2020

# DEEP LEARNING BASED AMHARIC GRAMMAR ERROR DETECTION

DESSALEGNE, YARED


**BAHIR DAR UNIVERSITY**

**BAHIR DAR INSTITUTE OF TECHNOLOGY**

**SCHOOL OF RESEARCH AND POSTGRADUATE STUDIES**

**FACULTY OF COMPUTING**

# DEEP LEARNING BASED AMHARIC GRAMMAR ERROR DETECTION

**BY**

**YARED DESSALEGNE ALEMU**

**BAHIR DAR, ETHIOPIA**
**August 29, 2020**

# DEEP LEARNING BASED AMHARIC GRAMMAR ERROR DETECTION

**BY**

**YARED DESSALEGNE**

**A thesis submitted to the school of Research and Graduate Studies of Bahir Dar Institute of Technology, BDU in partial fulfilment of the requirements for the degree of Masters in Computer Science in the Faculty of Computing**

**Program: MSc. Computer science**

**Advisor: Yaregal Assabie (PhD)**

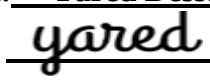
**Bahir Dar, Ethiopia**

**August 29, 2020**

## DECLARATION

**I, the undersigned, declare that the thesis comprises my own work. In compliance with internationally accepted practices, I have acknowledged and refereed all materials used in this work. I understand that non-adherence to the principles of academic honesty and integrity, misrepresentation/ fabrication of any idea/data/fact/source have constituted sufficient ground for disciplinary action by the University and can evoke penal action from the sources, which have not properly cited or acknowledged.**

**Name of the student:** Yared Dessalegne Alemu

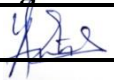
**Signature:** 

**Date of submission:** 8/6/2020

**Place:** Bahir Dar

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

**Advisor Name:** Yaregal Assabie (PhD)

**Advisor's Signature:** 

**BAHIR DAR UNIVERSITY**
**BAHIR DAR INSTITUTE OF TECHNOLOGY**
**SCHOOL OF RESEARCH AND GRADUATE STUDIES**

**FACULTY OF COMPUTING**

**Approval of thesis for defense result**

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student Yared Dessalegne Alemu Signature yared Date 08/08/2020

As members of the board of examiners, we examined this thesis entitled "Deep learning based Amharic grammar error detection "by *Yared Dessalegne*. We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of science in "Computer Science".

**Board of Examiners**

| Name of Advisor | Signature | Date |
|---|---|---|
| Yaregal Assabie(PhD) | | 08/08/2020 |
| Name of External examiner | Signature | Date |
| Million Meshesha (PhD) | million | 26/08/2020 |
| Name of Internal Examiner | Signature | Date |
| Tesfa Tegegne(PhD) | | 28/08/2020 |
| Name of Chairperson | Signature | Date |
| Abinew Ali | | 28/08/2020 |
| Name of Chair Holder | Signature | Date |
| Haileyesus Amsaya | | 28/08/2020 |
| Name of Faculty Dean | Signature | Date |
| Belete Biazen | | 28/08/2020 |

Faculty Stamp

## ACKNOWLEDGEMENTS

**First of all, I would like to thank Almighty God and Saint Mary for helping me in my life. Next I would like to express my deep gratitude to Dr.Yaregal Assabie, for giving me his endless support, motivation and valuable comments.**

## ABSTRACT

Nowadays, natural language processing became a hot research area, that is mainly focused on maximizing the capability of the computer to understand and communicate with human language or natural language. Therefore, to communicate with natural language or human language, grammatical correctness of the spoken language is important. So, developing a natural language application is important to identify the grammatical error that may occur in natural language texts. To say a sentence is grammatically correct, the arrangement of the words inside the sentence should agree in number, person, gender, tense, and other agreement rules. If the input sentence is incorrect, it may have agreement problems, such as subject-verb, object-verb, adjective-noun, incorrect word order or it may be adverb-verb agreement problems. In order to check the grammatical correctness of a sentence, several researches have been conducted for different languages with different grammar checking approaches, like rule-based, statistical-based and hybrid-based. Nowadays, deep learning becomes the most promising approach for natural language processing.

The objective of this proposed work is to develop deep learning based Amharic grammar error detection. To this end, we propose a deep learning grammar checker approach. We apply two deep learning approaches such as long short-term memory recurrent neural network and bidirectional long short-term memory recurrent neural network. We have used python 3.7, Keras TensorFlow as a backend, Pyqt5 to design the interface, and HornMorpho to analyze the feature of Amharic words. The evaluation is done for two test cases. The first one is for long short-term memory and the second one is for bidirectional long short-term memory recurrent neural network. Finally, the experimental result shows that, long short-term memory performs accuracy of 88.27%, recall 88.27%, precision of 88.33%, and f1 measure of 88.5%. The bidirectional long short-term memory performs 88.89% accuracy, 88.89% of recall, 89 % precision and 89% of f1 measure. The challenge of this research work was the quality of morphologically annotated Amharic sentence especially words having more than two meanings and words that tell respect. The grammar error detector can be more effective when we have a larger morphologically annotated sentence and hence further research needs to be done to enhance the result of this study.

**Keywords**: NLP, Deep Learning, LSTM, BILSTM, Amharic Morphological analysis, Amharic grammar, Amharic sentence disagreement, Amharic grammar error detection.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ALGORITHMS

## ACRONYMS/ABBREVIATIONS

| | |
|---|---|
| AND | Adjective Noun Disagreement |
| AVD | Adverb Verb Disagreement |
| Bi-LSTM | Bidirectional Long Short-Term Memory |
| CONLL | Computational Natural Language Learning |
| CNN | Convolutional Recurrent Neural Network |
| LSTM | Long Short-Term Memory |
| NLP | Natural Language Processing |
| OBJ | Object of a sentence |
| OVD | Object Verb Disagreement |
| POS | Part of Speech |
| RNN | Recurrent Neural Network |
| SBJ | Subject of the sentence |
| SVD | Subject-Verb Disagreement |
| SVO | Subject Verb Object |

# CHAPTER ONE: INTRODUCTION

## 1.1 Background

Nowadays, technology advancement is increasing from time to time, for instance, electronic documents are drastically increasing, such as word processing, Emails, webpages, and others. However, the majority of the data is appearing in textual format which highly unstructured (Gharehchopogh & A.Khalifelu, 2011), therefore to produce significant and actionable insides from this data, it is important to get appointed with the techniques of Natural Language Processing (NLP).

Natural language processing (NLP) is a part of computer science and artificial intelligence that deals with human language (e.g. English, Arabic, Chinese, Amharic, etc.). It represents texts which are occurred naturally into different levels of linguistic analysis. These include, morphological analysis, syntax analysis, semantic analysis, discourse, and pragmatic analysis (Khurana, Koli, Khatter, & Singh, 2017). Syntactic Analysis is one level of linguistics that analyzes how words combine to produce phrases and it determines how the input text is structured. Syntactic Analysis is important for many NLP application areas. These include Grammar Checking, Machine Translation, Information Extraction and Question Answering (Bhirud, Bhavsa, & Pawar, August 2017). Grammar checking is one application area of syntactic analysis that deals with whether the written text is grammatically correct or not, also it tells what the correct sentence should seem like (Bhirud, Bhavsa, & Pawar, August 2017). Grammar checker can be categorized into two branches such as grammar error detection and grammar error correction.in this research work we are considering grammatical error detection. Grammar error detection is defined as indenting the error in the given text.

The most common and widely used grammar error detection approaches are rule-based, statistical-based and hybrid-based (Aynadis & Yaregal, 2013). In a rule-based grammar checker approach, the input text is checked by manually generated rules. This approach requires linguistic experts to construct rules. The advantage of a rule-based grammar checker approach is simple to add, edit or remove a rule, it provides a detailed error message, and no need for training data is required for grammar error detection. A statistical-based grammar error detection approach requires a training corpus to learn what is 'correct' instead of using a manually designed rule. These approach uses N-

gram models to check the grammatical structure of the sequence of words. In this approach, it is difficult to provide detailed errors resulted from these systems. A hybrid grammar error detection approach is the combination of a rule-based and statistical approach. In this approach, some errors are solved by hand-crafted rules and some others are solved by N-gram models. Recently deep learning approach is coming up and it plays a vital role on NLP applications such as for machine translation, Question answering, information extraction, Summarization and other application areas. In contrast to machine learning approaches deep learning approaches didn't require future engineering manually rather it can learn automatically. Therefore, in order to check grammatical correctness of Amharic text we propose one of a deep learning approach which is Known as Bidirectional Long Short-Term Memory (BiLSTM).

## 1.2 Motivation

Amharic is the most spoken language in Ethiopia. It is the working language of the Federal Democratic Republic of Ethiopia. In addition to that, it is the working language of regional states and city administration such as Amhara regional state, Benishangul-Gumuz regional state, Southern Nations Nationalities and People Region (SNNPR) regional state, Gambella regional state, Addis Ababa city administration, and Dire-Dawa city administration (Gobena, 2011).

As a result, an enormous number of Amharic documents are produced and stored. To prepare and access such documents, it is important to have an Amharic grammar error detection. Amharic is the second language for many peoples. Due to that, they are prone to grammar errors. In addition to this Amharic grammar error detection is an important component to many NLP applications such as for Machine-Translation, Question-Answering, Information-Extraction, and others. This motivates me to develop an Amharic grammar error detection for Amharic text.

However, due to the absence of Amharic grammar error detection tool and Amharic is the second language for many peoples, they may make mistakes or miss grammatical structure of the sentence while writing texts. For example, አስቴር ልጁን ሰደበው። (aster insulted the boy). The problem with this sentence is the subject of the sentence is not agreed with the verb which is in gender information mismatch. The above sentence is corrected as አስቴር ልጁን ሰደበችው። so, in such like situations the user of Amharic language needs automated grammar error detection tool in order to identify grammatically incorrect sentences.

There are two research works done for Amharic grammar error detection by Aynadis and Yaregal (2013) and Abraham Gebreamlak (2019). both this works have been done using rule-based grammar error detection approach. However, in rule-based approach, it is difficult to detect errors specially for compound, complex and compound complex sentences. Since Amharic is morphologically rich language, it is difficult to extract each feature manually. Therefore, it needs a mechanism that learns features automatically using a deep learning approach. Therefore, this motivates me to develop Amharic grammar checker using bidirectional Long Short-Term Memory (BiLSTM).

### 1.3 Statement of the problem

A grammar error detection has been developed for various languages such as English (Naber, 2003), Arabic (Madia & Al-Khalifaa, 2018), Chinese (Yang, et al., 2017), Afaan-Oromo (Tesfaye, 2011), Amharic (Aynadis & Yaregal, 2013), among others. In addition to this, it is a stepping stone for many NLP applications, such as machine translation, question answering, dialogue system, and others.

Amharic language is morphologically rich and has unique features as compared to other languages. The order of the word in the Amharic language follows the subject-object-verb (SOV) structure (Kassa, 2010). In addition to this, the correct Amharic sentence should obey all agreement rules (Kassa, 2010). However, there are many agreement and word order problems that are faced by Amharic sentence. The common grammar errors are subject-verb disagreement, adjective-verb disagreement, adverb-verb disagreement, incorrect word order (Tensou & Assabie, 2014), ( Baye Yimam, 1995).

For example: look at the following Amharic sentences having agreement and word sequences problems.

1. **ተማሪው** ኢትዮጵያዊ **ናት**። (subject-verb disagreement).
2. አበበ **ቤቱን ሸጠችው** (object-verb disagreement).
3. ዩኒቨርሲቲው በተመደበለት በጀት ከአገር ውስጥና ከውጭ አገር መጻህፍት ፣ የምርምርና የትምህርት መርጃ መሳሪያዎችን ጨምሮ ለጥቁር አንበሳ ሆስፒታል **አዳዲስ** የህክምና መገልገያ **መሳሪያ** እንደሚገዛ ተጠቁሟል ። (adjective-noun disagreement).

4. የተወዳዳሪዎች ምዝገባ **በሚቀጥለው** ሳምንት ጀምሮ በወረዳና ቀበሌ ቢሮዎች ፣ ኮሌጆች ፣ ስፖርት ኮሚሽንና አትሌቲክስ ፌዴሬሽን **እንደተካሄደ** መገለጹን ዋልታ እንፎርሜሽን ማእከል ዘግቧል ። (adverb-verb disagreement).

5. ኢትዮጵያዊው ታደሰ ማርያምና ወርቅነሽ ቶላ በማራቶን **አሸነፉ ውድድር** ። (incorrect word order).

According to Amharic grammar rule, the subject and verb, object and verb, adverb and verb, adjective and noun should be agreed in order to say Amharic sentence is correct (Tensou & Assabie, 2014) ( Baye Yimam, 1995).

Accordingly, we need an Amharic grammar error detection to detect errors in Amharic texts for correction. Even if Amharic is a less-resourced language, there is a work on Amharic grammar error detection by Aynadis and Yaregal (2013) and Abraham (2019). However, due to the incompleteness of rules and quality of the statistical data, it displays false alarm. In addition to this, to get better performance, it depends on the knowledge of the researcher. Amharic is a morphologically rich and complex language. As a result, selecting features manually requires linguistic experts and enough time. in traditional grammar error detection approach, it is difficult to detect errors, especially when the sentence is compound, compound complex and complex sentences.

Therefore**,** it needs a mechanism that learns features automatically using a deep learning approach.

From deep learning grammar checker approaches, Long short term memory (LSTM) recurrent neural network and bidirectional long short term memory (BiLSTM) recurrent neural network performs better results than the existing one (e.g. for Chinese grammatical error detection (LEEa, LINb, YUb, & TSENGa, 2017) (Yang, et al., 2017)). As far as we know there is no research work developed for Amharic grammar error detections using deep learning approaches such as BILSTM. So, in order to solve the above problems and improve the performance of the existing work we have investigated the impact of long short-term memory recurrent neural networks and bi-directional long short-term memory recurrent neural networks on Amharic grammar error detection.

Generally, we have answered the following research question.

- What features are best describe most common errors in Amharic grammar?
- Which deep learning algorithms are better fit for Amharic grammar error detection?

## 1.4 Objective of the study

### 1.4.1 General Objective

The main objective of this research work is to develop a grammar error detection for Amharic language using deep learning approach.

### 1.4.2 Specific Objective

To meet the general objective of this research work, the following specific objectives have addressed.

- To collect and prepare suitable dataset (corpora) from different sources for training and testing purposes.
- To identify the grammatical structure, and morphology of the Amharic language.
- To design a model for automatic Amharic text grammar error detector.
- To develop a prototype for Amharic grammar error detection.
- To evaluate the performance of the proposed model.

## 1.5 Methodology of the study

This study followed design science research methodology. In design research methodology a problem is assessed based on that artifact is proposed and the evaluation of the artifact is a key contribution (Peffers, Tuunanen, A., Rothenberger, & Chatterjee, 2007). Design science research methodology has a certain phase such as problem identification and motivation, objectives for a solution, design and development, demonstration, evaluation, and communication.

### 1.5.1 Problem identification and motivation

In this design since methodology phase, the research problem and the value of the solution is defined and justified. Justifying the value of the solution used to motivate the researcher and the audience.in this study, we have identified the problem from literature review. We have reviewed related works which are done on grammar error detection for different languages, to better understand the problem and books or other resources to understand the grammatical structure of Amharic language.

### 1.5.1 Objective of solution

In this phase, the objective of the study followed from the first step in order to get a solution. generally, it inferred from a problem definition. Different researches are reviewed in order to know the statement of the problem.

### 1.5.2 Design and Development

In this phase, the artifactual solution Is designed.so, we have designed deep learning based Amharic grammar error detection model using the concept and procedures from step one and step two.in order to design this model and to develop a prototype of this model different supporting tools are required. These include python programming language, Keras, TensorFlow, Pyqt5, morphological analyzer. Python programing language is used to implement the Amharic grammar error detection. HornMorpho is used to morphologically analyze collected training corpus and input text. Pyqt5 is used to design the user interface of the propose system. We are used Notepad++ for dataset preparation. Our proposed system consists of preprocessing module such as tokenization, tag sequences splitting and sequence padding and learning phase. Under learning phase, the dataset is split into Training and testing phase. Furthermore, the Training phase is split to Training and validation. The validation and Training phase is input to word embedding, BiLSTM, Dense, SoftMax finally the model is trained. The testing phase is given to trained model in order to predict and evaluate the performance of the proposed model.

### 1.5.3 Demonstration

The developed model is demonstrated by simulating how the developed deep learning based Amharic grammar error detection system detect error and classify the type of errors. The system accepts Amharic texts and out puts the type agreement errors in the input sentence.in order to run the source code we have used Spider python IDE to analyze words and Jupiter notebook for running the source code.

### 1.5.4 Evaluation

In order to measure how the developed Amharic grammar error detection system well suited the model is evaluated. To evaluate the proposed system testing the learned model using morphologically annotated Amharic sentence, the sentence contains simple, compound and compound complex sentences. The model was evaluated using evaluation metric such as precession, recall, f1 measure and confusion metric.in addition we have compared the performance of two recurrent neural network algorithms such as long short-term memory and bidirectional long short-term memory.

### 1.5.5 Communication

The last step of design science research methodology is communication of the problem, the artifacts, and the effectiveness and other related information to relevant audiences when it is

needed. So, we will present our study to other researchers and experts. so that we can get feedback on the study. It is communicated through department of computer science in Bahir Dar University. We also will present it in different NLP conferences and try to publish an article on it.

## 1.6 Scope and Limitations of the study

The scope of this study is concerned with to the development of an Amharic grammar error detection for Amharic language which detects grammatical errors in Amharic sentence, such as for simple, compound, compound complex sentences. This study mainly concerned with grammar error detection especially agreement errors such as adjective-noun disagreement, adverb-verb disagreement, subject-verb disagreement, object-verb disagreement and incorrect word sequence errors; However, it does not consider with grammar error correction mechanism, and also, we didn't consider words that tell respect, and words having more than two different meanings.

## 1.7 Significance of the study

The final result of this research work is contributed to the development of many NLP applications. These include Amharic grammar error correction, machine translation, Question Answering, word prediction, information retrieval, text summarization, and others. In addition to this, the system has improved errors when users typing in word processor programs and social media. The proposed system is also enabling Amharic users especially the non-native ones, to prepare official documents, emails, letters and some other tasks.

## 1.8 Organization of the Thesis

The organization of this thesis comprises of five chapters including chapter one. Chapter two provides a detail explanation and discussion of literature review and related work. Chapter three presents the design of deep learning-based Amharic grammar error detection.in Chapter Four, presents the result and discussion. Result of experiment and test cases are discussed. Finally, the last chapter describes a conclusion and recommendation including future works.

# CHAPTER TWO: LITRATURE REVIEW

## 2.1 Overview

This chapter presents, the concept of grammar checker and theoretical concept or ideas related to Amharic language and grammar checker. The content of the chapter includes a brief introduction to Amharic Language, under Amharic Language section, Amharic Part of speech, Amharic Morphological characteristics, grammatical structure, and its agreement errors are presented. Furthermore, this chapter discusses, the different approaches used in grammar checker starting from rule-based up to deep learning. Finally, some related works on Amharic grammar error detection, and other local and foreign languages are discussed in detail.

## 2.2 Amharic Language

Amharic belongs to under a family of Semitic language, especially spoken in north and south part of Ethiopian. The Amharic language also belongs to under Afro-Asiatic language family. this family includes Amharic, Hebrew, Arabic, and Assyrian. It is also second most spoken language next to the Arabic language. Amharic language is mostly spoken in Ethiopia (ኢትዮጵያ), however, there are also other speakers throughout the world in other country's especially, in Eritrea, USA, Canada, and Sweden. Amharic is the working language of the Federal Democratic Republic of Ethiopia (Yifru & Menzel, 2009) and other regional states, and city administrations of Ethiopia such as. Amhara regional state, Benishangul-Gumuz regional state, Southern Nations Nationalities and People Region (SNNPR) regional state, Gambella regional state, Addis Ababa city administration, and Dire-Dawa city administration. (GOBENA, November 2010 ).

The Amharic language has its own writing system, which is known as Fidel (ፊደል). Fidel is simply it is alphabet or letters or it is also known as "letter", or "character" or *abugida* (አቡጊዳ). Fidel is a writing system of Amharic which contains consonants and vowels. The Ethiopian alphabet contains seven (7) vowels and thirty-three (33) basic shapes mostly represents by consonants followed by vowels. Amharic alphabet also is known as Ge'ez script or Ethiopic script which is one of the oldest alphabetic notations in the world. The alphabet of Amharic or Amharic letters is mostly appearing in a grid format which is the consonants are seen vertically and the vowels are lined up horizontally. The written system of Amharic is from left to right. The Amharic script is not exactly speaking it is an alphabet, but we can say it is a syllabary, which means each letter

mostly represents the whole syllable. So, using these systems anyone can easily learn or understand the Ethiopian alphabets. (Tensou & Assabie, 2014).

Figure 2.1 shows the list of Ethiopic or Amharic alphabets (Fidels) which consists of consonants and vowels. As shown in the figure Amharic alphabet has seven row-wise orders and the column shows a list of basic alphabets or characters and the rest one is Vowels which is derived from the basic symbols. (Meshesha & Jawahar, 2007)

| | *Ge'ez* ä | *Ka'eb* u | *Salis* ï | *Rab'e* a | *Hamis* é | *Sadis* i | *Sab'e* o |
|---|---|---|---|---|---|---|---|
| h | ሀ | ሁ | ሂ | ሃ | ሄ | ህ | ሆ |
| l | ለ | ሉ | ሊ | ላ | ሌ | ል | ሎ |
| ḥ | ሐ | ሑ | ሒ | ሓ | ሔ | ሕ | ሖ |
| m | መ | ሙ | ሚ | ማ | ሜ | ም | ሞ |
| s | ሠ | ሡ | ሢ | ሣ | ሤ | ሥ | ሦ |
| r | ረ | ሩ | ሪ | ራ | ሬ | ር | ሮ |
| s | ሰ | ሱ | ሲ | ሳ | ሴ | ስ | ሶ |
| q | ቀ | ቁ | ቂ | ቃ | ቄ | ቅ | ቆ |
| b | በ | ቡ | ቢ | ባ | ቤ | ብ | ቦ |
| t | ተ | ቱ | ቲ | ታ | ቴ | ት | ቶ |
| h | ኀ | ኁ | ኂ | ኃ | ኄ | ኅ | ኆ |
| n | ነ | ኑ | ኒ | ና | ኔ | ን | ኖ |
| a | አ | ኡ | ኢ | ኣ | ኤ | እ | ኦ |
| k | ከ | ኩ | ኪ | ካ | ኬ | ክ | ኮ |
| w | ወ | ዉ | ዊ | ዋ | ዌ | ው | ዎ |
| ẹ | ዐ | ዑ | ዒ | ዓ | ዔ | ዕ | ዖ |
| z | ዘ | ዙ | ዚ | ዛ | ዜ | ዝ | ዞ |
| y | የ | ዩ | ዪ | ያ | ዬ | ይ | ዮ |
| d | ደ | ዱ | ዲ | ዳ | ዴ | ድ | ዶ |
| g | ገ | ጉ | ጊ | ጋ | ጌ | ግ | ጎ |
| ṭ | ጠ | ጡ | ጢ | ጣ | ጤ | ጥ | ጦ |
| p | ጰ | ጱ | ጲ | ጳ | ጴ | ጵ | ጶ |
| ts | ጸ | ጹ | ጺ | ጻ | ጼ | ጽ | ጾ |
| ts | ፀ | ፁ | ፂ | ፃ | ፄ | ፅ | ፆ |
| f | ፈ | ፉ | ፊ | ፋ | ፌ | ፍ | ፎ |
| p | ፐ | ፑ | ፒ | ፓ | ፔ | ፕ | ፖ |

Figure 2. 1 The seven order of consonants and vowels of Ethiopic alphabets Adopted from (Meshesha & Jawahar, 2007)

### 2.2.1 Amharic Sentence

A sentence is a collection of phrases or words that express complete information or that tells a complete idea or thought. A sentence should have a subject and a verb or predicate. Amharic sentence may be a statement, exclamation or question or command. The subject of a sentence that may be noun phrase that is person or thing and the predicate tells what the subject does or it contains the object and the verb of a sentence. The sequence of words or the order of Amharic sentence is different from other languages. The word order in Amharic sentence is subject-object-verb structure, which means the verb goes to the end of a sentence. (BACH, 1970)

For example; አለሙ አበበን መታው።/Alemu beats Abebe.

እሱ ተማሪ ነው ።/he is a student.

If in the given sentence adjective appear, it has put before the object of the sentence.

For example: -

እሱ ጥሩ ተማሪ ነው ።/he is a good student.

In question also follows the same word order, that SOV.

For example: -

እሱ ተማሪ ነው? / is he a student?

ያች ተማሪ ማን ናት? / Who is that student?

Amharic sentence can be classified into simple, and complex sentences.

**Amharic simple sentence**

A sentence having only one subject and verb phrase that transfers a complete idea is called a simple sentence. A simple sentence can be categorized into four types such as declarative, negative sentence, interrogative, and imperative sentence.

Forexample:

አስቴር መምህር ሆነች ።/declarative sentence.

አለሙ ምሳ አልበላም።/negative sentence.

ካሳ ምን ሆነ? /interrogative sentence.

በሩን ዝጋው! /imperative sentence.

**Amharic complex sentence**

A complex sentence consists of one or more verb phrases combining with one or more noun phrase or adjective phrases. Amharic complex sentence can be simple complex, complex-complex, and compound-complex. A complex sentence contains a complex verb phrase and a noun phrase.

