SCHOOL OF GRADUATE STUDIES

YEMSA TO AMHARIC MACHINE TRANSLATION USINGDEEP LEARNING TECHNIQUES

MSc. THESIS

TEMESGEN HABTAMU ESHETU

DECEMBER, 2023

WOLKITE, ETHIOPIA

Wolkite University

School of Graduate studies

Yemsa to Amharic Machine Translation usingDeep Learning Techniques

A MSc Thesis Submitted to School of Graduate Studies, in Partial Fulfillment Requirement for the Degree of Master of Science in Computer Science and Engineering

Temesgen Habtamu Eshetu

Major Advisor: Mesfin Abebe (Ph.D.)

Co_Advisor: Adugna W/giorgis (M. Sc)

December, 2023

Wolkite, Ethiopia

APROVAL SHEET

School of Graduate Studies

Wolkite University

YEMSA TO AMHARIC MACHINE TRANSLATION USING DEEP LEARNING TECHNIQUES

Submitted by:

Temesgen Habtamu	______________	______________
Name of Student	Signature	Date

Approved by:

1. ___________________	______________________	________________
Major Advisors Name	Signature	Date
2. ___________________	______________________	________________
Co- Advisors Name	Signature	Date
3. _____________________	______________________	________________
Department Head	Signature	Date
4. _____________________	______________________	________________
Name of Chairman, DGC	Signature	Date
5. ___________________	______________________	________________
Name of Dean, SGS	Signature	Date

DEDICATION

This thesis is especially dedicated to my family who fulfilled my desire and supported my success.

DECLARATION

DECLARATION

By my signature below, I declare and affirm that this Thesis is my own work. I have followed all ethical principles of scholarship in the preparation, data collection, data analysis and completion of thesis. All scholarly matter that is included in the thesis has been given recognition through citation. I affirm that I have cited and reference all sources used in this document. Every serious effort has been made to avoid any plagiarism in the preparation of this thesis.

This thesis is submitted in partial fulfillment of the requirement for a degree from the School of Graduate Studies at Wolkite University. The thesis is deposited in the Wolkite University Library and is made available to borrowers under the rules of the library. I solemnly declare that this thesis has not been submitted to any other institution anywhere for the award of any academic degree, diploma or certificate.

Brief quotations from this Thesis may be used without special permission provided that accurate and complete acknowledgement of the source is made. Requests for permission for extended quotations from, or reproduction of, this thesis in whole or in part may be granted by the Head of the School or Department or the Dean of the School of Graduate Studies when in this or her judgment the proposed use of the material is in the interest of scholarship. In all other instances, however, permission must be obtained from the author of the thesis.

Name: Temesgen Habtamu Eshetu Signature: ____________

Date: 20/11/2023

Department: Computer science and Engineering

BIOGRAPHY

The author, Mr. Temesgen Habtamu, was born on June 16, 1987 E.C, in Semuawasho Kebele Yem Zone, Central Ethiopia Region, Ethiopia, from his father, Ato Habtamu Eshetu, and mother Woizero Senayit Wendimu. He attended his primary education at semuawasho primary school, his secondary education at Fofa secondary and preparatory school. Then he joined Kotebe Metropolitan University department of computer science and technology in 2007 E.C and obtained his BSc degree in a computer science in June, 2010 E.C. He was employed in the Yem Zone Trade and Market development served for two years. Then, in November, 2014, he joined the department of Software engineering,in college of computing and informatics at Wolkite University to pursue his Master of Science degree in computer science.

ACKNOWLEDGMENT

With sincere heartfelt gratitude I thank the Almighty **God** for allowing me to make my dream come true and giving me strength, determination, endurance and wisdom to accomplish this study and for being always with me in whole my journeys I passed.

Secondly, I would like to extend my deepest appreciation to my advisor, Dr. Mesfin Abebe, for his excellent and enduring support. He has given me so much inspiration for my future academic career. I taught more from him how to build up my academic brand, when to say no to distractors, what to give to others, and many, many more great things. I also wish to express my sincere gratitude to all whom through their supports contributed to the successful completion of this work.

