INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPU TER SCIENCE

SCHOOL OF GRADUATE STUDIES

MORPHOLOGY BASED SPELLING CHECKER FOR GEEZ LANGUAGE

MASTERS OF SCIENCE THESIS

FIKADE CHANE FELEKE

HAWASSA UNIVERSITY, HAWASSA, ETHIOPIA

NOVEMBER 2023

MORPHOLGY BASED SPELLING CHECKER FOR GEEZ LANGUAGE

MASTERS OF SCIENCE THESIS

FIKADE CHENE FELEKE

A THESIS SUBMITTED TO HAWASSA UNIVERSITY
INSTITUTE OF TECHNOLOGY,
FACULTY OF INFORMATICS,
DEPARTMENT OF COMPUTER SCIENCE,
SCHOOL OF GRADUATE STUDIES
HAWASSA, ETHIOPIA

IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE
DEGREE OF

MASTER OF SCIENCE IN COMPUTER SCIENCE

NOVEMBER 2023

Declaration

This thesis is my original work carried out by me and has never submitted to this or any other institution or university to get any other degree or certificates.

Name of the student Fikade Chane Signature _____

Date of submission: _____

Place: Hawassa

ADVISORS' APPROVAL SHEET

SCHOOL OF GRADUATE STUDIES

HAWASSA UNIVERSITY ADVISORS' APPROVAL SHEET

This is to certify that the thesis entitled "Morphology based Spelling checker for Geez language" submitted in partial fulfillment of the requirements for the degree of Master's with specialization in Computer Science the Graduate Program of the Department of Computer Science, and has been carried out by Fikade Chane , under our supervision. Therefore, we recommend that the student has fulfilled the requirements and hence hereby can submit the thesis to the department.

Name of major advisor	Signature	Date
Dr.Tesfaye Bayu Bati (PhD)	_____	_____
Name of co-advisor	Signature	Date
Teshager Kassa (MSc)	_____	_____

Acknowledgment

Before all, I would like to thank Almighty God, his mother holly Virgin Mary, and all the saints for leading and following me in each steps, for giving me the strength when I get weak and for showing me the way when I get lost.

Next, I would like to acknowledge my advisor Tesfaye Bayu Bati (PhD) for supporting and reading my document line by line and giving me comments. And, I would like to acknowledge my co-adviso Teshager kassa (MSc) for supporting from improving my title to reading my document.

I wish to express my deepest gratitude to my wife Beletie Asmare for supporting and encouraging me in my journey. And also, I would like to thank my father and mother and other family members who encouraged me to go on.

Abstract

Geez is one of the ancient languages. It belongs to Semitic language family. Many ancient literature and books have written in Geez. Currently, Geez course are offered in various colleges, universities and in some primary schools. However, still developed NLP applications are insufficient for this language. In order to write error free Geez text in less time, spelling checker application is a critical NLP application. Spelling checker is a tool used to detect spelling error in a block of text and gives closer suggestions to the error words. A previous attempt has made to develop a spelling checker for Geez language. This attempt was focus only homophone alphabet interchangeably error.

In this study, we proposed morphology based (dictionary lookup and morphological analyzer) approach to Geez language spelling checker. The system have three main compenents.These are text preprocessing, error detection, and error correction. To achieve the objective of this study the researcher builds one main dictionaries that contains Geez language lexicon and morphological feature. The researcher built 6115 unique Geez lexicon and 955 rules had defined. We adopt the Hunspell dictionary and affix file format to design a lexicon (i.e. the knowledge base component) and hashing algorithm for searching. Hunspell is an open source spelling checker tool. It has designed especially for languages that have complex morphology.

Finally, the researcher has developed a prototype of a system to test the functionality and performance of the Geez language spelling checker. The accuracy of error detection expressed in terms of precision and recall. In addition, the accuracy of suggestion expressed in terms of suggestion adequacy. Therefore, we got the result of lexical recall 91.9%, error recall 83.7%, lexical precision 97.2%, error precision 62.2% and correct suggestions provided by GLSC 87.5%. The overall performance of the system is 90.05%. We conclude that increase the size of the dictionary and develop well organized rule will increase the overall performance of the Geez language spelling checker.

Key words: Error Detection, Error Correction, Spell Checker, Morphology, Non Word Error, Real Word Error, Geez language, dictionary lookup

Contents

Acknowledgment iv

Abstract v

CHAPTER ONE 1

1. Introduction 1

1.1. Background 1

1.2. Motivation 3

1.3. Statement of Problem 4

1.4. Objective 8

1.4.1. General Objective 8

1.4.2. Specific Objective 8

1.5. Scope and Limitation of the Study 9

1.6. Significant of the Study 9

1.7. Methodology 10

1.7.1. Literature Review 10

1.7.2. Data Collection 11

1.7.3. Implementation Tool 11

1.7.4. Evaluation Metrics 11

1.8. Organization of the Thesis 12

CHAPTER TWO 13

2. Literature Review 13

2.1. Introduction 13

2.2. Overview of Spelling Checker 13

2.3. Technique of Spelling Checker 14

2.3.1. Error Detection 15

2.3.2. Error Correction 16

2.4. Performance Evaluation Technique for Spelling Checker 19

2.5. Related Works 21

2.5.1. Spell checker for foreign language 21

2.5.1.1. Arabic Spell Checker 21

2.5.1.2. Spelling Checker for Marathi 22

2.5.2. Spell checker for Ethiopian language 22
2.5.2.1. Afaan Oromo Spelling Shecker 22
2.5.2.2. Amharic Spelling Checker 23
2.5.2.3. Tigrigna Spelling Checker 27
2.5.2.4. Geez Spelling Checker 28
2.5.2.5. Morphological Analyzer for Geez language 29
2.5.3. Summary of Related Works 31
CHAPTER THREE 36
3. Overview of Geez Language 36
3.1.1. History of the Geez language 36
3.1.2. Writing System of Geez language 36
3.2. Geez word class 38
3.2.1. Affixation in Geez word 43
3.2.1.1. Prefixes 44
3.2.1.2. Suffixes 46
CHAPTER FOUR 48
Design of Geez Spelling Checker 48
4. Overview 48
4.1. Morphological analyzer for Geez Language 48
4.2. System Architecture for Geez Spelling Checker 51
4.2.1. Preprocessing Component 53
4.2.2. Error Detection Component 54
4.2.3. Error Correction Component 56
CHAPTER FIVE 59
5. Experiment 59
5.1. Overview 59
5.2. Testing Data Preparation 59
5.3. Prototype of the System 60
5.4. Performance Evaluation 63
5.4.1. Performance of Spelling Error Detection 63
5.4.2. Performance of Spelling Error Correction 64

5.5. Result and Discussion 65
5.6. Challenge and Limitation 66
CHAPTER SIX 68
6. Conclusion and Recommendation 68
6.1 Conclusion 68
6.2. Recommendation 69
Reference 71
Appendix A: Geez Alphabet and Number 75
Appendix B: Sample Affix file Rule in Hun spell 76
Appendix C: Sample Corpus 80
Appendix D: Sample Suggestion 83

List of Abbreviations

NLP: Natural language processing

EOTC: Ethiopian orthodox Tewahido church

GLSC: Geez language spelling checker

ASR: Automatic speech Recognition

MT: Machine Translations

OCR: Optical Character Recognition POS:

POS: Part of Speech

TTS: Text To Speech

NER: Named Entity Recognition

IR: Information Retrieval

LED: Levenshtein Edit Distance

List of table

Table1. 1 Geez homophone alphabet 3
Table1. 2 comparing Geez from Amharic and English 6
Table1. 3 Geez inflected in 10 pronouns 7

Table2. 1 Summary of literature related works 35

Tatble3. 1 sample example to show modern representation of Geez alphabet 37
Tatble3. 2 Example of singular to plural noun 38
Tatble3. 3 Example of Geez Amharic adjective suffix and prefix to change in to plural form 39
Tatble3. 4 Tense-mood Identified by Ethiopian Scholars [38] 40
Tatble3. 5 Geez root verb 41
Tatble3. 6 five stem verb with tense mode [46] 44
Tatble3. 7 Basic out breeding phones 45
Tatble3. 8 Geez subjective suffix adapted from 46
Tatble3. 9 Geez objective suffix 47

Table5. 1 Result of cause of pattern error 60
Table5. 2 Error detection result 64
Table5. 3 sample example to calculate SA 65

List of figure

Figure1. 1 sample screen shot from android 5

Figure4. 1 (a and b) Morphological feature (affix and rule) and Dictionary 49
Figure4. 2 Geez circumfix 50
Figure4. 3 General system architecture 52

Figure5. 1 sample example of error detection 61
Figure5. 2 sample prototype to rank suggestion 62

List Algorithm

Algorithm4. 1 for Tokenization process 53
Algorithm4. 2 Dictionary lookup 54
Algorithm4. 3 Algorithm for morphological analyzer [3] 55
Algorithm4. 4 for morphological generator 57

CHAPTER ONE

1. Introduction

1.1.Background

Language is one of the fundamental aspects of human behavior. It uses either in written form or spoken form. In written form, it serves as a means of recording information and knowledge for long time and transfer to the next generation. In spoken form, it serves as the way to support corporate human activities [1]. Language can classified in to two major parts. These are natural language and artificial language [2]. A language that, we learn from environment and used as a means of communication by human known as natural language like Amharic, English, Geez e.t.c. whereas artificial language is a language developed based on a set of rule to perform specific activity. For instance, Java, C++ e.t. c.

From natural language, Geez is one of the ancient languages. Geez language is the classical language of Ethiopia and belongs to the Semitic language family [3]. This language was a governmental language especially at the time of Axumite civilization for the kingdom and imperial court [4] .At that time Geez language becomes the only national language in Ethiopia and got an acceptance among people [5]. Due to this, different types of religious and secular books and literature have written in this language. From secular books, ancient literature, ancient philosophies, tradition, histories, romances, legal, mathematical, and medical texts are some of the examples [6]. From religious books, Bible, Apocrypha, liturgical literature, homiletic, theological, and magical texts, stories of martyrs and saints, religious poetry, hymns in honor of Christ, the Virgin, the martyrs, the saints, and angels are some of the examples.

After the end of Axum Empire Geez language have dominated by other Semitic language [4] and Geez language also becomes as liturgy language in Ethiopian orthodox Tewahido church (EOTC), Eritrean Orthodox Tewahido Church, the Ethiopian Catholic Church, the Eritrean Catholic Church [5] [7].

Although Geez language used as liturgy language after the end of Axum Empire, currently, Geez course are offered in various colleges, religious centers and higher institute and different Geez books and other reference materials are publishing [8]. For example, Ethiopian Orthodox Church Theology Colleges are giving Geez course and publishing various books. In addition to this, various higher institutes teach Geez language as a course and organize in its own department. In foreign country, like European and United States of America Universities [7] and In Ethiopia like Wollo University, Bahir Dar University starts to teach Geez language. In order to minimize errors and save time when we write Geez text, some Natural language processing (NLP) applications must be developed.

NLP is a branch of artificial intelligence that helps computer to understand, interpret and manipulate human language and how to program computer to process and analyze large language of natural language data [4]**.** NLP focuses on designing and implementing of tools, techniques, frameworks to enable computers communicate effectively with humans [3]. It is the means for accomplishing different types of applications. Such as grammar checking, spelling checker, Named Entity Recognition (NER), Information Retrieval (IR), Speech Recognition, Machine Translation, Question Answering etc. From these NLP applications, the proposed research focuses on developing the interactive spelling checker mod for Geez language.

In computing, spelling checker is a tool that detects misspelled word(s) in a paragraph or sentence and provides the alternatives to correct them [9]. The idea for spelling checker development is not new. Still many researchers conduct a research on spelling checker in foreign language as well as local language using different method [10]**.** For instance, for the foreign language like Kanada [11],Punjabi [12],Hindi [13].The same research had also been conducted on Ethiopian language like, Afan Oromo [9],Amharic [14] [10],Tigrigna [15],Hadiyyisa [16] [17] ,Kambatissa [17] ,Awngi [16] [17]. However, most of African languages have no complete spelling checker tool to integrate word processor [10].

Spelling checker application has two main phases [18]. The first phase is detect or identify the misspell word. The second phase is to suggest the alternative word form in order to replace the misspelled word.

1.2.Motivation

Now a day, some of foreign and Ethiopian universities start to teach Geez language and some organization and individual are developing android application that contains ancient books and bibles. Additionally, Geez course are offered in some primary schools. Therefore, the development of spelling checker application for this language is required for easy preparation of documents, notice and lecture note. The absence of a spelling checker tool for Geez language has made document preparation activities difficult, needs excessive effort to check and correct the misspelled from the document.

Previous attempt has made to develop a Geez language spelling checker. This attempt was made using N-gram technique to solve homophone alphabets interchangeably error for Geez language. In Geez some alphabets (ፊደሎች/fīdelochi) have the same sound but have different shape and meaning like 3Ha's (ሀ/hā፣ ሐ/ḥā፣ ኀ/ḫā), 2se's (ሰ/se፣ ሠ/še), 2tse's (ጸ/ts'e፣ፀ/t s'e) and 2A's (አ/ā, ዐ/'ā) are homophone [5] [4].

Look at the following table that shows homophone and known name.

Letter	Known name	Reason
ሀ	ሃሌታው "ሀ"	Since it is the beginning of the Geez word ሃሌ
ሐ	ሐመሩ "ሐ"	Since it is the beginning of the Geez word ሐመር
ኀ	ብዙኃኑ "ኀ"	When the word ብዙኃን written it is the one used
ሰ	እሳቱ "ሰ"	When the Geez word እሳት written it is the one used.
ሠ	ንጉሡ "ሠ"	When the Ge'ez word ንጉሡ written it is the one used
አ	አልፋው "አ"	The word አልፋ is always written using it
ዐ	ዐይኑ "ዐ"	The shape is like Eye and the Geez word ዐይን is written
ጸ	ጸሎቱ "ጸ"	The Geez word ጸሎት written it is the one used.
ፀ	ፀሐዩ"ፀ"	The shape is like sun and used to write the Geez word ፀሐይ

Table1. 1 Geez homophone alphabet

In Geez, these alphabets have their own meaning when we write a word [5] [20]**.** If the writers uses this alphabet interchangeable, it consider as invalid word **(ፀያፍ**) [20].

Example

Instead of **ንጉሥ/king/** the writer writes **ንጉሰ/king/**

Instead of **ሀገር/country/** the writer writes **ሐገር** or **ኀገር** are invalid (ፀያፍ) words.

Instead of **መጽአ/he came/** the writer writes **መፅዐ/metsia/**

Therefore, the previous study worked by Aleka T [4] focus on to fix Geez homophone alphabet interchangeably errors (**ፀያፍ**). However, interchangeably alphabets are not the only spelling errors in the Geez language. The work proposed by Aleka T [4] did not cover Geez abbreviations, Geez Non-word errors and other real word errors. There for performing Non-word error detection and correction can improve the performance of the Geez language spelling checker.

1.3.Statement of Problem

Currently, the world is digital world and people interact to digital computer and handheld device. In this digital world, the data must organize in digital way. Therefor inputting data is the critical activity for organize data in a digital way. Additionally, different organizations and individuals attempt to develop different mobile application like machine translation. In this case, a spelling checker is required to minimize error, save time and to motivate for all Geez writers and readers can easily communicate without misinterpretation caused by spelling errors. Look at the following figure that takes screen shot form mobile application .

Figure1. 1 sample screen shot from android

As shown in the figure above the underline word is Non-word error. This type errors and other type of homophone interchangeably errors makes the reader doubt to the developed application. We proposed a system to solve this problem.

In Geez, there is no dictionary that covers all possible inflection, derivation and compounds words. Because of it generates many words from a single verb. In practically, it is impossible to store all words in dictionary. There for dictionary based approach is not suitable for morphologically rich language. Error free readymade Geez corpora in electronic media also not available so far, so corpora-based spell-checker is not suitable. Although some Geez text (like bible) available in online, it contains a lot of Non-word and homophone error [4].The rich morphological nature of the language makes a morphology-based approach more suitable [21].

An attempt have already made on Semitic language using morphological information. like [14] Merhawit S was done Amharic language spelling checker based on morphology, Berhihun H. was done Tigrigna spelling checker using rule based and morphological analyzer [15]. However, for Amharic spelling checker and Tigrigna spelling checker cannot be applicable for Geez language. Since these languages did not incorporate some grammatical

rules of Geez and alphabet variation [5]. The main reason that makes it unique from the previous Semitic language spelling checker is:-

1. **The spelling checker application is language dependent**:-The spelling checker application for one language is does not work directly for another language.
2. **The nature of the language: -** Each verb in Geez can inflect in Gender (Masculine or Feminine), Number (Singular or Plural), nearness or farness [22]**.** All of this represent in different unique words. For example in English the word "**you**" represents may be plural or singular, may be represent female or male. However, in Geez male and female, plural and singular are representing in different unique word. Show the following table.