For example: -

አበበ ምሳ ሊበላ ሲል ስልክ ተደውሎለት ሳይበላ ሄደ ።

### 2.2.2 Amharic POS

Part of speech (POS) is a word-class for a given word. The word class of Amharic language can be classified as an adjective, adverb, conjunction, interjection, preposition, pronoun, noun, and verb. ( Baye Yimam, 1995)

**Amharic Noun**

Amharic Noun is one of the word class of Amharic language.it is mostly used to identify or to label person, thing, place, and others. For example, for a person (አበበ፣ከበደ፣አለሙ), for things (ወንበር፣ቤት), for a place (ባህር ዳር፣ደብረ ዘይት) and others ( Baye Yimam, 1995).

Table 2. 1 Amharic Nouns (Adopted from Baye Yimam,1995)

| | Number | | Gender | |
|---|---|---|---|---|
| Word | Singular | Plural | Feminine | Masculine |
| በግ | በግ | በጎቹ | በጊቱ | በጉ |
| ተማሪ | ተማሪ | ተማሪዎች | ተማሪቱ | ተማሪው |
| መምህር | መምህር | መምህራን | መምህሪቱ | መምህሩ |
| ልጅ | ልጅ | ልጆች | ልጅኢት | ልጁ |
| አህያ | አህያዋ | አህያዎቹ | አህያዎ | አህያው |

**Amharic Pronouns**

Amharic Pronoun is one part of word-class. is it also a subclass of a noun that means we can use a pronoun instead of a noun? A pronoun can be classified into personal, demonstrative, and interrogative pronouns. For personal pronouns 1$^{st}$person (እኔ፣እኛ), 2$^{nd}$ (አንተ፣አንቺ፣እናንተ), person 3$^{rd}$person (እሱ።እነሱ፣አንቱ). (Tensou & Yaregal, 2014)

For demonstrative pronouns: - ይህ፣ይቺ፣ያቺ፣ያ፣እነዚያ፣እነዚህ.

For interrogative pronouns: -ማን፣እነ ማን፣ምን፣የት፣መቼ.

For possessive pronoun: -የእኔ፣የእሱ፣ የእሷ፣ የእንቺ፣ ያንተ፣የእነሱ፣ የእናንተ.

Table 2. 2 Amharic personal pronouns (Adopted from Tensou & Yaregal,2014)

| Person | Number | | Gender | |
|---|---|---|---|---|
| | Singular | Plural | Masculine | Feminine |
| 1st Person | እኔ | እኛ | | |
| 2nd Person | አንተ | | M | |
| | አንቺ | | | F |
| | | እናንተ | | |
| 3rd Person | እሱ | | M | |
| | እሷ | | | F |
| | | እነሱ | | |

| Gender | Number | Far | Near |
|---|---|---|---|
| Masculine | Singular | ያ | ይህ |
| Feminine | | ያቺ | ይቺ |
| All | Plural | እነዚያ | እንዚህ |

Table 2. 3 Amharic demonstrative pronouns(Adopted from Tensou & Yaregal,2014)

**Amharic Verb**

A verb is another part of a class of words that tells about an action or condition. Amharic verbs can be perfective (e.g. ሮጠ፣ወሰደ), imperative(ይሩጥ፣ይውሰድ), imperfective, gerundive verbs(ሩጦ፣ ወስዶ) and other. (Yimam, 2009)

Table 2. 4 Amharic Verbs (Adopted from Michael Gasser,2011)

| Word | Number | Gender | Person | Tense |
|---|---|---|---|---|
| ሰከረ | Singular | Masculine | 3rd | Perfective |
| ዘመሩ | Plural | | 3rd | Perfective |
| ትረዝሚያለሽ | Singular | Feminine | 2nd | Imperfective |
| እንሰብራለን | Plural | | 1st | Imperfective |
| ትውረድ | Singular | Feminine | 3rd | Imperative |
| ልውረስ | Singular | Masculine | 2nd | Imperative |
| ቀድሰው | Plural | | 3rd | Gerundive |

**Amharic Adjective**

An adjective is a word that tells a piece of additional information to a noun. Amharic adjectives appear before a noun. For example (በጣም ቆንጆ፣ጥቁር በግ). Amharic adjectives are inflected for number and gender. Some sample Amharic adjectives are listed in the table. (Tensou & Yaregal, 2014)

Table 2. 5 Amharic Adjectives(Adopted from Tensou & Yaregal,2014)

| Adjective | Number | Gender |
|---|---|---|
| ቆንጂት | Singular | Feminine |
| ጥቋቁር | Plural | All |
| ረዣዥም | Plural | ALL |
| አጭሩ | Singular | Masculine |
| ነጫጭ | Plural | Plural |

**Amharic Adverb**

An adverb is a word that modifies verb as in terms of Place (ውጭ፣ ወዲህ፣ እዚህ), Time (ሞቼ፣ ዘወትር፣ ዛሬ፣ ቅድም፣ ቀትር), etc. (Yimam, 2009)

Table 2. 6 Amharic Adverb(Adopted from Baye Yimam,1995)

| Adverb | Tense marker |
|---|---|
| ነገ | Future |
| አሁን | Present |
| ቅድም | Past |
| ኋላ | Future |
| ትናንት | Past |

**Amharic Conjunctions**

Conjunctions are words that link or connect another word, phrases, or sentences. Example of Amharic adjectives are ስለ፣ ግን፣ እንደ፣ ወይም፣ ስለዚህ. For example, "የራስ ጸጉራችሁ እንኳ አንድ ሳይቀር የተቆጠረ ነው ስለዚህ አትፍሩ።" (Tensou & Yaregal, 2014)

**Amharic Preposition**

P**repositions are** words that are used to link pronouns, nouns, or phrases to other words. Most of the time prepositions are short in length and they are placed directly in front of nouns. For example (ከ፣ለ፣ወደ), etc. for example **ወደ** እግዚአብሔር ቤት እንሂድ ባሉኝ ጊዜ ደስ አለኝ።”. (Tensou & Yaregal, 2014)

### 2.2.3 Amharic Morphology

Morphological analysis is the process of finding the smallest unit of words like root, stem, and other (Aynadis & Yaregal, 2013). It is an essential component for grammar checking and, for other NLP applications. Morphemes can be free or bound morpheme. For example, from the word “ልጅነት” we can generate” ልጅ” is free morpheme and “ነት” is bounded morpheme. Amharic morphology can be derivational or inflectional morphology

**Amharic Nouns** are derived from adjectives and other nouns. Amharic nouns are inflected or marked for Person (1st person, 2nd person, 3rd person), number (singular, plural), gender (masculine and feminine), and other features. For example, for the number indicator, we can take the following examples በግ in the singular can be በጎች in plural. ያሬድ in the singular can be እነያሬድ in the plural, እሱ in the singular can be እነሱ in the plural. For gender example: በግ can be በጊት.

**Amharic verbs** are inflected for Person (1st person, 2nd person, 3rd person), number (singular, plural), gender (male and female), tense, and others. Adjectives are inflected for gender, number, definiteness and others, for example ጠባብ is singular, ጠባቦች is plural, ረኸም is singular, ረዣኸም is plural.

### 2.2.4 Amharic Grammar Errors

**Subject and Verb Agreement**

In the following sentence,” አስቴር ወደ ገባያ ሄደች ።”/’Aster wede gebaya hedech’/ the subject “አስቴር”/” Aster”/ is third-person singular feminine and the verb “ሄደች” is the third person singular feminine. The above sentence shows that the subject of a sentence and the verb agrees. Look this sentence “እኔ ተማሪ ነው ።” the subject “እኔ” is the first person singular and masculine in gender, and the verb “ነው” is the third person singular with masculine in gender. So, the subject of the sentence and the verb is not much so sentence has subject-verb disagreement error.

Another example “ተማሪው መጻፍ ገዛ ። ” this Amharic sentence is grammatically correct, that means the subject of the sentence is agree with the verb in number, gender, person.if we replace the subject of the sentence “ተማሪው” with the word “ተማሪይቱ” the sentence have grammatically

incorrect .The type problem is adjective-verb disagreement. Because the subject of the sentence the word "ተማሪይቱ" is feminine in gender however the verb of the sentence thew word "ገዛ" is masculine in gender. As a result, the subject of the sentence is disagreed with the verb in gender. Generally, if the sentence has such a problem the sentence is said to be adjective-verb disagreement.

**Adjective and Noun Agreement**

The Adjective and noun should agree be with a number and gender. Let us take an example "ነጫጭ ወፎች በረሩ" the adjective "ነጫጭ" is plural and the noun "ወፎች" is plural, so in this example, the adjective agrees with the noun. On the contrary look this example" ነጭ ወፎች" the adjective ነጭ is singular and the noun "ወፎች" is plural. Therefore, adjective-noun disagreement for the second example.

Another example: - "የደቡብ ዩኒቨርስቲ ስምንት አዲስ ፕሮግራሞችን ጀመረ ።" from this Amharic sentence we can understand that the sentence has grammatically incorrect, the problem is adjective-noun disagreement problem. Because the adjective "አዲስ" indicates singular however, the noun "ፕሮግራሞችን "indicates plural in number, So the adjective of the sentence doesn't agree with the noun. To make a sentence grammatically correct, replace the word "አዲስ" with "አዳዲስ" or the word "ፕሮግራሞችን" with "ፕሮግራም". If the sentence has such like a problem, we can say that the sentence has adjective-noun disagreement problem.

**Object and Verb Agreement**

The other rule of Amharic grammar error detection is the object of the sentence should be agreed with the verb in number (singular, plural), a person (1st, 2nd and 3rd), and gender (masculine and feminine). For example፣ "አበበ መኪናውን ገዛው ።" the object "መኪናውን" is 3rd person masculine and the verb "ገዛው" is 3rd person singular masculine, therefore the object of the sentence agrees with the verb. If we replace the verb "ገዛው" with "ገዛችው" the object has disagreed with the verb.

Another example:-"ያሬድ ልጁን መከረው" from this Amharic sentence the object of the sentence agrees with the verb of the sentence in number, gender and person, which means the object "ልጁን" agrees with the verb "መከረው" we replace the object of the sentence "ልጁን" replace with "ልጅቱን", the sentence has object verb disagreement problem. Because the word "ልጁን" is masculine indicates but the word "ልጅቱን" is a feminine indicator, so the object of the sentence is disagreed with the

verb in gender. Generally, the sentence having such like problem is knowns object-verb disagreement.

**Adverb and Verb Agreement**

The agreement error in Amharic sentence is adverb-verb disagreement or mismatch. For example: - "መምህሩ ትናንት ይመጣል ።" the adverb "ትናንት" and the verb "ይመጣል" disagree with each other because the adverb refers to past action and the verb refers to future action. If we replace the adverb "ትናንት" with "ነገ" the verb and the adverb is agreed with each other.

Another example: -

1.በከፋ ዞን በሚቀጥለው ሳምንት ምርጫ ተካሂዶል።

2.የቢዝነስ ማኔጅመንት ትምህርቱን ባለፈው አመት ይጀምራል።

From those two sentences, we can understand that the sentences are grammatically incorrect. When we see example one, the problem is the adverb in the sentence did not agree with the verb of the sentence. Because the adverb "በሚቀጥለው" indicates future action however the verb indicates past action. When we see example two the adverb of the sentence "ባለፈው" indicates past action however, the verb of the sentence shows future action. So, in both two-sentences. The adverb and the verb disagree with a tense. Generally, if the sentence has such like problem the problem is adverb-verb disagreement.

**Amharic Word sequence**

The structure of Amharic sentence follows subject object verb (SOV) sequence for example, in this sentence "ውሻው ልጁን ነከሰው ።" where "ውሻው" is a subject, "ልጁን" is an object, and ነከሰው is a verb. If the sentence is like this "ልጁን ውሻው ነከሰው" the sequence is OSV therefore it is word sequence error problem.

Most of the time incorrect word sequence problems may arise if the adjective-noun disorder, adverb-verb disorder, or subject and verb disorder, or object and verb disorder. For example: -

1. የኢትዮ - ዳች" የወዳጅነት ተመሰረተ ማህበር።
2. የተከበሩ ግርማ አቶ አረፉ።
3. አመት በቀጣይ ብቻ ከ300 በላይ አደንዛዥ እጽ አዟሪዎች ተይዘዋል።
4. ዲያቆን 5ኛ አንቀፅ የሰው ነው ስም ።
5. አበበ መከረው ልጁን ።

The above five examples have grammatically incorrect because all sentence have incorrect word sequence. So, if the sentence has such like a problem, we can say that incorrect word sequence.

## 2.3 Approaches to Grammar checker

Grammar is a set of structural rules governing the composition of sentences, phrases, clauses, and words in a given natural language processing (Aynadis & Yaregal, 2013).In natural language processing grammar is defined as a collection of structural rules which govern the composition of clauses, phrases, and words. In order to communicate and share information using text or in other communication mechanisms the flow of information must be grammatically correct. So, it needs to automate natural language processing. NLP has various application areas; from that one application area of NLP is grammar checking. Grammar checking is the process of checking the validity of the text or checking whether the text is correct or incorrect. A correct sentence is one in which the corresponding words inside the sentence agree with number, person, gender, tense, and other agreement rules.

So, in order to check the grammatical correctness of a text, various researches are done for different languages with different approaches. The most common widely used grammar checker approaches are rule-based, statistical-based and hybrid-based (Aynadis & Yaregal, 2013).

### 2.3.1 Rule-Based Approach

Rule-based grammar checking is the classical approach that needs manually design handcrafted features (Naber, 2003). In a rule-based grammar checker approach, the input text is checked by manually generated rules. This approach requires linguistic experts to construct rules. The advantage of a rule-based grammar checker approach is simple to add, edit, or remove a rule, it provides a detailed error message, and also no need of training data required for grammar checkers. The drawback of a rule-based grammar checker approach is time-consuming to formulate rules to and also it needs linguistic experts for specific language to write every rule manually. Many researches are done using a rule-based grammar checking approaches for different languages such as for English (Naber, 2003), Amharic (Aynadis & Yaregal, 2013), Afaan Oromo (Tesfaye, 2011), among others.

### 2.3.3 Statistical-Based Approach

Statistical based grammar checker approach is also known as data-driven grammar checker or machine learning-based grammar checker approach. Unlike a rule-based grammar checker approach, a statistical-based grammar checker approach requires a training corpus to learn what is correct instead of using a manually designed rule. the corpus may be collected from manually or

automatically from different sources such as journals, magazines, newspapers, and other online resources (Aynadis & Yaregal, 2013). These approach uses N-gram models to check the grammatical structure of the sequence of words. Statistical based approach uses a tagged corpus to generate a sequence of tags. This method checks a sequence of tags in the sentence if the sequence is familiar or usual the sentence is correct others ways if the tag sequence is the unusual or uncommon sequence the sentence incorrect. In this approach, it is difficult to provide detailed errors resulted from these systems. Many types of researches are conducted for different languages using statistical based grammar checker approaches such as for English (Alam, UzZaman, & Khan, 2006), Bangla (Alam, UzZaman, & Khan, 2006).

### 2.3.3 Hybrid Approach

A hybrid grammar checker approach is a combination of a rule-based and statistical-based approach that helps to improve the accuracy of the grammar checking system (Abraham Gebreamlak, 2019). In this approach, some errors are solved by hand-crafted rules and some others are solved by N-gram models. Different research is done using this approach for grammar checking such as for Tigrigna (Abrha Gebrekiros, 2018), Swedish (Rickard Domeij, 1999).

### 2.3.4 Deep Learning

Deep learning is one of the subfields of machine learning which consists of a successive layer or it is mainly concerned about algorithms inspired by the brain structure and function which is known as Artificial neural network (Brownlee, 2019). Deep learning is a large or deep neural network algorithm that is different from another machine learning algorithm. Deep learning is trained with large data as compared to other machine learning algorithms. If the given problem needs huge data deep learning can perform better than other machine algorithms (Brownlee, 2019). Deep learning can learn from labeled data and it extracted features automatically.

In deep learning, the word ‘deep’ tells the number of hidden layers which means in deep learning many hidden layers appear in the network depending on the nature of the problem. In traditional machine learning algorithms, the number of hidden layers is small in number, for example, it may be one or two but in deep learning the number of the hidden layer may be 100, 150,200, and others (Brownlee, 2019).

Many deep learning algorithms are used for various NLP applications. The most commonly used algorithms are Recurrent Neural Network, Convolutional Neural network, and deep belief network

have used for many NLP applications (Brownlee, 2017). This thesis mainly describes the algorithms used in this study and some related concepts.

#### *2.2.1.1. Recurrent Neural Networks (RNN)*

The recurrent neural network is one type of deep learning algorithm which mainly focuses on sequence labeling problem (Brownlee, 2017). The main drawback of traditional neural network algorithms is the limitation of considering sequential relations that meant the input and the output are independent. However, in recurrent neural network architecture, the output of the first input layer is input for the second layer and the output of the second layer is input for the third layer and others. For example, in part of speech tagging, in machine translation and in grammar checking to know whether the sentence is correct or not first we have to know each class of word sequentially. In order to remember for a short or long time, the recurrent neural network has its own memory.

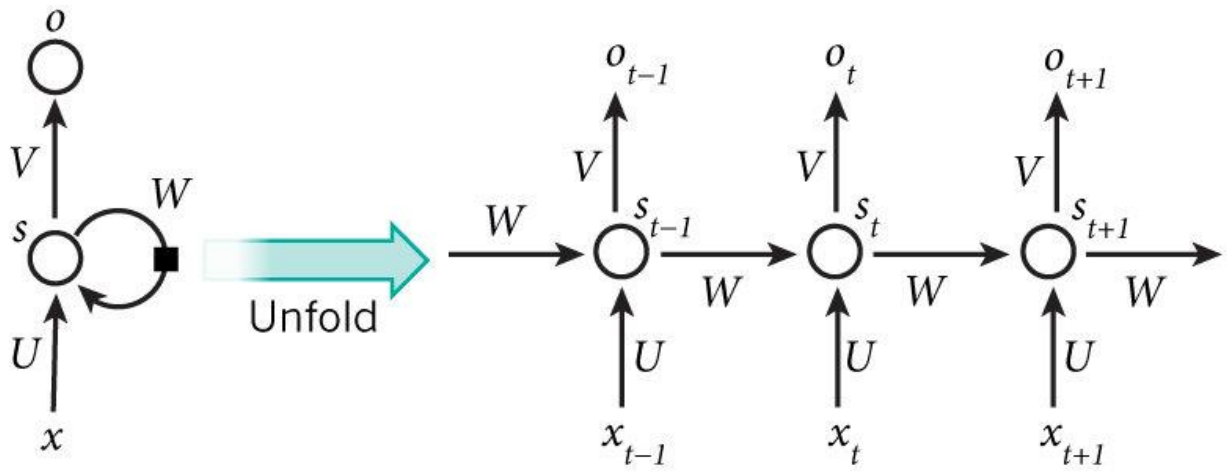



Figure 2. 3 A RNN and the unfolding in time of forward computation (Adopted from Britz, 2015)

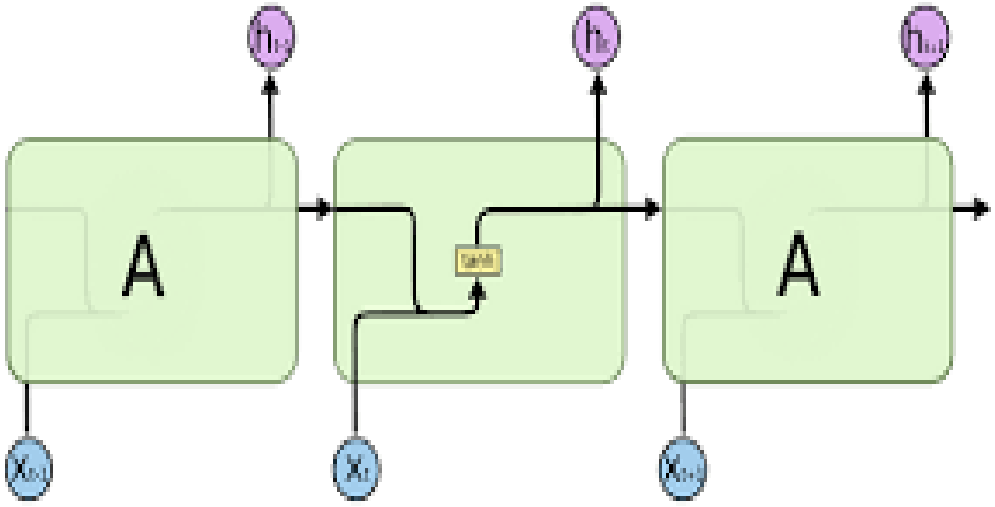
A recurrent neural network and the unfolding in time of the computation involved in its forward computation. From the above figure, xt is the input at a given time step t and st represents the hidden neuron at time t. Therefore the memory of the network can be calculated using the formula st=f(uxt+wst-1). F indicates the function such as sigmoid, relu or tanh. The output layer is represented by Ot at time recurrent neural network sheers the same parameters across a sequence of layers (Britz, 2015).

The recurrent neural network achieves great success in natural language processing applications. The common well known recurrent neural network algorithms are long short-term memory (LSTM), Bidirectional Long short-term memory (BI-LSTM), and GRU (Brownlee, 2017).

#### *2.2.1.2. Long Short-Term Memory Network (LSTM)*

The recurrent neural network has a capability to remember recent information and also unlike feedforward neural network, recurrent neural network has a recursive connection. However, RNN has some drawbacks which are unable to learn long term dependencies, which means when the gap between the information is very long the recurrent neural network is unable to learn the information (Brownlee, 2017).

The other limitation of the recurrent neural network is the problem of vanishing gradient problem. Long Short-Term Memory was proposed by Hochreitor and Schmidhuber in 1997 to handle long term dependency problems of the traditional recurrent neural network problems (Brownlee, 2017). Long Short-Term Memory is called "LSTM" is one kind of neural network that are better than the traditional recurrent neural network by handling the problem of long-term dependencies. In short Long Short-Term Memory have the capability to remember long time information. The chain-like structure of Long Short-Term Memory is the same as Recurrent Neural Network but different repeating module structure. This repeating module in LSTM consists of four layers interacting as compared to a traditional neural network (Olah,2005).



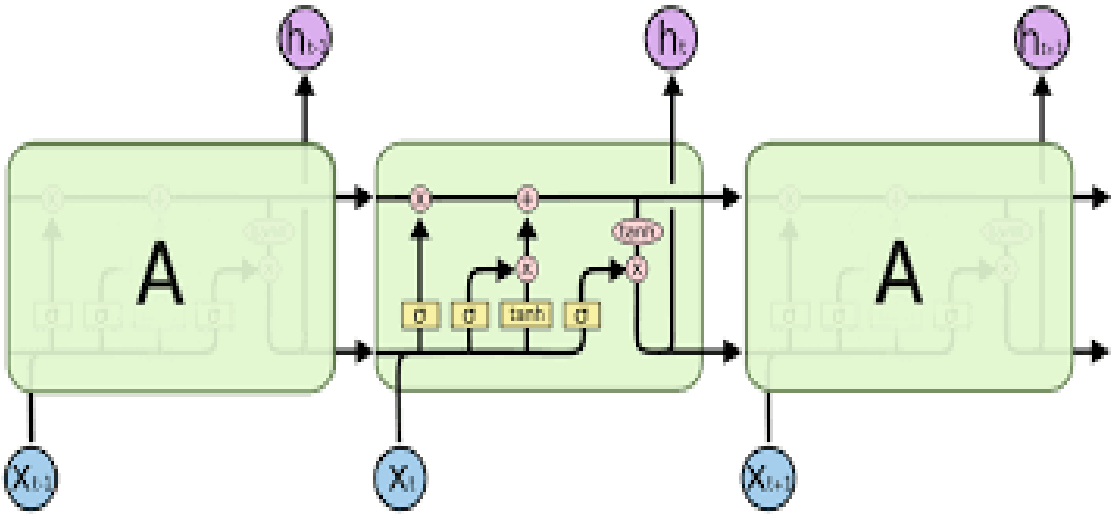



Figure 2. 5 Repeating modules in Long Short Term Memory Network (Adopted from Britz, 2015)

Long Short-Term Memory can add and remove information to the cell state in the help of gates. Gates are ways or methods to optionally let information through. Gates are composed of a pointwise multiplication operation and a sigmoid network layer.

#### 2.2.1.3.Bidirectional Long Short-Term Memory Networks (BLSTM)

The other type of recurrent neural network is the Bidirectional Long Short-Term Memory (BLSTM) network which an extended version of Long Short-Term Memory. LSTM and RNN get information from the previous or the past context only. However, Bidirectional Long Short-Term Memory can get information from past and future information.so BLSTM can handle the problem of Recurrent Neural Network and Long Short-Term Memory network. The BLSTM neural network process input sequence in both directions using two sub-layers which is known as forward and backward direction.