Thirdly, I would like to thank my co-advisor Adugna W/giyorgis for his deep guidance and constructive comments in all of my works.

My very exceptional and special thanks go to my family and friend for their endless and countless supports during my thesis write through moral, material and financial. There is no way I could have made it without their support. They are always with me and lettingme to do what I wished to do!

Next, my special thanks go to Yem Zone Administration, Yem Zone finance office, Yem Zone Education Department, Yem Zone culture and Tourism Department language directorate and Yem Zone Trade and market Department for their multidimensional support that makes me be at right way. Moreover, I like to highly acknowledge the affection, love and continued encouragement of all college of computing and informatics instructors for their dedication and tireless efforts and kind support.

ABBREVIATIONS AND ACRONYMS

ADAM	Adaptive Moment Estimation
ANN	Artificial Neural Networks
BERT	Bidirectional Encoder Representation
BI-LSTM	Bi-directional Long Short-Term Memory
BLEU	Bi-Lingual Evaluation Understudy
CBMT	Context Based Machine Translation
CBOW	Continuous Bag of Words
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Networks
EBMT	Example Based Machine Translation
FDRE	Federal Democratic Republic of Ethiopia
GD	Gradient Descent
GLOVE	Global Vectors for Word Representation
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HNMT	Hybrid Neural Machine Translation
LSTM	Long Short-Term Memory
MASS	Masked Sequence to Sequence a Pre-Training Model
METEOR	Metric for Evaluation of Translation with Explicit Ordering
ML	Machine Learning
MT	Machine Translation
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing

NMT	Neural Machine Translation
NN	Neural Network
PBSMT	Phrase Based Statistical Machine Translation
RBMT	Rule Based Machine Translation
RNN	Recurrent Neural Network
RNNM	Recurrent Neural Network Machine Translation
Seq2Seq	Sequence to Sequence
SGD	Stochastic Gradient Descent
SMT	Statistical Machine Translation
SNNPRS	Southern Nations Nationalities and Peoples Regional State
SOTA	State of the Art
SOV	Subject Object Verb
Word2vec	Word 2(to) Vector
XML	Cross-Lingual Language Model

