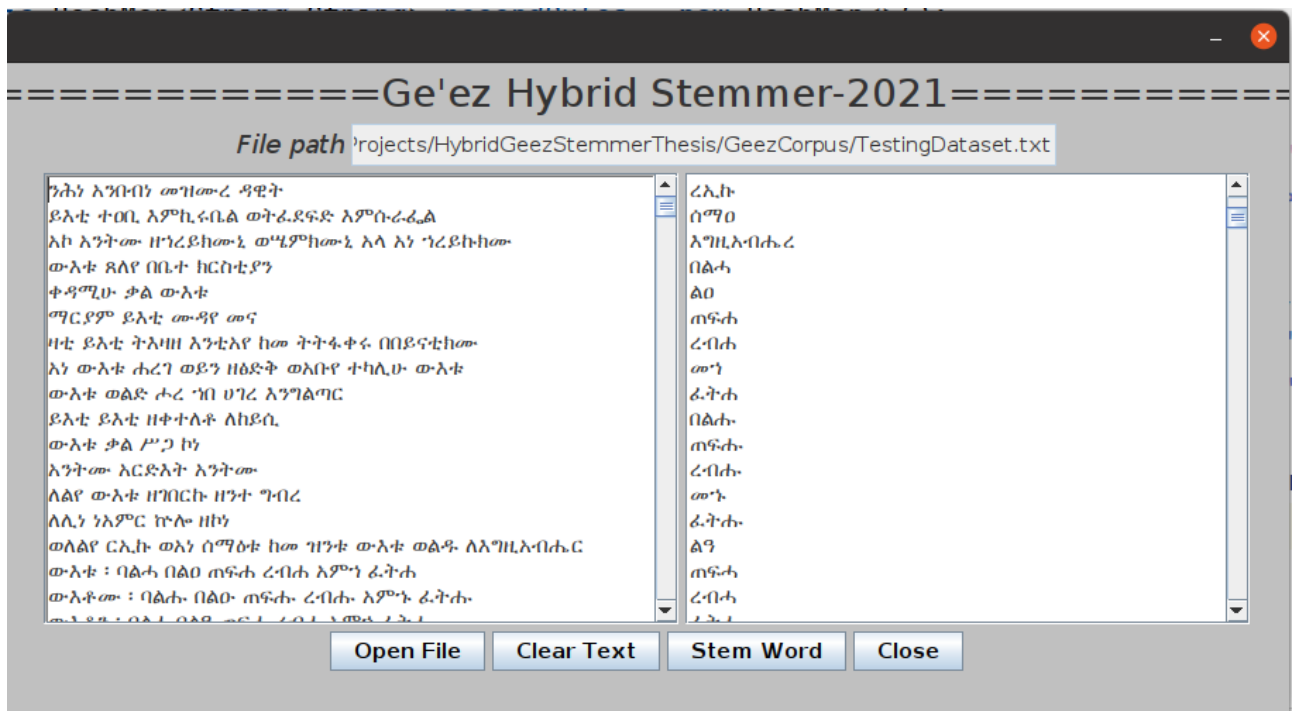


Figure 11 screen shot of the output of first version stemmer

Firstly the testing set is given to the first version (a stemmer without incorporating n-gam) of the stemmer and we have got the following results. According to the stemmer output, 119 (4.5%) words were over stemmed and 84 words (3.18%) were under stemmed, out of 2644 total words in the test sets. As a whole this stemmer generates 203 (7.68 %) words were incorrectly stemmed and out of total errors, most frequent errors were over stemming errors, which covers 58.62% of the errors; whereas, around 41.38% of the errors were observed due to under stemming errors.

On the other hand, 2441 words (92.32%) were correctly stemmed by the first version of the stemmer. As a result, the accuracy of the first version of proposed stemmer was evaluated 92.32%. The errors that were confronted by this stemmer were corrected or reduced by

incorporating other methods such as statistical techniques. So that, most of the above stated errors can be distinguished by integrating the first version (affix removal stemmer) and statistical (n-gram) stemmer in order to enhance the outputs and reduce the errors.

The following table shows that a sample of output from the first version of the stemmers with a sample of under stemmed and over stemmed terms.

Un inflected terms	Stemmer output	Expected output	Errors occurred
አዶናይ	አዶነ	አዶናይ	over stemmed
ይፈልጠኒኑ	ፈልጠ	ፈልጠ	-
ዘወሀብከኒ	ሀብ	ወሀብ	Over stemmed
ዘያድኅኖሙ	ደኅ	ደኅነ	Over stemmed
ዘወጽአ	ወጽአ	ወጽአ	-
ዘነድ	ዘነድ	ነድ	Under stemmed
እውራን	ውራነ	እውር	Under stemmed
ወትባርከ	ባረከ	ባረከ	-
ወደሰኪ	ደሰከ	ወደሰ	Over stemmed
ወረደ	ወረደ	ወረደ	-
ወለሊቃውንቲከ	ሊቅ	ሊቅ	-
ከርሰኪ	ከርሰ	ከርሰ	-
ትትናዘዘነ	ትናዘዘ	ናዘዘ	Under stemmed
ቤትነ	ቤትነ	ቤት	Under stemmed
ሰላምክሙ	ሰላም	ሰላም	-
አኮቴትከ	ኮቴት	አኮቴት	Over stemmed

Table 32 Sample output by the first version of Geez stemmer

Based on the the given test data-set given to the first version stemmer, most of the errors happened were over stemming. The affixes removal process take place after checking the given input text with corresponding stop word and if there is a matching affixes list are found. If the given input is not a part of stop word and any match is found from the affixes list, it further applied the rules and exceptional rules are checked; meanwhile the recording rules are considered for each steps of affixes removal process in order to conflate and produce the stems.

On the second pass, the same testing dataset was given to the hybrid version of the stemmer in order to see the impacts of the hybrid stemmer over the first version of stemmer which was an affix removal. The following figure shows the screen shot of output of the given testing dataset on the hybrid version stemmer.