	Subject	Plural and Singular	Male and Female	For Third person
English	You	You(may be plural or singular)	you(may be male or female)	She(for female)and He (for male) They (may be male or female)
አማርኛ	አንተ(you) አንቺ(you) እናንተ(you)	አንቺ/አንተ(for singular) እናንተ(plural)	አንቺ(for female)አንተ(for male) እናንተ(for male and female)	እሱ(for male) and እሷ(for female) እነሱ(for male or female)
ግዕዝ	አንተ(you) አንቲ(you) አንትሙ(you) አንትን(you)	አንተ(for singular male) አንቲ(for singular female) አንትሙ(for plural male) አንትን(for plural female)	አንተ(for male) አንቲ(for female) አንትሙ(for male) አንትን(for female)	ውእቱ(for singular male) ይእቲ(for singular female) ውእቶሙ(for plural male) ውእቶን(for plural female)

Table1. 2 comparing Geez from Amharic and English

The above table shows the word **“you”** in English may be plural or singular, female or male and the word **“እናንተ”** column 3 and row 3 in Amharic may be female or male. However, in Geez plural and singular, female and male are representing in different word. Look column 3 and row 4. This indicates Geez language have complex morphology from English as well as from sematic language, Amharic. Look at the following table the word “ ነበረ”/nebere.=ተቀመጠ

In Geez		Subject	In English	In Amharic
ነበርኩ	Neberku	I	I sat	ተቀመጥኩ
ነበርነ	Neberne	We	We sat	ተቀመጥን
ነበርከ	Neberke	You(singular male)	You sat	ተቀመጥክ
ነበርኪ	Neberki	You(singular female	You sat	ተቀመጥሽ
ነበርክሙ	Neberkimu	You(plural male)	You sat	ተቀመጣችሁ
ነበርክን	Neberkin	You(plural female)	You sat	ተቀመጣችሁ
ነበረ	Nebere	He	He sat	ተቀመጠ
ነበረት	Neberet	She	She sat	ተቀመጠች
ነበሩ	Neberu	They	They sat	ተቀመጡ
ነበራ	Nebera	They	They sat	ተቀመጡ

Table1. 3 Geez inflected in 10 pronouns

This table shows in English 6 unique subject (I, we, you, he, she, and they) represent 10 unique Geez words (ነበርኩ፣ነበርነ፣ነበርክ፣ነበርኪ፣ነበርክሙ፣ነበርክን፣ነበርነ፣ነበርት፣ነበሩ፣ነበራ) and 7 unique Amharic words. This indicates Geez language have high inflection.

3. **The existence of phonetic redundancies could create spelling errors;-** The third reason that makes it unique from the previous Semitic language spelling checker is, as mention above in Geez some alphabet (letter) have pronounced the same but different shape and meaning. This letter also uses to write Amharic text. Even though there is a convention to write Amharic text [14] . Currently, most of the people use this alphabet

interchangeably to write Amharic text. For instance, the word tsahay/ፀሐይ can be typed like ጸሐይ, ፀሐይ, ጸሃይ or ፀኻይ. However, according to Merhawit [14] there is a convention to write the word tsahay start by ፀ. For Geez use these letters as interchangeably considers as invalid **(ፀያፍ**) [20] [4] or change the meaning of the word. For instance, the word **ሰረቀ** and **ሠረቀ** are the same pronouncing but different in meaning and shape. **ሰረቀ**/sereqe means he/she stolen and **ሠረቀ**/sereqe means something raises.

4. **Some Geez word written in combination of alphabet and number:** In Geez, some words the collection of Geez number and alphabet, to represent number and abbreviation. For example ፲ወ፪ for 12 and ፳ኤል for **ኤስራኤል**, ፪ኤ for **ክልኤ.** This is also another challenge to develop spell checker application for Geez language. The research conducted on Semitic language previously not fill the gap, that mentioned in the above four challenge .There for the proposed morphology-based Geez spell checker solve the challenge that mention in the above,

At the end of conducting this research, the following basic research questions will have answered

- ✓ To what extent spelling errors in Geez can be detected using dictionary lookup and morphological analyzer?
- ✓ To what extent spelling error in Geez can be corrected using replacement rule and Levenshtein edit distance (LED) algorithm?

1.4.Objective

1.4.1. General Objective

The main objective of this study is to design and implement interactive spelling checker for Geez language using morphology-based approach.

1.4.2. Specific Objective

1. To explore Geez word formation and review related documents for Geez language spelling checker.

2. To build the lexicon and affix file of Geez language
3. To design the General architecture of the proposed system
4. To develop prototype of the proposed system
5. To conduct an experiment(on both error detection and error correction components)
6. To evaluate the performance of the system

1.5.Scope and Limitation of the Study

It is important to identifying the Scope or boundary of the study. In this study, the researcher attempt to collect the data and analyze different related documents in order to design and implement the Geez language spelling checker by considering internal Inflection of and compound words. The proposed study mainly focuses on Non-word error detection and error correction of Geez language. Non-word error is one class of error. It may be multiple/ single insertions, deletions, substitutions, transpositions in single word. After developing the system, the performance of the system has been evaluated.

Geez language has some common abbreviations like ፳ኤል for አስራኤል and ፲ቱ for ዐሥርቱ/10. There for this study consider for this abbreviations. However, real world error have not include in this scope. The first reason to exclude real word error is shortage of time to solve real world error or contextual error. The second limitation is lack of well-organized material in Geez language.

1.6.Significant of the Study

Spelling checker is a tool that used for everyday activity in today's generation. Students, teachers, organization and anyone who wants to learn and prepare documents use spelling checker application. In order to save time and reduce error in document preparation spelling checker is mandatory for every language. The proposed study offers different benefits from different perspectives. Some of the benefits are listed below.

- ✓ To motivate for all Geez writers and readers can easily communicate without misinterpretation caused by spelling errors and easily understand Geez literature.

Since, there are so many ancient literatures and book written in Geez language. This literatures and books used as sources of philosophy, creativity, knowledge and civilization both to Ethiopia and the rest of the worlds [6]**.**

- ✓ To organize the ancient literature and books in digital form with in less time and free error
- ✓ For academic exercise in preparation of handout, report, assignment and lecture note. Since Geez course are offered in various colleges, religious centers and higher institutes
- ✓ It uses an input for Researchers who want to conduct on higher-level application of NLP for Geez language such as Parsing, grammar checker, speech recognition, Question Answering, machine translation, text –to speech translation.
- ✓ To minimize spelling error and save time during document preparation
- ✓ Geez language is widely used in every activity of the Ethiopian orthodox Tewahedo church (EOTC) in worship of God. Holy books like bible, anaphora/kidassie/, psalm, miracles, biography of martyrs, wudasie Maryam are reading in Geez language. These ancient religious books are duplicated to address the newly established churches and monasteries. Therefore, correcting Geez spelling error is used to preserve the dogmatic and canonical faith of the religion.

1.7.Methodology

In order to achieve the research objective and provide valid and reliable result identifying the right methodology is a crucial step [14]. There for to conduct this study the following methods and techniques we have used.

1.7.1. Literature Review

Literature review has a vital role for finding the gaps, for exploring the comparison of different approach, detail understanding of the problem, identifying the methodology etc. We have reviewed global and local relevant literature Such as, book, article, conference paper, report and relevant resources from internet. In addition to this, the researcher assessed Geez language components such as phonemes, word classification and word formation.

1.7.2. Data Collection

So far, there is no readymade Geez word dictionary for Geez language spelling checker (GLSC). Therefor the researcher has constructed two dictionaries. One main dictionary contains stem word and another dictionary, which contains affix files. For this study the dictionary built from ሊቀ ሊቃውንት ያሬድ ሽፈራው መጽሐፍ "መጽሐፈ ግስ ወሰዋሰው መርዓ መጻሕፍት"(like likawunt Yared Shiferaw "metsihafe gis wesewasewu merho metsahift) [23].The testing data also directly collected from bible. In addition to this, the researcher communicated with the linguistics experts to understand the structure of the word, inflection and characteristics of the language words.

1.7.3. Implementation Tool

For implementing the proposed system, the researcher used Python programming language for designing prototype of GLSC. The major reasons for choosing a Python programming language are: the first reason, it is an object oriented programing language which is suitable and flexible for text processing and NLP applications. Second, the researcher has a good programming knowledge and skill in the selected programming language. Additionally, the researcher used another software and tool. Like

Sublime text editor:- used for developing dictionary and affix file.

Microsoft word office 2010:-for document preparation

Hunspell:-a tool used for morphological analyzer. It is a spell checker and morphological analyzer library and program designed for languages with rich morphology and complex word compounding or character encoding [24].

1.7.4. Evaluation Metrics

To determine the functionality and performance of the system, testing is a mandatory task. In order to measure the performance of the system, we have used recall and precision for spelling error detection and for suggestion, we have used suggestion adequacy (SA).

1.8. Organization of the Thesis

This study organized in six different but interrelated chapters. The first chapter starts by discussing the background of the research, motivation of the research, statement of the problem, objective of the study including general and specific, methodology of the study, scope and limitation of the study and significance of the study are discuss in detail.

Chapter two discusses basic theoretical concepts about the spelling checker, spelling error types, existing spell checking approaches and spell checking technique for Semitic language and other languages and some of the previous related works.

Chapter three discusses Geez language in detail. By including, explain Geez writing system, identify Geez word classification, explore Geez word formation and identify Geez word inflection. Like internal Inflection of words

Chapter four discusses the general architecture and design of the proposed system, the designed Geez spelling error detection and correction algorithm and the rule definition structure designed as an input for the algorithm. Moreover, stored knowledge base components used by the prototype are discusses detail.

Chapter five discusses the testing data collection and preparation for experiment, conduct experiment and measure the performance and discus the result. Chapter six discusses the conclusion of the whole work and future work.

CHAPTER TWO

2. Literature Review

2.1. Introduction

This chapter deals with the basic theoretical concept of type of spelling checker in written text, review previous work for spelling checker in local and foreign language, understand existing spelling checker approach.

2.2. Overview of Spelling Checker

Spelling checker is tools that identify misspelled word in a paragraph or in a sentence and provide an alternative word to correct the misspelled words. The spelling checker application is a basic requirement for any language to be digitized [25].In addition to this, the spelling checker application has a great role to develop higher NLP application. Like machine translation, question answering, Entity named Recognition, Text-to Speech (TTS) system, Automatic Speech Recognition (ASR) system e.t.c [12].

Type of spelling error

Many researchers try to develop spelling checker application using different method for different language. The error encounter in text may be either **Non-word** error or **Real word** error based on the meaning that gives [14].

Non-word error occurs when the user writes incorrect or misspelled word [10]. This type of error does not give any meaning. For example, **"we iss coming"**. The word "**iss**" is invalid word.

Real word (semantic) error occurs when a user writes correct word but not give meaning in the context. It is difficult to identify the word is wrong or correct without some contextual information [10]**.** For example, "**we is coming".** The word **"is"** is correct word in English language. But not have a meaning in the context of the above sentence. Because of in the

above sentence the subject/ we/ indicates plural person. But the verb “is” used for singular subject.

According to Merhawit sh [14] the spelling error also grouped in to three based on the error patterns produced .These are topographic error, cognitive error and phonetic error.

I. Topographic error

This type of error occurs the correct spelling of the word known by the writer but the writer make mistake unknowingly. The cause of this error mostly related to wrongly key press [13]. This type of error may be encounter in single character of a word or in multiple character of a word [14]. Typographic errors fall into one of the following four categories, such as substitution error, deletion error, insertion error, and transposition error.

II. Cognitive error

This error occurs when the correct spelling of the word is not known. In cognitive error the pronunciation of misspelled word is the same or similar to the pronunciation of the correct word [17].Cognitive error encounter especially in languages having homophone alphabet [4].

III. Phonetic error

Phonetic error one class of cognitive error [26].It occurs when a user substitute a character with its phonetically equivalent character [14] [4].

2.3.Technique of Spelling Checker

The spelling checker application has two main phases. This are error detection and error correction phases [18].Error detection concerns to identify wrongly spelled words in the paragraph or checking either the word is found in the dictionary or not. Whereas error correction concerns to provide suggestion or an alternative word in order to correct the misspelled word.

2.3.1. Error Detection

Spelling error detection is process of identifying misspelled words from a sentence or paragraph by using error detection technique. Dictionary lookup and N-gram analysis are the most common existing error detection techniques [10] [4].

A. dictionary lookup

Dictionary lookup technique used to check every word of input text present in the dictionary or not. If the word presence in the dictionary, then that word consider as correct word. Otherwise, put in to the list of error words.

All words with their inflection and derivation of the word store in the dictionary then, check each input text from the dictionary takes many times. As result, the dictionary looks up technique needs large amount of memory size and may take long time to search. Therefore, the drawback of this technique is needs more space and inefficient search time [15]. Because of practically, it is impossible to store all word with their inflection and derivation for morphologically rich language. The most significant searching techniques for dictionary lookup are hashing, binary search and finite state automata [27] [14].

Hashing involves searching for an input string from a pre-compiled hash table via a key. In the context of spelling checker, if the word stored at the hash address is the same as the input word, the input string marks as correct. However, if the input word and the retrieved word are not the same or the word stored at the hash address is null, the input word marked as a misspelled. Hashing technique provide fast access [26].

Another predominant method in dictionary lookup technique is binary search technique [28] [29].This method performs by using median split. Median split tree used to access high-frequency words faster than to low-frequency words without sacrificing the efficiency of operations.

Finite State Automata (FSA) [14] method is another dictionary lookup method. This method used to representing a language, as a set of strings with a sequence of notations of some symbol or script [30].

B. N-gram technique

N-grams are a method to find misspelled words in the text [31].It is a contiguous sequence of n items from a given sample text. These items can be phonemes, syllables, letters or words according to the application. Where n€ {1, 2, 3…}.If n = 1, 2, 3 they are referred to as unigrams, bigrams, trigrams respectively.

Instead of comparing each entire word in a text to a dictionary, just n-grams are controlled. N-gram technique is work by examining each n-gram in an input string and looking it up in a pre-compiled table of n-gram statistics to ascertain either its existence or its frequency. The advantage of n-gram algorithm they do not require knowledge of the language. Often it is called language independent [15].

2.3.2. Error Correction

After error detecting, the next step in spelling checker is error correcting. Spelling error correction is a process of generating or suggesting closer words to the misspelled word, ranks the suggested words and replaces the misspelled word with the correct one [14].There are two type of spelling error correction [26].These are automatic and interactive error correction. Automatic error correction is the system decides one best correction and automatically replaces the misspelled word. However, interactive error correction the system provide one or more suggestions and the user select one of them. Some of the existing error correction techniques are minimum edit distance, rule-based, similarity key, n-gram-based, probabilistic, noisy channel and neural networks techniques [14] [26]**.**

A. Minimum Edit Distance Technique

This technique stands on counting the minimum number of editing operations (i.e., insertions, deletions, and substitutions) required to transform one string into another. As stated by KAREN KUKICH [27] the first edit distance algorithm was implement by Damerau. This technique is a simple and the most effective technique to generate the suggestion. Minimum edit distance technique compute the minimum edit distance between a misspelled strings and dictionary entry. It have applied to virtually for all spelling correction tasks, including text editing, command language interfaces, natural language interfaces, etc. However, it is not

quite as good for correcting phonetic spelling errors [32].Given two-character strings s1 and s2, the edit distance between them is the minimum number of edit operations required to transform s1 to s2.Minimum distance technique perform by using different algorithm [15].Some of this algorithm are Levenshtein algorithm, Hamming, Longest Common Subsequence [33].

a. **The Levenshtein algorithm**: This algorithm is a weighting approach to appoint a cost of 1 to every edit operations. For instance, the Levenshtein edit distance between "cat" and "dog" is 3 (substituting "c" by " d" , "a" by "o", " t" by "g").

b. **The Hamming algorithm:** This algorithm used to measure the distance between two strings of equal length [34]. For instance, the hamming distance between "bad" and "bag" is 1.Becuase of the corresponding character mismatch (g and d).

c. **The Longest Common Subsequence algorithm:** The longest common subsequence algorithm is a popular technique to find out the difference between two words. The longest common subsequence of two strings is the mutual subsequence.

B. N-gram Technique

As stated by [35] N-gram analysis can work in two ways, either together with dictionary or without dictionary. When n-gram implemented without a dictionary, used to find in which positions the misspelled word occurs. If there is a unique way to change the incorrect word so that, it contains only valid n-grams, this is taken as the correction. The performance of this technique is limited. It is simple and does not require any dictionary. Together with a dictionary, n -grams are used to define the distance between words, but the words are always checked against the dictionary [35] [26] [31] [25].

C. Rule Based Techniques

Rule based technique works by defining a set of rule for common spelling errors patterns in the form of rule for transforming misspellings into valid words [14] [27]. This technique needs language expert to define the rule of the language. If a misspelled word falls in the set of

common spelling error patterns, candidate words will generated by applying all applicable rules that can be applied on the identified error pattern. Some of the drawback of rule-based technique is, it needs plenty of manual effort and requires complete grammar rules to cover all types of sentence constructions.

D. Similarity key

The similarity key technique works by mapping every string into a key such that similarly spelled strings will have identical or similar keys. This technique works based on transforming words into similarity keys that shows the relations between the characters of the words. Thus, when a key was compute for a misspelled string it will provide a suggestion to similarly words in the dictionary words. The importance of this similarity technique is, speed [27]. Because it is not necessary to directly compare the Non-word to every word in the dictionary. This similarity can be positional similarity, material similarity, and ordinal similarity [30] [14].

Positional similarity:-This similarity shows the degree of the matching character in two strings is in the same position.

Material Similarity:-This similarity shows two strings consist of exactly the same characters but the order is different.

Ordinal similarity: - This similarity shows the matching characters in two strings are in the same order. Similarity key techniques can found in the Soundex system. Soundex system has invented to solve the problem of phonetic errors [30].

E. Neural network

Another interesting technique and needs in small dictionary word is neural network [32]**.**This technique are potential candidates for spelling correction due to their ability to do associative recall based on incomplete or noisy input. Neural nets have the ability to adapt to the specific error patterns of a certain user's domain because they can be trained on actual spelling errors. Hence, the correction accuracy for that domain is potentially improved.

The back propagation algorithm is widely used in order to training the neural network. Back propagation has 3 layers. These are input layer, hidden layer and output layer. The nodes

within a given layer are not directly connected. However, each node in the input layer is connected to every node in the hidden layer by a weighted link and each node in the hidden layer follows the same fashion and is connected to every node in the output layer by a weighted link. The activities on input and output nodes of the network indicate input and output information respectively. In this input information is represented by on-off pattern (i.e. when a node is turned on, a 1 is indicated; when a node is turned off, a 0 is indicated [30]**).**

In spelling checker, a misspelled represented as binary in n-gram vector may be taken as input pattern. The output pattern is the activation of some subset of m output nodes, where m is the number of words in the dictionary [30] [26].Hidden layer connects with the input and output nodes.

F. Probabilistic Techniques

Probabilistic technique works based on statistical characteristics of the language [32]**.** There are two types of probabilities [30] [33]. These are transition and confusion probabilities.

Transition or Markov probabilities:- They represent a certain given letter will be followed by another given letter. These are language dependent. These probabilities can be estimate by collecting n-gram frequency statistics from a large corpus.