TABLE OF CONTENT

APROVAL SHEET ii

DEDICATION iii

DECLARATION iv

BIOGRAPHY v

ACKNOWLEDGMENT vi

ABBREVIATIONS AND ACRONYMS vii

LIST OF TABLES xiv

LIST OF FIGURES xv

LIST OF ALGORITHMS xvi

LIST OF TABLES IN THE APPENDIX xvii

LIST OF FIGURES IN THE APPENDIX xviii

ABSTRACT xix

1. CHAPTER ONE: INTRODUCTION 1

1.1. Background of the Study 1

1.2. Motivation of the Study 3

1.3. Statement of the Problem 4

1.4. Objective of the Study 7

1.4.1. General Objective 7

1.4.2. Specific Objectives 7

1.5. Research Questions 7

1.6. Significance of the Study 8

1.7. Scope and Limitation 8

1.8. Thesis Organization 9

2. CHAPTER TWO: LITRATURE REVIEW 10

2.1. Overview 10

2.2. Overview of Machine Translation 10

2.3. Approaches to Machine Translation 11

2.4. Evaluation of Machine Translation 12

2.5. Overview of Yemsa Language 12

2.7. Overview of Amharic Language 16

2.7.1. Amharic Punctuations 17

2.7.2. Amharic Morphology 17

2.8. Related Work 22

2.8.1. Machine Translation for Ethiopian Languages Pairs 22

2.8.2. Machine Translation for Foreign Language Pairs 23

2.8.3. Machine Translation for Foreign and Local Languages Pairs 25

2.8.4. Summary of Related Work 29

3. CHAPTER THREE: MATERIALS AND METHODS 30

3.1. Introduction 30

3.2. Proposed Approach 30

3.3. Literature Review 31

3.4. Parallel Corpus 31

3.5. Material and Tools 31

3.5.1. Hardware Tools 31

3.5.2. Software Tools 32

3.6. Data Pre-processing 34

3.6.1. Data Cleaning 34

3.6.2. Data Tokenization 36

3.6.3. Data Normalization 38

3.7. One Hot Vector Representation 39

3.8. Padding 40

3.9. Word Embedding 40

3.10. Encoder 42

3.11. Decoder 42

3.12. Attention Mechanism 42

3.13. Optimization Algorithms 43

3.14. Performance Measurement Methods 43

3.14.1. Accuracy 43

3.14.2. Loss Function 43

3.14.3. Cross-Entropy 43

3.15. Fine Tuning 44

3.15.1. Dropout 44

3.15. 2. Early Stopping 45

3.16. Evaluation Metrics 45

4. CHAPTER FOUR: RESEARCH DESIGN 46

4.1. Introduction 46

4.2. Parallel corpus Preparation 48

4.3. Pre-processing 48

4.4. Data Splitting 48

4.5. Vectorization 49

4.6. Model Selection 49

4.6.1. Long Short-term Memory Model 49

4.6.2. Bi-directional Long Short-term Memory Model 51

4.6.3. LSTM +Attention 52

4.6.4. Gated Recurrent Unit Model 53

4.6.5. Transformer Model 55

5. CHAPTER FIVE: EXPERIMENTATION 57

5.1. Introduction 57

5.2. Data Collection and Preparation 57

5.3. System Environment........57
5.4. Data Cleaning........58
5.5. Tokenization........59
5.6. Parameter Selection........62
6. CHAPTER SIX: RESULT AND DISCUSSION........64
6.1. Introduction........64
6.2. Experimental Result........64
6.3. Changing Training and Testing Dataset Ratio........68
6.4. Linguist Evaluation........72
6.5. Discussion on the Result of the Study........74
6.6. Summary........75
6.7. Answering Research Questions........76
7. CHAPTER SEVEN: CONCLUSION AND RECOMMENDATION........79
7.1. Overview........79
7.2. Conclusion........79
7.3. Research Output........79
7.4. Future Work........81
REFERENCES........82
8. APPENDIX........87
8.1. Appendix A:........87
8.2. Appendix B........89
8.3. Appendix C:........90
8.4. Appendix D:........91
8.5. Appendix E........91
8.6. Appendix F:........92
8.7. Appendix G........92

LIST OF TABLES

Table 2.1: Nominal derived from verb and nouns..13

Table 2.2: Nouns derived by compounding 14

Table 2.3: Noun derived from verbs 15

Table 2.4: Amharic Noun Plural Formation...18

Table 2.5: Amharic Nouns with Gender Marks 19

Table 2.6: Amharic Adjectives Inflection 20

Table 2.7: Summery of Related work..27

Table 3.1: Tools and material...33

Table 3.2: dimensional embedding vector...41

Table 4.1: Data splitting in percentages and purpose...48

Table 5.1: Source and size of the data collected for experiment..................................57

Table 5.2: Hyper-parameters and values of the proposed studies models 63

Table 6.1: Result of an experiment for training and testing dataset ratio......................68

Table 6.2: Accuracy, time and Loss when using latent and embedding dimension 69

Table 6.3: Experimental result of the proposed model...71

Table 6.4: Sample Yemsa-Amharic translation result..73

Table 6.5: Sample of Amharic_Yemsa machine translation by proposed model 74

LIST OF FIGURES

Figure 2.1: Approach of MT 11

Figure 3.1: Tokenization and integer representation..37

Figure 3.2: One hot vector representation 39

Figure 3.3: Padding 40

Figure 4.1: Architectural design of Yemsa to Amharic 47

Figure 4.2: LSTM encoder decoder architecture..51

Figure 4.3: Bi-directional Long Short-term Memory Model Architecture 52

Figure 4.4: LSTM with Attention Architecture..53

Figure 4.5: Workflow GRU...54

Figure 4.6: The work flow Transformer model...56

Figure 5.1: sample code to show cleaned data 58

Figure 5.2: Sample code for tokenization...59

Figure 5.3: Shows the total dataset read for training..60

Figure 5.4: Creating Embedding Layer and One-Hot Encoder....................................61