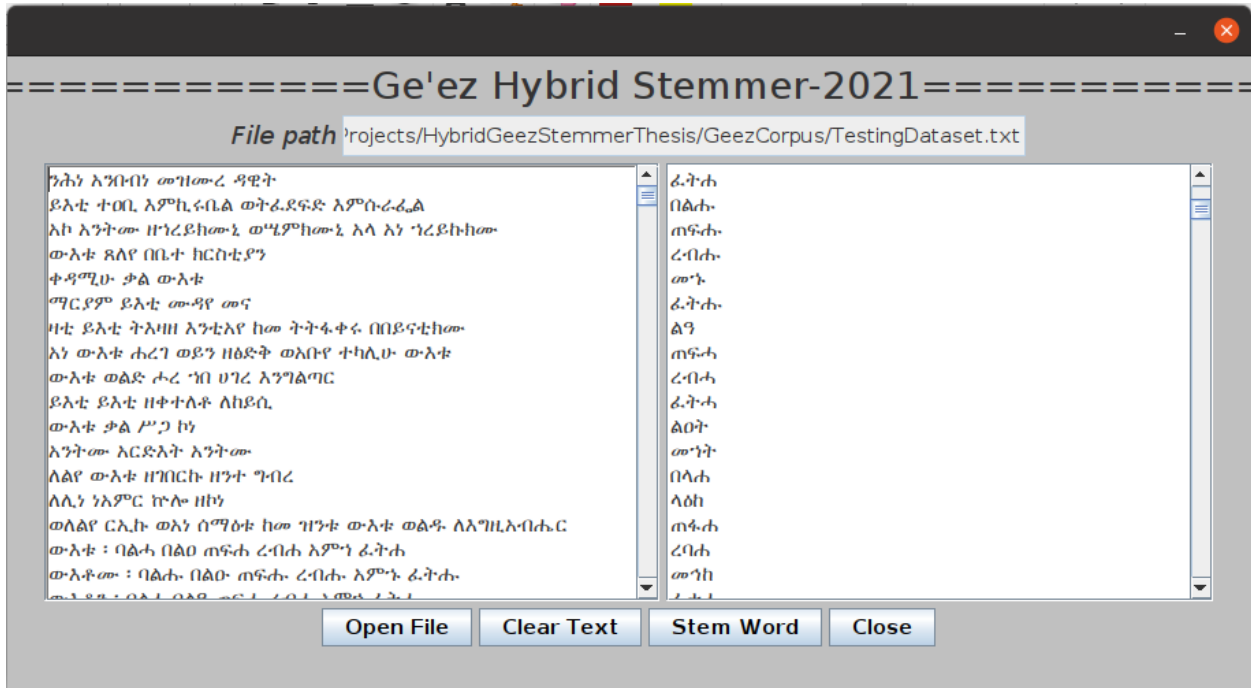


Figure 12 Screen shoot of the hybrid version stemmer

According to the hybrid version stemmer, the output found; 87(3.3%) words were under stemmed and 58(2.20%) words were over stemmed from the total of 2644 words. To this end the hybrid version stemmer generated 145(5.5%) words, that were incorrectly stemmed and 2499(94.5%) words were correctly stemmed.

Specifically, by way of the total errors found on the hybrid version, 60.0% of the errors were due to under stemming and 40.0% were found due to over stemming. As a result over-stemming errors are reduced by 2.3% (by 61 words) and whereas under stemming errors showed little increment particularly by 0.12% (by 3 words), when we integrating the first version(rule based) with statistical(n-gram). The reason of reducing over-stemming errors on the hybrid version stemmer were due to the errors that were made by the rule-based version were corrected/modified by using n-gram. In contrast, under stemming errors shown a little increment on the hybrid version stemmer was because of n-gram technique works only for stripping

suffixes parts. Some Geez affixes particularly prefix and infix that cannot be conflated by rule-based also cannot be conflated properly by n-gram technique i.e. this is due to the morphological complexity of the language.

Finally, an accuracy of the proposed hybrid stemmer was evaluated 94.5% on the testing dataset. Even though, the hybrid version of the proposed stemmer showed a few correction of errors occurred from the first version stemmer, we observed some errors like the first version stemmer. Unlike the first version stemmer the most frequent errors are under stemming. Comparatively, the hybrid version stemmer promoted by an accuracy of 2.18%. As a result integrating n-gram show few enhancements over the first version stemmer.

Un inflected terms	First version output	Hybrid output	version Expected output	Errors on first version	Errors on hybrid version
አድናይ	አድነ	አድናይ	አድናይ	over stemmed	-
ዘወሀብከኒ	ሀብ	ወሀብ	ዘወሀብ	Over stemmed	Under stemmed
እውራን	ውራነ	እውር	እውር	Under stemmed	-
ወደሰኪ	ደሰከ	ወደሰ	ወደሰ	Over stemmed	-
ትትናዘዘነ	ናዘዘ	ትትናዘዘ	ናዘዘ	-	Under stemmed
አኮቴትከ	ኮቴት	አኮቴት	አኮቴትከ	Over stemmed	-
አዘቅት	ዘቅት	አዘቅት	አዘቅት	Over stemmed	-
ገራህትክሙ	ገራህ	ገራህት	ገራህት	Over stemmed	-
ትዕቢትከ	ዕቢት	ትዕቢት	ትዕቢት	Over stemmed	-
ተትሕተኪ	ትሕተ	ተትሕተ	ተትሕተ	Over stemmed	-

Table 33 Sample output by the hybrid version of Geez stemmer

From the total test set given to the hybrid version stemmer, some of the errors happened on the first version stemmer were solved by the hybrid stemmer. Most of the time the over stemming errors shown by the first version stemmer is corrected by the hybrid one. In addition to that, the hybrid version of the proposed stemmer can stem the given texts that were not caught by the affixes removal stemmer. Like affixes removal stemmer, the stemming process take place after checking the given input text with corresponding stop word and affixes lists. If the given input is not a part of stop words and matching affixes list not found; n-gram stemming is applied for the

purpose of conflating the input text and producing the stems. The conflation process is based on the fulfillment the precondition stated on algorithm 5 on the previous chapter (section 4.4.5.4).

5.5.2. RESULTS AND DISCUSSION

In this study we have used 2644 word of testing dataset that are applied on two version of the stemmer. According to the manual evaluation of the first version stemmer results, 92.326% of words were correctly stemmed and 7.68% (203 words) were evaluated as incorrectly stemmed words. From the assessment, the errors revealed were under stemming which yields 41.38% and over stemming errors yields 58.62% of the total errors; that was unable to correctly conflate the words to get the desired stems based on the proposed affixes removal algorithm. From the evaluation we made, the most frequent errors were observed due to over stemming. The following table (table 34) shows as the summarized result for the evaluation of the first version stemmer.

Dataset	No of words	Correctly stemmed	Incorrectly stemmed	
			Under stemmed	Over stemmed
TestingDataset	2,644	2,441	84	119
Percentage (%)		92.32	3.18	4.5

Table 34 Evaluation results of First version stemmer

Furthermore, the same dataset was applied to the hybrid version stemmer and the manual assessment was made. From the assessment, the stemmers were correctly 2,499 words which yield 94.5% of the given dataset and the remaining 145 (5.5%) words were incorrectly stemmed. As a result the accuracy of the hybrid version stemmer was results 94.5%.

Over stemming and under stemming concerns are found after the stemming process is completed. From the total errors of hybrid version stemmer, under stemmed errors yields 60.0% whereas 40.0% of the errors were revealed from over stemming. From the evaluation we made; unlike the first version, the most frequent errors were observed due to under stemming. This is because of n-gram technique ignores affixes particularly prefix and infix in some case. Generally the hybrid version stemmer shows an enhancement of the accuracy by 2.18% and over stemmin errors are corrected in some way. The following table generalized the final results of evaluations for hybrid version stemmer.