Confusion or error probabilities:- estimate the probabilities that a certain letter substitutes another given letter, given that an error has been made. This probability is source dependent. Because different OCR devices have different techniques and features in order to recognize Characters. Each device will produce a unique confusion probability distribution [30].

G. Noisy Channel

This model assumes a natural language text passes through noisy communication channel [14].

2.4.Performance Evaluation Technique for Spelling Checker

Evaluation technique is a technique used to measure the performance of the proposed system or the model [29]. Different researchers have proposed various criteria for evaluation of a given spelling checker. The existing common performance evaluation metrics for spelling

checker is recall, precision and f- measure [29].Suggestion adequacy (SA) also another evaluation metrics, used to measure the performance of the spelling error suggestion [36].

Recall: - is a measure of completeness.

Lexical recall (R_c): is defined as the number of valid words in the text that are recognized by the spelling checker (i.e. true positives), in relation to the total number of valid words in the text (i.e. the sum of all true positives and false negatives)

$$R_c = \frac{TP}{TP+FN} \quad \text{……. …….(3.1)}$$

Error Recall (R_i): is defined as the number of invalid words in the text that are flagged by the spelling checker (i.e. true negatives), in relation to the total number of incorrect words in the text (i.e. the sum of all true negatives and false positives)

$$R_i = \frac{TN}{TN+FP} \quad \text{…….(3.2)}$$

Precision:- indicates exactness of the spelling checker in catching up and responding to the erroneous words.

Lexical Precision (P_c) is computed by dividing all correct non-flags (true Negative) by the total number of non-flags (i.e. true positives plus false positives)

$$P_c = \frac{TP}{TP+FP} \quad \text{……..(3.3)}$$

Error Precision (P_i): is defined as the number of correct flags (true negatives) in relation to the total number of flags assigned by the spelling checker (i.e. true negatives plus false negatives) gives an indication of the spelling checker.

$$P_i = \frac{TN}{TN+FN} \quad \text{……(3.4)}$$

Where TP- True Positive, TN- True Negative, FP- False Positive, FN- False Negative(see chapter five section 5.4.1 for detail)

F-measure: this also another performance measure that combines Recall and precision into a single measure of performance, this is just by taking into account the product of Precision and Recall divided by their sum

$$\boldsymbol{F\text{-}measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad \text{.....(3.5)}$$

2.5.Related Works

This section explains and summarizes the previous works on spelling checker. Many researchers tried to develop spelling checker application for both global and local language in order to minimize spelling error problem.

2.5.1. Spell checker for foreign language

2.5.1.1. Arabic Spell Checker

The authors of this study try to develop for an Independent Spell-Checking System from the Arabic Language Vocabulary. This study introduces morphological analysis concept in the Levenshtein algorithm using small size dictionary that contains Arabic language stem words [37]**.**These paper are mainly focus on comparing and evaluating their method with Levenshtein approach.

In order to compare the new approach with Levenshtein's, the researchers focus on the following three indicators. Those are-The correction average time, the rate of rectified words and the size of each system lexicon. According to his experiment the average correction time for the proposed approach is faster than the average correction time for Levenshte in distance is , the average correction rate by the proposed system is 33.7% higher than Levenshtein distance and the size of the lexicon in proposed system less than that of Levenshtein distance.

The authors concluded that the result of proposed approach highly positive and much better than that of Levenshtein distance. Due to this fact, the new approach appreciates the validity of new concept and shows the importance of their new approach.

2.5.1.2. Spelling Checker for Marathi

These authors [21] try to develop morphology-based spelling checker for Marathi language. For Indian languages such as Marathi and Hindi have complex morphology. For example, for a single noun in Marathi, over 200 words are formed that are either adjectives or adverbs. Similarly, a verb may exhibit over 450 forms [21]. Due to this reason, for morphologically complex language not maintain in the dictionary-based approach. Because it is impossible to store all root words with their possible inflection word in the dictionary. In addition to according to the author, there is no readymade corpus for this language. Due to this reason, corpora-based spelling checker may be achieving poor performance. The author collects 13,000 root words and classifies in to part of speech, and then apply morphological rule for all part of speech.

The author explain morphological rule for all class of part of speech such as postposition morphology, noun morphology, pronoun morphology, verb morphology, adjective morphology, Adverb morphology, Conjunction morphology and Interjections morphology. The performance of this work is 99.7%.The author conclude that the reason for traced this error is missing incomplete vocabularies and exceptional cases.

2.5.2. Spell checker for Ethiopian language

2.5.2.1. Afaan Oromo Spelling Shecker

a. Design And Implement Morphology Based Spell Checker for Afaan Oromo

Gaddisa Olani [9] try to design and implement of morphology based spell checker for afaan Oromo language.

The author uses morphology-based spell checker that combines dictionary look-up with morphological rules. The system architecture of this study contains Tokenizer, Error detection, Error correction, knowledge base, Morphological analyzer, Morphological generator, Suggestion ranker and Word Assembler components.

The tokenize component divide block of text in to individual words, digits and punctuation marks. The knowledge base component stores lexicons, affixes and rules of the language. Error detection receives a word from the word tokenize and looks for the word inside the root word. If the word found in the root word, it marks as correct word and no need to further process. If the word not found in root word, pass to Morphological analyzer. The Morphological analyzer components accept list of words from error detection and decompose in to root word and affixes and then pass to the error detection to check whether the decompose root word and affix are valid or not.

After affix stripping in morphological analyzer the error correction component, classify either the error encountered from affix or from root word. Then, make closer correction based on their class using LED algorithms. Morphological generators rebuild the decompose parts of words (morpheme). The word assembler to combine correct word with the one flagged as misspelled word. Finally, the evaluation of the proposed system resulted in 100% error recall, 88.62% lexical recall and 28.62% precision.

2.5.2.2. Amharic Spelling Checker

a. Automatic Amharic spelling error detection and correction using hybrid approach

Getinet A [18] developed spelling checker for Amharic language. This researcher tries to identify non-word error in the document and after identifying, correct them. To achieve his objective the author uses the Metaphone algorithm for error detection and edit distance algorithm for error correction. The main component of this study is data preprocessing, spelling error detection, spelling error correction and ranking.

The author use python programming for developing prototype that works single word at a time. In addition to this, the author uses VB programming language for integrating spelling checker with word processing. The prototype accepts a candidate word and checks the validity of the words against the dictionary of words. If it the word not found the candidate word mark as misspelled and the system will suggest closer words from the dictionary using edit distance algorithm.

The author collects about 125,000 words in order to construct dictionary and 500 words for testing. From this, 100 of them misspelled words. After prototype development the whole system tested by using testing data. Based on this, the author gets 98% effectiveness in its system error detection and correction functionalities.

The author concluded that the proposed system is effective and accurate in its functionality. However, the processing speed was not considered in this study. It needs improvement for future research. Then, the author recommends improving the processing time and integrating it with other Microsoft office systems in future works.

b. Multilingual Spelling Checker for Selected Ethiopian Languages

Wubetu B. [16] try to develop Multilingual Spelling Checker for Selected Ethiopian Languages such as Amharic, Afan Oromo, Tigrinya, Hadiyyisa and Awngi. This work done by using dictionary based approach and hashing. The proposed system marked as the candidate word is correctly spelled word; if the candidate word matches the word at the hash address. Otherwise, the candidate word is incorrect. The author build corpus-based dictionary for each language. This author collect 993,072 for Amharic,866,328 for Afan Ormo,966,328 for Tigrigna,987,176 for Hadiyyisa and 676,534 word for Awngi.

The performance of the system measured using evaluation metrics precision, recall and F-measure. There for the average performance of the system is 81%. The author concluded that the suggested approach is able to detect diverse classes of spelling errors in the mentioned six selected languages.

c. Automatic Spelling Checker In Amharic language

Melkamu T [10]try to design and develop an automatic Amharic spell checker integrated with Open office word processor. In this study the author, mainly focus on non-word spelling error detection with considering internal inflection of words, repeated words and compound words of Amharic word. In this paper, the spelling error correction is not included.

The author uses Hunspell tool for morphological analysis. Hunspell is morphological analyzer library and program designed for languages that have complex morphology. According to this

study, the spell checker has five components this are, input component, normalization, error detection, morphological analyzer, error correction and suggestion components.

The input component focus on tokenize the text file in to word and remove punctuation from file. Normalization identifies characters with syllographic redundancy or homophone alphabets that can be used interchangeably. Example ሀገር፣ሐገር፣ኀገር must change in to one common word.

Error detection this component search the candidate word inside the dictionary. If the word is found, it consider as valid word. Otherwise, it will be pas to morphological analyzer components. Morphological analyzer component receives the word that is not found in dictionary from the previous component, splits it into stem, and affixes. The author recommended to future research; need to full of error detection and error correction of Amharic word. However, in this paper, the author not conducts experiment on error correction component.

In order to evaluate the system, the author conducts five experiments. For the first experiment, the text had taken directly from Amhara Science Technology and information communication commission 2009 annual report. From this, source 199 Amharic words out of which 5(2.5%) words are misspelled. From this experiment, all misspelled words are detected by the system. and one correct word mark as misspelled.

For the second experiment had taken data directly from Amhara national regional state science technology and information communication commission annual report and from this source the author collected 840 Amharic words out of which 16(1.9%) words are misspelled. The result of this experiment all misspelled words are detected by the system and 20(2.4%) correct words are marked as misspelled. The third experiment, 181 words taken from afar Region ICT 2009 annual report out of which 7(3.9%) words are detected as misspelled. From this, all misspelled words are detected but 3 correct words are marked as incorrect.

The fourth experiment, 94 words taken from Harari Region ICT 2009 annual report out of which 9(9.5%) words are detected as misspelled. The result of this experiment all misspelled word is detected and 9.5% correct word are marked incorrect. The fifth experiment, 94 word

taken directly from experiment 4 evaluated by language expert out of which 7(7.4%) misspelled. From the last experiment precision and recall 97.75 and 92.55 respectively. The average performance of the system shows 97.4%

d. Amharic spelling detection and correction system morphology based approach

Merhawit Shi [14] present spelling checker for Amharic language using morphology based approach. This work is focus on Non-word spelling error detection and correction for Amharic word. In this work, rules are generated by analyzing the morphological behavior of the language. To formulating the rule for the morphology of language needs linguistic expert [15].The author develop morphological rule manually. This rule contains different parts of speech like nouns, adjectives, verbs, pronouns, and compound words.

The architecture of the system contains three components and three. These are preprocessors, error detectors, and error correctors. In addition to that, the architecture contains three knowledge bases. Those are morphological rules, dictionary lookup, and a database holding Amharic characters. The dictionary is one of the knowledge bases that stores stem words of Amharic language. The stem word may be noun, verb, adverb or adjective. If searching a word in the dictionary is empty, then the candidate word split into n-grams (unigram, bigram, and trigram) and checked in the morphological rules, which are used to derivate and inflect a word into different forms. All morphological rules and affix files are stored in morphological rules knowledge base. The author develops 717 morphological rules. The Amharic character database stores every Amharic character.

Preprocessing is the first phase of the architecture. This phase contains tokenization. Tokenization works by splitting block of text in tow word, punctuation mark removal except hyphen used for identifying compound word and replicated word removal. The second component is error detection component. Under error detection component three main activities were performed. Those are N gram splitter, stem extractor and rule filter. The error detection component receive candidate token (word) from preprocessing component and divide in to N-gram (unigram, bigram, and trigram) and comparing them with the defined affix rules. After that select the rule and extract the probable stem word. If the derivation of the root

word matches with the rule, then the word is flagged as valid otherwise it will be an invalid word.

The last component is the error correction component. This component have responsible to provide suggestions for the misspelled word by taking rule weight, location of the error, and the invalid word itself from the stem extraction and rule filter.

The challenge of this approach is time consuming to test every possible scenario in the language. Moreover, processing on a single word passes a long path starting from searching from dictionary, if not found split it into n-grams and comparing each n-gram with the derivation rules containing suffixes, prefixes, conjunctions and prepositions.

Finally, the author measures the performance of the spell checker prototype by using precision, recall, and accuracy. In order to test and evaluate prototype the author prepare two test cases. The first test case contains1551 valid words and 173 invalid words. The second test case contains 88 unique words Amharic bible. The result obtained from the evaluation is 96% precision, 99% recall, and 95% predictive accuracy.

2.5.2.3. Tigrigna Spelling Checker

a. Spell checker for Tigrigna language using rule based morphological analyzer and unsupervised approach

Berihun Hadi [3] try to develop spelling checker for Tigrigna language using rule based and morphological analyzer and unsupervised approach. The author performs two experiments. The first experiment conducted on rule-based dictionary look up and morphological analysis. Even though the language and size of data is difference, the dictionary and affix file preparation of this experiment is similar Gaddisa Olani as mention above [9] . Knowledge Base builds by adopt Hunspell dictionary and affix file format. It contains all information (morphological rule) and dictionary file.

The second experiment also conducted on unsupervised Morphological based approach using Morfessor tool. Unsupervised approach learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. In order to

conduct the second experiment the architecture containing, input component, morfessor tool, morfessor model, Viterbi algorithms, morfessor segmentation model, Knowledge base.

The input component is the contains Tigrigna text. morfessor tool: is a tool that used for morphological segmentation of a word in unsupervised machine learning. morfessor mod, Viterbi algorithms, morfessor segmentation model, Knowledge base in this experiment knowledge build from morfessor segmentation mode.

The author uses ½ million words collected from different source and extracted using Text STAT software. From this word use 787 words for testing the system and conduct two experiments as mention above. Based on the result of the experiment rule-based approach out performs well than the unsupervised approach. The author conclude that increasing the dictionary and morphological rule to the knowledge base and increasing corpus size to the unsupervised approach will increase the performance of the proposed methods.

2.5.2.4. Geez Spelling Checker

a. Context-Dependent Spelling Error Detection and Correction For Geez Language

Aleka T. [4] Try to developed Context-Dependent Spelling Error Detection and Correction for Geez language. In my knowledge, this work is the first work for Geez language spell checker. This study designed by using design science research methodology. The study focuses the problems of using homophone Geez alphabets incorrectly, that they bearing completely different meanings in a given sentence. In Geez, some alphabets have the same sound different shape and meaning called homophone alphabet [7] [4].In order to meet the objective of this paper, the author uses N-gram techniques. The proposed context-dependent spell checker architecture contains three main components: text preprocessing, error detection, and error correction.

Text processing component responsible to remove special character, punctuation marks, geez numbers and Arabic number. In addition to split in to sentence level using full stop (።) and question mark (፧ ?). For each sentence, a statement beginning (<s>) and ending with (</s>) are added to differentiate a sentence from another. In addition to that, the segment sentence split into different tokens to create unigram, bigram, and trigram using white space as a separation mark.

In the error detection component the author proposed and implemented an n-gram (bigram and trigram) language model to detect and correct context dependent spelling errors that occurred in the Geez language. To detect the error, the error detection algorithm checks the presence of every trigram [*w*i-1, *w*i, *w*i+1] and bigrams [*w*i-1, *w*i] by looking up into a precompiled trigram and bigram language model. Back off smoothing technique is use to eliminate the zero probabilities of unseen n-gram and estimates the probabilities higher-order model by also making using of information from the lower-order (n-1 gram) model.

In error correction component provide suggestion by ranking them based on their probability values. The most probable word pops at the top. Firs algorithm checks whether the word contains homophone Geez alphabets or not and the word is correctly spell or not based on bigram and trigram language model. Then, the algorithm generates all possible word combinations (bigrams and trigrams) and computes the most probable correction.

The performance of the system has measured by precision and recall. From 16918 sentences for training 650 sentences randomly selected for testing data. The author conduct bigram and trigram experiment. Based on his experiment the author concludes that the trigram-based context-dependent spell checker scored better performance than the bigram-based experiment.

2.5.2.5. Morphological Analyzer for Geez language

a. Morphological analysis for Geez language using memory based learning

Ytayal Abate [38] try to develop a morphological analyzer for Geez language using one of the supervised machine learning approaches known as memory-based learning. The proposed system is focus on only Geez verbs and trained with manually annotated sample verbs.

The proposed system architecture has two phases. This are training phase and morphological analysis phase. In the first phase of the architecture involves morpheme annotations, to detect the morpheme from a set of words. Inflected words are segments by prefix, stem and suffix. Each segment marked by their representative notations. In Geez, prefixes indicate many features [7]. Such as negation marker, preposition marker, conjunction marker, Causative marker, causative-reciprocal marker, Reflexive marker and reciprocal.

The representative letters represent those features. The annotated words are stored in a morphological database. Then, the instances of the annotated word extracted from the morphological database using windowing method. Then, a memory-based learning model is developed using a learning tool TiMBL. Classifier algorithms called TRIBL2 and IB1 are used to construct databases in memory. The suffix also indicates for two features in geez language. These feature either subject marker or object marker.

Once the learning model is developed, the second phase (morphological analysis) continues. In this phase, a new word is analyzed based on previous training data. The word is segmented and represented as instances and they are compared with the training set. The word will be classified as the closest instance. If the inflected word is unknown, its morpheme is identified and the class that shares the most common features is inferred and predicted as the class of the new instance.

The next step is reconstructing the given word into meaningful units. The system searches for similar stem patterns from the stored training set. If it is not found, distance to the most nearest neighbor is calculated. The last step is extracting the root from the stem. This involves removing the vowel in stems for words with more than three characters.

The author annotated manually 1105 verbs to be suitable to TiMBL algorithms. From these annotated verbs, the author extracted 12135 instances automatically. This data set was divided into training and testing data from which 90% for training and 10% for testing.

Generally, the author concludes that IB2 is good at memory usage on both default and optimized settings (with 91.72% and 93.24% accuracy). However, it has low processing speed, which in turn takes more time. On the other hand, TRIBL2 algorithm achieves a little bit different from IB2. It performs 91.19% and 92.31% with default and optimized parameter settings respectively.