Figure 5.5: Building encoder decoder and compile model 62

Figure 6.1: Accuracy and Loss..65

Figure 6.2: Snapshot of LSTM with Attention model Accuracy and Loss...................66

Figure 6.3: Snapshot of BI_LSTM model Accuracy and Loss 66

Figure 6.4: Snapshot of GRU model Accuracy and Loss 67

Figure 6.5: Training Accuracy and Loss for Transformer 68

Figure 6.6: Learning Rate and Loss Level for Training...70

Figure 6.7: BLEU score Experimental result for conducted model 72

LIST OF ALGORITHMS

Algorithms 3.1: Algorithms for cleaning a dataset 35

Algorithm 3.2: Algorithms for tokenization a dataset...36

Algorithm 3.3: Normalizing labialized characters 38

LIST OF TABLES IN THE APPENDIX

1. Sample of Parallel Corpus 88
2. Yemsa Language Orthographic… 90
3. Amharic Language Orthographic… 91

LIST OF FIGURES IN THE APPENDIX

1. Appendix-Figure: I Snapshot of python Code for Import all required library. 91

2. Appendix-Figure: II Snapshot of python code for normalization… 91

3. Appendix-Figure: III Snapshot code for BLEU Score 92

4. Appendix-Figure: IV Snapshot for sample output 93

ABSTRACT

In today's globalized world, the barriers of distance and language have been greatly diminished, transforming our world into a closely interconnected global community. As a consequence, human languages have taken on an international character, enabling effective communication across borders. Traditionally, human translation is costly and inconvenient; several kinds of research are currently conducted to resolve this problem with machine translation techniques. |So, it is automatic, which means it translates one language to another using a computer software system. In this study, Yemsa to Amharic machine translation and vice versa are used by deep learning techniques. Due to increased speaker numbers, to address the issue of endangered Yemsa language and enhance the language's content on the World Wide Web. A number of indigenous knowledge medicines called Samo Heta *and other traditional and religious names are found in the Yemsa language. We utilized the current STOA method of deep learning. The work was executed using a seq-to-seq encoder-decoder architecture. The proposed study was conducted experiments on LSTM, Bi-LSTM, LSTM with attention, GRU and transformer models. We collected a dataset of about 6,000 parallel sentences with 11690 and 12491 vocabularies. In order to translate text into sentence sequence, we applied the preprocessing technique and used Morfessor tools. The proposed studies utilize the 80/20 splitting technique for dividing the dataset into training and testing sets. The next step is training and testing models on a corresponding with training and testing dataset. The experiment was conducted on LSTM, Bi-LSTM, LSTM+ attention, GRU and Transformer models. Among those models, the transformer model outperforms other models by 99.4% accuracy, 0.0113 loss. And BLEU scores of 9.7 and 9.8 from Yemsa to Amharic and Amharic to Yemsa respectively. The primary limitation of the investigation is the insufficient availability of a substantial dataset to conduct comprehensive experimentation. As a result, there is a necessity to generate parallel corpora in order to conduct comparable research. Finally, the findings of the study show that utilizing deep learning techniques, particularly the transformer model, can significantly improve Yemsa to Amharic machine translation accuracy and BLEU scores.*

Keywords*:* Machine Translation, Yemsa language, Deep learning, Morffessor, TransformerLSTM, LSTM+attention, BLEU Score**.**