Dataset	No of words	Correctly stemmed	Incorrectly stemmed	
			Under	Over
TestingDataset	2,644	2,499	87	58
Percentage (%)		94.5	3.3	2.2

Table 35 Evaluation results of hybrid version stemmer

Finally the study tried to see the overall compression ratio of the stemmer. According to [14], the dictionary size reduction is calculated as follows:

$$C = \frac{W - S}{W} * 100$$

Where C= percentage of compression values

W= the number of total words

S= the number of distinct stem after conflation

According to this formula the compression ratio/dictionary reduction of the proposed stemmer based on the sample dataset are calculated as: the size of dataset are 2644 and the number of distinct stems are 1729. So we can calculate as

$$C = \frac{(2644 - 1729)}{2644} * 100 = 34.6\%$$

Therefore, the percentage of dictionary size reduced is quantified as 34.58%, i.e. Geez language morphology is highly inflected, it indicates that developing a stemmer for this language is recommended.

5.6. SUMMARY

This study tried to design and develop a hybrid stemming algorithm for Geez language text. Ultimately, the researchers tried to propose a list of rules that used to conflate derivational and inflectional affixes. For the sake of this study we have prepared a dataset from different source like text books, newspaper, and bible and from other print and non-printed materials. The dataset prepared were randomly taken from these sources. The test dataset was prepared to evaluate the number of valid words correctly accepted by the system and the number of invalid words incorrectly stemmed. Meanwhile, we have compiled list of stop word, prefixes and suffixes lists. For removing process some exceptional and recording rules are also prepared in order to check

the validity of the stemmed terms. The recording rules are stored on a hashMap of java program in order to handle the checkup process of affixes removal process.

The proposed stemmer has two versions, that integrate affixes removal and hybrid approaches. On the first half, affixes removal version was checked by giving a sample of 2644 testing dataset. Based on evaluation presented in section 5.4., the word that is correctly stemmed was scored an accuracy of 92.32%. During the stemming process over stemming and under stemming problems were observed from affix removal technique and it registered 7.68% as a whole. After the stemming process is completed, over stemming and under stemming problems were observed. From the investigation, over stemming errors were occurred more frequent than under stemming errors.

On the second half, hybrid version of the stemmer was evaluated. In order to achieve and see the impact of this stemmer over the first version, the same testing dataset were given. According to this stemmer, the performance of this stemmer registered 94.5% of the testing dataset. Like the previous version, under stemming and over stemming errors were found and also it has 5.5% coverage. As we discussed in the previous section, the hybrid version shows advancement by 2.18% accuracy and reduce some errors observed on affixes removal algorithm. Similarly the second version of the proposed stemmer has some errors which are common to all stemming algorithm. Conflation algorithms have intrinsic limitations and certain linguistic problems that are common to all conflation algorithms, irrespective of their ultimate use[32].

Additionally, there was 34.6% of compression ratio of stemmed words in these data sets of words. The following chapter summarizes the conclusions of this study and provides suggestions for future study.

CHAPTER SIX

CONCLUSSION AND RECOMMENDATION

In this chapter, the methods followed to conduct the study are summarized and the results found are summarized. The chapter also deals with what should be done to solve the problems are indicated.

6.1. CONCLUSION

Stemming is an extremely useful tool in the field of information retrieval (IR), almost all modern indexing and search systems support it. By reducing the word mismatch between the query and the document, stemming improves information retrieval and also it will result in more relevant documents being returned to the query. Stemming is important for highly inflected languages such as Geez for many applications that require the stem of a word.

This study aimed to design and develop a hybrid stemmer which was able to stem textual documents written with under-resourced languages (i.e. Geez). As explained earlier, this stemmer consists of several components like tokenization, normalization and stop word removal components. To this end, the possible prefixes, suffixes and prefixes-suffixes pair was compiled that were discussed earlier, which made this investigation achievable. According to the study, finding longest match is preferable for developing the stemmer for the Geez language. This is primarily due to the language's morphological complexity; most of the affixes are concatenated with each other i.e., obtaining the possible long lists of the affixes are recommended in order to get the final stems.

In this study, a hybrid stemming method was used that attempts to determine the stem of a word according to the compiled linguistic rules and by applying character n-grams. The method integrates two different stemming techniques to improve the overall performance of the stemming process. To evaluate the proposed system, a testing dataset were prepared from ready available sources randomly. The evaluations made on this investigation were in two phases. Firstly, the proposed rules were evaluated lonely and secondly, incorporating the linguistic rules (rule based) with a character n-gram techniques. The proposed stemmer is evaluated using the error counting method because; there were standard metrics for under resourced language like

Geez. With this method, the performance of a stemmer is computed by calculating the number of under stemming and over stemming errors.

According to the obtained result shows that, an overall accuracy of 92.32% for the first version and 94.5% for the integrated version of the proposed stemmer were resisted. Consequently encouraging result was found, which shows stemming process can be performed with low error rates in highly inflected languages specifically in Geez language. Without a doubt, it is possible to anticipate such considerable contributions and positive effects of the stemmer because; Geez is one of the morphologically rich and complex languages.

As the evaluation result shows, the proposed method generates some errors. These errors were examined and categorized into two different categories namely; under stemmed words and over stemmed words. The error rates were about 4.5% and 2.2% over stemming and then 3.18% and 3.3% under stemming for the first and the hybrid version of proposed stemmer respectively.

Additionally when compared to the rule (affixes only) based stemmer, the evaluation of the hybrid stemmer shows that; there was an accuracy increased by 2.18% with significant increment in computational time. As we observed from the evaluation, the hybrid version stemmer shows few enhancements in terms of accuracy. Finally the proposed hybrid stemmer outperforms by 12.26% and 8.28% accuracy with reducing the error rates by 12.08% and 7.28 % from a rule based and longest match approach that were done by former researcher [19] and [28] respectively. This is due to incorporating the rule-sets based on the detailed study of the languages morphology. Even if the proposed hybrid stemmer found an encouraging result, an error rate of 5.5% are facing i.e. the performance of this stemmer can be increased or the error rates will be reduced by incorporating additional rule sets and an other techniques with the detailed study of the language morphology.