2.5.3. Summary of Related Works

Title	author, Year	Approach/ methodology	Strength	Limitation
Automatic spell checker for Amharic language [10]	Melkamu T. ,2020	-Dictionary looks up technique. -The Hunspell tool is an open-source spell checker with a morphological analyzer library.	-The study considers the internal inflections of words, compound words, and repeated words. -Developing affix (prefix, infix, and suffix) rules: inflected words are generated by applying affix rules on the root words. This minimizes the problems of requiring extremely large spaces, searching time, and impracticability of listing all correct words to develop a dictionary. -Integrating the spell checker into an open office word processor	-Real word error detection and correction are not considered. -Even though developing an automatic spell checker is the researcher's aim, it is not implemented spelling correction.
Amharic Spelling Error Detection and Correction System [14]	Merhawit S. , 2010	-Morphology-based approach: for detecting misspelled words and generating suggestions. .	Define morphological rules to derive and inflect Amharic stem words into different forms. -Distance calculator	-Detecting and correcting real word errors are not included. -morphological rules

		-N-gram to split words unigrams, bigrams, and trigrams	algorithm used to find closer stem words	are prepared only for verbs. However, other parts of speech like nouns, adjectives, pronouns, and compound words are not considered. -Defining a complete set of rules for morphologically complex language is an inefficient task.
Multilingual Spelling Checker for selected Ethiopian Languages [17]	Wubetu B., 2020	-Dictionary-based approach with hashing	-The system is able to identify diverse classes of spelling errors. -Preparation of corpus from different source is good thing. -The speed of the system is well b/c Hashing search algorithm provide fast access.	-The system not well performed for the languages that have complex morphology. -Not include real word error detection and correction.
Automatic Amharic Spelling Error Detection and	Getnet A., 2018	-Metaphone algorithm for detection and edit distance for correction of misspelled word	-The proposed system is effective in its error detection and correction functionality.	-Metaphone algorithim mostly effective in less morphological

Correction using Hybrid Approach [18]				complexity language -The proposed system inefficiency in speed -The study not considers real word error.
Design and Implementation of Morphology-based Spell Checker [26]	Fikru T.,2018	-Dictionary look up and morphological analyzer for error detection -Levenshtein edit distance has used for correcting errors. -Hunspell tool for morphological analyzer	Developing affix (prefix, infix, and suffix) rules: inflected words are generated by applying affix rules on the root words. This minimizes the problems of requiring extremely large spaces, searching time, and impracticability of listing all correct words to develop a dictionary. Integrating the spell checker into an open office word processor	-Real word error detection and correction are not considered.
Designing Dictionary Based Spelling Checker for Afaan Oromoo [28]	Abduljebar K.,2019	-Dictionary lookup with hash function and n-gram approach to detect the error and correct	-Using the hash function to minimize the response time of the dictionary lookup -The corpus collected from difrent sources.	-Dictionary based method requires more space for languge, that have complex morphology. -Real word spelling

				error detetion and correction not include
Spell checker for Tigrigna language using rule based morphological analyzer and unsupervised approach. [3]	Berhun H.,2020	-Unsupervised machine learning approach -Hunspell for morphological analysis -Edit distance for error correction .	-Developing affix (prefix and suffix) rules: inflected words are generated by applying affix rules on the root words. This minimizes the problems of requiring extremely large spaces, searching time, and impracticability of listing all correct words to develop a dictionary. -Integrating the spell checker into an open office word processor	-Real word error detection and correction are not considered. -developed rule not concider infix and circumfix affixes
Context-Dependent Spelling Error Detection and Correction For Geez Language [4]	Aleka T.,2022	N-gram model	-Corpus preparation from different source -This work is the first attempt for Geez languge.	-The proposed systems only focus on homophone error. -This work not attempts to solve non-word error and real word error. -Addtinally, this work did not cover Geez abbreviations.Becuase of Geez language have many

				acceptable abbreviations.

Table2. 1 Summary of literature related works

CHAPTER THREE

3. Overview of Geez Language

This chapter explains the theoretical concept about Geez language. Especially, elaborates the historical background of Geez language, its scripts, writing system as well as its linguistics properties such as syllable structures, phonetics and morphology.

3.1.1. History of the Geez language

Ethiopian scholars [20] the word Geez has two meaning due to the variation of the letter. The first meaning of the word Geez is the first and foremost language when we write with as "ግዕዝ/giezi". For instance Arabic language belong to the people who live in Arab, French language belong to the people who live in France. However, Geez does not expect this practice [39], the first and foremost language. Geez does not describe any tribe (ንሳ), the language of all [39].The second meaning is the language spoken by free people when we write as "ግዕዝ/giezi".

Around 3600BC, Semitic peoples who live in the South Asia migrate to South Arab and Yemen then they migrate to Ethiopia, construct village at Aksum, develop, and spread their language around the country. At that time the most dominant and well-known languages are Saba (ሳባ) and Geez (ግዕዝ/gizi) [40]**.** Especially until Zagwe dynasty (13^{th} centuries), Geez becomes the only national language in Ethiopia, and used as governmental language. For this evidence, many stone inscription and monument inscription have found in Aksum. After the 13th century, as the remains of Latin were making the metamorphoses into the romance languages, spoken Geez also split into many closely related tongues, mainly Tigrigna in the north and Amharic in the south [5].

3.1.2. Writing System of Geez language

Writing systems shows how to represent sets of symbols to represent the sounds of speech and may also have symbols for such things as punctuation and numerals. In Ethiopia, the only language that has its own alphabet (ፊደል) is Geez. Other Semitics languages like Amharic and Tigrigna adopted these alphabets from Geez language [38]. Before 5 century

BC the ancient Geez writing system from right to left up to Abune Selama like Arabic [41]. Currently, the Geez language writing system changes (i.e., left to right).

In Geez, 26 base alphabet/ፊደል vertically listed with seven columns. Each of the columns are labeled as **ግዕዝ** /geez/ (first order), **ካዕብ** /*kaeb*/ (second-order), **ሣልስ** /*salɨs*/ (third-order), **ራብዕ** /*rabɨʔ*/ (fourth-order), **ኃምስ** /*hamɨs*/ (fifth-order), **ሳድስ** /*sadɨs*/ (sixth-order), and **ሳብዕ** /*sabɨe*/ (seventh-order) of alphabets [39].Look at the following example how to represent current Geez alphabet.

ግዕዝ	ካዕብ	ሣልስ	ራብዕ	ኃምስ	ሳድስ	ሳብዕ
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ

Tatble3. 1 sample example to show modern representation of Geez alphabet

In addition to 26 base alphabet, there are four alphabets which have not the second and the third order. So 26 *7+5*4=202. Totally 202 alphabets are used to type Geez text [38].This alphabet is known as ፊደል/fidel.

Geez language also has its own numerals for representing number.

፩/ አሐዱ/1 ፰/ስምንቱ/8 ፸/ስድሳ/ስሳ/60

፪/ክልዔቱ/2 ፱/ተስዐቱ/9 ፹/ሰብዐ/70

፫/ሠለስቱ /3 ፲/ዐሠርቱ/10 ፺/ሰማንያ/80

፬/አርባዕቱ /4 ፳/እሥራ/20 ፻/ተስዐ/90

፭/ኃምስቱ/5 ፴/ሠላሳ/30 ፼/ምዕት/100

፮/ስድስቱ /6 ፵/አርብዓ/40 ፼፼/እልፍ/10000

፯/ሰብዐቱ/7 ፶/ሃምሳ/50 ፼፼፼/አእላፋት/1000000

In Geez, has no symbol to represent zero [22]**.** Sometime Geez number combined with alphabet. For example, **፲ወ፪ => 12 ፣ ፫ተ => 3**.

3.2.Geez word class

Geez word can categorize into eight: Noun, Adjective, Pronoun, Verb, Adverb, Preposition, Conjunctions and Interjection [42].

Noun

Noun is a name that represents a person, animal places, feeling, things and idea. According to [42] there are different types of nouns. Such as common noun, proper noun, collective noun, abstract noun, pronoun, independent pronoun and suffix pronoun.

The objective of this study is not to explain types of nouns, rather how to derivate and inflect noun and identify the plural marker of noun in Geez language. In Amharic most of the noun have suffix -ኦች/ዎች(Plural marker) [10].But in Geez language there is no simple method in order to change singular noun to plural noun [7]. In addition to adding suffix at the end and prefix at the beginning, there is also internal inflection to change singular noun to plural noun.

Example ድንግል to ደናግል. Generally, there are two way of forming plural forms of noun. These are - 1. Pattern replacement 2. Addition of an ending

Pattern replacement: for example ደብር/ dabr --------አድባር /adbar/ ፤ ሀገር/heger to አህጉር/ahigur ፤ ቤት/bet to አብያት/abyat.

Addition of an ending: for example አመት/amet ---------- አመታት/ametat. In Geez we use the alphabet አ ፤አ……ት ፣ ው ፣ት ፣ል፤ሙ፤ይ to change a singular noun to Plural noun [23] [7] show the following table.

Alphabet(letter)	Singular noun	Plural noun
ል	ኪሩብ	ኪሩቤል
ሙ	አንተ	አንትሙ
ት	ካህን	ካህናት
አ	ደብር	አድባር
ው	እድ	እደው
አ…ት	ገብር	አግብርት

Tatble3. 2 Example of singular to plural noun

In addition to this plural formation of nouns can occur by changing the Fidel to ራብዕ ፣ ሳድስ. For example አንቀጽ to አናቅጽ ፣ ደብተራ to ደባትር ፣ መክሊት to መክልይ ፣ መድሎት to መዳልው. Some nouns end with Fidel ሣልስ ፣ ኃምስ ፣ ሳብዕ add ያት or ያን. For example ደቂ to ደቂያት ፣ አረጋዊ to አረጋዊያን ፣ ቅዳሴ to ቅዳሴያት.

Adjectives

An Adjective is a word that describes nouns or pronoun. It used to identify the behavior, color, shape, character and situation of noun's or pronouns [7]. Example ቀይሕ/red, ዐቢይ/big. In Geez, an adjective encounter in different state. Like in gender, case and number [42]. Most of The Geez adjective can be inflected and derivate to plural number by prefixing "እለ at the beginning and suffixing "ን ፣ያን ፣ ት. Show the following table. Table 3.2

Prefixing	Suffix	Geez Singular	Amharic	Geez Inflected	Amharic
እለ		መኑ	ማን	እለመኑ	እነማን
	ን	ፍንው	የተላከ	ፍንዋን	የተላኩ
	ያን	ሰማያዊ	ሰማያዊ	ሰማያውያን	ሰማያውያን
	ት	ቅድስት	የተመሰገነች	ቅዱሳት	የተመሰገኑ

Tatble3. 3 Example of Geez Amharic adjective suffix and prefix to change in to plural form

Verb

Verb is the word that used to indicate action or state of the sentence. According to [23] Geez verb (**ግስ**) begin by five alphabets (fidel). These are **ግዕዝ** /Geez/ (first order), **ራብዕ** /*rabi*/ (fourth-order), **ኃምስ** /*hamɨs*/ (fifth-order), **ሳድስ** /*sadɨs*/ (sixth-order), and **ሳብዕ** /*sabɨ*/ (seventh-order) of alphabets and finish in two alphabets. These alphabets are **ግዕዝ** /Geez/ (first order), and **ኃምስ** /*hamɨs*/ (fifth-order). The number of alphabet in Geez verb (**ግስ**) is between 2-7 [23].

According to the Ethiopian scholars, the types of verbs described as perfective, indicative, subjunctive and imperative verbs are called ዓብይት አናቅፅ /abeyt anaqtsi/ .This verbs can close the idea of a sentence independently without seeking a help of other verbs. In addition to these verbs, the scholars identify ንዑሳን አናቅጽ /niusan anaqtsi/ which cannot close a sentence unless other verbs are add to them.

As shown in Table 3.4, the tense-mood identified by the Ethiopian scholars are the same as that of those identified by the foreign scholars except that the Ethiopians identified extra moods which are categorized as ንዑሳን አናቅጽ.

Tense-mood Identified by Ethiopian Scholars			By foreign scholar	
No	Tense-mood	Category		
1	ቀዳማይ /ኃላፊ አንቀጽ	ዐቢይ አንቅጽ(abiy anqets)	Perfective	Perfective
	ካልኣይ /ትንቢት አንቀጽ		Indicative	Imperfective
	ሣልሣይ /ትእዛዝ (subjunctive)		Subjunctive	
	የቅርብ ትእዛዝ አንቀጽ		Imperative	
	ምክኒያታዊ አንቀጽ (Jussive)			
	ቦዝ አንቀጽ (Gerundive)	ንዑሳን አንቅጽ /nius anqets/		
	አርእስት አንቀጽ (Infinitive)			
	ሳቢ ዘር አንቀጽ (sabizər			
	ቅጽል አንቀጽ (ḱitsil anqetsi)			

Tatble3. 4 Tense-mood Identified by Ethiopian Scholars [38]

The inflection and derivation rule of verb follows the rule of their root verb. There are 8 type of root (modal) verb [43].The difference of each root (modal) verb based on the following criteria [22]**.**

1. Number alphabet**(የሆያት ብዛት)**
2. Shape of alphabet**(የሆያት አወቃቀር)**
3. And way of pronouncing**(አነባበብ)**

Geez root verb

No	**Root verbs (ግስ አርእስት)**	**No of alphabet(የሆያት ብዛት)**	**Shape of alphabet (የሆያት አወቃቀር)**	**Pronouncing (አነባበብ)**
1	ቀተለ	3	ግዕዝ ግዕዝ ግዕዝ	/ḱətələ/
2	ቀደሰ	3	ግዕዝ ግዕዝ ግዕዝ	/ḱəddəsə/
3	ገብረ	3	ግዕዝ ሳድስ ግዕዝ	/gəbɨrə/

4	አእመረ	4	ግዕዝ ሳድስ ግዕዝ ግዕዝ	/ʔəʔmərə/
5	ባረከ	3	ራብዕ ግዕዝ ግዕዝ	/barəkə/
6	ሤመ	2	ኃምስ ግዕዝ	/śemə/
7	ብህለ	3	ሳድስ ሳድስ ግዕዝ	/bihilə/
8	ቆመ	2	ሳብዕ ግእዝ	/ḱomə/

Tatble3. 5 Geez root verb

A root verb is used as a base for another verb. Geez verbs follow one root verb inflection and derivation rule. So to be clear if a verb similar to the root verb **ቀተለ/qetele,** then the inflection and derivation rule similar to **ቀተለ**. Look at the following derivation and inflection verb **ቀተለ** (qetele).

ቀተለ -- ገደለ (ቀዳማይ አንቀጽ)(perfective)

ይቀትል -- ይገድላል(ካልኣይ አንቀጽ/ትንቢት) (indicative) — ዐቢይ አንቀጽ

ይቅትል -- ይግደል/ ይገድል(subjunctive)

ይቅትል---- ይገድል ዘንድ(Jussive)

ቀቲል (ቀቲሎት) -- መግደል (infinitive)

ቀታሊ -- የገደለ .....Adjective

ቀታሊያን -- የገደሉ …adjective for plural male — ንዑስ አንቀጽ

ቀታሊት -- የገደለች.....adjective for singular female

ቀታልያት -- የገደሉ….adjective for plural femal

ቅቱል -- የተገደለ...adjective

ቅትለት -- አገዳደል..... sabizer

ቀቲሎ.... ገድሎ..... Gerundive

ንዑስ አንቀጽ

ቀታሊ -- ገዳይ.....Noun

ቀትል -- ውጊያ... noun(ጥሬ ዘር)

ስም ሲሆን

If some one knows how to inflect the verb ቀተለ/ qetele, all verbs similar to qetele follow the same rule.So the verb like ነበረ ፣ ወረደ follows the ቀተለ inflation and derivation rule.

Adverb (ተውሳከ ግስ)

Adverb is a word that modifies verbs for such categories as time, manner, place or direction [43]. A word gives additional meaning to verbs. Like how, why, where, etc [44]. In Geez there are eight main type of adverb [43] such as:

- ✓ Adverb of manner (ኩነታዊ) for example ፍጡነ -- በፍጥነት (quickly)
- ✓ Adverb of place (መካናዊ) for example ዝየ -- እዚህ (here)
- ✓ Adverb of time (ጊዛያዊ) for example ጌሠም -- ነገ (tomorrow)
- ✓ Adverb of frequency (የደጊመ ጊዜ) for example ወትረ _ _ በማዘውተር (always)
- ✓ Adverb of certainty (ርግጠኝነትን የሚገሌጽ) for example እሙነ --በርግጠኝነት (surely)
- ✓ Adverb of degree (የከፍታን ገሊጭ) for example ጽድቀ --በአግባቡ (fairly)
- ✓ Interrogative (መጠይቃዊ) for example ማዕዜ -- መቼ (when)
- ✓ Relative (ተዛማጅ) for example በጊዜ

Conjunction (አያያዥ)

Conjunction is a word that links clauses or sentences or to coordinate words in the same clause. ከመ፣ አምሳለ፣ በዘ፣ በእንተ፣ ሀየንተ፣ በይነ ፣አመ are some example of conjunction in Geez [45].

Preposition

According to [44]Geez proposition can divide in to two.

1. Prepositions that fall onto nouns to indicate start and end, direction, comparison, time, place .like እም (ከ)፣ ኀበ (መንገለ፣ ወደ)፣ ከመ (እንደ)፣ በ (በቁሙ)፣ለ (በቁሙ)፣ አሜሃ (ጊዜ) ፣ ውስተ (ውስጥ)፣ አፍአ (ውጪ).
2. The second Prepositions having (of, „s) meaning. Like ዘ ፣ እንተ ፣ እለ -- የ. Some preposition used as conjunction [44]

   Prepositions can be freestanding or prefixed to another POS [42]. For instance: በ /be/, ለ/le/, ወ/we/,ዘ/ze/ e.t.c are prefixed to another POS. ኀበ/hebe/፣መንገለ/Mengele/፣ ከመ/keme/፣ አሜሃ/ameha/ ፣ ውስተ/wiste/፣ አፍአ/afia/ e.t.c are free-standing preposition. The meaning of a preposition can vary depending on the case of its object.

Interjection

Interjection is words that express strong feeling, emotion, or surprise [42]. They are often capable of standing on their own.