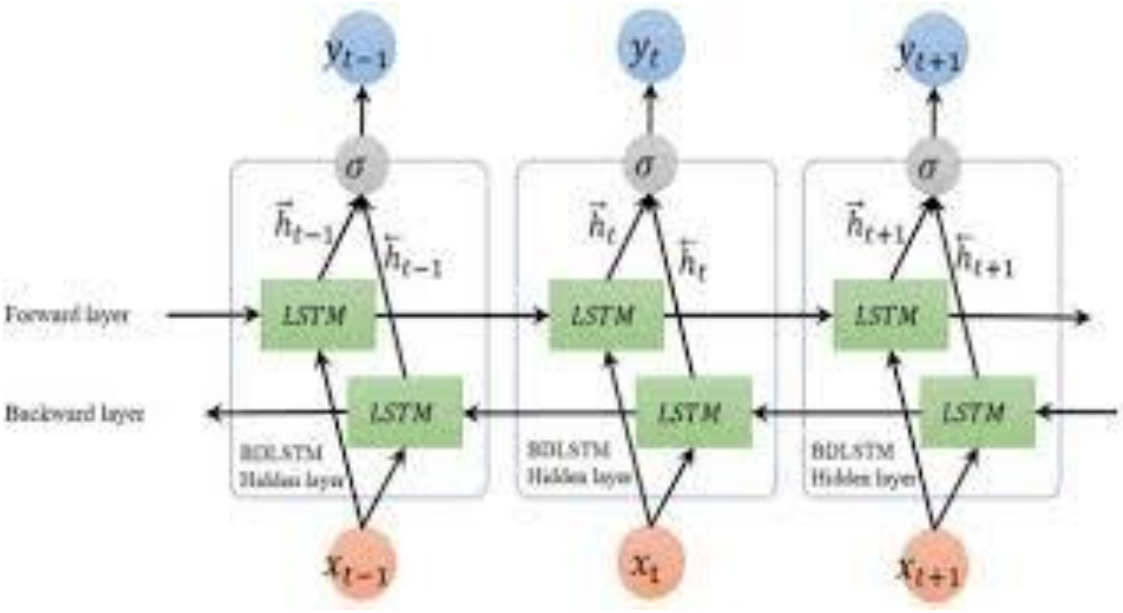



Figure 2. 6 The Bi-LSTM RNN model BiLSTM Recurrent Neural Network Adopted from (Cui, Ke, & Wang, 2018)

## 2.4 Related Works

Several types of researches have done on grammar checkers for different languages with different methods such as English (Naber, 2003), Arabic (Madia & Al-Khalifaa, 2018), Afaan-Oromo (Tesfaye, 2011), Amharic (Aynadis & Yaregal, 2013).

### 2.4.1 English Grammar Error Detection

Naber (2003) has conducted research work on style and grammar checker for English language. The objective of Daniel's study was to design and develop a style and grammar checker system for English text. His proposed system takes input text and displays a list of possible errors found in the input text. In order to detect errors, the text is split into a word and each word is assigned into its word-class such as noun, verb, adjective, determiner, adverb, and other. The style and grammar checker rules consist of 54 pre-defined grammar rules and five style rules. The input text after tagging is matched against all these predefined error rules. These rules describe the sequence of words, it's part of speech tag and chunks. The performance of the proposed system was not evaluated by precision and recall values. As the researcher described in the paper, to calculate the precision and recall values it requires a corpus of yet unedited text with all errors marked up. However, at the time of this research conducted, there was no such kind of resource is publicly available. Therefore, to evaluate the performance of the system the researchers prepared two

corpora. The first one was a corpus that contains texts with errors and the second corpus contains very few errors (BNC's text). When the English style and grammar checker evaluated with this text, it claimed 16 errors in 79,900 sentences. The system also evaluated by another corpus that contains the sentence with errors and compared with Microsoft word 2000 checker. The checker detects 42 errors whereas MS-word detects 49 errors.

### 2.4.2 Arabic Grammar Error Detection

Madia and Al-Khalifa (2018) have proposed a grammatical error detection for the Arabic language. In this study, they proposed Arabic grammatical error detection using a deep learning approach specifically using a recurrent neural network. In this study, they developed a web-based tool that can be used in a browser. The proposed system web page worked as follows it accepts input from the user to the web browser and sends it to the server to check grammatical correctness of the text, then the server hosts the deep learning model to detect errors and finally it displays to the web browser. However, the researcher does not show the performance of the system, simply they claim that the corpus makes challenges for this work because the corpus didn't tell about error types.

### 2.4.3 Swedish grammar checker

A grammar error detection have been developed for Swedish grammar error detection using a hybrid approach by Domeij. This study was done with the combination of a rule-based and statistical grammar checker approach. This proposed system is called Gransaka. The author uses a manually tagged corpus for training purposes. The proposed system is worked as follows first Gransaka tokenizes the given text into sentences and words. After tokenized the sentence into words the given word is tagged with the part of speech using the tagger module. The result of the tagger module is sent to manually prepared rules to check whether the given sentence is grammatically correct or incorrect. In addition to grammatical checker the Gransaka contains spelling checker and correction. Gransaka provides a better result than rule-based systems. However, the system displays false flags at the time of incorrectly tagged words.

### 2.4.4 Afaan Oromo grammar checker

Debela Tesfaye (2011) has conducted his research work on Afaan-Oromo language grammar checker which identifies grammatically incorrect texts in the Afaan-Oromo language. The researcher uses a rule-based grammar checker approach which is done using manually handcrafted

rules.in this proposed system there are five main modules these include tokenization, POS tagger module, stemmer module, grammatical relation finder, and suggestion creating a module. The tokenizer module is the first module of the proposed system and its function is to split the input text or paragraph into a sentence and further split into individual words. POS tagger is the second module of the proposed system which takes input words from the tokenization module and it performs part of speech to each word. The third component of the proposed system is the stemmer module, it takes tagged word form POS tagger module and finds the root and affixes for this tagged word. The fourth component of the system is grammatical relation finder, and it performs grammatical relation between words. The fifth component of this system is the suggestion of creating a module, and its function is to suggest correct sentence options. The proposed system is tested based on precision and recall, and he got an average performance of 88, 89%, and 80% respectively.

### 2.4.5 Tigrigna grammar checker

Abrha Gebrekiros and Yaregal Assabie (2019) have conducted a research work on Tigrigna grammar error detection to detect grammatically wrong errors. The proposed system is done using a hybrid approach which is both rule based and statistical based approach.in this wok they used 114 rules to check Noun-Modifier Agreement**,** Adverb-Verb Agreement**,** Object-Verb Agreement and Subject-Verb Agreement and to check word sequence grammar error they used N-gram probability. This proposed system has around five modules, the first module is tag sequence gathering, the function of this module is from Tigrigna corpus it extracts tag list and calculates the probability for a unique tag sequence from a given list. Preprocessing is the second module of the system which performs two tasks such as tokenization and tagging. Rule based grammar checking third module of this system, the function of this module is identifying grammar errors such as subject-verb, adverb-verb, object-verb and modifier noun agreement errors with manually designed rules. the fourth module is Statistical Grammar Checking the function of this module is used identify word sequence agreement errors using language model, the final and the last module of the proposed system is grammar Error Filtering, it accepts agreement errors from rule based and statistical based, and it tells sentence error type and error words. Finally, the researchers got average precision of 87.9%, recall 87.5% and average f-measure of 87.6%. However, the researchers claim that the performance of the system is less due to that the tagger uses manually

dictionary words and also the system splits the word in to a sentence, at this time the system classifies incorrect sentence as correct.

### 2.4.6 Amharic Grammar Error Detection

Aynadis Temesgen and Yaregal Assabie (2013) conducted the first research work on Amharic grammar error detection using two different grammar checker approaches which are rule-based and statistical-based. The rule-based approach is designed for a simple sentence which easier to formulate rules. However, the statistical approach is designed for both simple and complex a sentence which is difficult to formulate rules. Rule-based approaches are used to identify errors that are formulated by handwritten rules. They analyzed a sentence with phrase structure of words. The statistical or machine learning approach is used to check grammar errors for both simple and complex sentences using N-gram probability. In this research work, the rule-based approaches have certain modules. These include sentence splitter module, morphological analyzer module, grammar relation finder module, and grammar checker module. The function of the sentence splitter module is used to split the inserted text into a sentence, in addition to this it splits sentence into words, and the output of sentence splitter is given to the morphological analyzer module. The function of the morphological analyzer module is it accepts input from sentence splitter and it assigns its linguistic meanings to the input word such as a person, number, gender, and others. The function of the last module which is grammar relation finder is it accepts input from the output of morphological analyzer and it assigns grammatical relations such as subject-verb, object-verb, and others to words in a given sentence. Finally, the grammar checker module accepts input from the grammar relation finder and language model and matches both inputs. According to the evolution the proposed system is tested based on precision and recall, the result shows a precision of 92.45% and recall of 94.23% using a rule-based approach. The statistical-based the researchers got 59.72% precision and 82.69% recall for bigram and 90.38% recall and 67.14% precision for trigram. However, it is difficult to model Amharic sentence for complex sentence using rule-based approach. phrase structure grammar treats all sentences as a sequence of words linear relationship. And also, it is impossible to manually craft all grammar rules exhaustively.

Recently, Abraham Gebreamlak (2019) conducted research work on Amharic grammar error detection. The author proposed a dependency-based grammar checker for Amharic text to identify grammatically incorrect Amharic texts. In this study, the proposed system mainly contains three parts such as ConNll-U Formatter, Dependency parser, and the grammar Checker. ConNll-U Formatter is the first module of the system, in this module sentence tokenization, tagging and morphological features annotator are takes place, first, the input sentence is a structure in SOV format and the formatted sentence is tokenized to single words.to get exposed, universal post tagger and its morphological feature those words are analyzed with taggers. Generally, the function of this formatter is to change the input sentence to appropriate format that the dependency parser understands. The second module of this proposed system is Dependency parser the aim of this module is it takes a formatted sentence from the formatter module to predict head and dependent words, and its dependency relationship using Maltparser. The final part of the proposed system is a grammar checker that contains a relation extractor and agreement checker module. So, the function of this two modules is to check agreement errors, first, the relation extractor takes input from dependency parser and formats the result into a suitable format for agreement checking, and the agreement checker checks grammatical agreements such as Subject-verb, adverb-verb, adjective-noun, and object- verb agreements. Finally, the proposed system is evaluated with MaltEval 1.0, and the result shows 68.18% subject-verb agreement, 20% adverb-verb agreement, 81.25% object-verb agreement, and also the tokenizer performs 100% and the tagger performs 43%. However, according to the researchers they claim that due to the limitation of treebank and the performance of tools, the result of the proposed system is poor. Finally, they recommend the need to conduct further research using large treebank and language-independent tools.

### 2.4.7 Research gap

Generally, we have reviewed researches conducted on grammar checkers using different approaches such as rule-based, statistical-based, hybrid and deep learning-based approaches. Throughout the review, we understand that the rule-based approach has drawbacks, it displays error in case of a compound, compound-complex and complex sentences because of having false flags. The statistical-based grammar checker requires a large size annotated corpus. In addition to this in statistical grammar checker approach, features are extracted manually. However, unlike the machine learning approach deep learning approaches have the capability to learn features automatically. Hybrid grammar checker approach is the combination of rule based and statistical

based approach. However, still the grammar error detection cannot detect well because when the length of the sentence getting larger and larger or simply for compound and complex sentence it was difficult to extract features manually and N-gram probability method. Therefore, deep learning can solve the above problems by learning the sequence of words sequentially and extracting features automatically. From deep learning approaches the bidirectional recurrent neural network can learn the sequence of Amharic text in forward and backward direction. As a result, this algorithm can detect errors that can appear when a sentence having long sequence of words such as compound and complex sentence.

# CHAPTER THREE: METHODS AND APPROACHES

## 3.1 Overview

This chapter presents the designed deep learning-based grammar checker model to detect grammatical errors in Amharic sentence. We have used bidirectional long short-term memory recurrent neural network. Generally, in this chapter, the system architecture of the proposed Amharic grammar error detection model and its components of the proposed model is described in detail.

## 3.2 The architecture of the Proposed System

The overall system architecture of the proposed deep learning-based grammar checker is as shown below in Figure 3.1. The proposed system has sub-components such as Preprocessing, Tokenization, Morphological feature annotator, Word embedding, bidirectional long short-term memory recurrent neural network, grammar result. The functionality of those modules is described as follows; the first module of the proposed system is the preprocessing module. Under preprocessing module tokenization component, sequence padding, tag splitting, and morphology-based tagging is done. The function of this module is cleaning unnecessary texts such as non-Amharic texts or other unnecessary characters, and splitting the input text into a sentence and further split to tokens or words.

After splitting the input text into words, the next component of the system is the morphological tagger component. The main function of morphology-based tagging is morphologically analyzing words or morphologically annotation of words. After morphologically analyzing words the next component of the proposed system is tag splitting. The function of this component is splitting Amharic words from their tag value or extracting morphological annotation of words from words. After splitting sentence in to tokens the next component of the Amharic grammar error detection model is sequence padding. We use sequence padding in order to make each sentence the same length. After sequence padding, the other component of our model is word embedding. Word embedding module is representing sequence of padding in to dense vector representation vector value. After preprocessing and doing all of the above steps, the main and final component of the Amharic grammar error detection is applying bidirectional long short-term memory recurrent neural network. The function of bidirectional long short-term memory module is by taking input

from word embedding component learning sequence dense vectors in both forward and backward directions.

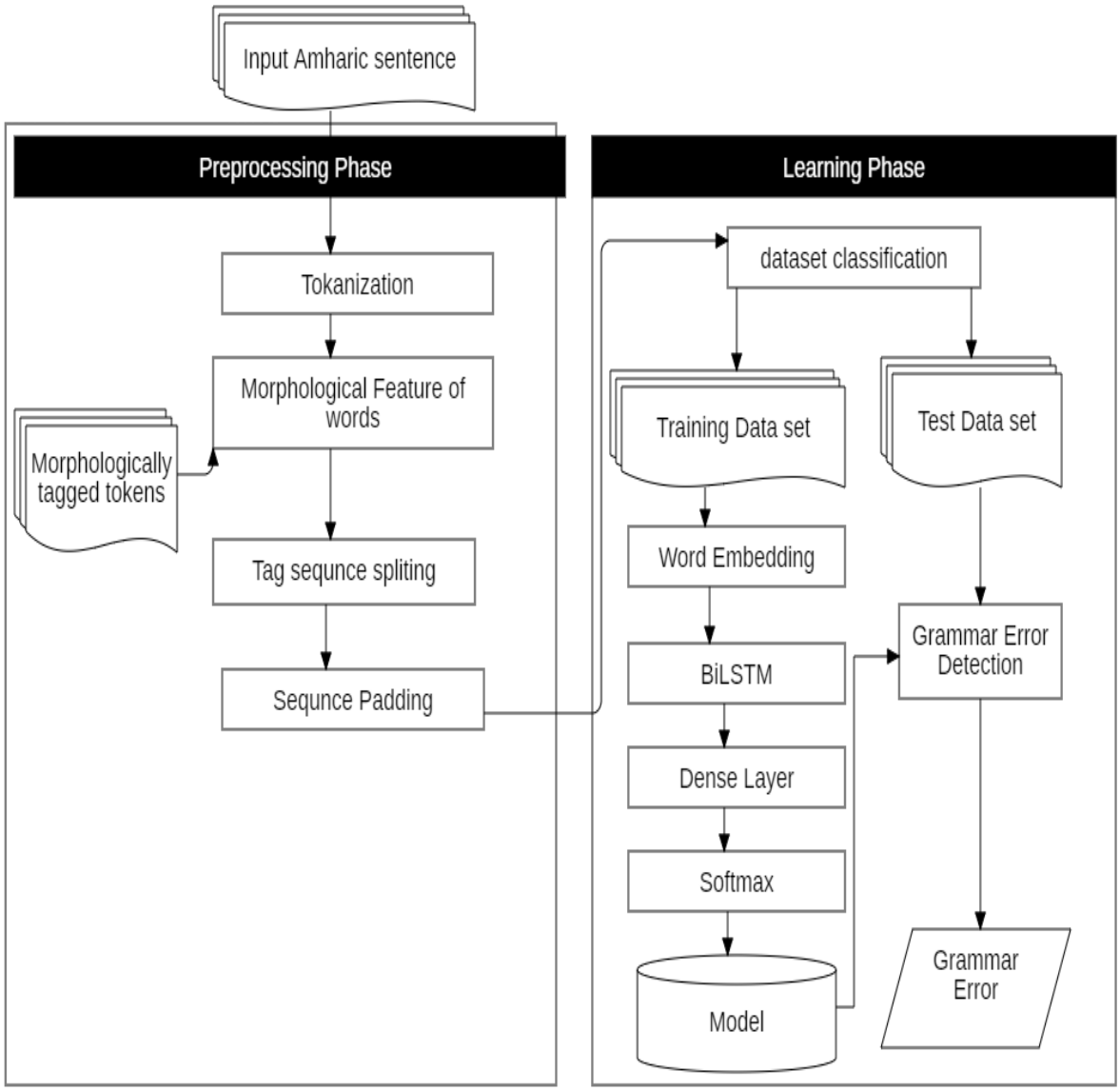



Figure 3. 1 The Architecture for Deep Learning-Based Amharic Grammar error detection

### 3.3 Preprocessing

The first component of the module of the proposed system is the preprocessing module. The preprocessing module mainly contains four main components such as Tokenization, Morphology based post tagging, and tag splitting component and sequence padding.

### 3.3.1 Tokenization

Sentence tokenization is the main part of natural language processing. In simple way tokenization is splitting a sentence in to list of tokens. We used Keras tokenizer, in order to split strings. To make words in sequence, Keras provides “texts to sequences” function. We are splitting the input sentence into words using space separator that means each word is split by space like this split= (“ ”). Generally, the main function of tokenization component is cleaning and splitting of the input text. Cleaning is removing unwanted characters such as non-Amharic texts and other unnecessary characters for this study and tokenizing the input text into sentences and further the sentence is further tokenized to words. In order to split the input text into a sentence, we have used three punctuation marks such as a double colon (።), an Exclamation mark (!), and Question Mark (?) to indicate the end of a sentence. After the input sentence is split to sentence, the sentence is further split into tokens. For example, if the input sentence is "አብርሃም ወደ ማርቆስ ሄደ።" and “ያሬድ ተማሪ ነው?” then the tokenization components split the input sentence as follows: - Suppose if the sentence is like this “አብርሃም ወደ $ ማርቆስ ሄደ ።” this sentence includes $ symbol inside Amharic sentence. So, this problem is solved under preprocessing module. the Keras provides tokenizer function in order to clear such type of symbol.

```
    READ InputText
OR sentences in inputText
  SPLIT sentences into SingleSentence//depend on end punctuation (e.g ።,?)
  FOR SingleSentence in Sentences
    SPLIT SingleSentence into Words// tokenized sentence using white space
    FOR each Wi in Words //Where Wi,i=1,2,3,…,n n is number of sentence.
        Add Wi to TokenizedSentence
      End For
    End For
    End For
```

Algorithm 3. 1 Tokenization module

### 3.3.2 Morphology based post tagging

The next component of the proposed Amharic grammar error detection is morphology-based post tagging. This component appears after segmenting or splitting the input text into tokens, the morphology-based post tagging component accepts input tokens from the tokenization component and it assigns the corresponding tag value for each token from morphologically annotated tokens database. Because to identify whether the sentence is correct or incorrect first the input sentence must be morphologically tagged. The feature of words may be number, gender, person, tense part of speech, and others. So, in order to identify the morphology of each word morphologically tagger corpus is prepared in consultation with language experts. The tagging procedure is taking place in word by word, after tagging each token in the sentence, the tagged tokens have combined to form a sentence. The algorithm 3.2 shows how the morphological information of tokens are tagged. It tells detail steps and procedures in order to tag morphological information words.

```
TokenizedSentence, MorphologyBasedTaggedCorpus
 FOR Wi in TokenizedSentence
       IF  Wi  in MorphologyBasedTaggedCorpus
        ADD Wi  tagOf Wi in TagedSentence
     ELSE
         ADD Wi tagOf Wi is Null
        -END For
```

Algorithm 3. 2 Morphology-based post tagging

Table 3. 1 Sample morphology-based tag description

| Tag Name | Description |
|---|---|
| ADJ_p | Adjective plural in number |
| ADJ_s | Adjective singular in number |
| ADJ_m | Adjective masculine in gender |
| ADJ_f | Adjective feminine in gender |
| ADJ_sm3 | Adjective singular in number masculine in gender third person |
| ADJ_sf3 | Adjective singular in number feminine in gender third person |
| ADV | Adverb |
| ADV_PERF | Adverb perfective in tense |
| ADV_IMPRF | Adverb in imperfective in tense |
| N_Ssm3 | Noun with subject singular masculine in number third person |
| PUNC | Punctuation marks excluding end Punctuation |
| ENDPUNC | End Punctuation tells the end of the sentence |
| N_p | Noun plural in number. |
| N_s | Noun singular in number. |
| N_sm3 | Noun singular in number, masculine in gender. |
| N_sf | Noun singular in number, feminine in gender |
| N_sm1 | Noun singular in number, masculine in gender, first-person. |
| N_sf2 | Noun singular in number, masculine in gender and second person. |
| V_GER_Ssm3 | Gerundive verb, singular in number and masculine in gender. |
| V_IMPF_Ssf3 | Imperfective verb, singular in number, feminine in gender and third person |
| V_PERF_Ssm2 | Perfective verb, singular in number, masculine in gender and second person. |
| V_JUSS_p3 | Jussive verb, plural in number, third-person |
| V_IMPF_Ssm3 _Osm3 | Imperfective verb with the subject (singular, masculine and third person) and object (singular masculine third person) |
| VP_PERF_Sp3 _Osm3 | Perfective verb with the subject (plural third person) and object (singular masculine third person) |
| V_GER_Ssm3 | gerundive verb with the subject (singular masculine third person) |
| VREL_PERF_S sm3_Osm3 | Perfective verb with a subject singular masculine third person and object singular masculine third person. |

Let us take one simples' sentence "እነሱ መጡ ።" in order to tag this sentence the tagger searches this two words እነሱ and መጡ from the morphologically tagged corpus. The result after morphologically tagged this sentence the result have like this "እነሱ PROP_p3 መጡ V_PERF_Sp3 ። PUNC," PROP_Sp3 tells the word is the third person plural pronoun and V_PERF_Sp3 tells the word is a perfective plural verb.

Let us see one complex sentence: - ሙሴ ሕዝቡን ሰብስቡና እናንተ በአምላክ የማትታመኑሰዎች ስሙ! አሮንና እኔ ከዚህ አለት ውኃ እንድናወጣላችሁ ትፈልጋላችሁ? አላቸው ። so, the result of this sentence after morphologically tagging is as follows.

N_sm3> ሕዝቡን<N_p> ሰበሰበና<V_IMPFA_p2> እናንተ<N_p2> በአንላክ<NP_sm3> የማትታመኑ<VP_IMPF_Sp2> ሰወች<N_p> ስሙ<V_IMPF_Sp2> !<PUNC> አሮንና<NC_sm3> እኔ<PRON_s1> ከዚህ<PRONP> አለት<N> ውሓ<N> እንድናወጣችሁ<VP_SP1_Op2> ትፈለጋላችሁ<V_IMPF_Sp2> ?<PUNC> አላቸው<V_PERF_Ssm3_Op3> ።<ENDPUNC>

Figure 3. 2 Sample Amharic morphology-based tagger

### 3.3.3 Tag splitting

After morphologically tagging the input text the next component of the proposed system is tag splitting. The function of this component is to split Amharic text from its morphology feature. For example, from this tagged sentence, we can filter its morphological feature from these two sentences.

Example 1. ሙሴ<N_sm3> ሕዝቡን <N_p> ሰብስቡና <V_IMPFA_p2> እናንተ <N_p2>በአምላክ <NP_sm3>የማትታመኑ<VP_IMPF_Sp2> ሰዎች <N_p> ስሙ<V_IMPF_Sp2>!<PUNC> አሮንና<NC_sm3>እኔ<PRON_s1>ከዚህ<PRONP>አለት<N>ውኃ<N>እንድናወጣላችሁ<VP_Sp1_Op2> ትፈልጋላችሁ<V_IMPF_Sp2>? <PUNC> አላቸው <V_PERF_Ssm3_Op3> ።<ENDPUNC>

Example 2. ትልቅ <ADJ> ልጆች<N_p> ሄዱ<V_PERF_Sp3> ።<ENDPUNC>

Finally, the above sentence after splitting component after the output of morphologically tagged sentence as follows:

Example 1. N_sm3 N_p V_IMPFA_p2 N_p2 NP_sm3 VP_IMPF_Sp2 N_p V_IMPF_Sp2 PUNC NC_sm3 PRON_s1 PRONP N N VP_Sp1_Op2 V_IMPF_Sp2 PUNC V_PERF_Ssm3_Op3 ENDPUNC.

Example 2.ADJ_p N_p V_PERF_Sp3 ENDPUNC

After splitting the morphological feature of words, the next component of the proposed system is sequence padding. The length of the dataset in our corpus is not equal. So, in order to predict each sequence equally the length of each sequence should be equal. In our case we used 100 as maximum length sequence. If the sequence of text is less than the maximum length of the sequence then the adding '0' to each sequence until it fulfils the maximum length of a sequence. The Keras library provides "pad sequences" function to pad sequences sentences. Padding may be pre or post. We are padding tag sequences with maximum length 100 sequences.

For example: from the above Amharic sentence. ትልቅ ልጆች ሄዱ። the result after tokenized and morphologically tagged is ትልቅ <ADJ> ልጆች<N_p> ሄዱ<V_PERF_Sp3> ።<ENDPUNC>.At the same time this sentence after tag splitting is ADJ_p N_p V_PERF_Sp3 ENDPUNC. After splitting morphological feature of words from Amharic text the next thing is sequence padding. So, let as pad two different length tag sequence from example 1 and example 2.