6.2. CONTRIBUTION

As the aim of this research work was to design and develop a hybrid stemming algorithm and implement a prototype of the proposed algorithm, i.e. the main contributions of the study are listed below:

- ✓ Prepared and analyzed Geez text corpus for the sake of implementation of the proposed stemmer, i.e. other researchers will benefit from this prepared corpus for evaluating their proposed studies.
- ✓ This work can help researchers for the purpose of developing application tools such as spell checker, parser, thesaurus, post tagger and dictionary on the language.
- ✓ Linguistic rules are proposed and tested for removing stop words and affixes of Geez text.
- ✓ Algorithms are developed for affixes removal, tokenization, normalization, stop word removal process of the language that are used for preprocessing Geez text.
- ✓ Hybrid version for stemming Geez text is designed as well as a prototype system is developed.
- ✓ We have tested the proposed system for demonstrating the performance and accuracy of each proposed approach.
- ✓ All of the rules described in this work can serve as a foundation for future research.
- ✓ In addition, the study contributes to the growth of research in the area of natural language processing as well as information retrieval system.
- ✓ Finally we believe that, this thesis work contributes in the stemming research and offers a retrieval tool for Geez text that can be used on the web.

6.3. RECOMMENDATION

According to the study we made, the research work was a prototype hybrid stemmer for Geez that appears to work with reasonably high accuracy. Although an encouraging result has been obtained in this study, the following recommendations are identified for further work in order to make the result useful in an operational retrieval environment.

- ✓ The observed 5.5 % error rate can be minimized by adding more stemming rules and exceptions rule-sets as well as by trying other approaches.
- ✓ Further study of the morphology of the language can increase the accuracy of the stemmer.

- ✓ Moreover, the stemmer has to be tested with large amount of texts to verify its real performance on IR system. This is because large size sample can represent the characteristics of the language more than small size sample.
- ✓ Evaluating the stemmer on text collection of large size collected from different sources my leads to see the real performance of the proposed stemmer.
- ✓ One can add more context-sensitive and recoding rules in order to increase the accuracy of this stemmer;
- ✓ This study concentrates on finding the longest possible affixes. Other algorithms can be implemented and the performance between them can be compared over a larger collection of data sets.
- ✓ Preparing adequate and better size corpus by incorporating from various domains must be one task in the future and having a standard dictionary with maximum word size is very important to see the accuracy of the proposed system.
- ✓ Machine learning approach like deep learning will be applied for future in order to see the performance of the stemmer for this language.

REFERENCE

- [1] M. Anjali and G. Jivani, “A Comparative Study of Stemming Algorithms,” vol. 2, no. 6, pp. 1930–1938, 2011.
- [2] C. Moral, A. de Antonio, R. Imbert, and J. Ramírez, “A survey of stemming algorithms in information retrieval,” *Inf. Res.*, vol. 19, no. 1, pp. 76–80, 2014, doi: 10.9790/0661-17367680.
- [3] F. Ahmed and A. Nürnberger, “Evaluation of N-Gram Conflation Approaches for Arabic Text Retrieval,” *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 1974, pp. 1448–1465, 2009, doi: 10.1002/asi.
- [4] H. B. Patil, B. V Pawar, and A. S. Patil, “A COMPREHENSIVE ANALYSIS OF STEMMERS AVAILABLE FOR INDIC LANGUAGES,” *Int. J. Nat. Lang. Comput.*, vol. 5, no. 1, 2016, doi: 10.5121/ijnlc.2016.5104.
- [5] M. H. Dianati, M. H. Sadreddini, A. H. Rasekh, S. M. Fakhrahmad, and H. Taghi-Zadeh, “Words Stemming Based on Structural and Semantic Similarity,” *Comput. Eng. Appl. J.*, vol. 3, no. 2, pp. 89–99, 2014, doi: 10.18495/comengapp.v3i2.57.
- [6] U. Mishra and C. Prakash, “MAULIK : An Effective Stemmer for Hindi Language,” *Int. J. Comput. Sci. Eng.*, vol. 4, no. 05, pp. 711–717, 2012.
- [7] A. Paul, “An Affix Removal Stemmer for Natural Language Text in Nepali,” *Int. J. Comput. Appl.*, vol. 91, no. 6, pp. 1–4, 2014.
- [8] A. Purwarianti, “A Non Deterministic Indonesian Stemmer,” *Int. Conf. Electr. Eng. Informatics*, no. July, pp. 1–5, 2011.
- [9] I. Shrestha, “A New Stemmer For Nepali Language,” pp. 0–4, 2016.
- [10] S. P. Meitei, B. S. Purkayastha, and H. M. Devi, “Development of a Manipuri stemmer: A hybrid approach,” in *2015 International Symposium on Advanced Computing and Communication, ISACC 2015*, 2016, no. August 1992, pp. 128–131, doi: 10.1109/ISACC.2015.7377328.
- [11] G. M. Fikremariam, “AUTOMATIC STEMMING FOR AMHARIC TEXT: AN EXPERIMENT USING SUCCESSOR VARIETY,” ADDIS ABABA UNIVERSITY, 2009.
- [12] D. Tesfaye, “Designing a Stemmer for Afaan Oromo Text: A Hybrid Approach,” ADDIS ABABA UNIVERSITY, 2010.
- [13] W. B. A. Karaa, “A New Stemmer to Improve Information Retrieval,” *Int. J. Netw. Secur. Its Appl.*, vol. 5, no. 4, pp. 143–154, 2013, doi: 10.5121/ijnsa.2013.5411.
- [14] J. Singh and V. Gupta, “Text Stemming: Approaches, Applications, and Challenges,” vol. 49, no. 3, 2016.
- [15] S. R. Sirsat, “Strength and Accuracy Analysis of Affix Removal Stemming Algorithms,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 4, no. 2, pp. 265–269, 2013.