Examples

- ✓ ሐዊሳ/hawisa/፣ሐቄሳ /hawisa/’sign of joy.’
- ✓ ለይልየ/leylye/፣ሌልየ/leylye/’woe is me.’
- ✓ ሶ/so/’sign of request, please!’.

3.2.1. Affixation in Geez word

Affixation indicates the addition of affixes on the root/stem words to cause word variations. Geez support all types of the affixes namely prefixes, suffixes, infixes and circumfixes.

3.2.1.1. Prefixes

Prefixes are a morpheme that attached at the beginning of a lexical item or base-morpheme. As described in [38] [46] Geez prefixes used to indicate Verbal-stem-marker, person-marker prefixes, negative marker and basic out breeding phones (አስራው ፊደላት) [5].

Verbal-Stem-Marker Prefixes

This type of prefixes are attached to the front of the base-stem verb to form four more derived stems namely causative, reflexive, reciprocal, and causative-reciprocal. For example the word ቀተለ/perfective, አቀተለ/causative, ተቀትለ/reflective, ተቃተለ/reciprocal,አስተቃተለ/causative-reciprocal.

Look the following table.

No	Tense mode	The Five Stem Types				
		ገቢር (Perfectve))	አገብሮ (Causative)	ተገብሮ (Reflexive)	ተጋብሮ (Reciprocal)	አስተጋብሮ(Causative Reciprocal)
	Perfective	ቀተለ	አቀተለ	ተቀትለ	ተቃተለ	አስተቃተለ
	Indicative	ይቀትል	ያቀትል	ይትቀተል	ይትቃተል	ያስተቃትል
	Subjunctive	ይቅትል	ያቅትል	ይትቀተል	ይትቃተል	ያስተቃትል
	Imperative	ቅትል	አቅትል	ተቀተል	ተቃተል	አስተቃትል
	Jussive	ይቅትል	ያቅትል	ይትቀተል	ይትቃተል	ያስተቃትል
	Gerundive	ቀቲሎ	አቀቲሎ	ተቀቲሎ	ተቃቲሎ	አስተቃቲሎ
	Infinitive(a)	ቀቲል	አቀትሎ	አስተቃትሎ	ተቀትሎ	ተቃትሎ
	Infinitive(b)	ቀቲሎት	አቀትሎት	አስተቃትሎት	ተቀትሎት	ተቃትሎት

Tatble3. 6 five stem verb with tense mode [46]

Person Marker Prefixes

These are prefixes attached in front of the stem verb, to show the indicative, subjunctive and jussive mood verbs to indicate the subject (doer) of the action of the verb. This prefixes are አ፤ ነ፤ተ፤ የ.Not only for this alphabet but also the **ካዕብ** /*kaʔb*/ (second-order), **ራብዕ** /*rabiʔ*/ (fourth-order) and **ሳድስ** /*sadsiʔ*/ (six-order) of the above alphabet is used as person marker prefixes.

For example, the verb ፈቀደ (he liked) has its indicative, subjunctive and jussive forms as **ይ**ፈቅድ (he will like), **ይ**ፍቅድ (he must like) and **ይ**ፍቅድ (for him to like). The alphabet 'ይ' the **ሳድስ** /*sadsɨʔ*/ (six-order) of የ. In both case 'ይ' indicates the subject of the action. So the subject is 'He'.

Negation makers (አሉታ)

Negation marker prefix used to show the negativity of verbs and nouns. Those alphabets are አል and ኢ. Only negative verbs can have the possibility of having two prefixes. The negation marker of the verb always present at the beginning of the verb. Example the word ሀ /present **አልሀ**/absent and **ተፈቀደ**/he is liked by) **ኢተፈቀደ**/he is not liked by).

Basic out of breeding phones (አስራው ፊደላት)

Basic out breeding phones are prefixes that used for making the verbs to be (future tense and past participle).This prefixes are /ይ፣ ት፣ ን፣ እ/

Basic out breeding phones (አሥራው ፊደላት(ይ፣ ት፣ ን፣ እ)		
Past	Future	Past participle
ቀደሰ (he sanctified)	**ይ**ቄድስ (he will sanctify)	**ይ**ቀድስ (he has sanctified)
ቀደሰት(she sanctified)	**ት**ቄድስ (she will sanctify)	**ት**ቀድስ (she has sanctified)
ቀደስነ (we sanctified)	**ን**ቄድስ (we will sanctify)	**ን**ቀድስ (we have blessed)
ቀደስኩ (I sanctified)	**እ**ቄድስ (I will sanctify)	**እ**ቀድስ (I have sanctified)

Tatble3. 7 Basic out breeding phones

Not only this but also the ግዕዝ /geez/ (first order) and ራብዕ /*rabɨʔ*/(second order) of the above alphabet(ይ፣ ት፣ ን፣ እ) is used as basic out breeding phones for some exceptional verbs. Example1 ሐተተ to **የ**ሐትት/ Example2 አቀመ to **ያ**ቀውም. There for the alphabet "የ "and "ያ" the ግዕዝ /geez/ (first order) and ራብዕ/*rabɨʔ*/(second order) of "ይ" respectively.

3.2.1.2. Suffixes

Those morphemes added at the end of the verbs. In Geez this morpheme shows Gender (Masculine or feminine), Number (Singular or Plural), nearness or farness of the verb. The suffix of the verb in Geez language indicates either subjectivity or objectivity. Suffixing morphemes at the end of a verb to show only the subject Gender, Number and nearness or farness called subjective marker suffixes [7]. In addition to this suffixes at the end of the verb to show the object's Number, Gender, nearness or farness known as Objective marker suffixes.

Subject markers (ባለቤት አመልካቾች/ [ክሙ፣ ክን፣ ት፣ ነ፣ ከ፣ ኩ፣ ኪ/kimu፣ kini፣ ti፣ ne፣ ke፣ ku፣ kī] are used for indicating the subject on verbs [41]. Example the word ቀተልኩ/I kill / the suffixes ኩ/ku indicates the subject I. For detail understand look the following table.

Geez		
Pronoun	Verb/ግሱ	Suffix
አነ	ቀተልኩ	ኩ
አንተ	ቀተልከ	ከ
አንቲ	ቀተልኪ	ኪ
ውእቱ	ቀተለ	ግዕዝ ድምጽ(ኧ)
ይእቲ	ቀተለት	ት
ንሕነ	ቀተልነ	ነ
አንትሙ	ቀተልክሙ	ክሙ
አንትን	ቀተልክን	ክን
ውእቶሙ	ቀተሉ	ካዕብ ድምጽ(ኡ)
ውእቶን	ቀተላ	ራብዕ ድምጽ(ኣ)

Tatble3. 8 Geez subjective suffix adapted from

Object markers suffixes (ተሳቢ አመልካቾች[ሁ፣ ሃ፣ ሙ፣ ሆሙ፣ ሆን፣ ን፣ ዋ፣ ዎ፣ ዎሙ፣ ዎን፣ ካ፣ ኮ፣ ቶ፣ ያ፣ ዮ፣ ዮሙ፣ ዮን/hu፣ ha፣ mu፣ homu፣ honi፣ ni፣ wa፣ wo፣ womu፣ woni፣ ka፣ ko፣ to፣ ya፣ yo፣ yomu፣ yoni] indicate the object of the verb [41] [5].

Pronoun	Geez verb	Subjective suffix	Objective suffix
አንተ	ቀተልኩከ	-ኩ	-ከ
አንቲ	ቀተልኩኪ	-ኩ	-ኪ
ውእቱ	ቀተልከዎ	-ከ	-ዎ
ይእቲ	ቀተልከዋ	-ከ	-ዋ
አንትሙ	ቀተልኩከሙ	-ኩ	-ከሙ
አንትን	ቀተልኩከን	-ኩ	-ከን
ውእቶሙ	ቀተልከዎሙ	-ከ	-ዎሙ
ውእቶን	ቀተልከዎን	-ከ	-ዎን

Tatble3. 9 Geez objective suffix

CHAPTER FOUR

Design of Geez Spelling Checker

4. Overview

In the previous chapters, the researcher describes some of the related works of spelling error detecting and correcting mechanisms for different languages. The main goal of this study is to design and implement interactive spelling checker for Geez language. In this chapter, the researcher has discussed the detail architecture of the system, the algorithm for Geez spelling error detection and correction and the rule definition structure.

4.1. Morphological analyzer for Geez Language

Morphology is a branch of linguistic that deals the word formation and internal structure of the word in the language [8]**.**The smallest unit of word, which has meaning or grammatical function in the language known as morpheme. So to be clear the word "**አቡኪ**" break in to two morphemes **"ኡኪ"** is morpheme and **"አብ**" is another morpheme. Morpheme can divide in two types. These are free morpheme and bound morpheme.

Free morphemes/lexical: this is a morpheme with complete meaning and they can stand alone as an independent word in a sentence. This morpheme does not need to attach to other words to give meaning. Example the word **አብ፣እም** this represents father, Mather respectively.

Bounded morphemes/affix: this is a type of morphemes cannot appear as a word by itself and cannot stand alone, but when connected to another morpheme, gives meaning. For example, the word **አቡኪ**/your father/ the morpheme **አብ** free morpheme were as the morpheme **ኡኪ** is bound morpheme.

As mentioned above Geez have complex morphology.so creating a dictionary having all the words of the language is very difficult task considering the large variations of affix combination in the language. To solve this problem, we use morphological features and dictionary of only the stem words and dictionary of affixes that are going to be attached to the stem word based on the provided rule. For example the stem word ቀተለ(he killed) ቀተሉ፣ቀተላ፣

ቀተለ፣ቀተለቶ፣ቀተልኪ፤ቀተልክ፤ቀተልኩ፤ቀተልኩክ e.t.c. From this example only the stem word ቀተለ will have kept in the dictionary and the other words inflected based on the morphological feature.

If the input word not found in the dictionary, the morphological analyzer split the word in to probable stem word and probable affix.

So far there is no full developed morphological analyzer for Geez language. Due to this reason, the researcher adopts the Hunspell dictionary and affix file format. Hunspell is an open source spell checker. It has been designed especially for language with rich morphology [24]. Hunspell requires two files to define the way of language being spell checked. These files are dictionary file(*.dic) and affix(*.aff) file.

The dictionary compiled from [23] book. The dictionary contains verb, adjective, adverb, noun and preposition. The researchers use this dictionary as base dictionary and add some country name and spiritual common name like ኢትዮጵያ/Ethiopia/፣ሚካኤል/Mikael/፣ገብርኤል/Gebriael/፣ አብርሐም/Abrham/፤ ኢየሩሳሌም/Eyerusalem. The word is listed one word per line followed by forward slash (“/”), then 0 or more flags, which represents the word attributes, for example affixes. Look the following example

```
4

ቀተለ/itLkeiu

አዘዘ/kitzei

ኢየሩሳሌም/WB

ፈተለ/itLkeiu
```

```
SET UTF-8

PFX  WB  Y    4

PFX WB   0    ወበ

PFX WB   0    እም

PFX WB   0    ዘ

PFX WB   0    ለ

SFX    it  Y    2

SFX  it   0     ት      [^ተት]

SFX  it   0     ቶ
```

Figure4. 1 (a and b) Morphological feature (affix and rule) and Dictionary

Figure 4.1a to shows how to build a dictionary. The first entry 4 indicates the number of estimate word in dictionary. Geez word start at the second line flowed by slash and flags. The slash ("/") indicates the end of dictionary word.**ቀተለ**/qetel/፣**አዘዘ**/azeze፣**ኢየሩሳሌም**/iyerusaliem and **ፈተለ**/fetele are dictionary words. **WB,it,k,ei,u** and **z** are flags that points to the rule of affix in morphological feature. For instance, **WB** points **ዘ፣ወበ፣ለ፣እም**. (See fig4.1b).

Geez language supports all affixes such as prefix, suffix, infix and circumfix. An affix file in hunspell may contain many attributes. For example, SET used for setting the character encoding of affixes and stem files. PFX and SFX defines prefix and suffix classes respectively named with affix flags.

At shown in the above fig4.1b **ወበ፤ዘ፣ለ and እም** attached at the beginning of the word (i.e., prefix). **ት** and **ኖ** attached at the end of the word(i.e., Suffix).as shown in fig 4.1a all accepted words for the word "**ኢየሩሳሌም/WB"** is **ዘኢየሩሳሌም፤ወበኢየሩሳሌም፤ለኢየሩሳሌም፤ እምኢየሩሳሌም**.

```
CIRCUMFIX X

PFX  h Y 1

PFX  h  0  ይ /X

SFX  u  Y   1

SFX  u  ተለ   ትል/uh    [ተለ]
```

Figure4. 2 Geez circumfix

In addition to prefix and suffix, Geez also support infix and circumfix. Look in fig 4.2 shows if the word ends with **ተለ** and **ይ** attached at the beginning of the word, and then change the alphabet **ተለ** to **ትል**. For example from the fig 4.1a the word ቀተለ, ፈተለ found in the dictionary, then ይፈትል፣ ይቀትል are valid word.

4.2. **System Architecture for Geez Spelling Checker**

The proposed system has three main components. These are preprocessing component, error detection component and error correction component. Preprocessing component are responsible for accept a block of Geez text from the user, then tokenize each word by using word delimiter, remove punctuation marks, remove duplicate words and passes the candidate words to the error detection component. The error detection component is responsible to accept the candidate words from preprocessing component and check whether the candidate word is valid or not based on the Geez dictionary and morphological feature. The error correction component is responsible for providing suggestion for misspelled word by using replacement rule and LED. Then after provide probable word suggestion, the user will select correct list from the list.

The proposed system has two back end database. These are dictionary and morphological feature. The dictionary database contains Geez language dictionary words, this includes noun, verb, adjective and preposition as mentioned above in this chapter. Morphological feature database also contains affix of Geez and defined morphological rule. Morphological rule used to show how to attach each affix to the stem word to generate valid word (see fig4.1a and b). General system architecture of Geez Language spelling checker are provided in the figure 4.3

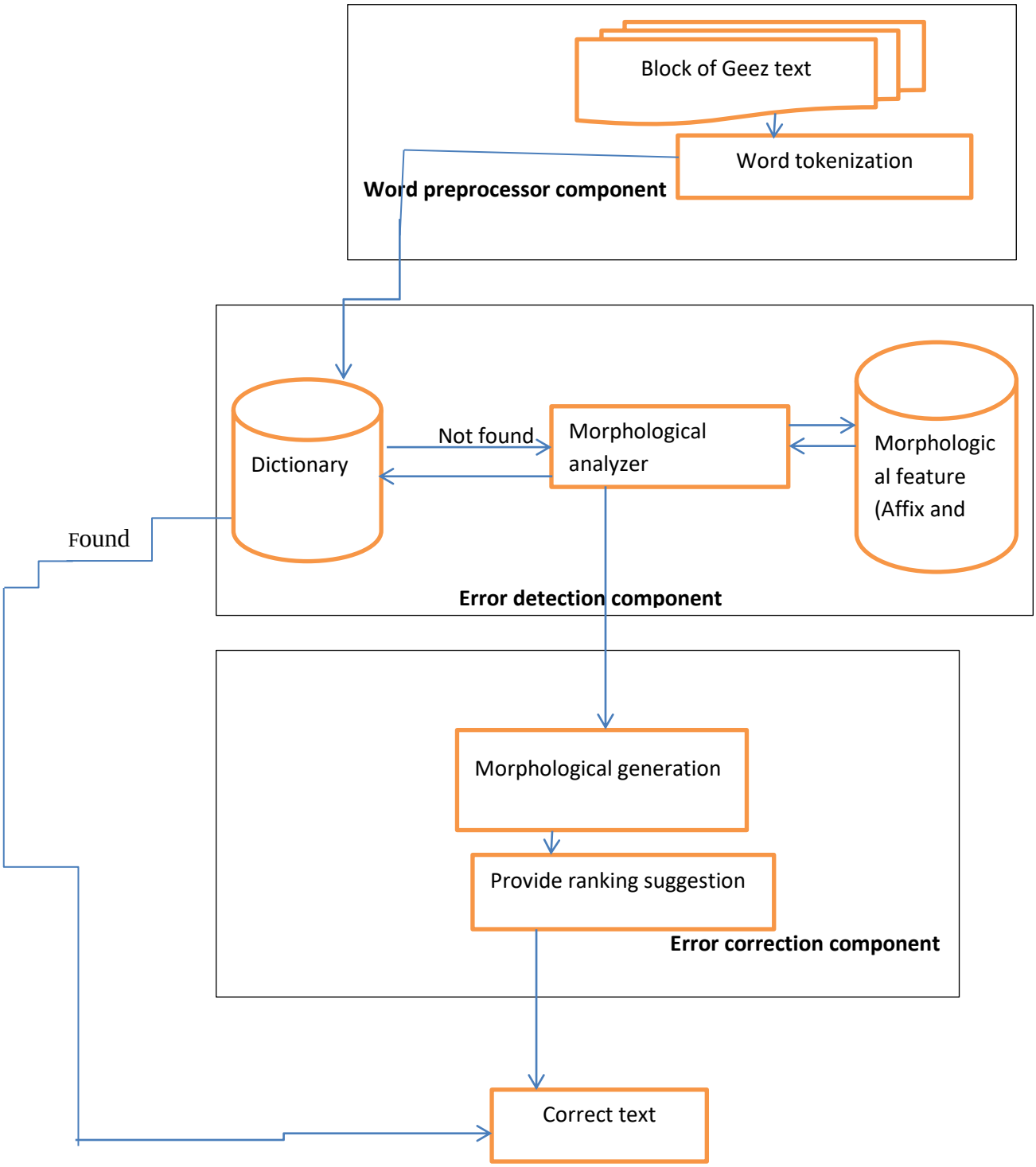

Figure4. 3 General system architecture

4.2.1. **Preprocessing Component**

The first component of this study is preprocessing component. According to modern informational retrieval book [47] there are five types of document preprocessing. These are tokenization, elimination of stop word, stemming, index term selection and thesauri. Form this document preprocessing, only tokenization used for this study.

Tokenization

Tokenization is responsible to split a block of text in to individual tokens. These tokens may be paragraphs, phrases, sentences, punctuation marks, digits or individual words [26]. The text is broken based on some boundary delimiter such as white space and punctuation marks. The Geez language words are separate or ended by blank space or punctuation. After the splitting text is done, all punctuation marks are removed.