Example 1. PRON N NP ADV_IMPF NP CONJ NP N N N V_PERF_Ssm3 N V_PERF_Ssm3 NC_sm3 VP_PERF_Sp3_Osm3 NP_sf3 N_p N NP N VN_sm3 V_GER_Sp3 ENDPUNC, Example 2. ADJ_p N_p V_PERF_Sp3 ENDPUNC

After padding the sequence with maximum length is as follows: -
Example 1: [46, 1, 2, 18, 2, 41, 2, 1, 1, 1, 7, 1, 7, 85, 51, 52, 4, 1, 2, 1, 14, 12, 3]
Example 2. [33, 4, 13, 3]

When the maximum pad length =100

Example 1 [ 0 0 0 ... 4 13 3]
Example 2 [ 0 0 0 ... 14 12 3]

In this study, Algorithm 3.3. presented below is used for Amharic morphological Tag splitting.

```
Read TaggedSentence, TaggedCorpus
    Set TaggSplit=" "
FOR each TaggedSentence in TaggedCorpus
     split TaggedSentence by ENDPUNC
     End For
FOR each Tagglist in TaggedSentence
         split Tglist from Tokens
         ADD taglist in TaggSplitCorpus
          End for
```

Algorithm 3. 3 Tag splitting module

### 3.4 Word Embedding

In order to represent words in dense representation or in word vectors we use word embedding layer. Word embedding is defined as representation of words or documents in dense vector representation. Keras library provides embedding layer in order to represent words in unique integer. Therefore, the function of the word embedding component in this proposed system is to represent the output of the sequence padding component into real dense representation. The main function of embedding layer in this model is reducing the size input to low dimensional space

After padding the sequence of tags, we need to change those morphological features into dense vector representation. Because, bidirectional long short-term memory network accepts input in vector representation only. In Keras library, the embedding layer has 3 arguments, such as input dimension, output dimension and input length. We have chosen 100 input length because the maximum number of sequences in our dataset is near to hundred but not greater than this. The number of vocabulary size is total number of unique tags sequences. We have used 32 output dimension and 100 input length. So, after representing sequence into dense vector representation, the output have input to BiLSTM module.so embedding (vocabulary size, 32, 100) is input to the Bidirectional recurrent neural network.

### 3.5 Bidirectional LSTM

BiLSTM is the modified version of long short-term memory recurrent neural network. The BiLSTM recurrent neural network checks the sequence of vectors in forward direction and backward direction. so, this type algorithm is better for sequence checking because the error may be at the beginning or at the end of the sentence. and BiLSTM solves the problem of vanishing gradient problem that arises from recurrent neural network.

This proposed model has certain layers such as input layer, embedding layer, BILSTM layer, Dense Layer, dropout, SoftMax. BILSTM are followed by in our model after representing the sequence of tags in vector representation. After changing the tag sequence into a sequence of real value vectors, the next component of the proposed system is BiLSTM module. BiLSTM accepts input from word embedding component, that means the output of word embedding component is the input to long short term recurrent neural network. As shown in figure 3.1 the output of the embedding layer is given to BiLSTM layer, which means (None, 100, and 100) is the output of the word embedding layer, which contains features vectors. For example, in our model the shape of embedding layer is (None, 32, 100). So, the BILSTM layer changes this shape in to (None, 32), so we have used 32 neurons. The BILSTM network learns the sequence feature vector in forward and backward direction this helps to predict correctly. Therefore, in our proposed model has both LSTM and BILSTM. During training additional layer are important such as dropout layer in order to minimize overfitting and underfitting of a model. So, in this proposed model 0.5 dropout layer are used because during training, the gap between training accuracy and validation accuracy is high, and also the gap between training loss and validation loss is high, so in order to minimize this problem we add dropout layer to my model. The dense layer changes the input data to output data by adding bias and some activation function.so, the output of BILSTM layer is given to dense layer. The dense layer accepts input from BiLSTM layer with a dimension of 32 and produce six dimensions. Because we have six classes such as adjective-noun disagreement, adverb-verb disagreement, correct class, incorrect word sequence, subject-verb disagreement and object-verb disagreement. Therefore, the function of dense layer is minimizing the dimensionality of bidirectional long short-term memory recurrent neural network into exact number of classes. Finally, SoftMax classifier assigns to each number of class based on probability distribution. So,

this classifier displays six different decimal points since we have six labels in our dataset, so, after SoftMax classifier the type of grammar error disagreement have displayed.

Suppose T1, T2, T3, …, TN, where N is number of morphologically tag sequences in the sentence. After changing the TN into integers values using dictionary mapping of tags into D1, D2, D3, …, DN where D is dictionary mapping of tags. then the corresponding sequences of integer values have padded.

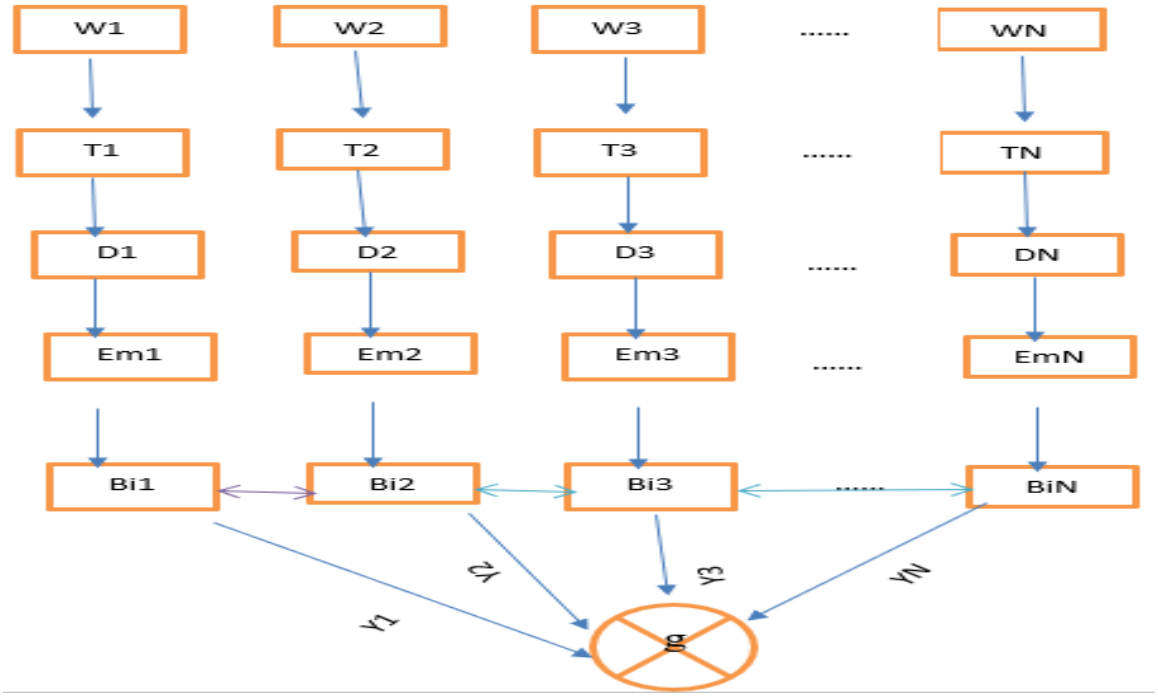



Figure 3.3 sample flow diagram for proposed system

Generally, the above figure worked as follows: -

T(W1,W2,W3, …, WN)=T1,T2,T3, …, TN.

D(T1,T2,T3, …, TN)=D1,D2,D3, …, DN.

E(D1,D2,D3, …, DN )=Em1,Em2,Em3, …, EmN.

B(Em1,Em2,Em3, …, EmN)=Bi1,Bi2,Bi3,…,BiN.

L(Bi1, Bi2,Bi3,…,BiN)=y1,y2,y3, …, yn.

∑ Y1+Y2+Y3, …+yN=g

Where T1, T2, T3, ..., TN refers to sequences of morphologically tagged tokens.

D1, D2, D3, …, Dn refers to dictionary mapping of sequences of integers

Em1, Em2, Em3, …, EmN refers to embedding of sequences of Integers.

Bi1, Bi2, Bi3, …, BiN refers to Bidirectional recurrent neural network.

Y1, Y2, Y3, …., YN refers to the output of each bidirectional recurrent neural network, and g represents the final result of grammar checker.

### 3.6 Demonstration

Generally, the overall model of our proposed Amharic grammar error detection is worked as follows. Let us see the following six Amharic sentence as an example.

1. አለሙ a ትናንት ያገባል ።
2. ከበደ ነገ ያገባል ።
3. ያሬድ ምሳ በላ ።
4. አበበ በሶ በላች ።
5. ሰለሞን ጥቋቁር ላም አሉት ።
6. ሚኪያስ ነጫጭ _በግ አሉት ።

As shown in the figure 3.1, the first component of the proposed model is tokenization. Under the tokenization component, the function of this component is cleaning non-Amharic sentences and tokenizing the sentence into words. For example, in the first sentence, there is an English letter ‘a’, therefore we have removed from Amharic sentence under this component. After tokenizing the sentence into tokens, the next component is finding the morphological feature of the tokens. So, the morphological information of the above six sentences are as follows: -

1. አለሙ N_Ssm3 ትላንት ADV_PERF ያገባል V_IMPRF_Ssm3 ። EDNPUNC
2. ከበደ N_Ssm3 ነገ ADV_IMPRF ያገባል V_IMPRF_Ssm3 ። EDNPUNC
3. ያሬድ N_Ssm3 ምሳ N በላ V_PERF_Ssm3 ። EDNPUNC
4. አበበ N_Ssm3 በሶ N በላች V_PERF_Ssf3 ። EDNPUNC
5. ሰለሞን N_Ssm3 ጥቋቁር ADJ_p ላም N አሉት V_PERF_Sp3 ። EDNPUNC
6. ሚኪያስ N_Ssm3 ነጫጭ ADJ_p በግ N አሉት V_PERF_Sp3 ። EDNPUNC

After finding the morphological information each word in a sentence, the next component is tag splitting, which means splitting the tag sequence from a morphologically tagged sentence. Removing Amharic sentence from tag sequence is important to reduce or minimize the computational time of the system and also the model have learned training data easily, because many different Amharic words may have the same morphological information. For example, when we the first word of all six sentences have a different name but all of the words have the same morphological information as shown below.

1. N_Ssm3 ADV_PERF V_IMPRF_Ssm3 EDNPUNC
2. N_Ssm3 ADV_IMPRF V_IMPRF_Ssm3 EDNPUNC
3. N_Ssm3 N V_PERF_Ssm3 EDNPUNC
4. N_Ssm3 N  V_PERF_Ssf3 EDNPUNC
5. N_Ssm3 ADJ_p N V_PERF_Sp3 EDNPUNC
6. N_Ssm3 ADJ_p N V_PERF_Sp3 EDNPUNC

The next component of the proposed Amharic grammar error detection is sequence padding. From the example, we can see that all six sentences have different length. However, in deep learning the length of the data should be the same size, we must feed a matrix of normalized values, that means all the tokens in the sentence should have the same length. For example, the result of the sentence after padding is as shown below.

```
[[ 1  7  4  2  0]
 [ 1  8  4  2  0]
 [ 1  3  9  2  0]
 [ 1  3 10  2  0]
 [ 1  5  3  6  2]
 [ 1  5  3  6  2]]
```

The result shows that the first four sentences have the same length and the last two sentences have the same length. So, it needs to pad the first to sentence by replacing “0” to each sentence, because the largest sentence in this example has five tokens to, which is greater than the other four sentences so the value of padding have five or ‘5’. So, it needs an additional one ‘0’ at the end of the begging of the sentence.

After padding, the input data have given to the embedding layer. The function of this component is representing the data with dense vector representation. Because in order to feed the data to deep learning model the input data should be normalized and it must be in the form of vector representation. The other main advantage of word embedding is representing making the large dimensional space to low, so the model has learned efficiently.

In order to use deep learning models for natural language processing, it needs to transform words into a numeric representation. There are many methods in order to convert words into vector space such as word2vec, Glove, Keras embedding layer and others. So, in our case in order to represent words in vector representation, we use Keras embedding layer which little bit similar to word2vec, but the difference is word2vec is the unsupervised method in order to make similar words together in the embedding space. However, Keras embedding is a supervised method which learns depends on the data in input (Jason, 2019). Generally, in order to find words into the context, we have train word2vec. For example, To tell if "ወተት" is a likely word given the "ድመቲቱ ጠጣች።" So, in this sentence we expect word2vec. However, Keras embedding is learned as a layer of LSTM, the LSTM is trained in order to predict whatever you went. For example, we train for grammar checking or sentiment analysis, the difference is the data that we input. So, embedding layer learn features for a specific problem. The other importance of embedding layer is to minimize high dimensional space to low dimensional space that means, minimizing the input feature space to smaller one.

Suppose if we give vocabulary size '11' which is the number of unique words and the embedding length is '5' and the size of the input length should be the same with the length of the largest sentence. so, in our case the input length is '.so when we see the result, each token of a sentence has represented with five values. And also, words having the same name are represented with the same vector representation. All of the sentence having padding value '0 'also represented with the same vector. Look at the first four-sentence, when we see the embedding value of the last word is the same which is represented to '0'


Figure 3.4 A dense vector representation of pad sequences

Finally the last component of our model is BILSTM. The function of this component is learning the sequence the input data to predict the exact class of the sentence. so, in this example, the input for BILSTM is the output of the embedding layer. Let us take the last sentence from the example above, input diminution of BiLSTM have “5,5,5”. Therefore, the Bidirectional recurrent neural network has worked as follows below in the figure. every word in the sentence is its unique vector representation. Each of the dense vectors is input to the forward and backwards direction. after learning the sequence. We have a dense layer having six neurons and SoftMax classifier. Finally, the classifier classifies based on each probability distribution. so, the output of this sentence has “AND”, which stands for adjective-noun disagreement. Because the adjective of the sentence tells plural but the noun is singular.

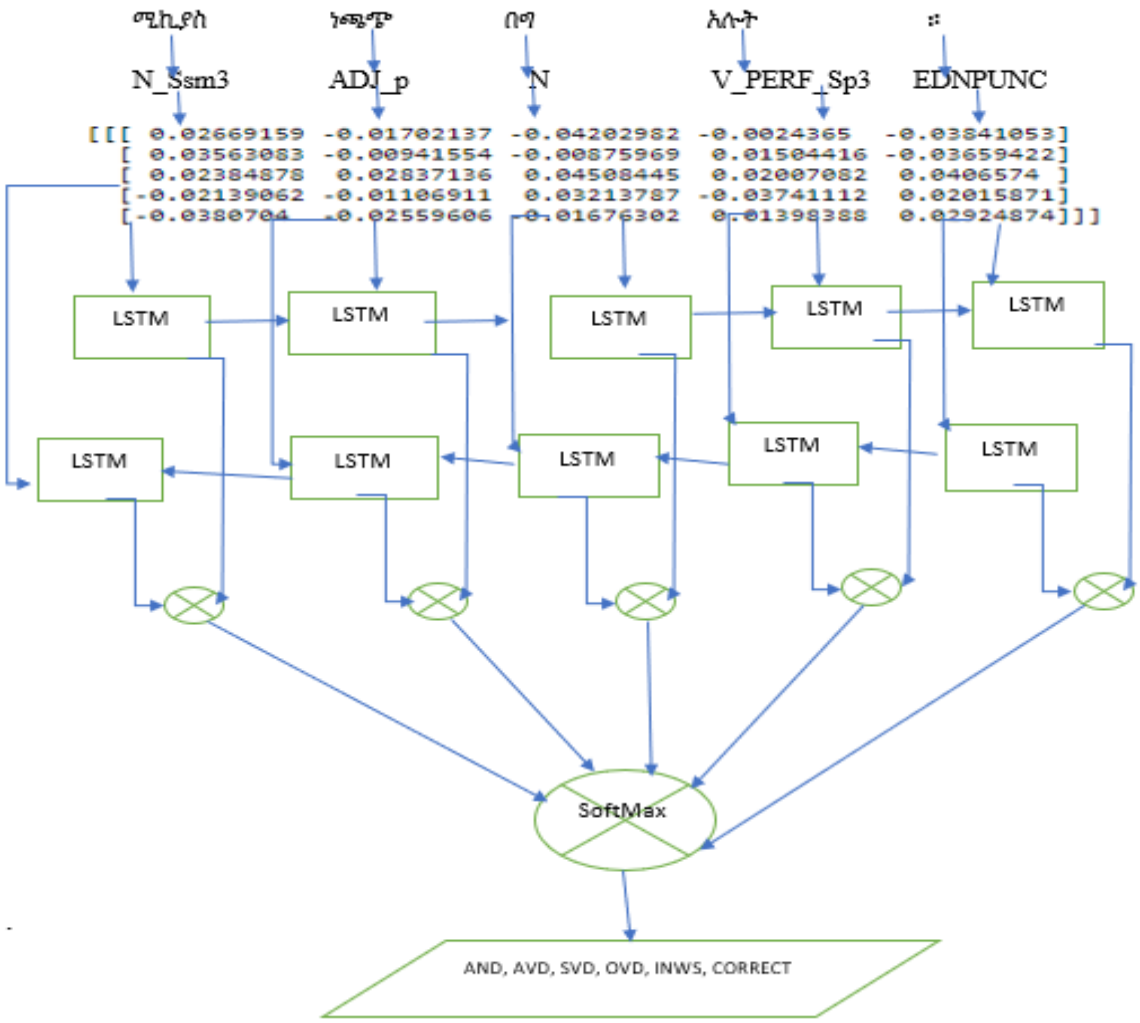



Figure 3.5 steps to grammar checker using BILSTM

Finally, the last component of our model is Bidirectional long short-term memory recurrent neural network. The function of this component is learning the sequence the input data to predict the exact class of the sentence. so, in this example, the input for BILSTM is the output of the embedding layer. Let us take the last sentence from the example above, input diminution of BiLSTM have “5,5,5”. Therefore, the Bidirectional recurrent neural network has worked as follows below in the figure. every word in the sentence is its unique vector representation. Each of the dense vectors is input to the forward and backwards direction. after learning the sequence. We have a dense layer having six neurons and SoftMax classifier. Finally, the classifier classifies based on each probability distribution. so, the output of this sentence has “AND”, which stands for adjective-noun disagreement. Because the adjective of the sentence tells plural but the noun is singular.

# CHAPTER FOUR: RESULTS AND DISCUSSION

## 4.1 Overview

One of the aims of this research is to develop a prototype of a deep learning-based Amharic grammar error detection. The prototypes include Preprocessing, tokenization, morphology-based tagger, word embedding, and Long Short-Term Memory models. So, in this chapter, we have discussed grammar checker evaluation metrics, corpus preparation for training data, and morphology-based tagger. We have also discussed experimental details and evaluation results of grammatical error checking.

## 4.2 Data Collection and Preparation

The first activity of a deep learning-based grammar checker system is data collection and preparation. We have collected around 7,000 sentences from different sources such as Contemporary Amharic Corpus (CACO), Walta Information Center and we have collected from various sources such as Newspapers, sport, News articles, Ethiopian News Agency, Magazines, Fictions, Historic Novel, Short stories, History books, Politics book, Children's book, and Amharic Bible. However, we have used only 3,881 sentences having 50,000 tokens. The data set has contained both the correct and incorrect sentence. The incorrect sentence contains different types such as incorrect word sequence, subject-verb disagreement, object-verb disagreement, Adverb-verb disagreement, and subject-verb disagreement. To make the data suitable for my proposed system it needs further processing, so we prepared a morphologically annotated corpus to detect agreement errors such as number, gender, person, and tense agreement errors.

After collocating data from a different source, it needs further preparation or preprocessing of the data. So, we have prepared a morphologically tagged corpus which contains around 120,000 words. The feature of words may be gender, number, person, and tense. For instance, the morph syntactic information of the word "መጣች" can be written as "መጣች V_PERF_Ssf3" that means the given word is a perfective verb, singular in person, masculine in gender and third person.

We have prepared three corpus files the first one is morphing syntactically tagged corpus. The corpus contains collection sentences that are morphed syntactically each word in a sentence is given. The second corpus that we have prepared is a morph syntactically tagged tokens or a tagger

database. The third corpus is our proposed system training data. The dataset that we have prepared have different size for each class for example for correct 1312, for Adjective-Noun disagreement 249 sentence, for Adverb-Verb disagreement 504 sentence, for Object-Verb disagreement 564 sentence, for Subject-Verb disagreement 750 sentences, and incorrect word sequence, 500 sentences are prepared for both training and testing case  Generally, figure 4.1 shows a sample morphology-based annotated corpus database which contains a list of tokens and its corresponding token information.