- [16] M. N. Al-kabi, P. O. Box, and Z. Jordan, “Towards Improving Khoja Rule-Based Arabic Stemmer,” *Jordan Conf. Appl. Electr. Eng. Comput. Technol. Towar.*, 2013.
- [17] Y. FISSEHA, “DEVELOPMENT OF STEMMING ALGORITHM FOR TIGRIGNA TEXT,” ADDIS ABABA UNIVERSITY, 2011.
- [18] Y. Abate, “Morphological Analysis of Ge’ez Verbs Using Memory Based Learning,” ADDIS ABABA UNIVERSITY, 2014.
- [19] A. B. ADEGE, “DESIGNING A STEMMER FOR GE’EZ TEXT USING RULE BASED APPROACH,” ADDIS ABABA UNIVERSITY, 2010.
- [20] (መምህር)ዮሴፍ ቀሥብ, “ትኩሣኤግእዛ.” በኢትዮጵያ ኦርቶዶክስ ተዋሕዶ ቤተክርስቲያን በሰነድ ትምህርት ቤቶች ማዕረጃ መምሪያ ማኅበረ ቅደሳኑ, ኦዲዮአቦ, 2002.
- [21] ዋቅጅ-ራመ/ት ኑሳሚን, ማኅተ ገበየብ □ ሌሳነ ግእዜ 2ኛ እትም. በኢትዮጵያ ኦርቶዶክስ ተዋሕዶ ቤተ ክርስቲያን ማኅበረ ቅደሳን, 2008.
- [22] መ/ር ነጋሲ ግደይመ/ር ኃይለ ኢየሱስ መንግሥቱ, ልሳነ ግእዛ. በኢትዮጵያ ኦርቶዶክስ ተዋሕዶ ቤተ ክርስቲያን ማኅበረ ቅደሳን.
- [23] ደሴ ቀለበበዲያቆን, ግዕዝ ሕያው ልሳን በቀላል ዘዴ, no. የካቲት. ኦዲዮ አቦ.
- [24] D. Garg, “Improved Stemming approach used for Text Processing in Information Retrieval System,” no. May, pp. 1–71, 2012.
- [25] T. Misikir, “Developing a Stemming Algorithm for Awngi Text: A Longest match approach,” ADDIS ABABA UNIVERSITY, 2013.
- [26] O. T. OMER, “DEVELOPMENT OF A STEMMER FOR AFARAF TEXT RETRIEVAL,” ADDIS ABABA UNIVERSITY, 2015.
- [27] M. K. ABEDO, “DESIGNING A STEMMING ALGORITHM FOR SILT’E LANGUAGE,” ADDIS ABABA UNIVERSITY, 2012.
- [28] A. H. Bahta, M. J. R. Hayag, and T. Gebremeskel, “An enhanced stemmer algorithm for geez text: a long match approach,” vol. 15, 2019.
- [29] A. Mateen, M. Kamran Malik, Z. Nawaz, H. M. Danish, M. Hassan Siddiqui, and Q. Abbas, “A Hybrid Stemmer of Punjabi Shahmukhi Script,” 2017.
- [30] H. Taghi-Zadeh, M. H. Sadreddini, M. H. Diyanati, and A. H. Rasekh, “A new hybrid stemming method for Persian language,” *Digit. Scholarsh. Humanit.*, vol. 32, no. 1, pp. 209–221, Apr. 2017, doi: 10.1093/llc/fqv053.
- [31] M. and Mayfield, “Character N-gram based Tokenization for European Language retrieval.” 2004.
- [32] D. Sharma, “Stemming Algorithms: A Comparative Study and their Analysis,” *Int. J. Appl. Inf. Syst.*, vol. 4, no. 3, pp. 7–12, 2012, doi: 10.5120/ijais12-450655.
- [33] P. Majumder, M. Mitra, and B. Chaudhuri, “N-gram: a language independent approach to IR and NLP,” ... *Knowl. Lang.*, vol. 3, no. November, 2002, [Online]. Available:

<http://www.mt-archive.info/ICUKL-2002-Majumder.pdf>.

- [34] A. Jabbar, S. Iqbal, A. Akhuzada, and Q. Abbas, “An improved Urdu stemming algorithm for text mining based on multi-step hybrid approach,” *J. Exp. Theor. Artif. Intell.*, vol. 30, no. 5, pp. 703–723, 2018, doi: 10.1080/0952813X.2018.1467495.
- [35] K. J. Dhawan, Chandni Singh, ashanpreet Garg, “Hybrid Approach for Stemming in Punjabi,” *Int. J. Comput. Sci. Commun. Netw.*, vol. 3, no. 2, pp. 101–104.
- [36] C. Moral, A. de Antonio, R. Imbert, and J. Ramírez, “A Survey of Stemming Algorithms in Information Retrieval, Information Research: An International Electronic Journal, 2014-Mar,” p. 22, 2014, [Online]. Available: <https://eric.ed.gov/?id=EJ1020841>.
- [37] B. M. H. A, “N-gram-Based Automatic Indexing for Amharic Text By,” ADDIS ABABA UNIVERSITY, 2002.
- [38] J. H. Paik, D. Pal, and S. K. Parui, “A novel corpus-based stemming algorithm using co-occurrence statistics,” in *SIGIR’11 - Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011, pp. 863–872, doi: 10.1145/2009916.2010031.
- [39] V. K. Sarkania and V. K. Bhalla, “International Journal of Advanced Research in,” *Android Internals*, vol. 3, no. 6, pp. 143–147, 2013.
- [40] K. Boukhari and M. N. Omri, “SAID: A new stemmer algorithm to indexing unstructured Document,” in *International Conference on Intelligent Systems Design and Applications, ISDA*, 2016, vol. 2016-June, pp. 59–63, doi: 10.1109/ISDA.2015.7489180.
- [41] V. Gupta and G. S. Lehal, “Punjabi Language Stemmer for nouns and proper names,” in *Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), IJCNLP 2011*, 2011, pp. 35–39, [Online]. Available: <https://www.aclweb.org/anthology/W11-3006.pdf>.
- [42] T. Brychcín and M. Konopík, “HPS: High precision stemmer,” *Inf. Process. Manag.*, vol. 51, no. 1, pp. 68–91, 2015, doi: 10.1016/j.ipm.2014.08.006.
- [43] Y. Hegde, S. Kadambe, and P. Naduthota, “Suffix stripping algorithm for Kannada information retrieval,” in *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013*, 2013, pp. 527–533, doi: 10.1109/ICACCI.2013.6637227.
- [44] D. Rajalingam, “A Rule Based Iterative Affix Stripping Stemming Algorithm For Tamil,” pp. 28–34.
- [45] J. H. Paik, S. K. Parui, D. Pal, and S. E. Robertson, “Effective and robust query-based stemming,” *ACM Trans. Inf. Syst.*, vol. 31, no. 4, 2013, doi: 10.1145/2536736.2536738.
- [46] V. Gupta and G. S. Lehal, “A survey of common stemming techniques and existing stemmers for indian languages,” in *Journal of Emerging Technologies in Web Intelligence*, 2013, vol. 5, no. 2, pp. 157–161, doi: 10.4304/jetwi.5.2.157-161.
- [47] A. Mahmoud, A. Dawut, P. Tursun, and A. Hamdulla, “A Survey on the Methods for