In this study tokenization, consider at word level. First read block of Geez texts and split in to word, Then after, the candidate word pass to the dictionary to check either the candidate word is found in the dictionary or not.

Input: Block of Geez Text

Output: List of Tokenize Geez words

1. Empty _list=[ ]
2. Read a character
3. For each character in a block of text up to end of document is reached
4. If a character is punctuation mark or space
5. Skip
6. Else append the read character to empty_list
7. End if
8. End for

Algorithm4. 1 for Tokenization process

The algorithm for tokenization to split a block of text starts working by reading block of text. First, read a character (line 2-4) up to punctuation mark or space is reached. When the process

reached to the space or punctuation mark (line 4), the algorithm will skip (line 5) and store the word in to empty_list (line 6).

4.2.2. **Error Detection Component**

The error detection components detect whether the individual word is valid or not. This component contains stem word in dictionary database and affix with rule in another database. To achieve this component the system performs two main activities. These are dictionary lookup and apply morphological analyzer.

A. Dictionary lookup

Accept the candidate word from preprocessing component and check whether the word is found in the dictionary or not. If the word founds in the dictionary, the candidate word will be marked as valid and no need further process. However, the candidate word not found in the dictionary, it needs further process on morphological feature. Then pass to morphological analyzer. Look at the following algorithm to show dictionary lookup.

Input: tokenize words from preprocessing component

Output: correct or incorrect

1. Read tokenize words
2. For word in tokenize word
3. If the word present in the dictionary
4. Return correct
5. Else
6. Return incorrect

Algorithm4. 2 Dictionary lookup

The algorithm search the word from dictionary (line1-3).If the word found in the dictionary. The word consider as correct word (line 4).But if the word is not found in the dictionary, it consider as the word is incorrect(line 5-6).The word not found in dictionary, pass to morphological analyzer for further processing.

C. Morphological analyzer

The task of morphological analyzer in this study is to accept words from dictionary and decompose these words in to stem word and affix morphemes. Then send the stem word to dictionary and the affix morpheme to affix file to check the valid stem word and valid affix. The researcher develop knowledge base morphological analyzer algorithm based on hunspell dictionary and affix file format. Look at the following algorithm.

Input: word I_word from dictionary

Output: list of affix and stem words

1. Scan input word from right to left and left to right to look for valid suffix and prefix

   For each valid suffix in I_Word strip them and store result in a buffer

   For each valid prefix in I_Word strip them and store result in a buffer

   //pass list of affix and stems to the error detection module

   Return stem and affix

   //If there is no valid suffix and prefix

2. Scan input word from left to right and right to left and then look for a possible stem

   For each valid roots in I_Word strip the root and store result in a buffer

   //pass list of possible stem to the error detection module

   Return stem and affix //valid stem and invalid affix

Algorithm4. 3 Algorithm for morphological analyzer [3]

This algorithm makes use of rules stored in the knowledge base to strip a given word into its stem words and affix. In this process, each individual word have scanned from right to left and left to right in order to get affix and valid stem. The exact affix stripping is possible only for a correctly spelled word. In the case of misspelled word, as there is an ambiguity as to whether the error exists in the stem or affix, only probable affix stripping can be done. To illustrate how the error detection and morphological analyzer works, consider the unknown word **መጽአት**/she came/. The error detection module first check if **መጽአት** is found in the dictionary or

not. Since it is not found in the dictionary, the system cannot automatically say that it is misspelled, for it may be inflected or derivate. Then the word passes to the morphological analyzer. The morphological analysis algorithm will first scan from the right to left and left to right to search for a valid suffix and prefix. Since /-ት/ is a valid suffix in Geez. Now the System checks the unknown morpheme መጽአ for its presence in the dictionary. Hence, it is found in the dictionary. Finally, to determine if the unknown word መጽአት is an acceptable word, all the rules required to append suffix /-ት/will be checked in the affix file. Then, the system will recognize መጽአት as a valid word, and no further process is needed. The same processes have done for a misspelled word.

4.2.3. Error Correction Component

The main function for this component is to provide suggestion for misspelled words. In order to achieve this component, the system perform morphological generation, suggestion and ranking.

A. Morphological generation

Morphological generator is the process of rebuilding of decomposed word that come from error detection module. It is the task of producing the appropriate inflected form of word according to a given morphological features, first accepts a Pairs of morphemes obtained after affix stripping in morphological analyzer. Then classify the error in one of the following classes.

I. Correct stem and correct affix

First, the algorithm scan from left to right and right to left to get stem word and affix. Then the stem word is searched in the dictionary and affix is searched in morphological features, if both the stem word and affixes are found in the dictionary and the combination satisfies the rule, then the word is considered as correct word and no further processing.

II. Incorrect stem and correct affix

If the stem word is incorrect, it is difficult to determine the exact affix that match with the stem word. However, after strip the word, we get a list of possible pairs of stripped stem word

and valid affix. Then, the valid affix shows a specific category of the probable stem word. After getting the possible valid affix, then possible stem word is checked for a word match. If a match found, the possible suggestion are generated for the misspelled word by using the replacement rule and LED. Rules in the knowledge base and LED techniques had used for error correction.

III. Correct stem and incorrect affix

Similarly, in this combination again the word is scanned form left to right and right to left, then the word stripped into a stem word and possible affix. If the stem is valid and the affix is invalid, the valid stem shows a specific category of the possible affix. After that, as mentioned above use replacement rule and LED to get probable correction.

IV. Incorrect stem and incorrect affix

For the case of misspelled word that is occur due to the miss combination of invalid stem and invalid affix, the morphological analyzer tries to split the word into stem and affix, but the stripped stem word and affix does not exist in the provided class and the word is recognized as misspelled word. Some nearest word are displayed as suggestions for that misspelled word by taking nearest stem word and affix that match to this stem word class. Look at the following algorithm.

Input: morphemes from error detection (morphological features and dictionary) component

Output: list of words for suggestion

1. For each error word given from error detection
2. Identify stem word class and affix class of the error word
3. If match found
4. Append affix to stem word
5. Generate list of words
6. End if
7. Return list of words
8. End for

Algorithm4. 4 for morphological generator

D. Provide Ranking and Suggestion

After the morphological generator was done, the next step is providing suggestion for misspelled word and ranking for the suggestions. Replacement rule and LED are used for suggestion and ranking. LED is the most efficient and known algorithm for many language to rank suggestion [3] [26]. For providing suggestion, we have uses maximum suggestion 5 as the threshold value.

CHAPTER FIVE

5. Experiment

5.1. Overview

In this chapter, we have discussed how to collect the test data, detailed implementation of spelling checker and the prototype of the GLSC system. In addition to this, measure the performance of the GLSC system and discussed the challenge and limitation to implement GLSC.

5.2. Testing Data Preparation

Geez language does not have electronic error free text for spelling checker and any other NLP applications. Even though, some of EOTC books available in online and in android mobile applications. Such as EOTC bible, mezigebe haymanot, Wudasie Maryam, Teamre Maryam,Sinksar , But this electronic source have expose to error. Especially homophone alphabet interchangeably error, almost all [4].Due to this fact, we do not use electronic source to build dictionary. Because of preparing error-free Geez dictionary helps to maximizing the performance of spelling checker system. As mentioned in chapter 4, build lexicon of the language from [23] book and identify the morphological feature of each word by linguistics experts.

To evaluate the system, we have taken 3000 words directly from EOTC bible from online. After removing repeated word, we got 1318 unique words. The linguistic expert identify the misspelled words manually, got 185 misspell words. From these misspelled words, the majority errors are encounter on (substitution, deletion and insertion). Look at the following table.

Total Number of Non word error	Cause of error	Total number of error	Percentage in (%)
	Substitution	70	37.8

185	Deletion	32	17.2
	Insertion	26	14.05
	Other type error(transposition ,split word , combined word and mixed)	57	30

Table5. 1 Result of cause of pattern error

5.3.**Prototype of the System**

The prototype of the system developed to show how to implement the GLSC system architecture that presented in Chapter 4. The system developed by using python programming language. The reason for selecting python as development environment is, python has known for its efficiency in the implementation of data processing applications and it can be integrate with hunspell dictionary format. The prototype uses two reference files stored in hunspell dictionary format (i.e., dictionary of the language and morphological feature (affix and rule)).

The prototype worked by accepting input word from the user. After accepting the word from the user, tokenize the sentence in word level and remove punctuation marks by preprocessing module. Then detect the misspelled word. If the word is misspelled, highlight by red color. Look at the following figure to show the prototype of the system for error detection .

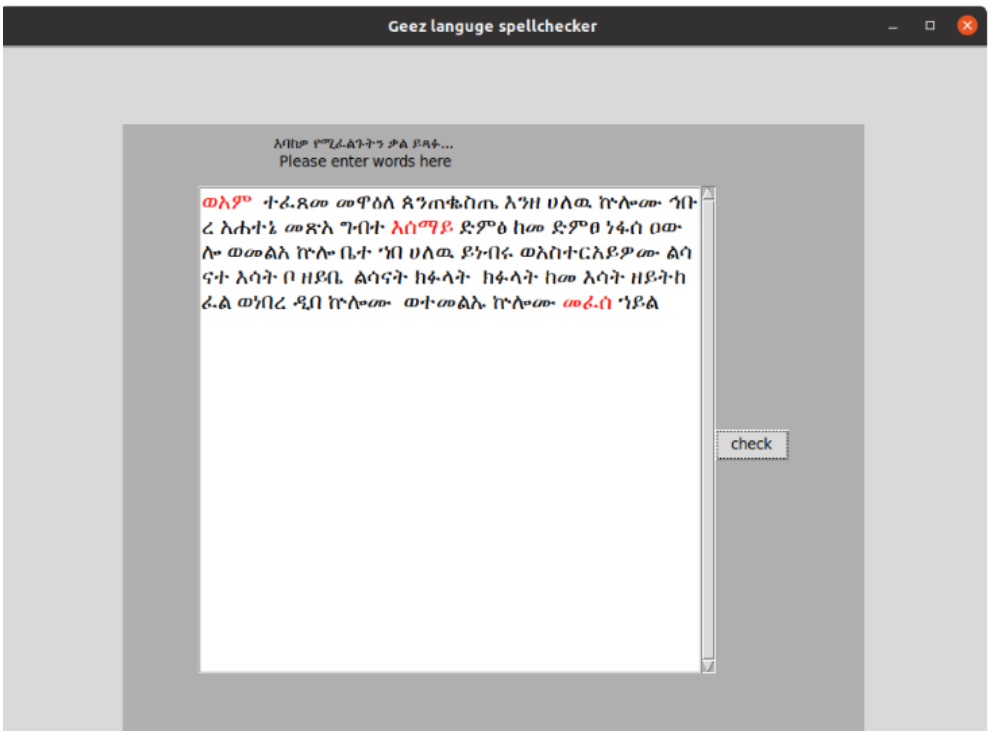

Figure5. 1 sample example of error detection

Fig 5.1 shows, how Geez language spelling checker system detects the misspelled words. The user writes Geez text in the provided space and click **check button.** At that time the system identify the misspelled word and if misspelled is found shade in red color.

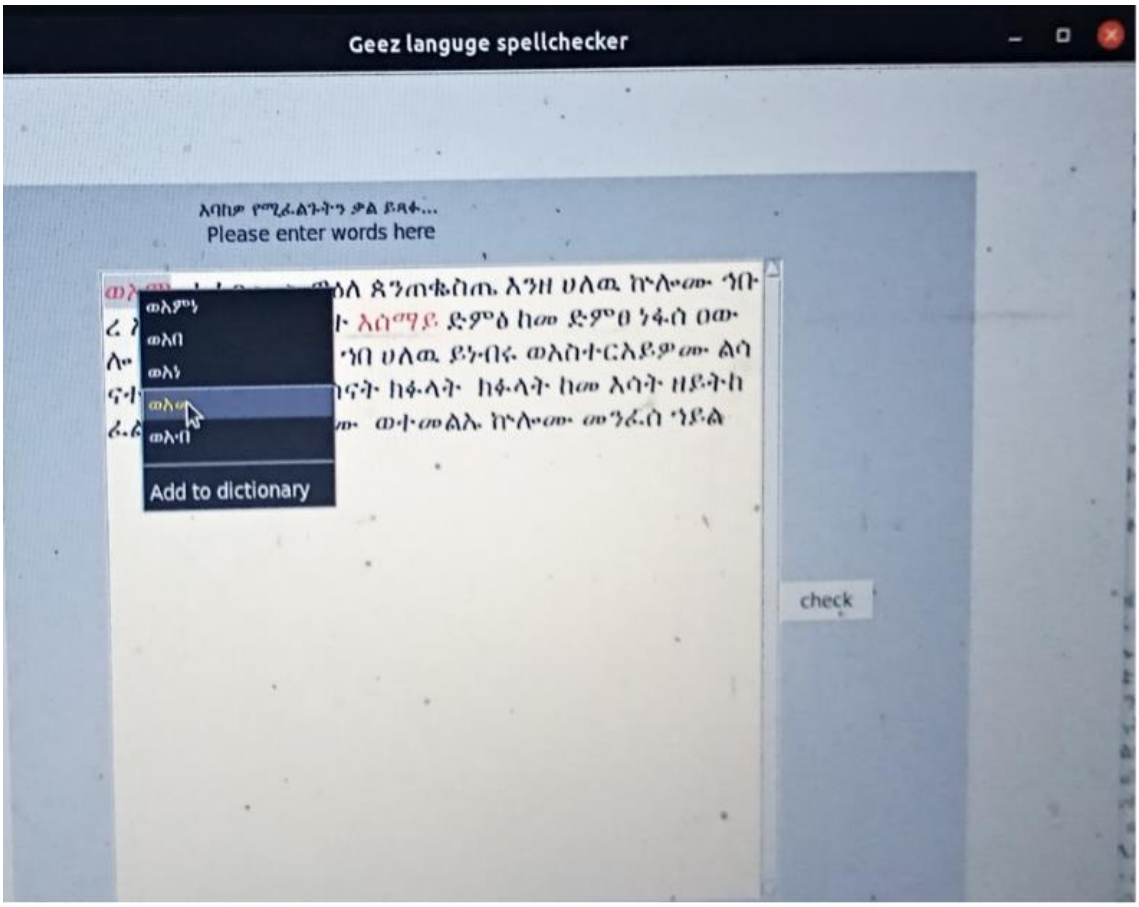

Figure5. 2 sample prototype to rank suggestion

Fig 5.2 shows, how to suggest misspelled word by the system. As mentioned in chapter 2 the first phase of spelling checker is identify misspelled words and the second phase of spelling checker is providing suggestion for misspelled word.

After writing a block of text in provided space, click **check button**. If the word is misspelled words, highlight in red color. Then in order to get suggestion for misspelled word, Double-click a highlighted word to select it. Then right click on the selected word to pop the suggestion .Then choose one candidate correction word.

5.4. Performance Evaluation

To see the functionality and performance of the system, we conduct an experiment on GLSC system. Functionality indicates what the GLSC should do (i.e., error detection and error correction). Performance shows to what extent the functionality of GLSC system achieve the objective. In this work, we conduct two experiments. The first experiment conducted on error detection component and the second experiment conducted on error correction component.

5.4.1. Performance of Spelling Error Detection

The experiment has conducted to measure the performance of the GLSC.As we clearly explain in chapter 4 sub section 4.3.2, the Geez language spelling error detected by error detection component (dictionary lookup and morphological analyzer). So the first experiment conducted on spelling error detection to see the performance of error detection component.

As mentioned above in this chapter sub section 5.2, we have collected 1318 unique words from bible. From these words, linguistics expert identifies 185 words as misspelled words. In order to evaluate the proposed system, we use the following parameters.

- ✓ **True positive (TP)** indicates valid words recognized by the spelling checker as valid
- ✓ **True Negative (TN)** indicates invalid words are detected by the spelling checker as invalid.
- ✓ **False Positive (FP)** indicates invalid words not detected by the spelling checker as invalid.
- ✓ **False negative (FN)** indicates valid words not recognized by the spelling checker as valid

According to this parameter, we can calculate the lexical recall, error recall, lexical precision and error precision. The total number of correct and incorrect word identified by linguistics expert is 1133 and 185 respectively. But the GLSC system detected only 155 words from 185 misspelled identify by linguistics expert, 94 correct words detected by GLSC as misspelled and 30 misspelled words have not detected by GLSC. Look at the following table

Result		Lexical recall	Error Recall	Lexical Precision	Error precision	Accuracy
True positive(TP)	1069	$\frac{1069}{1069+94}$ =91.9%	$\frac{155}{155+30}$ =83.7%	$\frac{1069}{1069+30}$ =97.2%	$\frac{155}{155+94}$ =62.2%	90.07%
True negative(TN)	155					
False negative(FN)	94					
False positive(FP)	30					

Table5. 2 Error detection result

As shown in table 5.2 to measure the overall performance (accuracy) of error detection component, uses the following formula

P= (TP+TN)/(TP+TN+FN+FP)=(1069+155)/(1069+155+94+30)=90.07%

5.4.2. Performance of Spelling Error Correction

The Spelling error corrector component of the proposed system is responsible to provide an alternative word(s) for misspelled word(s) as we mention in chapter four sub sections 4.3.3. In this section, we measure how well the spelling checker can suggest corrections for all true negatives (i.e. incorrect words flagged by the GLSC). To measure the spelling error suggestion we use suggestion adequacy (SA).SA refers to a spelling checker's ability to display the end-user with relevant suggestions for all true negatives.

Although SA often not used to measure spelling checker evaluations, some researchers have already attempted to express the SA of spelling checkers as a metric [48].

$$SA= \frac{\text{No of correct suggestions flagged for invalid words}}{\text{Total No of flagged for invalid words}} \quad \ldots\ldots\ldots (5.1)$$

Look at the following sample example to show how to calculate SA.