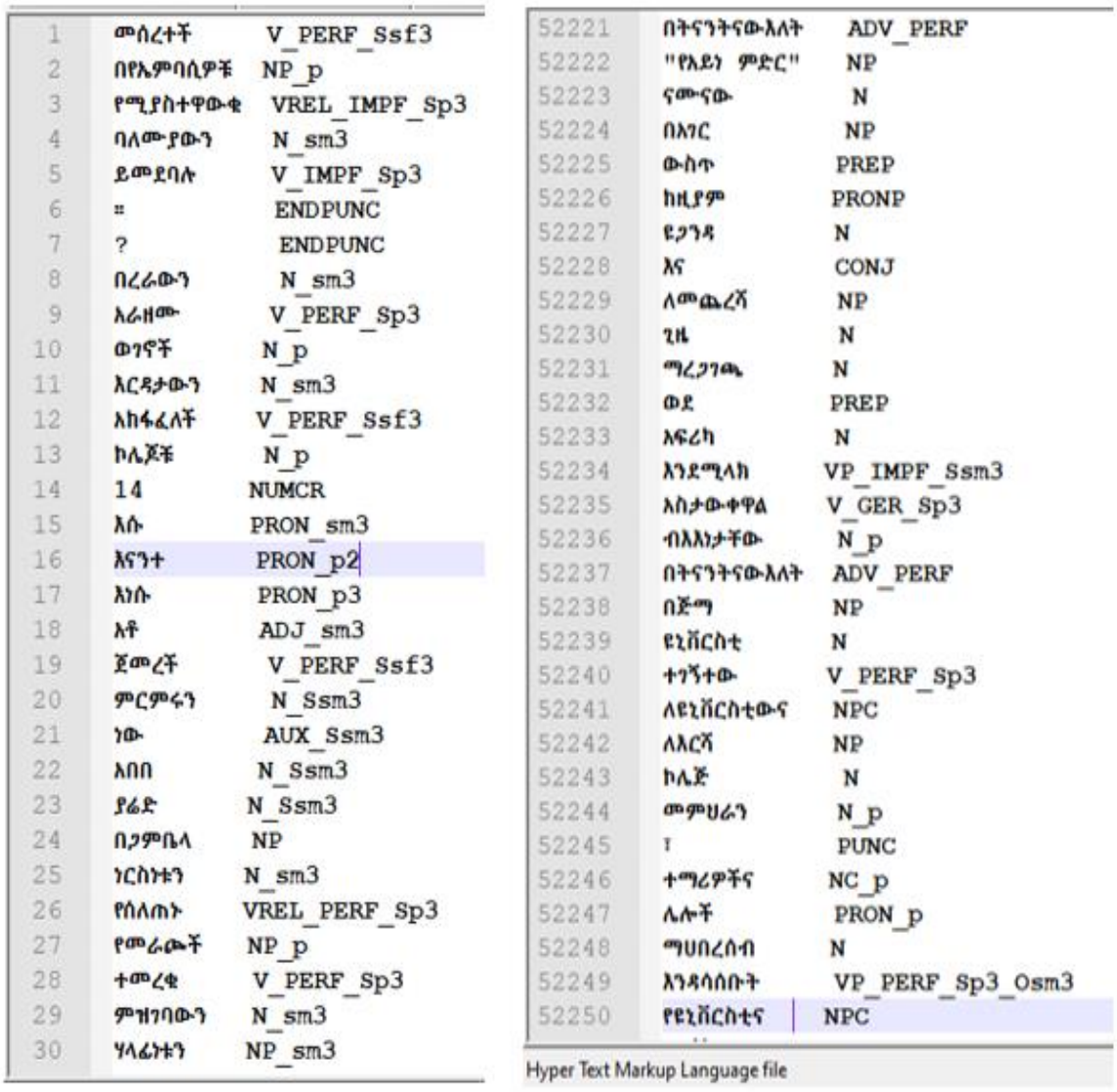



Figure 4. 1 sample morphology-based annotation dictionary database

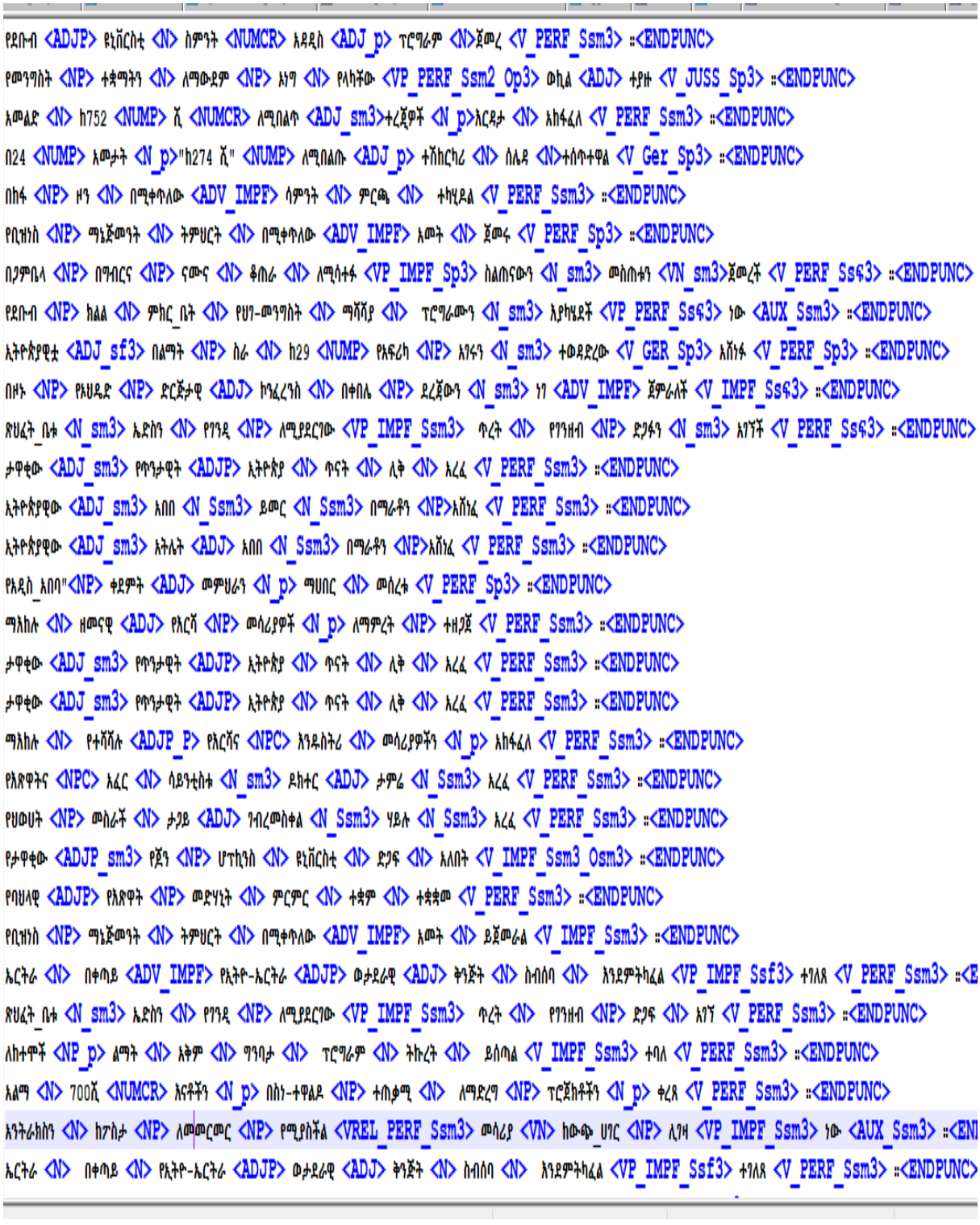



Figure 4. 2 Sample morph-syntactic information of sentence.

```
NUMP ADJ_p N ADJ VP_PERF_Ssm3 N V_PERF_Ssm3 AUX_Ssm3 ENDPUNC,AND
NP ADJ N_p PREP N PREP VN_sm3 V_PERF_Ssm3 ENDPUNC,AND
N ADJP_P NPC N N V_PERF_Ssm3 ENDPUNC,AND
ADJP N_Ssf3 N_Ssm3 NP N V_PERF_Sp3 ENDPUNC,AND
NP N N ADJ_p NP N V_PERF_Ssm3 ENDPUNC,AND
N NPC N VREL_PERF_Sp3 ADJ_p N V_PERF_Ssm3 ENDPUNC,AND
ADV_IMPRF NUMCR N_p NUMP N N NP N VREL_PERF_Sp3 NUMCR N_p VN VN_p3 NP N N N V_PERF_Ssm3 ENDPUNC,AVD
NP N N ADV_IMPRF NUMCR N_p NUMP N N NP N NC VREL_PERF_Sp3 NP_p N N_p VN_sm3 V_PERF_Ssm3 ENDPUNC,AVD
N ADV_IMPRF NUMCR N_p ADJ NP N PRONP NPC N N_p NC_p NUMCR NP N_p VN_sm3 ADJ_sm3 N_Ssm3 V_GER_Sp3 ENDPUNC,AVD
NP N NP ADV_IMPRF<NUMCR N_p VREL_PERF_Ssm3_Opm3 N ADJ VN_sm3 NP N ADJP NP N_p N N V_PERF_Ssm3 ENDPUNC,AVD
NP N ADV_IMPRF NUMCR N_p N N NP N VP_JUS_Ssm3_Op3 NUMCR N_p NP N VN_sm3 NP N N V_PERF_Ssm3 ENDPUNC,AVD
NP N ADJ N N N VP_IMPF_Ssm3 V_PERF_Ssm3 ENDPUNC,CORRECT
PRON ADJ_sm3 N_Ssm3 NP N N ADJ V_PERF_Ssm3 ENDPUNC,CORRECT
NUMP ADJ_p N_p ADJ VP_PERF_Ssm3 N V_PERF_Ssm3 AUX_Ssm3 ENDPUNC,CORRECT
NP ADJ N PREP N PREP VN_sm3 V_PERF_Ssm3 ENDPUNC,CORRECT
N ADV_IMPF ADJP ADJ N N VP_IMPF_Ssf3 V_PERF_Ssm3 ENDPUNC,CORRECT
NUMCR NP N NP NUMCR N NUMCR V_PERF_Ssm3 ENDPUNC,INWS
N_p NP PREP AUX_Ssm3 N_p NP N NC_p ENDPUNC,INWS
NUMCR NP NUMCR V_PERF_Ssm3 N NP NUMCR N ENDPUNC,INWS
N VREL_IMPF_Ssm3_Osm3 VREL_IMPF_Ssm3_Osm3 NP N AUX_Ssm3 N ADV ENDPUNC,INWS
NP N N_sm3 PRON NP VP_IMPF_Ssm3 N N_sm3 N V_PERF_Ssf3 ENDPUNC,SVD
N N_sm3 NP NC N ADJ_p N_p ADJ VN V_PERF_Sp3 ENDPUNC,SVD
NP ADJPC ADJ NP N_sm3 NP N_p VP PREP AUX_Ssm3 ENDPUNC,SVD
NP N_sm3 N N NUMCR NC_p N N NP V_PERF_Ssf3 ENDPUNC,SVD
ADJ_sm3 N N_Ssm3 N_Ssm3 NP N NP N VP_PERF_Ssm3_Osm3 V_PERF_Sp3 ENDPUNC,SVD
NP_sm3 N VREL_IMPF_Ssm3 N VN_sm3 N_p NP NP VP_IMPF_Sp3 V_PERF_Ssm3 ENDPUNC,SVD
NUMCR VREL_IMPF_Ssm3 NP N_sm3 V_PERF_Ssf3 ENDPUNC,OVD
N NP_p N_p VN_sm3 V_PERF_Ssf3 ENDPUNC,OVD
NP N NP_p N_Ssm3 V_PERF_Ssf3 ENDPUNC,OVD
N VREL_PERF_Sp3 NUMCR N_sm3 V_PERF_Ssf3 ENDPUNC,OVD
N N VREL_IMPF_Sp3 N_sm3 V_PERF_Ssf3 ENDPUNC,OVD
NP N NUMCR NUMCR N_p V_PERF_Ssm3 ENDPUNC,OVD
NUMCR N_p N PREP N_sm3 V_PERF_Sp3 ENDPUNC,OVD
```

As

The

Figure 4. 3 Sample training data  sequence of tags and its target value.

The figure 4.3 shows, the dataset contains six class such as AVD, AND, INWS, CORRECT, OVD and SVD. Because the most common errors in Amharic are Adjective-noun disagreement, adverb-verb disagreement, subject-verb disagreement, object-verb disagreement and incorrect word sequence error (Tensou & Yaregal, 2014), ( Baye Yimam, 1995)

### 4.3 Development environment

In this study, we have used certain supporting tools such as Keras, Tensorflow, HornMorpho. HornMorpho is used to morphologically analyze Amharic words. Tensorflow is a powerful library that makes deep learning faster. Keras is user friendly and provides certain python libraries .and other python libraries that used to enable developers to use optimized algorithms and implements popular machine learning techniques in classification. Some of the important python libraries used in this study are discussed below

a) **Scikit-learn** is the main machine learning library that contains features of various classification, regression, and clustering. It also built on NumPy, SciPy, and Matplotlib provides tools for data analysis and data mining.

b) **NumPy** is provided with mathematical functions that can be used in many calculations and also with an n-dimensional array object in the dataset.

c) **Matplotlib** is the most important visualization libraries to analyze the data. It is a scientific plotting library usually to plot histograms, scatter graphs, lines, ROC curves, and other graphic diagrams.

d) **Pandas** is used for data analysis it can take multi-dimensional arrays as input and produce charts/graphs. Pandas may take a table with columns of different datatypes. It may ingest data from various data files and databases like SQL, Excel, CSV, and so on.

### 4.4 Implementation

Experiments are done based on the prototype developed with Keras (TensorFlow as a backend) on Intel Core ™ i7-7500U CPU, and 8 GB of RAM. The proposed model is trained for 100 epochs, a batch size of 32, and a starting or initial learning rate of 0.001 (1e-3). The data is partitioned into training and testing dataset such that 70 percent of the data is assigned for training the model and 30 percent of the data is allotted for testing and 80 percent of the data is assigned for training the model and 20 percent of the data is allotted for testing.

Deep learning based Amharic grammar error detection is implemented using long short-term memory and bidirectional long short-term memory recurrent neural network. In this study, the prototype of the system used Tokenization of words, morphology-based POS tagged corpus of Amharic language, Tag sequence splitting, Sequence padding, word embedding, Amharic grammar error detector and several development tools.

Components available on a User interface is designed for tagging a sentence (see figure 4.4) and for tag splitting (see figure 4.5) are described below.

**Input area**: is used to accept Amharic texts from users.

**Tag sentence button**: is used to morphological analysis the feature of each Amharic word.

**Split tagged sentence button**: It is used to split sequence Amharic words from its morphological feature.

**Check Grammar button**: is used to predict Amharic grammar error the input sequence of morphological feature of the corresponding Amharic sentence using trained model. Finally, it displays one of the following class like noun-adjective disagreement, subject-verb-object disagreement, adverb-verb disagreement, object-verb disagreement, incorrect word sequence and correct class.

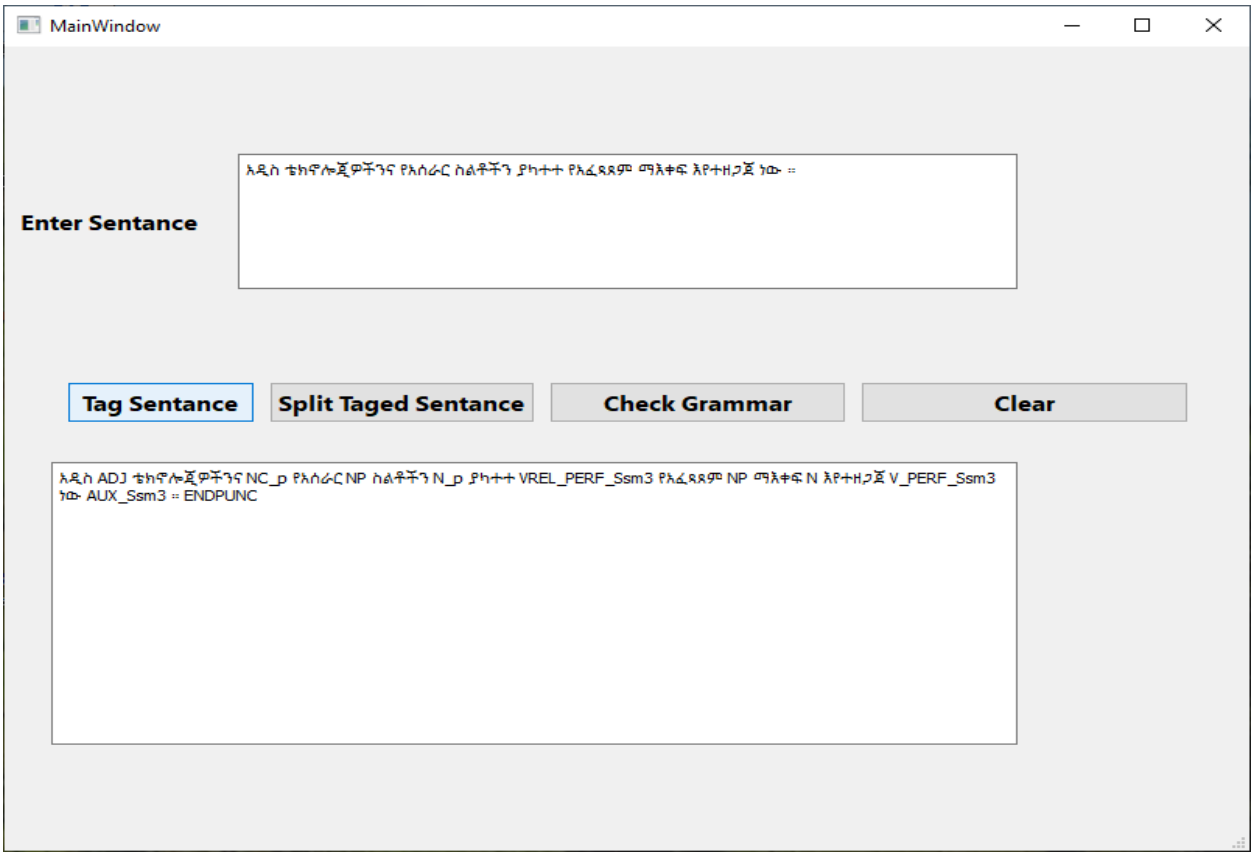



Figure 4. 4 user interface for tagging a sentence

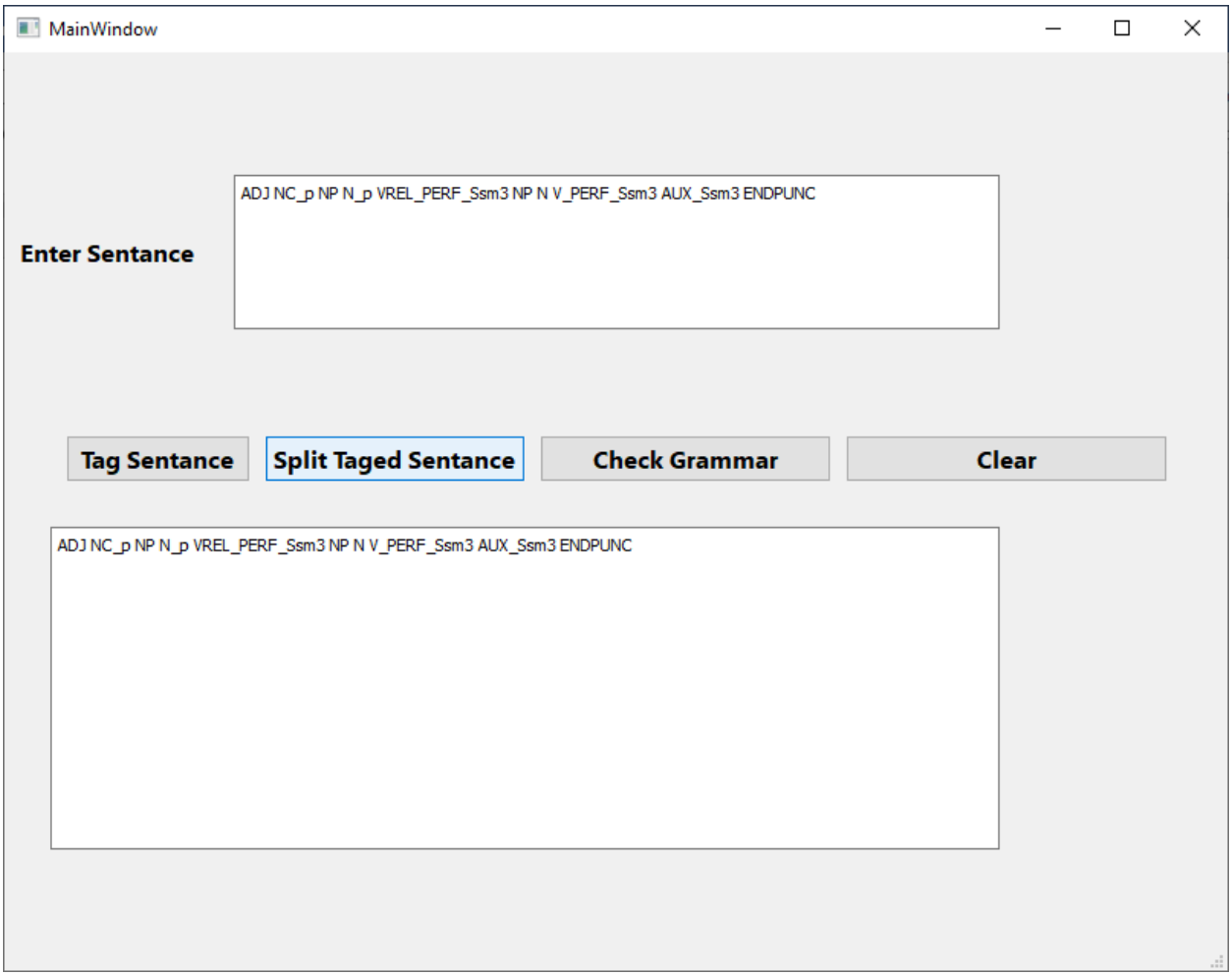



Figure 4. 5 user interface for tag splitting

### 4.5 Performance evaluation and Testing

To check the performance of deep learning-based Amharic grammar error detection evaluation and testing is important. So, we have used the following hyper parameters, such as epoch, batch size, learning rate, drop out, and Adam optimizer. We have evaluated the performance of the system using confusion metrics such as recall, precision and f1 measure (see table 4.1). We selected those evaluation metrics because the size of the dataset is an imbalance; as a result accuracy may not predict exactly, so confusion matrix is mandatory to evaluate the performance of the system.

**True Positives -** These are the correctly predicted positive as positive which means that the value of the actual class is positive and the value of the predicted class is also positive.

**True Negatives -** These are the correctly predicted as negative values which means that the value of the actual class negative and the value of the predicted class negative.

**False Positives –** It shows the actual class is negative but the classifier predicted as positive.

**False Negatives –**The actual class is positive but the classifier predicts as negative. The other evaluation metrics are described.

Table 4.1. evaluation matrix

| **Metric** | **Formula** | **Description** |
|---|---|---|
| Accuracy | $\frac{\text{TP+FN}}{\text{TP+FN+FP+TN}}$ | Overall performance of the model |
| Precision | $\frac{\text{TP}}{\text{TP} + \text{FP}}$ | How the positive description is accurate |
| Recall | $\frac{\text{TP}}{\text{TP} + \text{FN}}$ | Coverage of actual positive samples |
| F1-Score | $2 * \frac{Precison * Recall}{Precison+Recall}$ | The harmonic means of precision and recall |

The prepared dataset contains six class those are Adjective-Noun-Disagreement (AND), Adverb-Verb-Disagreement (AVD), Object-Verb Disagreement (OVD), Incorrect Word Sequence and Correct class. The size of the dataset that we have prepared is 4321. The number of data for each class is different, that means the data set is imbalanced. So, to improve the performance of deep learning-based Amharic grammar error detection it needs to increase or balances the dataset. Therefore, to increase the size of the dataset, we have resampled the dataset from 3881 to 5059. Generally, the figure below shows the total number of data for each class.

```
CORRECT    1559
INWS        700
AVD         700
SVD         700
AND         700
OVD         700
Name: Target, dtype: int64
```

Figure 4. 6 total number of the data for each class

After resampling the dataset we have split the data set into a training set and test set using 70/30 and 80/20 ratio .which means 70% and 80% of the data is for training and 30% and 20% of the data is allocated for testing .further more from the training dataset we have split 20 % for validation. The training dataset in we checked the performance of the system with the above ratio. We have checked the dataset with different epoch values and batch size values. However, we got better result when the epoch value is 100 and 64 batch size. So, the proposed deep learning-based Amharic grammar error detection is trained with epoch 100. We got a better result with those epoch values. And also, we check the batch size with 64. We add dropout layer in order to minimize overfitting and underfitting.

Generally, we checked the system with different hyperparameters such as epoch, batch size, learning rate, optimizer, dropout with 70/30 ratio, and 80/20 ratio. We have checked those parameters in both long short-term memory and bidirectional long short-term memory network. The reason for choosing both LSTM and BiLSTM is a long-short term memory network checks the sequence of tags in a forward direction but bidirectional long short-term memory checks the sequence of tags in both forward and backward direction. The class names or labels are represented with a number starting from 0 to 5. Adjective noun disagreement is represented by 0, adverb verb by 1, correct by 2, incorrect word sequence by 3, object-verb by 4 and subject-verb disagreement represented by 5.

### 4.5.1 Test result using bidirectional long short-term memory (BiLSTM)

In bidirectional long short-term memory network, we have used the following parameters such as epoch=100, batch size=64, Adam optimizer with learning rate 0.01 and dropout=0.5.

The above figure shows the performance of the proposed system when we are applying bi-direction long short-term memory with epoch 64 and batch size 100. We are splitting the dataset with 80% for training and 20% for validation and 20% for testing.as the performance shows the training accuracy is around 91%, and the validation accuracy is 86% and the testing accuracy shows approximately 88.89%.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, 100)               0
_________________________________________________________________
embedding_3 (Embedding)      (None, 100, 32)           2752
_________________________________________________________________
bidirectional_3 (Bidirection (None, 64)                16640
_________________________________________________________________
dense_3 (Dense)              (None, 6)                 390
=================================================================
Total params: 19,782
Trainable params: 19,782
Non-trainable params: 0
_________________________________________________________________

Train on 3079 samples, validate on 770 samples
Epoch 1/100
3079/3079 [==============================] - 4s 1ms/step - loss: 1.3957 - acc: 0.4479 - val_loss: 0.9300 - val_acc: 0.6065

Epoch 00001: val_acc improved from -inf to 0.60649, saving model to weights.hdf5
Epoch 2/100
3079/3079 [==============================] - 3s 1ms/step - loss: 0.9411 - acc: 0.6161 - val_loss: 0.8526 - val_acc: 0.6688

Epoch 00002: val_acc improved from 0.60649 to 0.66883, saving model to weights.hdf5
Epoch 3/100
3079/3079 [==============================] - 3s 1ms/step - loss: 0.7759 - acc: 0.6999 - val_loss: 0.6981 - val_acc: 0.7247

Epoch 00098: val_acc did not improve from 0.86883
Epoch 99/100
3079/3079 [==============================] - 4s 1ms/step - loss: 0.2947 - acc: 0.8876 - val_loss: 0.4087 - val_acc: 0.8597

Epoch 00099: val_acc did not improve from 0.86883
Epoch 100/100
3079/3079 [==============================] - 4s 1ms/step - loss: 0.2827 - acc: 0.8954 - val_loss: 0.4199 - val_acc: 0.8636

Epoch 00100: val_acc did not improve from 0.86883
training Accuracy is 0.9187.
testing Accuracy is 0.8889.
F-measure  is 0.8904.
Recall is 0.8889.
Precision is 0.8983.
Keppa Score is 0.8983.
              precision    recall  f1-score   support