- Uyghur Stemming,” *Int. J. Control Autom.*, vol. 9, no. 11, pp. 143–158, 2016.
- [48] T. M. T. Sembok and Z. A. Bakar, “Characteristics and retrieval effectiveness of n-gram string similarity matching on malay documents,” in *10th WSEAS International Conference on Applied Computer and Applied Computational Science, ACACOS’11*, 2011, vol. 5, no. 3, pp. 165–170.
- [49] W. B. Cavnar, J. M. Trenkle, and A. A. Mi, “N-Gram-Based Text Categorization.”
- [50] A. Rahimi, “Hybrid stemming for Persian.”
- [51] A. Ismailov, M. M. Abdul Jalil, Z. Abdullah, and N. H. Abd Rahim, “A comparative study of stemming algorithms for use with the Uzbek language,” in *2016 3rd International Conference on Computer and Information Sciences, ICCOINS 2016 - Proceedings*, 2016, pp. 7–12, doi: 10.1109/ICCOINS.2016.7783180.
- [52] R. Mahmud *et al.*, “A Rule Based Bengali Stemmer,” in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2014, pp. 2750–2756.
- [53] H. A. Alatabi and A. R. Abbas, “Sentiment analysis in social media using machine learning techniques,” *Iraqi J. Sci.*, vol. 61, no. 1, pp. 193–201, 2020, doi: 10.24996/ijs.2020.61.1.22.
- [54] A. Jabbar, S. Iqbal, M. U. G. Khan, and S. Hussain, “A survey on Urdu and Urdu like language stemmers and stemming techniques,” *Artif. Intell. Rev.*, vol. 49, no. 3, pp. 339–373, 2018, doi: 10.1007/s10462-016-9527-1.
- [55] J. B. Lovins, “Development of a stemming algorithm,” *Mech. Transl. Comput. Linguist.*, vol. 11, no. June, pp. 22–31, 1968, [Online]. Available: <http://journal.mercubuana.ac.id/data/MT-1968-Lovins.pdf>.
- [56] D. Waegel, “The porter Stemmer,” *Appl. Nat. Lang. Process.*, pp. 1–29, 2011.
- [57] D. Kumar and P. Rana, “Design and Development of a Stemmer for Punjabi,” *Int. J. Comput. Appl.*, vol. 11, no. 12, pp. 18–23, 2010.
- [58] O. M. Elrajubi, “An Improved Arabic Light Stemmer,” in *3rd International Conference on Research and Innovation in Information Systems*, 2013, vol. 2013, pp. 33–38.
- [59] B. A. Asma Al-Omari, “Arabic light stemmer (ARS),” *J. Eng. Sci. Technol.*, no. December 2014, 2015.
- [60] W. B. Demilie and A. Class, “Implemented Stemming Algorithms for Six Ethiopian Languages Implemented Stemming Algorithms for Six Ethiopian Languages,” vol. 29, no. August, pp. 5–10, 2020.
- [61] G. F. Scelta, “The Comparative Origin and Usage of the Ge’ez writing system of Ethiopia,” *A Pap. Submitt. to Profr. Pilar Quezzaire-Belle Partial fulfillment Requir. Arts Africa*, pp. 1–9, 2001.
- [62] በጅ/ዩ/ዋ/ግ/ግ/ጉ-ባዔ, ፪ኛ ዓመት ለግእዝ መግሪያ የተዘጋጀ. .

- [63] A. Dillmann, *Ethiopic Grammar, Second Edition Ancient Language Resources*. 2005.
- [64] መንግሥቱመ/ር ኃይለ ኢየሱስ, የልሳነ ግእዝ መማሪያ. በኢትዮጵያ ኦርቶዶክስ ተዋሕዶ ቤተ ክርስቲያን ማኅበረ ቅዱሳን, 2012.
- [65] C. Moral, A. De Antonio, R. Imbert, and J. Ramírez, “A survey of stemming algorithms in information retrieval,” 2014.
- [66] በጅ/ዩ/ዋ/ግ/ግ/ጉባዔ, *1ኛ ዓመት ለግእዝ መማሪያ የተዘጋጀ*. 2006.
- [67] ቅዱሳንማኅበረ, “Hamere News Papare,” በኢትዮጵያ ኦርቶዶክስ ተዋሕዶ ቤተ ክርስቲያን ማኅበረ ቅዱሳን, 2013.
- [68] E. B. Assosiation, “Plasm,” in *Plasm*, Addis Ababa, Ethiopia: Ethiopian Orthodox Tewahido Church, 1980.

APPENDIXES

APPENDIX I: SAMPLE GEEZ STOP WORD LIST

አንቲ	ዛቲ	ላዕለ	እመ	ኩልክን
አንትሙ	እሎንቱ	ታህተ	አላ	አል
አነቲን	እላ	መትህተ	እግዚአ	ቦቶ
ውእቱ	እሉ	ውስጠ	ወይ	ቦሙ
ውእቶን	እለመኑ	ውሳጤ	አሌ	ቦቶሙ
ውእቶሙ	መኑ	ማዕከለ	አህ	ቦን
ይእቲ	እላንቱ	ቦይነነ	አይ	ቦቶን
አነ	ዝክቱ	ቦእንተ	ኅድግ	ብከ
ንህነ	ዝኩ	እም	አሆ	ቦክሙ
ኪያየ	ነዋ	እምነ	ዝስኩ	ብኪ
ኪያከ	ነዩ	ቤዛ	ዝክቱ	ብክን
ኪያኪ	ነያ	ህየንተ	እንታክቲ	ብየ
ኪያሁ	ነየከ	ኅበ	እንትኩ	ብነ
ኪያሃ	ነየኪ	መንገለ	እልክቱ	አልቦ
ኪያነ	ነየ	ጊዜ	እልኩ	አልቦቱ
ኪያክሙ	ነየሙ	መጠነ	እማንቱ	አልቦሙ
ኪያክን	ነየን	እንበለ	ሎቱ	አልባቲ
ኪያሆሙ	ነየክሙ	አምጣ	ኩሉ	አልቦን
ኪያሆን	ነየክን	ከመ	ኩላ	አልብከ
ለሊሁ	ነየነ	አመ	ኩልክሙ	አልብክሙ
ለሊሆሙ	ዲበ	ሶበ	ኩሎሙ	አልብኪ
ዝንቱ	መልዕልተ	እንዘ	ኩሎን	አልብየ

አልብነት	ለልዩ	ለፌ	እንተ	ቅድም
እወ	ለሊሆን	ወለፌ	እለ	ዳዕሙ
አንጋ	ለሊከሙ	የም	አው	ብሂል
እንቢ	ለሊነ	ትማልም	ወሚመ	ወትረ
እንቢየ	ዚአየ	ጌህም	እንዘ	ዘልፈ
እንቢ	ዚአነ	ይእዚ	አያት	እስኩ
እንቢየ	እንተአነ	ናሁ	አይቴ	ነዓ
እንቆዕ	እንተአከ	ዓዲ	በአፎ	ሀብ
ጥቀ	ኤቴ	ጽባህአሜሃ	ባሕቱ	አንቢ
ሰይ	እፎ	ዘልፈ	እንጋ	እንተአየ
ሰፍን	ማእዜ	እወ	እንዳኢ	እሊአየ
ምንት	ዝየ	አሜን	ዕንቆዕ	ካልእ
እስፍንት	ሀየ	አማን	ጥቀ	ባሕቱት
ምንታት	ኩለኔ	እንቆዕ	ንስቲት	
ለሊሃ	ድህረ	እስመ	ሕዳጥ	
ለሊከ	የማን	አምጣነ	ሕቀ	
ለሊኪ	ጸጋም	አኮኑ	ፅሚተ	