Invalid word flagged by GLSC	Total No of suggestion flagged by GLSC	Only correct Suggestion flagged by linguistic expert
በንተ	4	3
ለእግዚኢነ	1	1
ዘተጸውኡ	1	1
ወተፈልጠ	2	2
በሊሰን	2	2
ለትፃምተ	5	3
መላኪተ	3	3
ነጉረ	4	4
በአፈነቢያቲሁ	2	1

Table5. 3 sample example to calculate SA

SA determined by summing all the scores (where S is a score for a suggestion), and dividing it by the total number (N) of true negatives (Tn):

$$SA=\frac{\sum_{k=1}^{n} Sk}{NTn} \quad \text{.....(5.2)}$$

Table 5.3 shows, the GLSC flagged 155 true negative words. In order to correct them, the GLSC suggest an alternative word. As we can see from the results, it is impossible to say that all the words suggested by GLSC are correct words.So we used linguistic expert to identify which word is correct and which one is in correct. For instance, from table 5.3, the first row the word **"በንተ"** is misspelled word. GLSC suggest 4 words for this word. From these suggestion, only 3 are (በእንተ/beente፤በነት/benete፤በቤተ/bebiete) correct suggestions and one is (ብንተ/binte) incorrect word .There for We use SA to measure the extent to which spelling correction of performance is achieved and got correct suggestion 87.5%.

5.5.**Result and Discussion**

The experimental results show value of lexical recall 91.9%, error recall 83.7%, lexical precision 97.2%, error precision 62.2% and correct suggestions provided by GLSC 87.5%. The overall performance of the system is 90.05%. This shows that the GLSC system perform

well in flagging words as valid/invalid and suggestion. However, needs some improvement to increase the performance of both error detection and suggestion generation. We have identified some of the reason that reduces the performance of the GLSC system. The first reason that out of vocabulary problem. As researcher knowledge, there is no readymade dictionary for Geez language so far. Due to this reason, our dictionary not contains all Geez words. Additionally, the testing data randomly have taken from bible. In this case, some of personal name and place name may include in testing data. For example, the word **ቂሳር**/qisar and **ቄሬኔዎስ**/qierenewos are spiritually recognizable names. However, The GLSC detect as an error. For this reason, some of recognizable personal names or place names flagged as invalid. Therefore, increase the size of the dictionary will increase the performance of the GLSC system.

The second reasons are affix rule mismatching. Geez verb inflected and derivate based on their root verb as mentioned in chapter 2 section 2.1.3.But some verb are follow exceptional rules. For example the root verb ቀተለ(he killed), to indicates future time add prefix ይ at the first and the last two character change in to ሳድስ(sadis)(six order). There for it will be ይቀትል(he will kill).So the Geez word found under root verb ቀተለ follow the same rule. For example, ነበረ/nebere፤ነገረ/negere፤ ፈተለ/fetele follows this rule. The word ሐተተ(he examine) also found in the root verb ቀተለ. However, it follows exceptional rule to inflect and derivate. To indicate future time add the prefix የ instead of ይ. There for it will be የሐትት (he will examine).if we add ይ at the prefix of the word ሐተተ we will get invalid (ይሐትት).There for developed well organized affix rule increase the performance of GLSC.

5.6. **Challenge and Limitation**

The first challenge we face is Geez have complex morphology and some word follow exceptional rule. For example to inflect the verb ቀተለ/qetele/ to show the subject 'I' as the doer of the action, we change the last alphabet to ሳድስ /sadis and add prefix ኩ/ku to get ቀተልኩ(I killed).Most words follow the same rule of changing the order of their last alphabet to ሳድስ /sadis and add prefix ኩ/ku. However, there is an exception for verbs ending in ቀ, ገ, ከ. For example, applying the same derivation to the verbs 'አጥመቀ'/atimeqe/, 'ሐደገ/hedege',

‘ነሰከ/neseke’ results in ‘አጥመቅኩ’, ‘ጎደግኩ’, ‘ነስክኩ’ respectively. These are invalid words. These and other exceptions in Geez language make the system flagged invalid word as correct and valid word as incorrect.

The second challenge we faced is, lack of error free corpus to build lexicon of the language. Manually reviewing the collected Geez data from book is very tiring and time-consuming task.

CHAPTER SIX

6. Conclusion and Recommendation

6.1 Conclusion

Spelling checker is an NLP application, used to detect misspelled word(s) in a sentence or a paragraph and provide an alternative for misspelled word(s). The spelling checker is language dependent. (i.e., a spelling checker for one language cannot work for another language). There are different techniques of for spelling error detection and correction. For error detection, dictionary lookup and N-gram analysis are common techniques. Rule-based, N-gram, similarity key, edit distance, neural network, probabilistic and noisy channel are some of the techniques used to provide suggestions.

In this research, the researcher applied morphology-based approach for the implementation of Geez spelling checker, particularly to detect and correct non-word error. The error was detected using dictionary lookup and morphological analyzer and corrected by using replacement rule and LED. We adopt the Hunspell dictionary and affix file format to design a lexicon (i.e. the knowledge base component) and hashing algorithm for searching. Hunspell is an open source spelling checker tool. It has designed especially for languages that have complex morphology. Hunspell needs two files. These files are dictionary and morphological feature (affix) with defined rule. The dictionary saved as Geez.dic and morphological feature (affix) saved as Geez.aff. A dictionary built from 6115 words, and 955 rules have defined from affix file.

The developed GLSC was implemented using python programming language. A user can provide a block of text by typing in space provided. Finally, testing has carried out in order to prove the spelling checker system using sample data of Geez language words collected from bible. To measure the performance of the system, we have used information retrieval evaluation metrics of recall and precision for error detection and suggestion adequacy for suggestion. For testing, 1318 unique words randomly have taken from bible to conduct experiment.

From the conducted experiment, we got the result of lexical recall 91.9%, error recall 83.7%, lexical precision 97.2%, error precision 62.2% and correct suggestions provided by GLSC 87.5%. The overall performance of the system is 90.05%.

Generally, we concluded that the performance of the Geez spelling checker is well with some limitations. Increase the size of the dictionary and develop well organized rule will increase the overall performance of the GLSC system.

Contribution of this study

This study has the following contribution: -

- ✓ This experimental study shows that a morphology-based approach for spelling error detection and correction for the Geez Language with an experimental performance result of 90.05% overall error detection and 87.5% error suggestion is a good candidate for development of spelling checker for the language.
- ✓ This study shows the lexicon and affix of Geez language for error detection and correction
- ✓ We developed a system that detects error words for Geez language
- ✓ We developed a system that provides a list of words a suggestion for correcting misspelled words for Geez language

6.2. Recommendation

We would like to recommend the following key points for further research that helps to improve the performance and include some functionality.

- In this study, we tried to solve Non-word error in Geez language, one can extend it to work on spelling checker that can detect and correct real-word errors by considering Semitics and grammar of the language to make spelling checker system more interactive.
- One of the challenges we faced in this thesis was finding standard corpus and free from homophone alphabet error of Geez language. Some of the corpus that found in online

faced for homophone error. For instance instead of ሐመር write ሀመር or ኀመር is an error. Therefore, future study may need to give attention to the development of standard Geez language corpus that can motivate the researchers to work on any NLP application not only on spelling checker application and minimize time to collect corpus in order to evaluate their system performance by the researchers.

- We develop morphology-based spelling checker for Geez language in limited dictionary and test in limited data. In order to increase the performance of the system, one can increase the size of dictionary and develop well organize rule
- For future research will by including part of speech tagging to overcome the real word error detection and correction problems will increase the performance of the system.
- Integrating the full GLSC with word processor and search engine is a future research.

Reference

[1] M. Abeje, "DESIGNING AUTOMATIC PRONUNCIATION DETECTION FOR GEEZ LANGUAGE," Bahir Dar University(Msc Thesis), Bahir Dar, 2018.

[2] f. Gizachewu, "Developing part of speech tagger for guragginga language," Bahir Dar University(Msc thesis), Bahirdar , 2017.

[3] B. Hadis, "SPELL CHECKER FOR TIGRIGNA LANGUAGE USING RULE BASED MORPHOLOGICAL ANALYZER AND UNSUPERVISED APPROACH," Bahir Dar university (thesis Msc), Bahir Dar, 2019.

[4] A. Tesfie, "Context-Dependent Spelling Error Detection and Correction for Geez language," Gondar University, Gondar University Department of Computer Scienc Msc thesis, 2022.

[5] W. Manaye, "DESIGNING GEEZ NEXT WORD PREDICTION MODEL USING STATISTICAL APPROACH," Bahir Dar University (MSc thesis), Bahir Dar, 2020.

[6] G. H., "Enhanced Morphological Analayzer for geez verb," international Journal of Science & Technoledge, china, 2020.

[7] T. Kass, "Morpheme-Based Bi-directional Ge'ez -Amharic Machine Translation," Addis Ababa,, 2018.

[8] A. Belay, "DESIGNING A STEMMER FOR GE'EZ TEXT USING RULE BASED APPROACH," Ababa University(Msc Thesis), Addis ababa, 2010.

[9] G. Olani, "Design And Implementation Of Morphology Based Spell Checker," *International Journal of Scientific & Technology Research,* vol. 3, no. 12, 2014.

[10] M. Tilahun, "Automatic Speeling Checker In amharic languge," Bahir Dar University(Msc Thesis), Bahir Dar, 2019.

[11] Rajashekara Murthy S,Vadiraj Madi,Ramakanth Kumar P, "A NON-WORD KANNADA SPELL CHECKER USING MORPHOLOGICAL ANALYZER AND DICTIONARY LOOKUP method," *International Journal of Engineering Sciences & Emerging Technologies,* vol. 2, no. 2, 2012.

[12] AmanjotKaur, Dr. ParamjeetSingh,Dr. ShavetaRani, "Spellchecking and Error Correcting System for text paragraphs written in Punjabi Language using Hybrid approach," *International Journal of Advanced Research in Science, Engineering and Technology,* vol. 2, no. 11, 2015.

[13] Baljeet kaur ,Harsharndeep Singh, "Design and Implementation of HINSPELL -Hindi Spell Checker using hybrid approch," *International Journal of scientific research and management (IJSRM) ,* vol. 3, no. 2, 2015.

[14] M. Shimelis, "Amharic spelling detection and correction system morphology based approch," Addis Ababa University(MSc thesis), Addis ababa, 2020.

[15] B. H, "Spell Checker for tigringa language using rule based using morphological analayzer and unsupervised approch," Bahir Dar University(MSc thesis) , Bahir Dar, 2019.

[16] W. Barud, "Multilingual Spelling Checker for Selected Ethiopian Languages," *International Journal of Advanced Science and Technology,* vol. 29, 2020.

[17] W. Barud, "Dictionary Based Spelling Corrector System the Case of Six Ethiopian Languages," *International Journal of Aquatic Science,* vol. 12, no. 02, 2021.

[18] G. Asefa, "AUTOMATIC AMHARIC SPELLING ERROR DETECTION AND CORRECTION USING HYBRID APPROACH," *Journal of Emerging Technologies and Innovative Research,* vol. 5, no. 6, 2018.

[19] S. TEGEN, "OPTICAL CHARACTER RECOGNITION FOR GE'EZ SCRIPTS WRITTEN ON THE VELLUM," UNIVERSITY OF GONDAR, UNIVERSITY OF GONDAR Msc Thesis, 2017.

[20] መ. አ. ተክሌ, መጽሐፈ ታሪክ ወግስ, አዲስ አበባ, 1997.

[21] Veena Dixit, Satish Dethe, Rushikesh K. Joshi, "Design and Implementation of a Morphology-based Spellchecker for Marathi,an Indian Language," Mumbai, Indian Institute of Technology Bombay.

[22] H. Haile, "ግእዝ መማሪያ," dirzon.com, [Online]. Available: www.dirzon.com/Doc/Details/Habtamu Haile%3Aየግእዝ መማሪያ.pdf.

[23] ሊ. ሊ. ያ. ሽፈራው, መጽሐፈ ግስ ወሰዋሰው መርኖ መጻሕፍት.

[24] January 2022. [Online]. Available: http://hunspell.github.io/.

[25] S. M. El Atawy,A. Abd ElGhany, "Automatic Spelling Correction based on n-Gram Model," vol. 182, August 2018.

[26] F. T, "Morphology Based Spell Checker for Kafi Noonoo Language," Addis Ababa University (MSc thesis), Addis Ababa , 2018.

[27] K. KUKICH, "Techniques for Automatically Correcting Words in Text," *ACM Computing Surveys,*

vol. 24, 1992.

[28] A. Kedir, "Designing Dictionary Based Spelling Checker for Afaan Oromo," IMMA, 2019.

[29] T. KUMERA, "CONTEXT BASED SPELLCHECKER FOR AFAN OROMO WRITING," JIMMA UNIVERSITY SCHOOL OF GRADUATE STUDIES DEPARTMENT OF INFORMATION SCIENCE Msc thesis , JIMMA , 2018.

[30] H. L. Liang, "SPELL CHECKERS AND CORRECTORS:A UNIFIED TREATMENT," UNIVERSITY OF PRETORIA,, SOUTH AFRICA , 2008.

[31] Er. Sumreet Kaur Randhawa Er.Charanjiv Singh Saroa, "STUDY OF SPELL CHECKING TECHNIQUES AND AVAILABLE SPELL CHECKERS IN REGIONAL LANGUAGES: A SURVEY," *International Journal For Technological Research In Engineering,* vol. 2, no. 3, 2014.

[32] Neha Gupta Pratistha Mathur, "Spell Checking Techniques in NLP: A Survey," *International Journal of Advanced Research in,* vol. 2, no. 12, 2012.

[33] Ritika Mishra,Navjot Kaur, "A Survey of Spelling Error Detection and Correction Techniques," *International Journal of Computer Trends and Technology,* vol. 4, no. 3, 2013.

[34] "www.geeksforgeeks.org/hamming-distance-two-strings/," geeksforgeeks. [Online]. [Accessed January 2022].

[35] M. S. Nonghuloo, "Spell Checker for Khasi Language," *International Journal of Software Engineering,* vol. 8, 2017.

[36] Gerhard B. van Huyssteen*, Roald Eiselen & Martin Puttkammer, "Evaluating Evaluation Metrics for Spelling Checker Evaluations," North-West University, Potchefstroom, .

[37] MOHAMMED ATTIA1,2, PAVEL PECINA3,YOUNES SAMIH4, KHALED SHAALAN2,and J O S E F V A N G E N A B I T H1, "Arabic spelling error detection and correction," 2015.

[38] Y. A, "Morphological Analysis of Ge'ez Verbs Using Memory Based Learning," Addis Ababa University(MSc thesis), addis ababa, 2014.

[39] ዶ. ለ. ብርሃኑ, የግእዝ መማሪያ, ዋሽንግተን ዲሲ, 2006.

[40] ኪ. ወ. ክፍሌ, መጽሐፈ ሰዋስው ወግስ, አዲስ አበባ , 1909.

[41] T. ABEBE, "DESIGNING AUTOMATIC SPEECH RECOGNITION FOR GE'EZ LANGUAGE," BAHIR DAR, ETHIOPIA , 2018.

[42] M. Kebede, "Development of Part of Speech Tagger for Ge'ez Languag," Addis Ababa, Ethiopia, 2017.

[43] መ. ደ. ቀለብ, ትንሣኤ ግእዝ, አዲስ አበባ: በኢትዮጵያ አርቶዶክስ ተዋህዶ ቤተክርስቲያን ማኅበረ ቅዱሳን , 1996.

[44] B. Abe, "Geez to Amharic Machine Translation," Addis Ababa, 2018.

[45] W. Wolde, "INFORMATION EXTRACTION MODEL FROM GE'EZ TEXT," BAHIR DAR, 2021.

[46] D. Berihu, "DESIGN AND IMPLEMENTATION OF AUTOMATIC MORPHOLOGICAL ANALYZER FOR GE'EZ VERBS," ADDIS ABABA UNIVERSITY SCHOOL OF GRADUATE STUDIES DEPARTMENT OF COMPUTER SCIENCE, 2010.

[47] Ricardo Baeza Yates Berthier Ribeiro-Neto, Modern information retrival, Nework: ACM press.

[48] Menno M van Zaanen and d Gerhard B van Huyssteen, "Various Uses of a Spelling Checker Project: Practical Experiences, Teaching learning," *southern African Linguistics and Applied Language Studies,* 2009.

[49] ዳ. ደ. ቀለብ, ግእዝ ሕያው ልሳን በቀላል ዘዴ, አዲስ አበባ, 1993 .

[50] N. Ytayal, "Context based spell checker for amharic," Jimma Univsrsity(Msc thesis), Jimma, 2016.

[51] Andargachew Mekonnen Gezmu, Andreas Nürnberger, Binyam Ephrem Seyoum, "Portable Spelling Corrector for a Less-Resourced Language: Amharic," in *Otto-von-Guericke-Universität Magdeburg, Addis Ababa University*, Otto-von-Guericke-Universität Magdeburg, Addis Ababa University, 2018.

[52] Atakilti Brhanu Kiros1,Petros Ukbagergis Aray2, "Tigrigna language spellchecker and correction system for mobile phone device," vol. 11, 2021.

[53] Itziar Aduriz, Iñaki Alegria, Xabier Artola, Nerea Ezeiza, Kepa Sarasola and Miriam Urkia, "A spelling corrector for Basque based on morphology," Literary and lingustics computing .

[54] M. Asadullah, "FINITE STATE RECOGNIZER AND STRING SIMILARITY BASED SPELLING CHECKER FOR BANGLA," BRAC University Department of Computer Science and Engineering Msc thesis, Dhaka, 2007.

[55] Bakkali Hamza,Gueddah Hicham ,Yousfi Abdellah,Belkasmi Mostaf, "For an Independent Spell-Checking System from the Arabic Language Vocabulary," *International Journal of Advanced Computer Science and Applications,,* vol. 5, 2014.