1. CHAPTER ONE: INTRODUCTION

1.1. Background of the Study

Natural language is one of the fundamental aspects of human behavior and a basic component of daily activities [1]. It is a means of exchanging thoughts, feelings, and information through spoken, written, and signed communication. Natural language processing is a subfield of artificial intelligence and linguistics that makes computers understands human language. MT is among the applications for NLP that used by computer programs and software to convert one natural language into another as an instance of speech or text. The current globalized environment may nonetheless make it challenging to obtain information due to language restrictions. In some circumstances, it is impossible to satisfy the demand for translation by relying entirely on human translators; as a result, tools like MT are growing in popularity because they can address this problem. NLP enables this by utilizing the structures and patterns of human communication. One area of NLP is concerned with creating proof systems, such as machine translation.

MT is a branch of NLP that studies how to use computer programs to automatically translate text or speech between two languages without the use of intermediaries, which comes under the broad area of AI[2].The development of MT is more often used in the deep learning approach because it requires huge computational linguistic resources and high result accuracy compared to the other earlier machine translation approaches and relies to a great extent on parallel datasets of both input and target languages. it is possible to quickly translate huge texts, including paragraphs and documents. NMT is a new sort of method that has recently been developed in response to the development of DL techniques. Fundamentally MT uses artificial intelligence and complex algorithms to examine and comprehend the content, structure and grammar of an essay in its original languages.

Deep learning can also be taken as NN which has multi-layer architectures and very huge parameters on which it works [3]. As the name suggests, this is a new way of focusing on how our brain and the human nervous system worked [4]. The latest advancement in computational science and the appearance of massive language resource data have increased the demand for automatic semantic analysis utilizing data-driven techniques. It is easier to interpret data when it is understood in its context, which facilitates text analysis and mining. Considering a sufficiently large and relevant input dataset, DL is a set of techniques that, in theory, can determine the best solution to any problem. Currently, the survey conducted on languages indicates that worldwide, a remarkable number of 7,000 languages are spoken across the globe [5].

In this era where most of our daily chores are performed by computer software systems, why not machines work as middleman translators for different people. Given a large enough and pertinent input dataset deep learning is a set of techniques that in theory can determine the best solution to any problem. It works the inspiration of human brain. The task can range from simple classification to complex reasoning. Every language possesses its unique encoder and decoder, the model itself is considered a member within this linguistic family. Better comprehension of the brain system and communication is aided by using this, we can obtain to know how the normal human brain thinks and to map algorithm for and solve problems through a machine just as it has been solved by a human brain.

According to the historical content of the language and the history of the nation and the ancient great Enrayiya state and people, the language of Yemsa, namely Farsi, Latin, Hebrew, Arabic, geez, and Greek, was one of the ancient languages. The Yemsa language has lexical similarities with other omotic languages such as Dawuro, Konta, Wolaita, Kaficho, Sheko, bench and Cushitic language such as Afaan Oromo language. The verbal system of Yemsa is highly inflectional, compounding, reduplication, tone changes with prefixes and suffixes indicating categories such as person, number, gender and tense. Although the Yemsa dialects have been influenced by Cushitic the phonology of Yemsa includes its phonemic inventories phonological processes and syllable patterns [6] The Yemsa language uses the Latin writing system, which utilized a total of 34 letters made up of 22 single 7 paired, and 5 vowel consonants [7]. However, it is challenging to find Information studies done on the language. Amharic uses a script which originated fromthe Ge'ez alphabet [8]. Amharic uses the Ethiopic script and has arich morphology. It is a Semitic language family and lingua franca in our country.

DL is a subset of AI methods that analyzes and organizes data to find patterns, gain new perspectives, and improve predictions. DL is especially helpful jobs like machine translation, which involves translating input text from one language into another. This process requires sophisticated algorithms to determine the contextual meaning of words and phrases before translating them accurately.