           0       0.99      0.85      0.91       163
           1       1.00      0.98      0.99       147
           2       0.74      0.90      0.81       219
           3       0.96      0.97      0.96       138
           4       0.94      0.90      0.92       144
           5       0.83      0.74      0.78       152

    accuracy                           0.89       963
   macro avg       0.91      0.89      0.90       963
weighted avg       0.90      0.89      0.89       963
```

4. 7 shows the performance of BiLSTM with epoch 100 and  80/20 splitting ratio

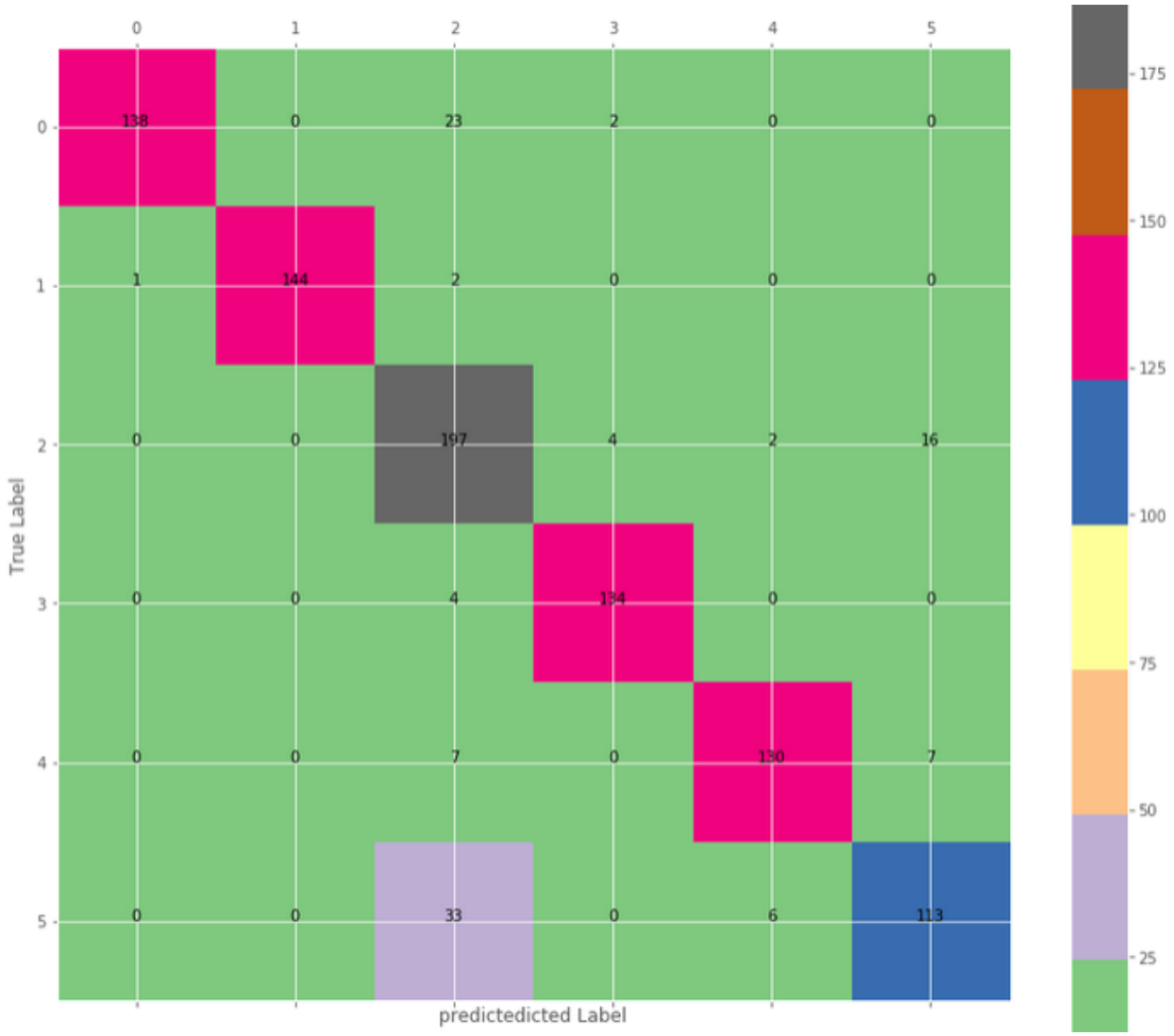



Figure 4. 8 confusion matrix for BiLSTM with 80/20

The above figure 4.8 shows the confusion matrix and loss value. The confusion matrix shows that how many of the data is predicted actual class and incorrectly classified for example the first class in Adjective-Noun disagreement, 138 sentences are predicted correctly and 25 sentences are incorrectly classified.in adverb-verb disagreement, the model predicts 144 sentences correctly and the 3 sentences are wrongly classified as a correct class. From 219 correct sentence, the model correctly predicts 197 sentences and 22 sentences are classified incorrectly. Generally, when we are splitting the dataset with 80/20 ratio and validated with 20% and applying on bidirectional long short-term memory, the performance of the proposed system is 88.89 % accuracy, 89 % f1,88.89% recall, and 89% precision.

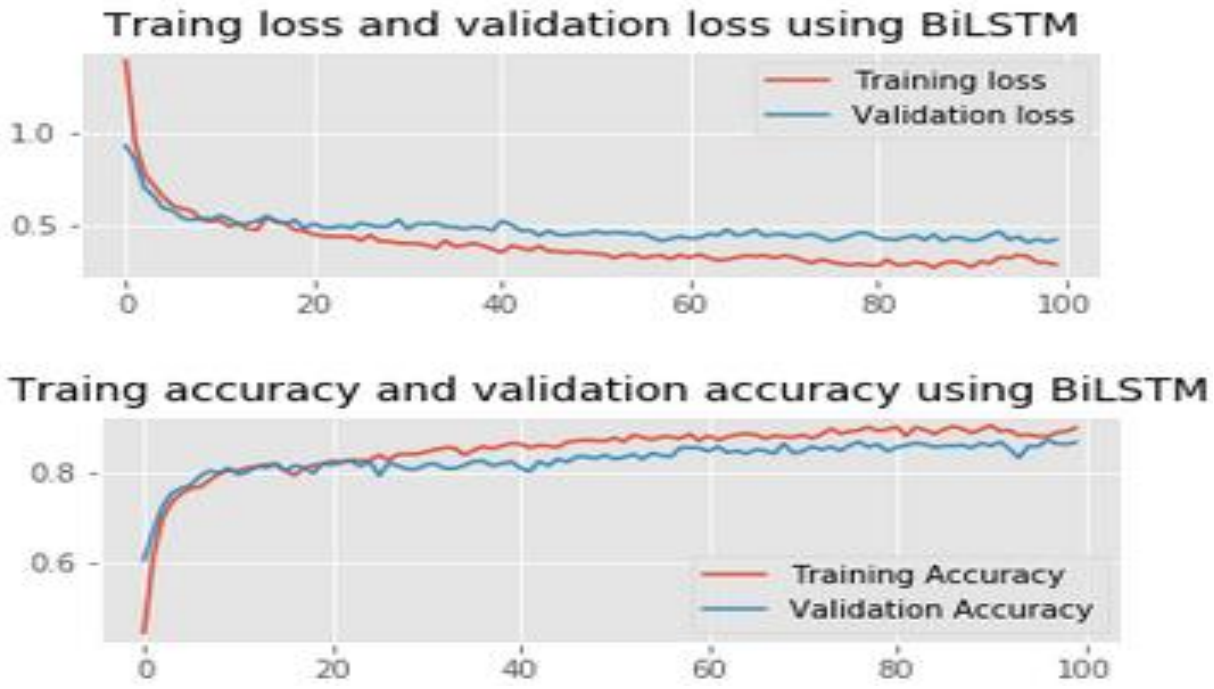


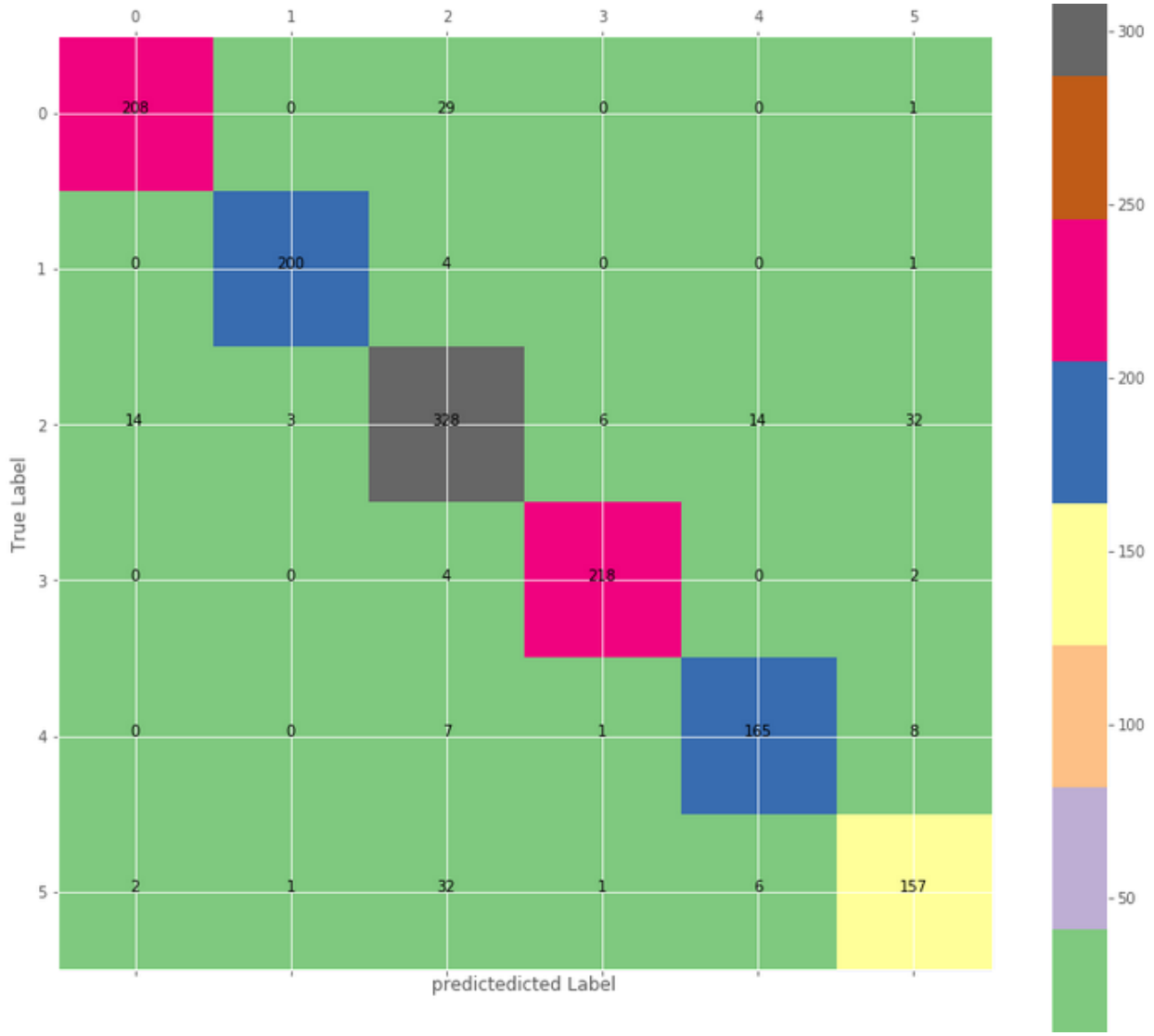

Figure 4. 9 Accuracy and loss for training and validation with BiLSTM 80/20 ratio

Figure 4.9 shown above tells about the performance of bidirectional long short-term memory with 80% of the data set is allocated for training and 20% the data is assigned for testing. So, training loss and validation loss of the model is 0.28 and 0.41 respectively.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_4 (InputLayer)         (None, 100)               0
_________________________________________________________________
embedding_4 (Embedding)      (None, 100, 32)           8192
_________________________________________________________________
bidirectional_4 (Bidirection (None, 64)                16640
_________________________________________________________________
dense_4 (Dense)              (None, 6)                 390
=================================================================
Total params: 25,222
Trainable params: 25,222
Non-trainable params: 0
_________________________________________________________________
Epoch 95/100
2694/2694 [==============================] - 5s 2ms/step - loss: 0.2004 - acc: 0.9313 - val_loss: 0.4401 - val_acc: 0.8665

Epoch 00095: val_acc did not improve from 0.88576
Epoch 96/100
2694/2694 [==============================] - 5s 2ms/step - loss: 0.2075 - acc: 0.9239 - val_loss: 0.4125 - val_acc: 0.8724

Epoch 00096: val_acc did not improve from 0.88576
Epoch 97/100
2694/2694 [==============================] - 5s 2ms/step - loss: 0.2354 - acc: 0.9146 - val_loss: 0.4349 - val_acc: 0.8769

Epoch 00097: val_acc did not improve from 0.88576
Epoch 98/100
2694/2694 [==============================] - 5s 2ms/step - loss: 0.2365 - acc: 0.9154 - val_loss: 0.4187 - val_acc: 0.8739

Epoch 00098: val_acc did not improve from 0.88576
Epoch 99/100
2694/2694 [==============================] - 5s 2ms/step - loss: 0.2479 - acc: 0.9087 - val_loss: 0.4150 - val_acc: 0.8739

Epoch 00099: val_acc did not improve from 0.88576
Epoch 100/100
2694/2694 [==============================] - 5s 2ms/step - loss: 0.2096 - acc: 0.9258 - val_loss: 0.4387 - val_acc: 0.8605

Epoch 00100: val_acc did not improve from 0.88576
training Accuracy is 0.9513.
testing Accuracy is 0.8837.
F-measure  is 0.8839.
Recall is 0.8837.
Precision is 0.8845.
Keppa Score is 0.8845.
              precision    recall  f1-score   support

           0       0.93      0.87      0.90       238
           1       0.98      0.98      0.98       205
           2       0.81      0.83      0.82       397
           3       0.96      0.97      0.97       224
           4       0.89      0.91      0.90       181
           5       0.78      0.79      0.78       199

    accuracy                           0.88      1444
   macro avg       0.89      0.89      0.89      1444
weighted avg       0.88      0.88      0.88      1444
```

Figure 4. 10 Performance of BiLSTM with  70/30 splitting ratio

From the above figure 4.10 shows the performance of Amharic grammar error detection when 70% of the dataset is for training with epoch 100 and batch size 64. So, the experiment shows 88.37% testing accuracy, 88.39% f1 measure, recall 88.37 % and 88.45% precision.




4. 11 confusion matrix for BiLSTM with 70/30

The experiment in figure 4.11 shows that the confusion matrix when we are using bidirectional long short term memory with 70% the dataset is for training and 30% the data is for testing.as shown in the figure, the confusion matrix tells the number of the sentence correctly classified and

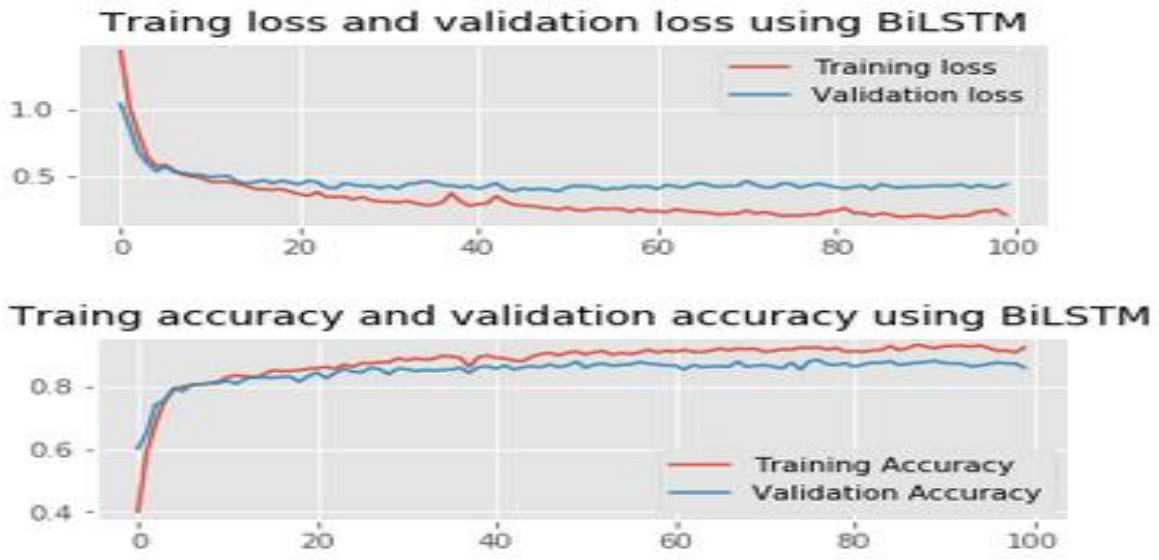



Figure 4. 12 Accuracy and loss for train and validation with BiLSTM 70/30 ratio

incorrectly classified by other class. For example, for adjective-noun disagreement class from a total of 238 sentences 208 sentences are classified correctly and the other 30 sentences are classified incorrectly. If we take object-verb disagreement class from a total of 181 sentences only 165 sentences are predicted correctly and the other 16 sentences are classified incorrectly.

Figure 4.12 shows the accuracy and loss for training and validation using bidirectional long short-term memory with 70% of the data is for training and 30% of the data is for testing the model. So, training loss is 0.21 and 0.43 validation loss.

### 4.5.2 Test result using long short-term memory (LSTM)

The other type of deep learning algorithm is long short-term memory that checks sequence with forwarding direction only. Hire also we are checked the performance with 80/20 and 70/30% ratio.so the performance of the model is described below. Figure 4.13 above shows the performance of long short-term memory network with 80% of the dataset is allocated for the training set and 20% of the data is assigned for the test set.so the model performs 88.27 testing accuracy and 94% training accuracy.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_9 (InputLayer)         (None, 100)               0
_________________________________________________________________
embedding_9 (Embedding)      (None, 100, 32)           8192
_________________________________________________________________
lstm_9 (LSTM)                (None, 32)                8320
_________________________________________________________________
dense_9 (Dense)              (None, 6)                 198
=================================================================
Total params: 16,710
Trainable params: 16,710
Non-trainable params: 0
_________________________________________________________________
Train on 3079 samples, validate on 770 samples
Epoch 1/100
3079/3079 [==============================] - 5s 2ms/step - loss: 1.3511 - acc: 0.4498 - val_loss: 1.0555 - val_acc: 0.5325

Epoch 00001: val_acc improved from -inf to 0.53247, saving model to weights.hdf5
Epoch 2/100
3079/3079 [==============================] - 4s 1ms/step - loss: 0.8999 - acc: 0.6574 - val_loss: 0.7735 - val_acc: 0.6857

Epoch 00002: val_acc improved from 0.53247 to 0.68571, saving model to weights.hdf5
Epoch 3/100
3079/3079 [==============================] - 3s 1ms/step - loss: 0.6599 - acc: 0.7528 - val_loss: 0.6614 - val_acc: 0.7532

...

Epoch 97/100
3079/3079 [==============================] - 4s 1ms/step - loss: 0.1996 - acc: 0.9347 - val_loss: 0.5238 - val_acc: 0.8623

Epoch 00097: val_acc did not improve from 0.87273
Epoch 98/100
3079/3079 [==============================] - 4s 1ms/step - loss: 0.1922 - acc: 0.9337 - val_loss: 0.5126 - val_acc: 0.8714

Epoch 00098: val_acc did not improve from 0.87273
Epoch 99/100
3079/3079 [==============================] - 4s 1ms/step - loss: 0.1978 - acc: 0.9328 - val_loss: 0.5413 - val_acc: 0.8545

Epoch 00099: val_acc did not improve from 0.87273
Epoch 100/100
3079/3079 [==============================] - 4s 1ms/step - loss: 0.2057 - acc: 0.9289 - val_loss: 0.5099 - val_acc: 0.8688

Epoch 00100: val_acc did not improve from 0.87273
training Accuracy is 0.9465.
testing Accuracy is 0.8827.
F-measure  is 0.8833.
Recall is 0.8827.
Precision is 0.8851.
Keppa Score is 0.8851.
              precision    recall  f1-score   support

           0       0.94      0.87      0.90       153
           1       0.98      0.99      0.98       146
           2       0.82      0.86      0.84       250
           3       0.96      0.97      0.97       144
           4       0.91      0.84      0.87       145
           5       0.73      0.77      0.75       125

    accuracy                           0.88       963
   macro avg       0.89      0.88      0.89       963
weighted avg       0.89      0.88      0.88       963
```

Figure 4. 13 performance of LSTM with epoch 100 and 80/20 splitting ratio

Figure 4.14 shows confusion matrix that shows the overall performance of the LSTM model. It shows the performance of each label. For instance, if we see the first label, from a total of 153 sentences 133 sentences are predicted correctly and 19 sentences are incorrectly classified as correct class and one sentence are classified as adverb-verb disagreement.

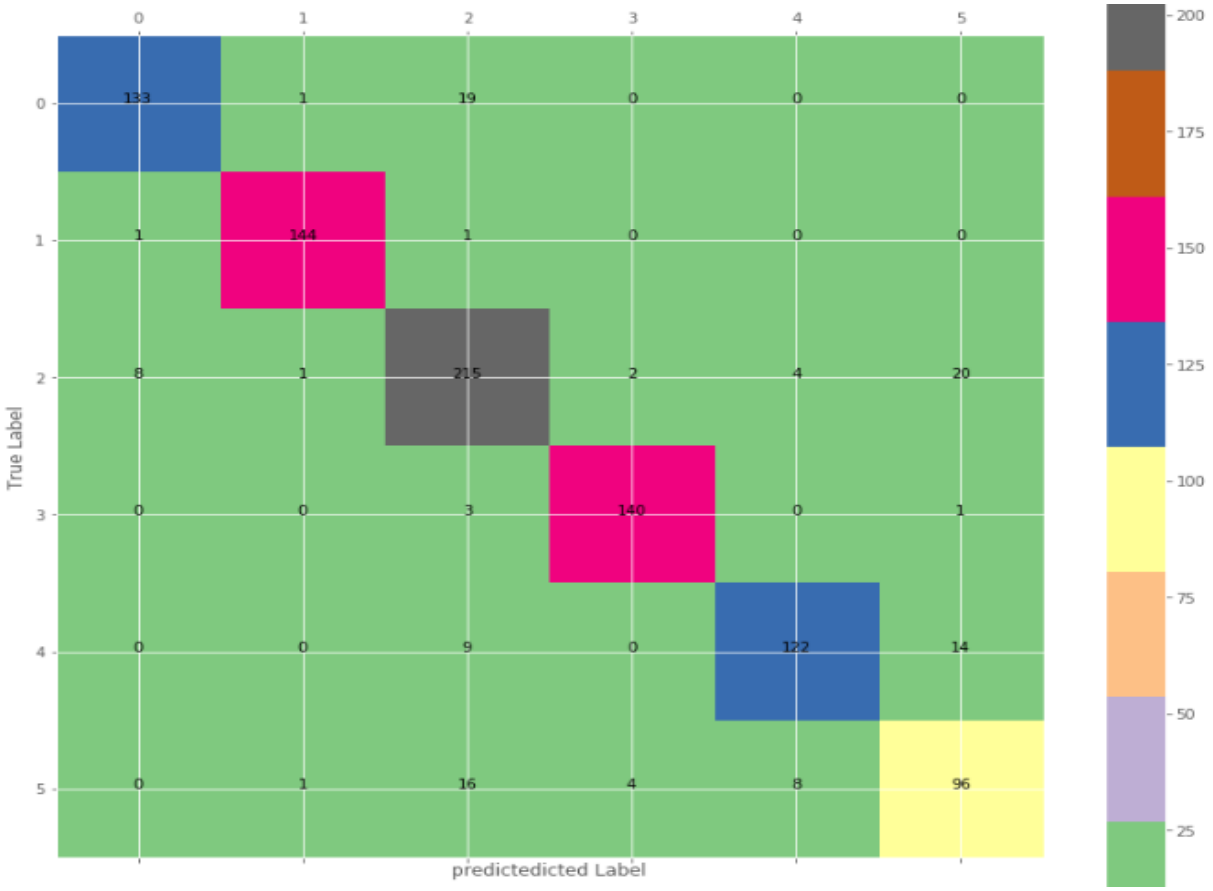



Figure 4. 14 confusion matrix  for LSTM with 80/20

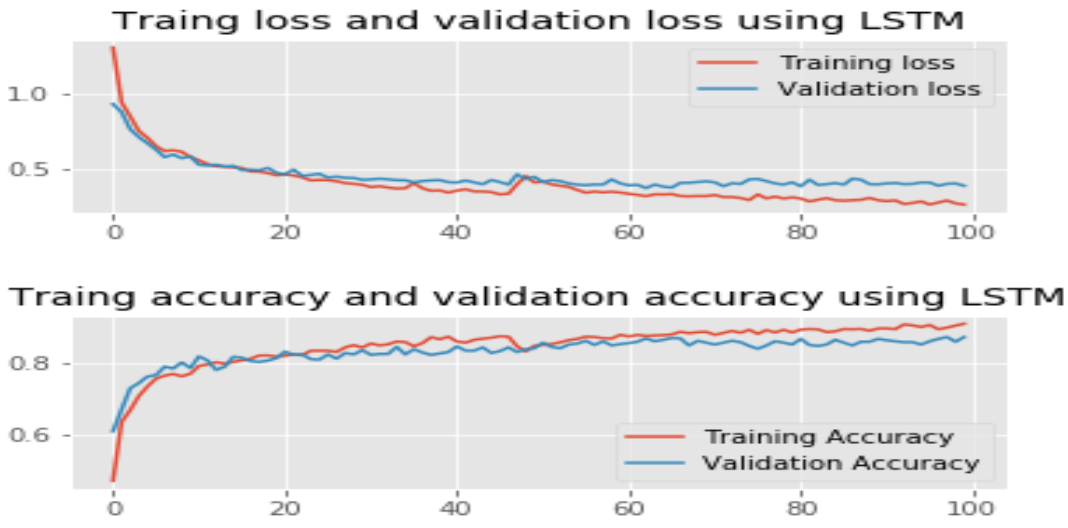



Figure 4. 15 Accuracy and loss for train and validation with LSTM 80/20 ratio

Figure 4.15 shows the experimented result for long short-term memory with 70% of the dataset is for the training set and 30% of the dataset is allocated for the testing set. We have trained the model with 100 iterations with batch size 64. The performance of the model is 88% testing accuracy, 94% training accuracy and, the model is evaluated with confusion matrix f1 score of 88%, recall 88% and precision 88%.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_5 (InputLayer)         (None, 100)               0
_________________________________________________________________
embedding_5 (Embedding)      (None, 100, 32)           8192
_________________________________________________________________
lstm_5 (LSTM)                (None, 32)                8320
_________________________________________________________________
dense_5 (Dense)              (None, 6)                 198
=================================================================
Total params: 16,710
Trainable params: 16,710
Non-trainable params: 0
Train on 2694 samples, validate on 674 samples
Epoch 1/100
2694/2694 [==============================] - 4s 2ms/step - loss: 1.3928 - acc: 0.4321 - val_loss: 1.0158 - val_acc: 0.5786

Epoch 00001: val_acc improved from -inf to 0.57864, saving model to weights.hdf5
Epoch 2/100
2694/2694 [==============================] - 3s 1ms/step - loss: 0.9458 - acc: 0.6388 - val_loss: 0.6728 - val_acc: 0.7626

Epoch 00002: val_acc improved from 0.57864 to 0.76261, saving model to weights.hdf5
Epoch 3/100
2694/2694 [==============================] - 3s 1ms/step - loss: 0.7343 - acc: 0.7275 - val_loss: 0.5896 - val_acc: 0.8027
....
Epoch 97/100
2694/2694 [==============================] - 6s 2ms/step - loss: 0.1927 - acc: 0.9287 - val_loss: 0.4157 - val_acc: 0.8665

Epoch 00097: val_acc did not improve from 0.87537
Epoch 98/100
2694/2694 [==============================] - 5s 2ms/step - loss: 0.2221 - acc: 0.9195 - val_loss: 0.4302 - val_acc: 0.8665

Epoch 00098: val_acc did not improve from 0.87537
Epoch 99/100
2694/2694 [==============================] - 6s 2ms/step - loss: 0.2331 - acc: 0.9139 - val_loss: 0.4424 - val_acc: 0.8635

Epoch 00099: val_acc did not improve from 0.87537
Epoch 100/100
2694/2694 [==============================] - 7s 2ms/step - loss: 0.2112 - acc: 0.9187 - val_loss: 0.4595 - val_acc: 0.8605

Epoch 00100: val_acc did not improve from 0.87537
training Accuracy is 0.9421.
testing Accuracy is 0.8809.
F-measure  is 0.8807.
Recall is 0.8809.
Precision is 0.8814.
Keppa Score is 0.8814.
              precision    recall  f1-score   support

           0       0.94      0.91      0.92       233
           1       0.98      0.99      0.98       200
           2       0.81      0.86      0.83       368
           3       0.98      0.95      0.96       210
           4       0.87      0.88      0.88       207
           5       0.78      0.73      0.75       226

    accuracy                           0.88      1444
   macro avg       0.89      0.89      0.89      1444
weighted avg       0.88      0.88      0.88      1444
```

Figure 4. 16 performance  of LSTM with epoch 100 and  70/30 splitting ratio

Below figure 4.17 shows the confusion matrix that evaluates the overall performance of each label. For instance, for label 0 total number of the tested sentences are 211, however, 22 sentences are correctly classified as label 0 and others are wrongly classified as other classes. if we see label 1, from a total of 200 sentences 198 sentences are correctly classified and 2 sentences are incorrectly classified as a correct class.