APPENDIX II: SAMPLE GEEZ TEST SET

ተንሥእ እግዚአ አምላኪያ ወኡድኅነኒ እስመ አንተ ቀሰፍኩሙ እለ ይጻረሩኒ በከንቱ። ሰነኒሆሙ ለኃጥዓን ሰበርከ።

ወኢያጎድሩ እኩያን ምስሌከ። ወኢይነብሩ ዐማፅያን ቅድመ አዕይንቲከ።

ሰላም ለርእስኪ በቅብእ ቅዳሴ ርሑሰ። አኮ አኮ በቅብዐ ደነሰ።

ሰላም ለእንግድአኪ ለእሳተ መለኮት ምርፋቁ ዘይቄድስ ነፍሰ ወያጥዒ ሞቁ።

እደውየ ይገብራ መሰንቆ ወአፃብየ ያስተዋድዳ መዝሙረ።

ሰላም ለኩርናዕከ ኩርናዐ ከቡድ አንበሳ ዘቀጥቀጠ ኩሎን አርእስተ ሰቡሐን እንሰሳ።

ኅረየ ዐሥርተ ወክልዔተ አርድእተ

አቅረብኩ ለከ እሳተ ወማየ ደይ እዴከ ኅበ ዘፈቀድከ

ንሴብሐከ እግዚአ

ወዳግመ ይመጽእ አምላክነ በግርማ መንግሥቱ።

መጋቤ ብርሃናት ውእቱ ራጉኤል ሊቀመላእክተ።

ዝማሬ መላእክት ያስተፌስሕ አልባበ ቅዱሳን።

አልቦሙ እዝን ዘይሰምዑ ቦቱ ቃለ እግዚአብሔር።

ሰላም ለስዕልኪ እንተ ሰዐላ በእዱ ሉቃስ ጠቢብ እምወንጌላውያን አሐዱ

ገሪማ ገረምከኒ

አነኑ ዐቃቢሁ ለአቤል

ኅበ ሀለዉ ክልዔቱ ወሠለስቱ ግቡአን በስምየ ሀሎኩ አነ ማእከሌክሙ

እመሰ ትፈቅድ ፍጹመ ትኩን ሑር ወሚጥ ኸሉ ዘብከ ሀብ ለነዳያን ወታጠሪ መዝገበ በሰማያትወንዓ ትልወኒ

አንተ ኩከህ ወዲበ ዛቲ ኩከህ አሐንጻ ለቤተክርስቲያን ወአናቅጸ ሲጋል ኢይኔይልዋ

እለ ውስተ ሲጋል 90 ወእለ ውስተ ጽልመት ተከሥቱ

ጊዜ ተሰዓቱ ሰዓት የዐርጉ መላእክት ምግባሮ ለሰብእ

ውድስት አንቲ በአፈ ነቢያት ወስብሕት በሐዋርያት

ኦ ኣኃው ተዐቀቡ እምስካር እስመ ሰካር ይዘሩ ልበ ወያደክም ሥጋ ወይሬስዮ ለብእሲ ይኩን ማኅደሮ ለሰይጣን ወያረስዮ

ትሕርምታተ ዘእግዚአብሔር ወያተነትኖ ማዕከለ ሰብእ ወያመጽዕ ላዕሌሁ ኩሎ ነገረ ኃፍረት ወኃሣር ወዓዲ ይከውን ዕልወ
ወይመርሖ ውስተ ቀትል ወይስሕቦ ውስተ ዝሙት ወዘይመስሎ ለዝንቱ ወያሐጉል ንዋዮ ወያረሲዖ ጸሎተ ወኢያዜክሮ አምላኮ
ወለብእሲ ስኩር ይሴሰል ይርኅቁ እምኔሁ-መላእክት ወዝኩሉ ይረክቦ በዝ ዓለም
ለመኑ ተውህቦ ሀብተ መንፈስ ቅዱስ ወለእለመኑ ተውህቦ ዝንቱ መካን።
ለመኑ እለመኑ ተሰቅሉ ምሰለ ክርስቶስ።
ለምንት ይደክም ሰብእ ኩሎን ጊዜያት።
አልአዛር ምንቶን ውእቱ ለማርያም ወለማረታ።
ዳዊት ምንቱ ውእቱ ለሰሎሞን።
አ ወልድዮ ምንተከ ውእቱ ዘሐመከ።
ኤልሳቤጥ ምንትኪ ይእቲ ወለተ እምኪ።
በመኑ ተገብረ ዝንቱ ግብረ ኃጉል።
እለመኑ ውእቶሙ ዘልሣነ ግእዝ መምህራኒክሙ።
እለመኑ ተዐስሩ ወእለመኑ ተፈትሑ።
እለመኑ አንትሙ ዘታስተዋርዱ ክብረነ።
መኑ ፈጠሮ ለእጻለእመሕያው ወመኑ ሣረሮ ለሰማይ።
በምንት ይመጽእ ኤርምያስ በሰረገላኑ ወሚመ በእግር።
አ አንስት ምንክን ውእቱ ዝንቱ ወሬዛ።
ሀሎኑ በዝ ሰማይ አምላክ እስራኤል አዶናይ።
አኮኑ ለእግዚአብሔር ትገኒ ነፍሰዮ።
በይነምንት ይዜኅር ባዕል ላዕለ ነዳይ። ኢይነዲሁ ዘብዕለ ወኢይብዕልሁ ዘነድዮ።
በእፎ ያፈቅር ሰብእ ዓለመ ቦኑ ይመስሎ ኢዮነልፍ።
ቦኑ በከነቱ ፈጠርኮ ለእጻለእመሕያው።
ማእዜ ይመጽእ ክርስቶስ ቀማእዜ ይከውን ኅልፈተ ዓለም።
እስከማእዜኑ እግዚአ ትረስአኒ ለግሙራ።