At its core, MT using DL, translation is enabled by huge amounts of structured data the more training data available, the translations will be processed more accurately. DL-based machine translation typically achieves a much higher accuracy rate than traditional approaches because it's able to identify patterns at a deeper level that can capture nuances better; this makes it far better suited for understanding context over basic dictionary look-ups which are used in older methods of translation such as those employed by Google translate. In addition to improving accuracy when compared to traditional methods of translation, DL also makes machine translation scalable in a way that most traditional approaches cannot match due to the sheer size of modern-day data sets. With deep learning-powered translation systems, businesses no longer need worrying about manually entering each phrase or word into software applications; instead, they can now automate this task using those same NLP models trained on massive datasets previously mentioned.

1.2. Motivation of the Study

Yemsa is taught in the Yem Zone at the Central Ethiopia Region for both native speakers and non-native speakers, such as students from Oromo regional state and Hadiya zone linguistic backgrounds. The lessons started in first grade in 2019 and went up to sixth grade [9]. In recent years, NLP has emerged as a field of study interest for AI and machine learning. Therefore, there are two approaches to translating between two languages. These are experienced human-based translation and machine translation. This focus was enabled by the availability of datasets and computational capabilities required by AI and different ML algorithms. Although the start is great, it ishoped that the lack of studies on the language is present difficulties for teachers and students. For beginners, lexical items are one of the most important because they serve as the sole foundation for language learning. The only dialect that is considered slightlydifferent is that of Toba [10].

The number of speakers and the importance of the Yemsa language are growing all the time; the language is currently used in education services as well as other media agencies and radio stations: Wolkite FM 89.2 and Saja Min Media as a communication medium. In Ethiopia Amharic is the FDRE official working language. Several regional states like Amhara region, Addis Ababa city Administration and other southern region used Amharic as their official language [11]. In Ethiopia, it is the primary and most widely spoken language, after Arabic the second most widely used Semitic language worldwide and one of the five biggest languages onthe African continent [12].

Today it is the largest language in Ethiopia and one of the five largest languages on the African continent. Following the Constitution drafted in 1995, Ethiopia is categorized into thirteen independent regions and two chartered cities, each with its own regional working language [13]. Due to these reasons, creating an intelligent machine translation model for these languages will help millions of people. Translation from Yemsa to Amharic enables individuals to document their scientific discoveries, cultural knowledge and other insights in their native language and disseminate them on a global scale. In our country have grouping of several nationalities with various languages and customs. Additionally, the district's official working language is Amharic. Thus, several kinds of official documents newspapers job advertisements are written and produced in Amharic both at the federal and regional levels [13]. That is why this study is initiated to explore the possibility of developing DL models that can produce fluent and natural language translation between Amharic and Yemsa languages that wewill encounter when doing our experimental investigation.

1.3. Statement of the Problem

MT is a growing technology, much like any other technology. The growth in MT technology inspired different researchers to explore applications of MT for different languages and understanding its limitations is also key to successful implementation [1]. As our world becomes increasingly connected, language translation provides a critical cultural, political and economic bridge between people from different countries and ethnic groups. It is impossible to learn and understand every language spoken around the globe.

According to [15] there are currently around 7000 languages spoken worldwide, yet only a handful high resource languages have been the focus of most translation research tasks. It helps people to transfer their knowledge, culture, tradition,history and religious and philosophical writings from one natural language to another language using MT. It is commendable to meet the demand for translation, despite the inherent challenges in this active research field. In today's globalized world, language barriers can hinder knowledge acquisition. In several cases, using only human translators would not be able to meet the demand for translation. It takes much longer to translate something by hand, especially when searching up specific phrases in a dictionary.

Due to globalization, tourism, commerce, governance, and education, the increasing demand for translation has made automation indispensable. As a result, the machine translation system is becoming a solution to this technology use is becoming more and more important in people's daily lives. With the growing need to use technology, so did the require for translation in a multi nation and nationalities country, we need to have some automated languages translation systems. To help in knowing one nation's culture,thoughts, beliefs, democracy practice, and cultural heritages that facilitate each other's relationships, developing this kind of model helps in creating the idea of nationalism and making citizens stand up for their country's sovereignty. Thus, to bridge the gap, there is a need to investigate a MT that translates materials and documents into multiple languages, making every effort to facilitate efficient information and knowledge sharingwith the general population.