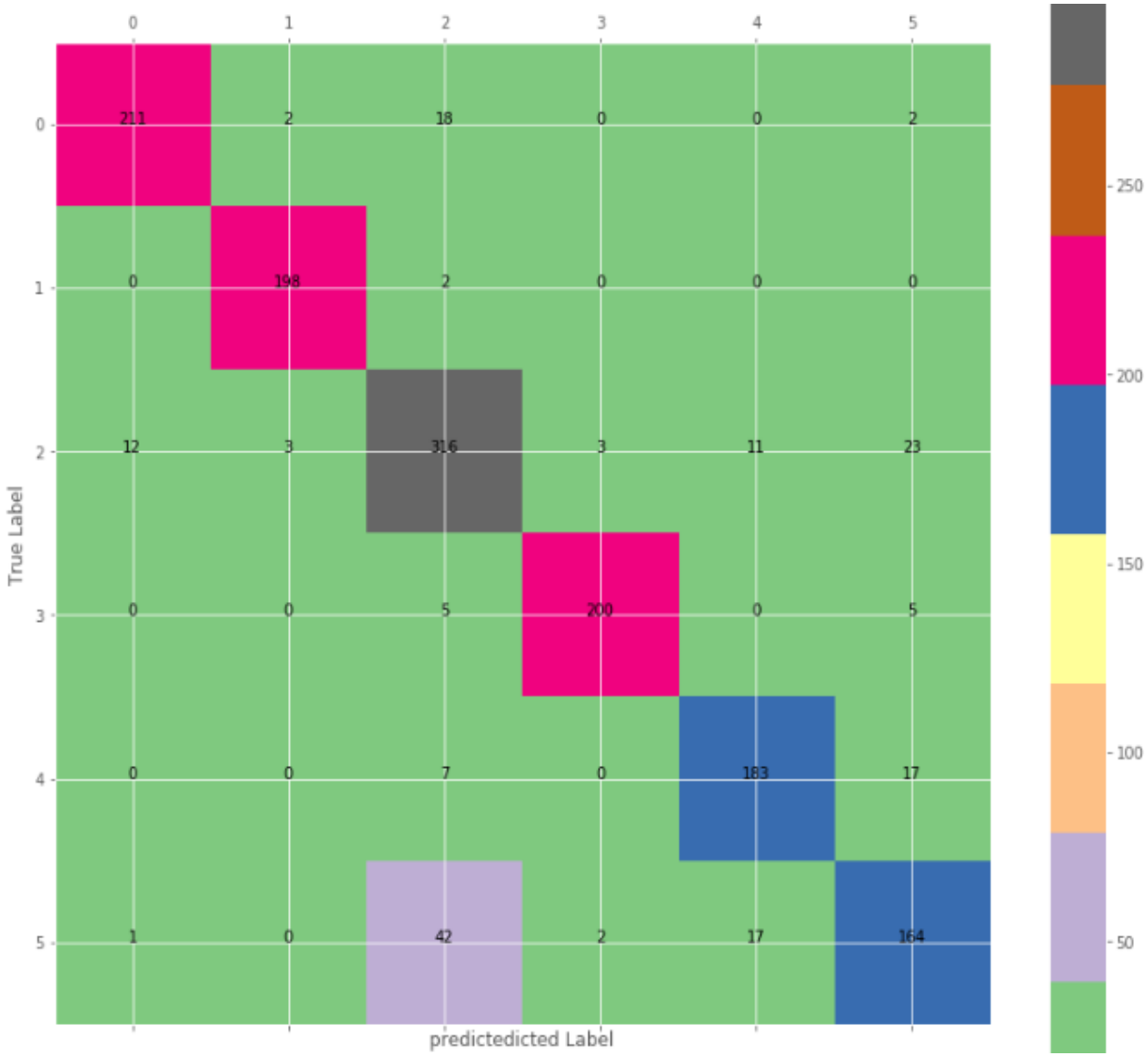



Figure 4. 17 confusion matrix  for LSTM with 70/30

Figure 4.18 tells about the training accuracy and testing accuracy, training loss and validation loss. When we are increasing the epoch the training loss and validation loss is decreasing, but training

accuracy and validation accuracy are increasing. Generally, the training loss is 0.23 and validation loss is 0.49.

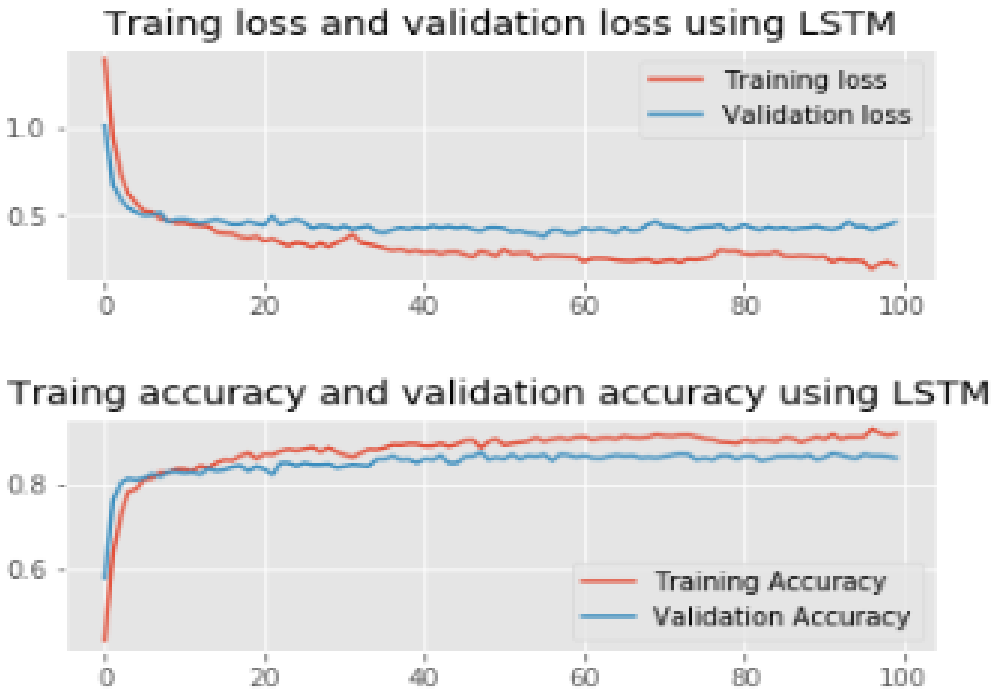



Figure 4. 18 Accuracy and loss for training and validation with LSTM 70/30

### 4.6 Discussion of Results

In this study, we have designed and developed a deep learning-based Amharic grammar error detection that uses long short-term memory recurrent neural network and bidirectional long short-term memory recurrent neural network. The proposed deep learning based Amharic error detection system is evaluated for adjective-noun disagreement, adverb-verb disagreement, object-verb disagreement, subject verb disagreement and incorrect word sequences. Generally, we have detected Amharic sentence errors when adjective and noun disagree with number and gender, adverb and verb disagree in tense, subject and verb (disagree in number, gender and person), object and verb (disagree in number, gender and person), and incorrect word sequences (adjective-noun disorder, adverb-verb disorder, object-verb disorder and, subject-object disorder ). Finally, we add correct class in order to know if the input sentence is valid or not.

we have evaluated the model by splitting the data set with 70/30 and 80/20 ratio for training and testing respectively, and from training data we have taken 20% for validation. In order to train, validate and test the model the following hyper parameters are used such as epoch =100, batch size=64 and dropout=0.5, and Adam optimizer with default learning rate.

As the experiment shows that, the performance of the long short-term memory and bidirectional memory is almost similar. However, bidirectional long short-term memory is a little bit better than long short-term memory. BiLSTM performed 92% training accuracy and 88.89% testing accuracy with 80/20 splitting ratio and 95% training accuracy and 88.37% testing accuracy with 70/30 splitting ratio. And also, the performance of splitting the dataset with 80/20 is better than splitting the data with 70/30%. The experiment shows that during training the model from one iteration to another iteration the accuracy and loss is fluctuating. The reason behind this is because for each iteration different sampled data are taken. So, in order to minimize this problem, we apply dropout layer with 0.5 value. Generally, the detail comparison of the proposed system is shown in the table below.

Since the model is evaluated through recall, precision, f1 measure, and confusion matrix it tells how many of the sentence is correctly or incorrectly predicted by the model. The confusion matrix shows the true positive, false positive, true negative and false negative. For example, figure 4.8 confusion matrix shows from a total of 163 randomly selected adjective noun disagreement sentence 138 sentences are predicted as correctly and 25 sentences are predicted as in correctly. From 147 adverb-verb disagreement sentences 144 sentences are correctly predicted and 3 sentences are in correctly predicted. From 219 correct sentence 197 sentence are correctly predicted by the model and 22 sentences are incorrectly predicted. When we the false positive and false negative for each class, from adjective-noun disagreement class 25 sentence are false negative which are classified as others class. That means the actual class of those sentence are adjective-noun disagreement calls but the model predicted as correct class and incorrect word order class. 1 sentence is false positive, while the actual class of the sentence is adverb-verb disagreement class but the model predicts as adjective-noun disagreement class. When we see adverb-verb disagreement class ‘3’ sentences are true positive and there is no sentence incorrectly predicted as other sentence which means false positive is ‘0’. for correct class, 22 sentences are false negative and 69 sentences are false positive. Which means the actual class of 22 sentences

are correct class but the model predicts as other class. For incorrect word sequence, 4 sentences are false negative and 6 sentences are false positive. For object-verb disagreement class 14 sentences are false negative and 8 sentences are false positive. Finally, when we subject-verb disagreement class 39 sentence are false positive and 23 sentences are false positive.

Generally, as confusion matrix shows that the model predicts most of the data correctly. However, there are sentences which wrongly classified. The reason is that one sentence may have more than one disagreement problems. For example, let see this simple sentence "አበበ ነገ አገባች ። " when we see subject of the sentence the word "አበበ" it tells singular in number masculine in gender and third person however the verb "አገባች" is singular in number, famine in gender, third person and perfective tense. So, there is subject-verb disagreement because the subject of the sentence does not agree in gender with the verb that means the subject is masculine but the verb is feminine gender information. In the other case, when we see the tense agreement, the adverb "ነገ" refers to future tense but the verb refers to perfective tense. So, the sentence have adverb-verb disagreement problem because the adverb refers future but the verb tells past tense.

Therefore, the above Amharic simple sentence have two disagreement problems which is adverb-verb disagreement and subject-verb disagreement.so, in such like case the model may classify in to both classes. Let as see another sentence "ተማሪው ከትምህርት ቤት ሲመልስ የሉም ።" when we see this sentence actually it is correct sentence however the model predicts as subject-verb disagreement problem because the subject "ተማሪው" is singular in number but while the verb "የሉም" tells plural in number. So, in such like case the proposed model may not predict correctly. One of the cause is the quality of morphological information of words which leads the model to predict incorrectly, and also there is miss labeling of a training data.

Generally, as shown in the table above the performance of the two recurrent neural network algorithms are a little bit different. So bi-directional long-short term memory network performs better than long short-term memory. Because bi-directional long short-term memory checks the sequence of tags in a forward direction and backward direction. However, the long short-term memory model checks sequence only in the forward direction. When we see the data split ratio 80/20 ratio is better than 70/30, that means 80% of the dataset is for training and 20% of the dataset for testing is better suited for our model.

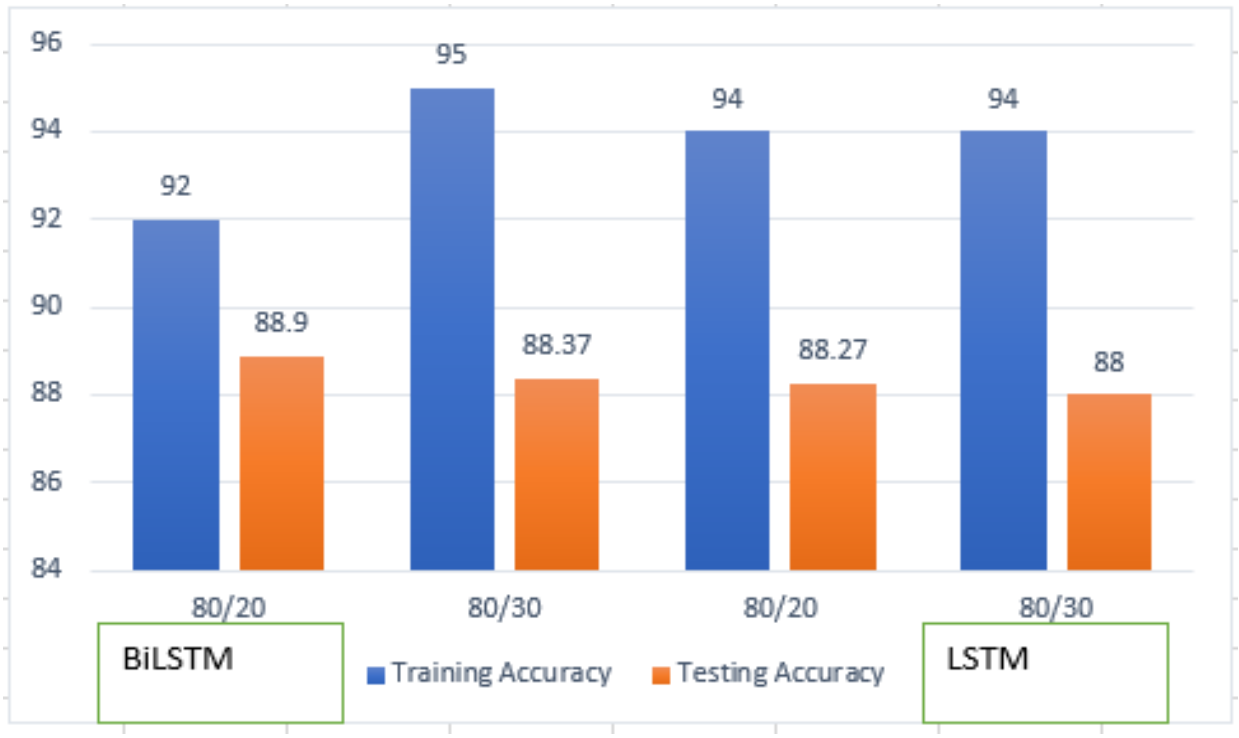



Figure 4. 20 comparsion of  BiLSTM and LSTM

When we compared the proposed deep learning-based grammar error detection system with the exiting grammar error detection systems the proposed model performed better than the existing one. For example, in the previous research's which is done by Aynadis & Yaregal (2013) achieves 92% precision and 94% recall for simple sentence and 67% precision and 90% recall. However, the statistical-based Amharic grammar checker approach is tested using complex sentences in the second test case and it achieves 63.76% of the errors are detected. In the second work by Aynadis & Yaregal (2019), the grammar error detector result shows 68.18% subject-verb agreement, 20% adverb-verb agreement, 81.25% object-verb agreement. Those two works are better performing only for simple sentence but not for compound, complex and compound complex sentences. So, the proposed model performed 89% recall,89% precision and 89% accuracy for all Amharic simple, compound, complex and compound complex sentences. In the previous research's features are generated manually which is difficult to list down all rules because Amharic is morphologically rich and complex language. However, the proposed system can generate and learn features automatically.

# CHAPTER FIVE: CONCLUSION AND FUTURE WORK

## 5.1 Conclusion

This research work aimed to design and develop a deep learning-based Amharic grammar error detection. The grammar checker's modular components are preprocessing module (Tokenization and morphology-based tagger, tag splitting), word embedding and Long Short-Term Memory/bidirectional long short-term memory module. The preprocessing module is to clean the input sentence and to tokenize the sentence and finally to tag the input sentence. After tagging component, the next one is tag splitting. Tag splitting is to extract tags from sentences. After extracting the next component is changing tags into word vector using word embedding. the final component of the LSTM component, this component takes input from word embedding component and predict the input sentence, to check the grammatically correct or not. We have trained and test the proposed deep learning-based Amharic grammar error detection using a manually prepared sentence. We have prepared three corpora such as morphologically tagged sentence, morphologically tagged tokens and tag sequence corpus. The study has six labels such as subject-verb disagreement, adjective verb disagreement, and incorrect word sequence, correct, object-verb disagreement and adverb-verb disagreement, and also, we have prepared additional data set from a collected corpus.

We have implemented a proposed system using python 3.7, Keras TensorFlow as a backend, Pyqt5 to design the user interface and HornMorpho to morphologically analyze the feature of Amharic words. Finally, the performance of deep learning-based Amharic grammar error detection is evaluated with confusion metric. The proposed model performs 88.89% accuracy, f1 measure of 89%, and recall of 88.89% and precision of 89%. However, the quality of morphologically annotated Amharic sentence needs improvement specially for words having more than two meaning and words that tells respect. Since Amharic is morphologically rich and complex language it needs large morphologically annotated corpus. Due to those reasons the proposed model did not correctly detect some sentences.

## 5.2 Contribution of the study

The contribution of this study is listed below.

- We prepare a morphologically tagged corpus that contains 4321 sentences and 60,000 tokens.
- The study proposes the state-of-the-art deep learning approach to check Amharic grammar errors.
- The study contributes to the tagger model for the Amharic language.

## 5.3 Future work

We have done many works to design a model for deep learning-based Amharic grammar error detection, and also the developed model and the morphologically annotated corpus will have used to develop Amharic grammar error correction, machine translation and other NLP application areas. however, in order to increase the performance of the system, still, it needs more work on this area.in our point of view, and we suggest the following things.

- In this study we have considering only Amharic grammar error detector so we suggest to do Amharic grammar error correction.
- We have used only 4300 morphologically annotated sentences, better result will be achieved by building a large Amharic corpus that contains morphology feature of words, automatic spelling checker and morphological analyzer.
- This study is done only by using LSTM and BiLSTM. However, we suggest checking other deep learning algorithm such as encoder-decoder, Transformer neural network and adding attention.
- We recommend also to extend the proposed approach for other local languages so as to design a grammar checker.

# REFERENCES


Baye Yimam. (1995). Yamarigna Sewasiw (Amharic Grammar). *EMPDA Publications*.

ABEBE MIDEKSA, E. A. (2015). STATISTICAL AFAAN OROMO GRAMMAR CHECKER. MSc thesis, *Department of Computer Science,Addis Abeba Universtiy, Ethiopia*.

Abraham Gebreamlak, Y. A. (2019). DEPENDENCY BASED AMHARIC GRAMMAR checker. MSc thesis,*Department of computer sccience,Addis Ababa University, Ethiopia*.

Abrha Gebrekiros, Y. A. (2018). Development of Tigrigna grammar checker using hybrid approach. MSc thesis, *Department of computer science,Addis Ababa University*.Ethiopia

Alam, M. J., UzZaman, N., & Khan, M. (2006). N-gram based Statistical Grammar Checker for Bangla and English. *Center for Research on Bangla Language Processing,* Bangladesh.

Appleyard, D. (1995). *Colloquial Amharic: The Complete Course for bignners.* New York.

Arppe, A. (2000). Developing a Grammar Checker for Swedish. *The 12th Nordic conference computational linguistic*, (pp. 13 – 27).

BACH, E. (1970). Is Amharic an SOV Language ? *Journal of Ethiopian Studies, 8*(1), (pp. 53-66).

Bhirud, N. S., Bhavsa, R., & Pawar, B. (August 2017). GRAMMAR CHECKERS FOR NATURAL LANGUAGES:A REVIEW. *International Journal on Natural Language Computing (IJNLC), Vol. 6*(4), *August 2017*.

Britz, D. (2015). *Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs.* WildML.wordpress,17 Sept.2015.Web. 04 May 2016.

Brownlee, J. (2019). *What is Deep Learning?* Machine Learning Mastery.[Online] Available:https://machinelearningmastery.com/what-is-deep-learning/,Augest 16,2016.

Cui, Z., Ke, R., & Wang, Y. (2018). Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. In 6th International Workshop on urban computing(UrbComp 2017),2016.

Daniel W. Otter, J. R. (2019). A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*. VOL. XX, NO. X, CoRR abs/1807.10854 (2018).

Fitehalew Ashagrie, D. S. (2019). Ancient Geez script recognition using deep learning. *SN Applied Sciences* .https://doi.org/10.1007/s42452-019-1340-4

Gasser, M. (2011). *HornMorpho.* Indiana University, School : School of Informatics and Computing. Bloomington, Indiana.

Gebreamlak, A. (2019). *DEPENDENCY BASED AMHARIC GRAMMAR checker.* MSc thesis, *Department of Computer Science,Addis Abeba Universtiy, Ethiopia*.

Gezmu, A. M., Seyoum, B. E., Gasser, M., & Nürnberger, A. (2018). Contemporary Amharic corpus: automatically morpho-syntactically tagged Amharic corpus. *In: Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*.

Gharehchopogh, F. S., & A.Khalifelu, Z. (2011). Analysis and Evaluation of Unstructured Data:Text Mining versus Natural Language Processing. *Apllication of information and communication technology(AICT).*

Gobena, K. (2011). Implementing an open source amharic resource grammar in GF. *Chalmers University of Technology*.

GOBENA, M. K. ( November 2010 ). IMPLEMENTING AN OPEN SOURCE AMHARIC RESOURCE GRAMMAR IN G. *Chalmers University of Technology*.

Kassa, M. (2010). Implementing An Open Source Amharic Resource Grammar In Gf. *Chalmers University of Technology.* Sweden.

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2017). Natural Language Processing: State of The Art, Current Trends and Challenges.

LEEa, L.-H., LINb, B.-L., YUb, L.-C., & TSENGa, Y.-H. (2017). Chinese Grammatical Error Detection Using a CNN-LSTM Model. *Proceedings of the 25th International Conference on Computers in Education.* New Zealand.

Madia, N., & Al-Khalifaa, H. S. (2018). A Proposed Arabic Grammatical Error Detection Tool Based on Deep Learning. *The 4th International Conference on Arabic Computational Linguistics (ACLing 2018).* Dubai.

Md, J., Uzzaman, N., & Khan, M. (2006). N-Gram Based Statistical Grammar Checker For Bangla And English. *Center for Research On Bangla Language Processing*.

Meshesha, M., & Jawahar, C. V. (2007). Optical Character Recognition of Amharic Documents. *Addis Ababa Univeristiy*.

Naber, D. (2003). A Rule-Based Style and Grammar Checker. *Diplomarbeit. Technische Fakultät Bielefeld*.

Peffers, K., Tuunanen, T., A., M., Rothenberger, & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*.

Rickard Domeij, O. K. (1999). Granska an efficient hybrid system for Swedish grammar checking. *Proceedings of NODALIDA* , (pp. 49-56). Stockholm University .

sak, H., Senior, A., & Francoise, B. (2014). Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. *In Proceedings of the Annual Conference of International.* USA.

Temesgen, A., & Assabie, Y. (2013). Development of Amharic Grammar Checker Using Morphological Features of Words and N-Gram Based Probabilistic Methods. *Department of computer science ,Addis Abeba Univeristy*.

Tensou, & Yaregal. (2014). *Word Sequence Prediction for Amharic Language.* Addis Ababa: Addis Ababa University.

Tensou, T., & Assabie, Y. (2014). *Word Sequence Prediction for Amharic Language.* Addis Ababa University.

Tesfaye, D. (2011). rule-based Afan Oromo Grammar Checker. *(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2* (No. 8).

Yang, Y., Xie, P., Tao, J., Xu, G., Li, L., & Luo, S. (2017). Embedding Grammatical Features into LSTMs for Chinese Grammatical Error Diagnosis Task. *roceedings of the 8th International Joint Conference on Natural Language Processing.* Taipei.

Yifru, M., & Menzel, W. (2009). Amharic part-of-speech tagger for factored language modeling. *in International conference RANLP.* bulgaria.

Yimam, B. (2009). Amharic Grammar Third Edition. *Addis Ababa University Press*.