Appendix A: Geez Alphabet and Number

First order	Second order	Third order	Fourth order	Fifth order	Six order	Seven order
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
ወ	ዉ	ዊ	ዋ	ዌ	ዉ	ዎ
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ

፩	፪	፫	፬	፭	፮	፯	፰	፱	፲
1	2	3	4	5	6	7	8	9	10
፳	፴	፵	፶	፷	፸	፹	፺	፻	፼
20	30	40	50	60	70	80	90	100	10,000

ቈ	ቊ	ቋ	ቌ	ቍ
ጐ	ጒ	ጓ	ጔ	ጕ
ኰ	ኲ	ኳ	ኴ	ኵ
ጐ	ጒ	ጓ	ጔ	ጕ

Adapted from [5] with little modification

Appendix B: Sample Affix file Rule in Hun spell

#prefix

PFX ei Y 5

PFX ei 0 ኡ

PFX ei 0 ዘኡ

PFX ei 0 ወዘኡ

PFX ei 0 በዘኡ

PFX ei 0 ወኡ

PFX Ys Y 8

PFX Ys 0 ይ

PFX Ys 0 ት

PFX Ys 0 እ

PFX Ys 0 ዘይ

PFX Ys 0 ኢይት

PFX Ys 0 ኢተ

PFX Ys 0 ኢይ

PFX Ys 0 ተ

PFX bz Y 4

PFX bz 0 በዘ

PFX bz 0 ዘበ

PFX bz 0 ለዘ

PFX bz 0 ዘለ

PFX WB Y 7

PFX WB 0 ወበ

PFX WB 0 እም

PFX WB 0 ወ

PFX WB 0 ለ

PFX WB 0 ዘ

PFX WB 0 በ

PFX WB 0 ወኢ

#suffix

SFX pi Y 8

SFX pi 0 ነ

SFX pi 0 ኪ

SFX pi 0 ክሙ

SFX pi 0 ከ

SFX pi 0 የ

SFX pi 0 ት

SFX pi 0 ሁ

SFX pi 0 ሆሙ

SFX Li Y 10

SFX Li ለ ሌሁ ለ

SFX Li ለ ሌሃ ለ

SFX Li ለ ሌክን ለ

SFX Li ለ ሌሆሙ ለ

SFX Li ለ ሌሆን ለ

SFX Li ለ ሌክሙ ለ

SFX Li ለ ሌኪ ለ

SFX Li ለ ሌከ ለ

SFX Li ለ ሌየ ለ

SFX Li ለ ሌነ ለ

#infix

PFX IN Y 11

PFX IN ከዐ ተከዕ ከዐ

PFX IN ም እም ም

PFX IN ከሠ ክሡ ከሠ

PFX IN በጽ ይብጽ በጽ

PFX IN ደመ ድሙ ደመ

PFX IN በዝ ብዙ በዝ

PFX IN ፈለ አፍለ ፈለ

PFX IN ተሰ አሰ ተሰ

PFX IN ነበ አንበ ነበ

PFX IN አንቀ አናቅ አንቀ

PFX IN ጽን አጽና ጽን

PFX ts Y 6

PFX ts ጸሕ ጻሕ ጽሕ

PFX ts ጸን አጽ ጸን

PFX ts ጸን ጽኑ ጸን

PFX ts ተን አን ተን

PFX ts ነበ ነቢ ነበ

PFX ts ድን ደና ደን

Appendix C: Sample Corpus

እምጳውሎስ ገብሩ ለእግዚኢነ ኢየሱስ ክርስቶስ ወሐዋርያ ዘተጸውአ ወተፈልጠ ለትምህርተ ወንጌሉ እግዚአብሔር ዘአቅደመ ነጉረ በአፈ ነቢያቲሁ ወመጻሕፍቲሁ ቅዱሳት በእንተ ወልዱ ዘተወልደ ወመጽአ እምዘርዐ ዳዊት በሥጋ ሰብእ ግብረ ወአርአየ ከመ ወልደ እግዚአብሔር ውእቱ በኀይሉ ወበመንፈሱ ቅዱስ ከመ ተንሥአ እሙታን ኢየሱስ ክርስቶስ እግዚእነ ዘቦቱ ረከብነ ጸጋ ሐዋርያተ ከመ ናስምዖሙ ለአሕዛብ በስሙ ከመ አንትሙኒ ይእዜ ኮንክሙ ጽዉዓን በኢየሱስ ክርስቶስ።

በእንተ ጽዉዓን ወበእንተ ሃይማኖሙ

ለኵሎሙ እለ ሀለዉ ብሔረ ሮሜ ፍቁራኒሁ ለእግዚአብሔር ወኀሩያኒሁ ወቅዱሳኒሁ ሰላም ለከሙ ወጸጋ እግዚአብሔር አቡነ ወእግዚእነ ኢየሱስ ክርስቶስ አቀድም አእኵቶቶ ለእግዚአብሔር በኢየሱስ ክርስቶስ በእንተ ኵልክሙ እስመ ተሰምዐት ሃይማኖትክሙ ውስተ ኵሉ ዓለም ወእግዚአብሔር ሰማዕትየ ዘኪያሁ አመልክ በኵል መንፈስየ ወበትምህርተ ወልዱ ከመ እዜከረክሙ በጸሎትየ ዘልፈ ወእስእል ከመ ይሠርሐኒ እግዚአብሔር በፈቃዱ እምጻእ ኀቤክሙ እስመ እፍቅድ ወከመ ትርክቡ ጸጋሁ ለመንፈስ ቅዱስ በእንተዝ ከመ ይትፈሣሕ ልብክሙ እስመ ኀበርክሙ አሚነ ምስሌየ።

በእንተ ጻሕቀ መምህራን ወዕሴቶሙ ወእፈቅድ ባሕቱ ታእምሩ አኀዊነ ከመ ዘልፈ እፈቅድ እምጻእ ኀቤክሙ ወይሰአነኒ እስከ ይእዜ ወለእመኒ ቦ ከመ እርከብ ዕስትየ በላዕሌክሙ ከመ በላዕለ አሕዛብኒ ወበላዕለ አረሚኒ ወበላዕለ ሐቃልኒ ወለጠቢባንኒ ወለአብዳንኒ እስመ ለኵሉ ሰብእ እምሂር ወዓዲ ፈድፋደ እጽሕቅ ለከሙ ለእለ ብሔረ ሮሜ እምሀርከሙ እስመ ኢየኀፍር ምህሮ ወንጌሉ እስመ ኀይሉ ለእግዚአብሔር ውእቱ ለኵሎሙ እለ የአምኑ ቦቱ ለአይሁዳዊ ቀዳማዊ ወለአረማዊኒ ደኃራዊ ወቦቱ ያስተርኢ ጽድቀ እግዚአብሔር ወርትዑ እስመ ያጸድቆሙ ለእለ የአምኑ ቦቱ በአሚን እስመ ከማሁ ይብል መጽሐፍ በአሚን የሐዩ።

በእንተ መቅሠፍት ዘይመጽእ ላዕለ ኃጥኣን

ወይመጽእ መቅሠፍተ እግዚአብሔር እምሰማይ ላዕለ ኵሉ ሰብእ ኃጥእ ወዐማፂ እለ ለጽድቅ ወይመይጥዋ በዓመፃሆሙ እስመ አእምሮ እግዚአብሔር ከሠት በኀቤሆሙ ወአርአየ እግዚአብሔር ላዕሌሆሙ ወዘኢያስተርኢ እግዚአብሔር እምፍጥረተ ዓለም ይትዐወቅ በፍጥረቱ በኀልዮ ወበአእምሮ ወከመዝ ይትአመር ኀይሉ ወመለኮቱ ዘለዓለም ከመ ኢይርከቡ እስመ እንዘ የአምርዎ ለእግዚአብሔር አኮ ከመ እግዚአብሔር ዘአእኰትዎ ወሰብሕዎ ዘእንበለ ዘሐሰውዎ ወረኵሱ በኀሊናሆሙ ወተጸለለ ልቦሙ ወእንዘ ይፈቅዱ ይጥብቡ አብዱ ዕስመ ወለጡ ስብሐቲሁ ለእግዚአብሔር ዘኢይመውት ወአምሳለ ርእየ ሰብእ መዋቲ ረሰዩ ወከመ እንስሳ ወከመ አራዊት ወከመ አዕዋፍ ወበእንተዝ አግብኦሙ ወኀደጎሙ ከመ ያርኵሱ ርእሶሙ ለሊሆሙ ነፍስከሙ እስመ ሐሰተ ረሰይዎ ለጽድቀ እግዚአብሔር ወአምለኩ ወተፀአፀኡ ተግባሮ ወኀደግዎ ለፈጣሬ ኵሉ ዘውእቱ አምላክ ቡሩክ ለዓለመ ዓለም አሜን ወበእንትዝ ወሀቦሙ እግዚአብሔር መቅሠፍተ እኩየ ወአንስቲያሆሙኒ ኀደጋ ፍጥረቶን ወተመሰላ በዘኢኮነ ፍጥረቶን ወከማሁ ዕደዊሆሙኒ ኀደጉ አንስቲያሆሙ ወነዱ በፍትወቶሙ ወገብኡ በበይናቲሆሙ ብእሲ ላዕለ ብእስ ኀሣሮሙ ገብሩ ወባሕቱ ፍዳሆሙ ወይገብእ ጌጋዮሙ ዲበ

ርእሶሙ ወበከመ ኢኀለይዎ ለእግዚአብሔር በልቦሙ ከማሁ እግዚአብሔርኒ ወሀቦሙ ልበ እበድ ከመ ይግበሩ ዘንተ ዘኢይደሉ እዘ እሙንቱ ጽጉባነ ኵሉ ዓመፃ ወእከይ ወጹግ ወትዕግልት ወጽጉባነ ቅንአት ሐማምያን ቀታልያን ጉሕላውያን መስተመይናን ዝኑራን እኩያነ ልማድ ወግዕዝ ሐማይያን መስተሣልቃን መስተሐብባን ዕቡያን ዝሉፋን ዐላውያን ሐሳውያን ጸላእያነ እግዚእ ሥሑጻን አብዳን ወዝንጉዓን ወመስተኃሥሣን ለእከይ ወአልቦሙ ምሕረት እንዘ ለሊሆሙ የአምሩ ኵነኔሁ ለእግዚአብሔር ከመ ይደልዎ ሞት ለዘገብረ ዘንተ ከመዝ አኮ ባሕቲቶሙ ዘይገብርዎ ዓድ ይዌሕከዎ ያግብእዎ።

ኅበ ሰብአ ሮሜ

በእንተ ዘይግዕዝ ቢጾ ዕጓለ እመሕያው በምንትኑ ለመኰንነ ጽድቅ ሶበ አንተ ዝኰ ዘትግዕዝ በላዕለ ባዕድ ሶበ ለልከ ትገብር ዝኰ ዘትጸልእ በላዕለ ቢጽከ አኮኑ ርእሰከ ትግዕዝ እስመ ለሊከ ትግበር ከመ ጽድቅ ኵነኔሁ ለእግዚአብሔር ዘያመጽእ መቅሠፍተ ላዕለ እለ ይገብሩ ዘንተ ከመዝ ኀልዮ እስኩ ዕጓለ እመሕያው ለዝንቱ ለእመ ብከ ኀበ ታመሥጥ እምኵነኔሁ ለእግዚአብሔር ሶበ ለሊከ ትገብር ለዝኩ ዘትግዕዝ በላዕለ ባዕድ ወዘትጸልእ ትትሔዘብኑ ታስተዐብዶ ለእግዚአብሔር ምሕረቱ ወበትዕግሥቱ ወበአሆ ብሂሎቱ ላዕሌከ ከመ ምሕረቱ ለእግዚአብሔር ኪያከ ያገርር ኀበ ንስሓ ወባሕቱ በአምጣነ ታጸንዕ ልበከ ወኢትኔስሕ ትዘግብ ለከ መቅሠፍተ ለአመ ይበጽሐከ ኵነኔሁ ለእግዚአብሔር እስመ ውእቱ ይፈድዮ ለኵሉ በከመ ምግባሩ በኵነኔ ጽድቁ ለባዕድን።

በእንተ ተዐጋሥያን ወከሓድያን

ወለእለሰ ተዐገሡ በምግባር ሠናይ ወየኀሥሡ ክብረ ወስብሐተ ውእቱኒ ይሁቦሙ ሕይወተ ዘለዓለም ወለእለሰ ከሓድያን ወዓላውያነ ጽድቅ ወመፍቀርያነ ዐመፃ ፍዳሆሙ መቅሠፍት ወመንሱት ወቀኖጥዓ ወገዓር ወምንዳቤ ለነፍሰ ኵሉ ሰብእ ዘእኩይ ምግባሩ እመኒ አይሁዳዊ ወእመኒ አረማዊ ክብር ወስብሐት ወሰላም ለኵሉ ዘሠናይ ምግባሩ እመኒ አይሁዳዊ ወእመኒ አረማዊ እስመ ኢያደሉ እግዚአብሔር ለገጸ ሰብእ እስመ ኵሙ እለ ፈድፈደ እምሕግ አበሆሙ ፈድፈደ እምሕግ ኵነኔሆሙ ወኵሉሙ እለ ዘእንበለ ሕግ አበሳሆሙ ዘእንበለ ሕግ ኵነኔሆሙ።

በእንተ እለ ይፈቅዱ ይጽደቁ በአጽምዖ መጻሕፍት

ቦኑ በአጽምዖ መጻሕፍት ይጸድቁ በቅድመ እግዚአብሔር አኮኑ ዳእሙ ይጸድቁ አሕዛብኒ እለ አልቦሙ ሕግ ይሠርዑ ሎሙ ሕገ ሎሙ ለሊሆሙ ወይገብሩ ዘበሕጎሙ ገቢረ ሕግ እንዘ ጽሑፍ ውእቱ ውስተ ልቦሙ እምግባሮሙ ወያርሰሐስሐሙ ልቦሙ ወይቀልዮሙ ወየምሩ ከመ አልቦ ዘይብሉ ወዘይወቅሱ አመ እግዚአብኄር ለዕጓለ እመሕያው ዘየኀብኡ ወዘይከብቱ ውስተ ልቦሙ በከመ መሀርኩ በእንተ ኢየሱስ ክርስቶስ።

በእንተ ዘለባዕድ ይሜህር ወለሊሁ ኢይገብር

ወሶበ አንተ አይሁዳዊ ዘተዐርፍ በኦሪትከ ወትትሜካሕ በእግዚአብሔር ወተአምር ፈቃዶ ወትፈልጥ ዘይኄይስ ወምሁር አንተ ኦሪተ ወትትአመን ከመ አንተ መርሖሙ ለዕዉራን ወብርሃኖሙ ለእለ ውስተ ጽልመት ወመጥብቢሆሙ ለአብዳን ወመምህሮሙ ለሕፃናት ወትትሜሰል ጻድቀ ወተአምር ሕገ ኦሪት በዘትጸድቅ ወእፎኑ እንከ ዘኢትሜህር ርእሰከ ዘለባዕድ

ትሜህር ኢትስርቁ ትብልዋ ወለሊከ ትሰርቅ ኢትዘምዉ ትብል ወለሊከ ትዜሙ ወተሐውር ኀበ ብእሲተ ብእሲ ጣዖተ ወለሊከ ትሰርቅ ቤተ መቅደስ ወትትሜካሕ በኦሪት ወለሊከ ዐላዊሃ ለኦሪት ለእግዚአብሔር።

በእንተ እለ ይፀርፉ ላዕለ ስመ እግዚአብሔር

ወናሁ በእንቲኣከሙ ይፀርፉ አሕዛብ ላዕለ ስመ እግዚአብሔር በከመ ጽሑፍ ግዝረትሰ ትበቍዐከ ለእመ ገበርከ ኦሪት ወእመሰ ኢገበርከ ኦሪት ግዝረትከ ቍልፈተ ትከውነከ ወለእመሰ እንዘ ቈላፍ አንተ ወዐቀብከ ኦሪት ቍልፈትከ ግዝረተ ትከውነከ።

Appendix D: Sample Suggestion

invalid word flagged by GLSC	Total No of suggestion by GLSC	Only correct Suggestion
በንተ	4	3
ለእግዚኢነ	1	1
እየሱስ	3	1
ዘተጸውኡ	1	1
ወተፈልጠ	2	2
በሊሰን	2	2
ለትምተ	5	4
መላኪተ	3	3
ነጉረ	4	4
በአፈነቢያቲሁ	2	1
ወሊዱ	2	2
ሰብ	1	1
ወበመ ንፈሱ	1	1
አትሙኒ	1	1
አም	5	5
ሃይማኖሙ	1	1
እሰማይ	2	2
እለለ	5	5
ኩልከሙ	2	2
በኵል	2	1
መፈስየ	2	2
ዘልፈፈ	2	2
ትርቡ	2	2
በንተዝ	1	1
እስከይእዜ	3	3
ዕስትየ	1	1
አሕዛቡኒ	3	3
እምሂር	1	1
ፈደፋደ	2	2
ወንደሱ	2	2
ይትወወቅ	1	0
ወመለከቱ	2	2
ዕስመ	3	3
ስብሀቲሁ	1	1
ለግዚአብሔር	2	2
ንስሳ	1	1
አግብሙ	1	1

ወጎደጎአሙ፡	2	2
ወሀሙ፡	1	1
ጎደጋኣ	2	2
በበእይናቲሆሙ፡	2	2
ብእስ	2	2
ርይእሶሙ፡	1	1
ከሚሁ፡	4	4
እዘ	4	4
ወጽጉባኣነ	2	2
በጾ	4	2
ስጽድቅ	3	3
ዞበ	2	2
ለልከ	5	5
ላለ	3	1
ትብር	5	5
ትትኤሐዘብኑ	1	1
ምህሕረቱ	1	1
ወበወሆ	2	2
ብይሎቱ	3	2
ትእዘግብ	2	2
ለባድን	2	2
ከ ብረ	5	5
አበሆሙ፡	2	2
ዘንበለ	3	3
እግዚአብኄር	2	2
ርህሰከ	1	1
ጣወተ	2	2
ወትትሜካ	2	2
ለኣወሪት	3	3
ሕገኦሪት	3	2
መጻሐፍ	1	1
በሐበ	3	3
ቦእለ	5	5
ነቢከ	3	3
ወሰገእደ	2	2
ወጎበር	5	5
ወአወሰበ	1	1
ከመጥንተ	3	3
ባህቲ	2	2
ለበል	5	5
አበሁ፡	4	3