አይ ዕለት ተሰቅሎ ክርስቶስ ዐለተ ዓርብኑ ወሚመ ዕለተ ረቡዕ።

ማእዜ መጻእከሙ ኅብ ዝንቱ መካን ወማእዜ ተሐውሩ እምዝንቱ መካን።

እፎ መጻእከ ጳውሎስ ኅብ ዝንቱ መካን ኢተሐፍርኑ በዘገበርኩ።

እፎ ተሐንጸ ዝንቱ ቤተ መቅደስ በእደ ሰብእኑ ወሚመ በግብረ መንፈስ ቅዱስ።

እፎ እፎ ተዋነዩ አብያጽዮ።

እፎ ኅደርከሙ አዝማድዮ በይእቲ ሌሊት።

ውእቱ ቃለ አብ እግዚአብሔር ሥጋኪ ተዐፅፈ በመንክር ምሥጢር ማርያም ድንግል ወላዲተ ክርስቶስ ክቡር።

መኑ ይነግረከ ወይዘንወከ ዜናዊ ገዓረ ወለትከ ትስማዕ ኢያቄም እስራኤላዊ።

ይጽብበኒ እግዝእትዮ ከመ ጸበበኪ ዓለም በምልዑ አመ ዐገቱኪ ፈያት ልብሰ ወልድኪ ይንሥኡ።

ጉዮ ዮሴፍ ብሔረ ግብጽ ተንሥኦ እም ንዋሙ መልአከ እግዚአብሔር ሌሊተ ከመነገሮ በሕልሙ ነሥኦ ሕፃነ ምስለ እሙ።

ወእምዝ አዘዘ መስፍን ያምጽእዎ ለዘይንዕስ ወልድ እኑሆሙ ወያብእዎ ውስተ ቤተ ዐቀብት በከመ ይቤ ቅዱስ ሚካኤል።

ወይእዜኒ ኦ አጋውዮ ንስአሎ ለእግዚአብሔር መፍቀሬ ሰብእ በእንተ ሰሙ ለቅዱስ ሚካኤል ሊቀ መላእክት ዘውኩፍ

ስእለቱ ቅድመ እግዚአብሔር

ይቀትል ይቄድስ ይገብር የአምር ይባርክ ይዴግን ይክህል ይጦምር

ይቅትል ይቀድስ ይግበር ያእምር ይባርክ ይድግን ይክህል ይክል ይጦምር

ዖደ ዖዱ ዖደት ዖዳ ዖድከ ዖድከሙ ዖድኪ ዖድከን ዖድኩ ዖድነ

የዐውድ የዐውዱ ተዐውድ የዐውዳ ተዐውድ ተዐውዱ ተዐውዲ ተዐውዳ አዐውድ ነዐውድ

ይዐድ ይዐዱ ትዐድ ይዐዳ ትዐድ ትዐዱ ትዐዲ ትዐዳ እዐድ ንዐድ

ይዐድ ይዐዱ ትዐድ ይዐዳ ዐድ ዐዱ ዐዲ ዐዳ እዐድ ንዐድ

አልቦ አብ ወአልቦ እም

ወፅአ ትእዛዝ እምነብ ቄሳር

እምነ ጽዮን ይብል ሰብእ

እምነ ረሀብ ይኔይስ ኩናት

ወአዝነመ ሎሙ መና ይብልዑ

ንሳኢ መና ምስሌክ ወሑር ወተቀበሎ

ማዕዜ ይኸውን ነግህ

ወሑራ ወአንግሃ ጥቀ ሐዊረ ኅበ መቃብር

ወቤተት ኅበ እገረሁ እስከ ይጸብሕ

ኢትጻብሐ ወእመቦ ዘኢተጻባሕክ ኢታውሥኦ

መቅድሕተ እሳት ይክዕወ ላዕሌሆሙ

ትኸውን ምቅዳሐ ለኸሉ ነገሥተ ዓለም

ለምንት ትቴክዚ ወለምንት ተሀውክኒ እመኒ በእግዚአብሔር ከመእገኒ ሎቱ

ዘየአምን ብየ እመኒ ሞተ የሐዩ

ኸሉ ትውልድ ይብሉ እምነ ጽዮን ሰላም ለኪ

ነግሠ ሞት እምነ አዳም እስከ ሙሴ

አርባሕክ ትምክሕተ በርእስክ ወበሰብዕክ ኅሳረ ወዝንጋጌ

ለ.ወበስቴክ መስተፍስሔ አርብሐኒ ተድላ

ቦ እለ ይብሉክ ዮሐንስሃ መጥምቅ።

አነ ውእቱ ገብርኤል መልአክ ዘእቀውም በቅድመ እግዚአብሔር።አነ ፡ ወጠንኩ በየንኩ ማሰንኩ አድነንኩ መነንኩ ከነንኩ

አመ ተሰቅለ ክርስቶስ መልዕልተ መስቀል ፀሐይ ጸልመ።

ናሁ ተወልደ ዮም ቤዛ ኸሉ ዓለም በከመተነበየ ኢሳይያስ ነቢይ።ነግሠ ሞት እምነ አዳም እስከ ሙሴ

አመ ይመጽዕ ወልድኪ ምስለ አእላፍ መላእክቲሁ።

ሰአሊ ለነ ማርያም በቅድሜሁ።

ተወውቀ ዝንቱ ብእሲ በኅበ ሊቃውንትአምጣነ ምሁር ውእቱ።

ወይቤላ መልአክ ኢትፍርሒ ማርያም እስመ ረከብኪምገሰ በኅበ እግዚአብሔር

ሰላም ለኪ እንዘ ንሰግድ ንብለኪ ማርያም እምነ ናስተበቀኦኪ

ናሁ እም ይእኬሰ ያስተበጽዑኒ ኸሉ ትውልድ

አእምሩ ከመ ኮነ እግዚአብሔርኔር

APPENDIX III: SAMPLE GEEZ PREPOSITION AND CONJUNCTIONS

በእንተ	ወእደ	አዳምሰ	ዮጊ	ማዕከለ	ግንጽሊት
በይነ	ኅበ	እፎ	ከመ	ኅበ፤መንገ	ግፊተቲት
እንበይነ	መንገለ	በእፎ	ጽመ	ለ	ጽፍሕ
ህየንተ	ውእደ	ለምንት	እንቋዕ	ውስተ	ጽንፍ
ተውላጠ	ምንት	ስፍን	ጥቀ	እንተ	ጽንፊፍ
ፍዳ	መኑ	እስፍንቱ	እምዮም	ለ	ጽላፍ
በቀለ	አይ	ሁ	ኬ	በ	ከንፍ
በዘ	እፎ	ኑ	ሶ	አመ	ከንፊፍ
እስመ	ማዕዘ	ቦኑ	ሰ	እንተ	ከነፊ
አምጣኑ	አይቴ	አሌ	ዲበ	በበ	ከንፈር
አኮኑ	ስፍን	ወይ	ላዕለ	እንበለ	ሐይቅ
ከመ	በሀ	ሰይ	መልዕልተ	ኢ	ድንጋግ
አምሳለ	አይ	አው	ታሕተ	አመሂ	ትርጋጽ
እስከ	ምንት	ዮጊ	ታሕቲቱ	ጸጉ	ኩለንታ
አላ	አያት	ሚመ	መትሕተ	ከሀት	ወእምከሀ
ዳዕሙ	ምንታት	እመ	ውስተ	ትካት	ከሀ
ባሕቱ	ኦ	አኮ	ውሳጤ	የማን	ወትረ
እንበለ	ሚ	ሶበአኮ	ቅድመ	ይምን	ዘልፈ
አዲ	መኑ	እስኩ	ድኅረ	ድኅሪት	ለዝሉፉ