## APPENDIX A

## SAMPLE AMHARIC PART OF SPEECH TAGGING

| | POS name | Description |
|---|---|---|
| 1 | <ADJ> | Adjective |
| 2 | <N> | Noun |
| 3 | <VREL> | relativized verb |
| 4 | <NUMCR> | Number |
| 5 | <V> | Verb |
| 6 | <ENDPUNC> | sentence end punc (Aratnetb) |
| 7 | <NP> | noun phrase |
| 8 | <VP> | verb phrase |
| 9 | <NUMP> | number phrase with conjuction |
| 10 | <PREP> | Preposition |
| 11 | <VN> | Verb |
| 12 | <ADJP> | Adjective phrase |
| 13 | <NC> | noun with conjuction |
| 14 | <ADV> | Adverb |
| 15 | <PUNC> | Punctuation |
| 16 | <NPC> | noun phrase with conjuction |
| 17 | <AUX> | auxilary verbs |
| 18 | <PRONP> | pronoun phrase |
| 19 | <CONJ> | Conjuction |
| 20 | <NUMOR> | Number |
| 21 | <VPC> | verb phrase with conjuction |
| 22 | <PRON> | Pronoun |
| 23 | <PRONPC> | pronoun phrase with conjuction |
| 24 | <ADJC> | adjective with conjuction |
| 25 | <VC> | verbs with conjuction |
| 26 | <PRONC> | pronoun with conjuction |
| 27 | <UNC> | |
| 28 | <ADJPC> | pronoun with conjunction |
| 29 | <INT> | |
| 30 | <NUMC> | number with conjunction |
| 31 | <NUMPC> | number phrase with conjunction |

## APPENDIX B

## SAMPLE CORRECT AMHARIC MORPHOLOGICALLY TAGGED CORPUS

ኢማተልፈ <N> በአማራ <NP> ያስገነባችው <VREL_PERF_Ssm3_Opm3> መስኖ <N> ፕሮጀክቶች <N_p> ለአገልግሎት <NP> ተዘጋጁ <V_PERF_Sp3> ።<ENDPUNC>
ማህበሩ <N> የስድስት <NUMCR> አመት <N> ስልታዊ <ADJ> እቅዶች <N_p> መጎደፉን <VN_sm3> አስታወቀ <V_PERF_Ssm3> ።<ENDPUNC>
አንድ <NUMCR> የአሜሪካን <NP> ዶላር <N> ብር <N> "8 ነጥብ 5573" <NUMCR> ተሸጠ <V_PERF_Ssm3> ።<ENDPUNC>
በቤንሻንጉል <NP> ጉሙዝ <N> የተገነባው <VP_PERF_Ssm3> መስኖ <N> አገልግሎት <N> መስጠት <VN>ጀመረ <V_PERF_Ssm3> ።<ENDPUNC>
አመልድ< N> ለ1 <NUMP> 3_ሚሊየን <NUMP> ተረጂዎች <N_p> የእርዳታ <NP> እህል <N> አከፋፈለ <V_PERF_Ssm3> ።<ENDPUNC>
የዞኑ <NP> ቤተክህነት <N> ውጣቶችን <N_p> ስለኤድስ <NP> መከላከል <VN> እያስተማረች <VREL_PERF_Ssf3> ነው <AUX_Ssm3> ።<ENDPUNC>
የአውሮፓ <NP> ህብረት <N> አባል <N> አገሮች <N_p> አምባሳደሮቻቸው <N_p> ከአስመራ <NP> አስወጡ <V_JUSS_Sp2> ።<ENDPUNC>
በቦረና <NP> ዞን <N> ለፖርክ <NP> አገልግሎት <N> የሚውሉ <VREL_IMPF_Sp3> ቦታዎች <N_p> ተከለሉ <V_PERF_Sp3> ።<ENDPUNC>
በደቡብ <N> ከቁሪዝም <NP> ከ7_ሚሊየን <NUMP> ብር <N> በላይ ገቢ <N> ተገኘ <V_PERF_Ssm3> ።<ENDPUNC>
አዲስ አበባ <N> ዩኒቨርስቲ <N> ከ125_ሚሊየን <NUMP> ብር <N> የሚጠጋ <VREL_IMPF_Ssm3> ገንዘብ <N> ተመደበለት <V_IMPF_Ssm3_Osm3> ።<ENDPUNC>
ተሃድሶው <N> እስከታችኛው <ADJP> የሀብረተሰብ <NP> ክፍል <N> መሰራት <VN> እንዳለበት <VP_PERF_Ssm3_Osm3> ተጠቆመ <V_PERF_Ssm3> ።<ENDPUNC>
ኢኒስቲትዮቱን <N> ለማጠናከር <NP> የ64ሚሊየን <NUMP> ብር <N> ኮንጥራት <N> ስምምነት <N> ተፈረመ <V_PERF_Ssm3> ።<ENDPUNC>
የደቡብ <NP> ክልል <N> ህዝብ <N> በምርጫው <NP> በንቃት <NP> እንዲሳተፍ <VP_IMPF_Ssm3> ተጠየቀ <V_PERF_Ssm3> ።<ENDPUNC>
አልማ <N> የህክምና <NP> መሳሪያዎችና <NC_p> የወሊድ <NP> መከላከያ <N> መድሃኒቶችን <N_p> አሰራጨ <V_PERF_Ssm3> ።<ENDPUNC>
ቢሮው <N> በ600 <NUMCR> አብያተ_ክርስቲያናት <N> የቅርስ <NP> ምዝገባና <NC> ቆጠራ <N> ያካሂዳል <V_IMPF_Ssm3> ።<ENDPUNC>
ከአምስት <NUMP> አገሮች <N_p> ለአዲሱ <ADJP_sm3> ፕሬዚዳንት <N> የደስታ <NP> መግለጫዎች <N_p> ተላኩ <V_PERF_Sp3> ።<ENDPUNC>
በአማራ <NP> ክልል <N> በመስኖ <NP> የለማው <VREL_PERF_Ssm3_Opm3> አናሳ <ADJ> መሆኑ <VN_sm3> ተጠቆመ <V_PERF_Ssm3> ።<ENDPUNC>
በከፋ <NP> ዞን <N> የአንድ_ሚሊየን <NUMP> ቡና <N> ችግኞች <N_p> ተከላ <N> ተካሄደ <V_PERF_Ssm3> ።<ENDPUNC>
ውሳኔው <N_sm3> የተሰጠው <VP_PERF_Ssm3> በዳኞች <NP_p> ልዩነትና <NC> አብላጫ <ADJ> ድምፅ <N> ነው <AUX_Ssm3> ።<ENDPUNC>
የቬትናም <NP> እና <CONJ> የቻይና <NP> ፕሬዚዳንቶች <N_p> የደስታ <NP> መግለጫዎች <N_p> አስተላለፉ <V_PERF_Sp3> ።<ENDPUNC>
የአፋር <NP> ህዝብ <N> የነቃ <ADJ> ተሳትፎ <V_GER_Ssm3> እንዲያደርግ <VP_IMPF_Ssm3> አብዴፓ <N> ጠየቀ <V_PERF_Ssm3> ።<ENDPUNC>
የኤርትራ <NP> መንግስት <N> ሁለት <NUMCR> የአሜሪካ <NP> አምባሲ <N> ሰራተኞችን <N_p> አሰረ<V_PERF_Ssm3> ።<ENDPUNC>
ድርጅቱ <N> ምርቱን <N> ወደ <PREP> መከከለኛው_ምስራቅ <N> ለመላክ <NP> ሁኔታ <N> አመቻች <V_PERF_Ssm3> ።<ENDPUNC>
ትምህርታቸውን <N_p> ያቋረጡ <VREL_PERF_Sp3> የዩኒቨርስቲ <NP> ተማሪዎች <N_p> በአንገኛ <ADJP> መምህርነት <N> ተሰማሩ <V_PERF_Ssm3> ።<ENDPUNC>
በኬንያ <NP> የሚኖሩ <VREL_IMPF_Sp3> ኢትዮጵያውያን <N_p> 35ሺ <NUMCR> 797 <NUMCR> ዶላር <N> ለገሱ <V_PERF_Sp3> ።<ENDPUNC>
ድርጅቶቹ <N_p> ከ57_ሚሊየን <NUMCR> ብር <N> በላይ <NP> የፕሮጀክቶች <NP_p> ግንባታ <N> አስጀመሩ <V_PERF_Sp3> ።<ENDPUNC>
በቤንሻንጉል <NP> ጉሙዝ <N> መረጃዎችን <N_p> የሚያሰባስበው <VP_IMPF_Ssm3_Osm3> ተቋም <N> ስራ <N> ጀመረ <V_PERF_Ssm3> ።<ENDPUNC>
ዩስትሪክቱ <N> በ36_ሚሊየን <NUMP> ብር <N> ወጪ <N> የመንገድ <NP> ጥገና <N> አከናውኗል <V_GER_Ssm3> ።<ENDPUNC>
የተወካዮች <NP_p> ምክር_ቤት <N> የሀገሪቱን <NP_sf3> ፕሬዚዳንት <N> መተዳደሪያ <N> አዋጅ <N> አጸደቀ <V_PERF_Ssm3> ።<ENDPUNC>
መስተዳድሩ <N> ከፍተኛ <ADJ> ውጤት <N> ላስመዘገቡ <VP_PERF_Sp3> ተማሪዎች <N_p> ሽልማት <N> ሰጠ <V_PERF_Ssm3> ።<ENDPUNC>
የእስልምና <NP> ሃይማኖት <N> በሰላም <NP> ተምሳሌትነት <N> ላይ <PREP> እንደሚያተኩር <VP_IMPF_Ssm3> ተጠቆመ <V_PERF_Ssm3> ።<ENDPUNC>
ኢማተልፈ <N> ያስገነባችው <VREL_PERF_Ssm3_Opm3> 17 <NUMCR> ትምህርት_ቤቶች <N_p> አገልግሎት <N> መስጠት <VN>ጀመሩ <V_PERF_Sp3> ።<ENDPUNC>
ማቱዎሮሌጀ <N> ተምች <N> ለከሰት <VP_IMPF_Ssm3> ስለማችል <VP_IMPF_Ssm3> ጥንቁቁ <N> እንደደ/ማ <VP_IMPF_Ssm3> አሳሰበ <V_IMPF_Ssm3> ።<ENDPUNC>

# APPENDIX C

# SAMPLE ADJECTIVE-NOUN DISAGREEMENT TAGGED SENTENCE

```
እህቴ <N_Ssf1> እረጂጅም<ADJ_p> ልብስ <N> ትወዳልች<V_IMPF_Ssf3>።<ENDPUNC>
ውንድሜ<N_sm1> እረጂጅም<ADJ_p> ነው<AUX_Ssm3>።<ENDPUNC>
አለሙ <N_Ssm3>ሰነፍ <ADJ>ተማሪወች <N_p> ነው<AUX_Ssm3>።<ENDPUNC>
አለሙ <N_sm3>ሳበዝ<ADJ> ተማሪወች <N_p> ነው<AUX_Ssm3>።<ENDPUNC>
እህቴ <N_sf1> ቀይ <ADJ> ልብሶች<N_p> ትወዳለች<V_PERF_Ssf3>።<ENDPUNC>
እናቴ <N_sf1>ጥቁር <ADJ> ከብቶች<N_p> አትወድም<VN_IMPF_Ssm3>።<ENDPUNC>
አርጎንዋዴ <ADJ> ልብሶች<N_p>አለኝ<VREL_PERF_Ss1>።<ENDPUNC>
ታዋቂው <ADJ_p> የጥንታዊት <ADJP> ኢትዮጵያ <N> ጥናት <N> ሊቅ <N> አረፉ <V_PERF_p3> ።<ENDPUNC>
ኢትዮጵያዊው <ADJ_sm3> አበበች <N_Ssf3> ይመር <N_Ssm3> በማራቶን <NP>አሸነፈ <V_PERF_Ssm3> ።<ENDPUNC>
ኢትዮጵያዊው <ADJ_sm3> አትሌት <ADJ>  አበበች <N_Ssf3> በማራቶን <NP>አሸነፈ <V_PERF_Ssm3> ።<ENDPUNC>
የአዲስ አበባ"<NP> ቀደምት <ADJ> መምህራን <N_p> ማህበር <N> መሰረቱ <V_PERF_Sp3> ።<ENDPUNC>
የባህላዊ <ADJP> የእጽዋቶች <NP_p> መድሃኒት <N> ምርምር <N> ተቋም <N> ተቋቋመ <V_PERF_Ssm3> ።<ENDPUNC>
የደቡብ <ADJP> ዩኒቨርስቲ <N> ስምንት <NUMCR> አዲስ <ADJ> ፕሮግራሞችን <N_p> ጀመረ <V_PERF_Ssm3> ።<ENDPUNC>
የደቡብ <ADJP> ዩኒቨርስቲ <N> ስምንት <NUMCR> አዲስ <ADJ> ፕሮግራሞችን <N_p> ጀመረ <V_PERF_Ssm3> ።<ENDPUNC>
የኔዘርላንድስና <NPC> የሞሮኮ <NP>  መንግስታት <N_p> የደስታ <ADJP> መግለጫወች <N_p> አስተላለፉ <V_PERF_Sp3> ።<ENDPUNC>
ተዘዋዋሪ <ADJ> ችሎቶች <N_p> አራት <NUMCR> ተከሳሾችን <N_p> በነጻ <NP> አሰናበተ <V_PERF_Ssm3> ።<ENDPUNC>
በሶስት <NUMCR> ከተሞች <N_p> ችግረኛ <ADJ_p> ህጻጓን<N> ማቋቋም <VN> ተጀመረ <V_PERF_Ssm3> ።<ENDPUNC>
እነአቶ <ADJP> ስዬች <N_Ssf3> አብርሀ <N_Ssm3> የክስ <NP> መቃወሚያ <N> አቀረቡ <V_PERF_Sp3> ።<ENDPUNC>
 የኦሮሚያ <NP> ልማት <N> ማህበር <N>  አዳዲስ <ADJ_p> የልማት <NP> ስትራቴጂ <N> ነደፈ <V_PERF_Ssm3> ።<ENDPUNC>
 መምሪያው <N> ስለጅማና <NPC> አባጅፋር <N> የተጻፉ <VREL_PERF_Sp3> ታሪካዊ <ADJ_p> መጻህፍ<N> አሳባሰበ <V_PERF_Ssm3> ።<ENDPUNC>
 የደቾችን <NP_p> አቅም <N> የማሳደበት <NP> ልዩልዩ<ADJ_p> ትኩረት <N> እንዲሰጥ <VP_IMPF_Ssm3> ተጠቆመ <V_PERF_Ssm3> ።<ENDPUNC>
 በ24 <NUMP> አመታት <N_p> "ከ274 ሺ" <NUMP> ለሚበልጡ <ADJ_p> ተሽከርካሪ <N>  ሰሌዳዎች <N_p> ተሰጥተዋል <V_Ger_Sp3> ።<ENDPUNC>
ለተፈናቀሉ <ADJ_p> ተማሪ <N> የትምህርት <NP> መርጃ <N> መሳሪያዎች <N_p> ድጋፍ <N> ተደረገ <V_PERF_Ssm3> ።<ENDPUNC>
 ማእከሉ <N>  31_ሺ_130 <NUMCR> ኩንታል <N> የበቆሎ <NP> ምርጥ <ADJ> ዘሮች <N_p> እያሰራጨ <V_PERF_Ssm3> ነው <AUX_Ssm3> ።<ENDPUNC>
 ከምክርቤት <NP> አባላት <N_p> በርካታ <ADJ_p> ጥያቄ <N> ቀርበው <V_GER_Sp3> ሚኒስትሩ <N_sm3> መልስ <N> ሰጥተውበታል <V_PERF_Sp3_Osm3> ።<ENDPUNC>
 ባለስልጣኑ <N> መለስተኛ <ADJ> ላቦራቶሪ <N> በ2 <NUMP> ሚሊየን <NUMCR> ብር <N>  ሊገነባ <VP_IMPF_Ssm3> ነው <AUX_Ssm3> ።<ENDPUNC>
በአማራ <NP> ክልል <N> መሰረታዊም<ADJ> አደረጃጀቶች <N_p> የአሰራር <N> ለውጥ <N> እንደሚደረግ <VP_IMPF_Ssm3> ተጠቆመ <V_PERF_Ssm3> ።<ENDPUNC>
የእነ <PRON> አቶ <ADJ_sm3> ስዬች <N_Ssf3> የዋስትና <NP> መብት <N> ጥያቄ <N> ውድቅ <ADJ> ሆነ <V_PERF_Ssm3> ።<ENDPUNC>
"ከ29 ሺ" <NUMP> ለሚበልጡ <ADJ_p> ወገን <N>  መደበኛ <ADJ> ያልሆነ <VP_PERF_Ssm3> ትምህርት <N> እየተሰጠ <V_PERF_Ssm3> ነው <AUX_Ssm3> ።<ENDPUNC>
"ከ29 ሺ" <NUMP> ለሚበልጡ <ADJ_p> ወገን <N>  መደበኛ <ADJ> ያልሆነ <VP_PERF_Ssm3> ትምህርት <N> እየተሰጠ <V_PERF_Ssm3> ነው <AUX_Ssm3> ።<ENDPUNC>
"በምስራቅ ጎጃም" <NP> አዲሱ <ADJ> አደረጃጀቶች <N_p> እስከ <PREP> ቀበሌ <N> ድረስ <PREP> መዘርጋቱ <VN_sm3> ተጠቆመ <V_PERF_Ssm3> ።<ENDPUNC>
 ማእከሉ <N>  የተሻሻሉ <ADJP_P> የእርሻና <NPC> እንዱስትሪ <N> መሳሪያ <N> አከፋፈለ <V_PERF_Ssm3> ።<ENDPUNC>
```

# APPENDIX D

# SAMPLE ADVERB-VERB DISAGREEMENT TAGGED SENTENCE

በከፋ <NP> ዞን <N> በሚቀጥለው <ADV_IMPF> ሳምንት <N> ምርጫ <N> ተካሂዶል <V_PERF_Ssm3> ።<ENDPUNC>
የቢዝነስ <NP> ማኔጅመንት <N> ትምህርት <N> በሚቀጥለው <ADV_IMPF> አመት <N> ጀመሩ <V_PERF_Sp3> ።<ENDPUNC>
በተያዘው <ADJP_sm3> አመት <N> መጨረሻ <ADJ> ሙሉለሙሉ <ADV> ተጠናቀው <V_GER_Sp3> በሚቀጥለው<ADV_IMPF> አመት <N> ለአገልግሎት <NP> በቅተዋል <V_PERF_Sp3>።<ENDPUNC>
ኤርትራ <N> በቀጣይ <ADV_IMPF> የኢትዮ-ኤርትራ <ADJP> ወታደራዊ <ADJ> ቅጎጅት <N> ስብሰባ <N> እንደተካፈለች <VP_PERF_Ssf3> ተገለጸ <V_PERF_Ssm3> ።<ENDPUNC>
አሁን <ADV> ሩብ <ADJ> ያህሉ <N> መሰብሰቡንና <NC> በቀጣይም <ADV_IMPF> እንቅስቃሴው <N_sm3> እንደቀጠለ <VP_PERF_Ssm3> አስረድተዋል <V_GER_Sp3> ።<ENDPUNC>
በቀጣይ <ADV_IMPF> የተለያዩ <VREL_PERF_Sp3> ፕሮጀክቶች <N_p> አከናውነ <V_PERF_Ssm3> መዘጋጀቱን <VN_sm3> አስታውቀዋል <V_GER_Sp3> ።<ENDPUNC>
ተልእኮው <N> <PUNC> በቀጣይም <ADV_IMPF> ሃላፊነቱን <N_sm3> ለመወጣት <NP> እንደጣረ <VP_PERF_Ssm3> አስታውቀዋል <V_GER_Sp3> ።<ENDPUNC>
ኤርትራ <N> በቀጣይ <ADV_IMPF> የኢትዮ-ኤርትራ <ADJP> ወታደራዊ <ADJ> ቅጎጅት <N> ስብሰባ <N> እንደተካፈለ <VP_PERF_Ssm3> ተገለጸ <V_PERF_Ssm3> ።<ENDPUNC>
የመሳሪያ <N> ግዢው <N> ቀጣይነት <ADV_IMPF> እንዳለው <VP_PERF_Ssm3> አስታውቀዋል <V_GER_Sp3> ።<ENDPUNC>
ከአምናው <ADV_PERF> ውይይቱ <N> በየወረዳው <NP> ይካሄዳል <V_IMPF_Ssm3> ።<ENDPUNC>
በኢትዮጵያ <NP> የ106ኛው <NUMP> የአደዋ <NP> ድል <N> በአል <N> በቀጣይ <ADV_IMPF>እለት <N> በድምቀት <NP> ተከብሮ <V_GER_Ssm3> ውሏል <V_GER_Ssm3> ።<ENDPUNC>
በቀጣይ <ADV_IMPF>እለት <N> አንድ <NUMCR> የአሜሪካን <NP> ዶላር <N> 8 <NUMCR> ብር <N> ከ5513 <NUMP> ሳንቲም <N> ተሸጠ <V_PERF_Ssm3> ።<ENDPUNC>
በቀጣይ <ADV_IMPF> ለተጠቃሚው <ADJP> ሀብረተሰብ <N> በማቅረብ <NP> ላይ <PREP> ይገኛል <V_PERF_Ssm3> ።<ENDPUNC>
በባሌ <NP> መስህቦችን <N_p> የሳበኙ <VREL_PERF_Sp3> ቱሪስቶች <N_p> ቁጥር <N> በቀጣይ <ADV_IMPF> በ2ሺ <NUMP> ጨመረ <V_PERF_Ssm3> ።<ENDPUNC>
የዘንድሮ <NP> ተጋቢዎች <N_p> ቁጥር <N>በቀጣይ <ADV_IMPF>ጥር <N> ተጋቢዎች <N_p> በ29 <NUMP> ብልጫ <N> ማሳየቱን <VN_sm3> አመልክተዋል <V_GER_Sp3> ።<ENDPUNC>
ለቀጣይ <ADV_IMPF> ስራዎች <N_p> አመቺ <ADJ> ሁኔታ <N> እንደፈጠሩ <VP_PERF_Sp3> መጠቆማቸውን <VN_p3> ዋልታ <N> እንፎርሜሽን <N> ማእከል <N> ዘግቧል <V_GER_Ssm3> ።<ENDPUNC>
ቀጣይ <ADV_IMPF> የማቋቋሚያ <NP> ገንዘብ <N> ለማከፋፈል <VP_IMPF> ተጨማሪ <N> ዝግጅት <N> መጠናቀቂን <VP_PERF_Sp3> አስታውቀዋል <V_GER_Sp3> ።<ENDPUNC>
በቀጣይ <ADV_IMPF> አመታት <N_p> ዘርፉን <N_p> ለማሳደግ <NP> የተደረጉ <VREL_PERF_Sp3> ጥረቶች <N_p> ውስን <N> መሆናቸውን <VN_p3> ጠቁመዋል <V_GER_Sp3> ።<ENDPUNC>
የአዲስ አበባ"<NP> ቀደምት <ADV_PERF> መምህራን <N_p> ማህበር <N> ይመሰርታሉ <V_IMPF_Sp3> ።<ENDPUNC>
ቀደምሲል <ADV_PERF> ጣሊያንና <NC> ኤርትራ <N> ርስበርሳቸው <PRON> አምባሳደሮቻቸውን <N_p> ማስወጣታቸው <VN_p3> ያስታውሳል <V_IMPF_Ssm3> ።<ENDPUNC>
የጀርመን <NP> ቦር <N> በቀጣይ <ADV_IMPF> ጥር <N> መጀመሪያ <N> ወደ <PREP> አፍጋኒስታን <N> መጓዙ <VN_sm3> ይታወሳል <V_PERF_Ssm3> ።<ENDPUNC>
በቀጣይ <ADV_IMPF> አመት <N> ብቻ <ADV> ከ300 <NUMCR> በላይ <PREP> አደንዛዥ <ADJ> እጽ <N> አዟሪዎች <N_p> ተይዘዋል <V_GER_Sp3> ።<ENDPUNC>
የአገሪቱ <NP> የባንክ <NP> ግብይት <N>በቀጣይ <ADV_IMPF> ጥቅምት <N> 14 <NUMCR> መጀመሩ <VN_sm3> ይታወቃል <V_IMPF_Ssm3> ።<ENDPUNC>
በቀጣይ <ADV_IMPF> አርብ <N> አንድ <NUMCR> የአሜሪካን <NP> ዶላር <N> 8.5555 <NUMCR> መሸጡን <VN_sm3> ባንኩ <N> አስታውሷል <V_GER_Ssm3> ።<ENDPUNC>
በአዲስ_አበባ <NP> በቀጣይ <ADV_IMPF> አመት <N> ብቻ <ADV> በ8 <NUMP> ሺ <NUMCR> ሀጻናት <N> ላይ <PREP> ጥቃት <N> ተፈጽሟል <V_GER_Ssm3> ።<ENDPUNC>
በቀጣይ <ADV_IMPF> አመትም <N> የአንበሳ <NP> መጎታ <N> በወረዳው <NP> ስምንት <NUMCR> የቀንድ <NP> ከብቶች <N_p> መብላቱ <VN_sm3> ይታወሳል <V_PERF_Ssm3> ።<ENDPUNC>
በባሌ <NP> መስህቦችን <N_p> የሳበኙ <VREL_PERF_Sp3> ቱሪስቶች <N_p> ቁጥር <N> በቀጣይ <ADV_IMPF> በ2ሺ <NUMP> ጨመረ <V_PERF_Ssm3> ።<ENDPUNC>
የዘንድሮ <NP> ተጋቢዎች <N_p> ቁጥር <N> በቀጣይ <ADV_IMPF> ጥር <N> ተጋቢዎች <N_p> በ29 <NUMP> ብልጫ <N> ማሳየቱን <VN_sm3> አመልክተዋል <V_GER_Sp3> ።<ENDPUNC>
የሺወርቅ <N> ደስታ <N> ባለፈው <ADV_PERF> በሀኪም <NP> እንድትታይ <VP_IMPF_sf2> ተወሰነ <V_PERF_Ssm3> ።<ENDPUNC>
አልቃይዳ <N> በሶማሊያ <NP> ባለፈው <ADV_PERF> እንዳይደራጅ <VP_IMPF_Ssm3> አሜሪካ <N> ቅኝቷን <N_sf3> አጠናክራለች <V_GER_Ssf3> ።<ENDPUNC>
የኢንቨስትመንት <NP> አዋጅ <N> ባለፈው <ADV_PERF> ለማሻሻል <NP> የሚረዳ <VREL_IMPF_Ssm3> ዝርዝር <ADJ> ጥናት <N> ተዘጋጀ <V_PERF_Ssm3> ።<ENDPUNC>
በሞስኮ <NP> የኢትዮጵያውን <NP_p> ኮሚኒቲ <N> ባለፈው <ADV_PERF> ተዋቀረ <V_PERF_Ssm3> ።<ENDPUNC>
ከአምናው <ADV_PERF> ለዘንመት <NP> እንደማ የበቀ <VP_IMPF_Ssm3> ተናግሯል <V_GER_Sp3> ።<ENDPUNC>