Lack of information with misunderstanding of language to keep people informed and lead to unnecessary costs, such as time-consuming research. The challenges are that the translation of low-resource and morphologically rich languages such as Afaan Oromo get limited attention, due to the unavailability of standardized parallel corpus which has a great effect on alignment of source and target languages [16]. The human translation method is boring and time-consuming. It takes a very long time, editing and evaluation is too costly (in terms of time, money and raw materials like paper and ink), and translation errors cannot be handled easily.

It can be very difficult to translate, evaluate and deliver a translation in a short period. The primary driving force behind this investigation was Yemsa inability to obtain such technology or systems. The structure of Yemsa is characterized by a subject object verb (SOV) word order and a pervasive head-final syntax in general. Dependent clauses precede main clauses. In morphology, suffixes are widely used. These traits are common among other omotic and among Ethiopian languages in general. Clause chaining is frequent and can result in rather long chains. Gender is a prominent category in Yemsa language, as it is marked on nouns, adjectives, pronouns, adverbs and verbs. It can be marked in a variety of ways, be it by morphology or tone. While the category refers to natural gender in nouns and adjectives with feminine as default gender is partly grammatical in verbal agreement: Where person is not marked in a given verb form, the different persons pattern either with the masculine or feminine 3rd person forms[17]. Yemsa language can translate with Amharic language.Besides, there are different biblical documents available in Amharic language. So, thatalso needs to be translated to Yemsa language.

But currently people use human translation, which is hard to accomplish and takes much time compared to machines. Currently, primary schools in the Yem Zone offer Yemsa as a subject from kindergarten up to grade 8, allowing students to learn and study the Yemsa language throughout their primary education. This also necessitates efficient translation of Amharic educational documents to Yemsa, promoting accessibility and inclusivity. And there are also Yemsa documents that Amharic speakers need to share for Amharic speakers, like culture, indigenous knowledge of the society, etc. The indigenous knowledge medicine which is called Samo Heta contains many plant diversity names and apart from its economic importance, the nation's traditional medicine collection is a tuba culture that can be a tourist attraction, so the state government will pay attention to it. So, this necessitates studying Yemsa-Amharic machine translation and vice versa.

Different MT research was done on local languages corresponding to Amharic such as Amharic-Wolayitta [18], Amharic-Ge’ez [19], Awgni to Amharic [20], Geez to English[21], English to Awngi[22] and English to Wolaita[23], However as researcher knowledge, there is no research conducted on Yemsa to Amharic and its vice versa machine translation. Currently, the progress in technology is shifting from traditional MT approaches to Deep learning approach.

based NMT approaches. NMT has recently performed well in standard benchmarks, yet it faces challenges when sentences contain longer, complex words. Therefore, the aim of this study is to design Yemsa to Amharic and Amharic to Yemsa machine translation. There is Yemsa to Amharic in Globse website but no any data is entered in that system and no prior research conducted on development of Yemsa to Amharic machine translation has been done on the creation of a machine translation system for Yemsa to Amharic and vice versa.

1.4. Research Questions

This study addresses the problem by answering the following question:

RQ1. How to develop a parallel corpus for Yemsa to Amharic Machine translation?

RQ2. Which vectorization techniques are used in Yemsa to Amharic MT?

RQ3. Which deep learning model is better for Yemsa to Amharic MT?

1.5. Objective of the Study

1.5.1. General Objective

The main objective of this study is to design Yemsa to Amharic machine translation and vice versa using deep learning techniques.

1.5.2. Specific Objectives

The specific goals that the following should be realized to achieve the main objective:

- To collect and prepare a parallel corpus for Yemsa and Amharic languages.
- To study syntactic relationships between Yemsa and Amharic languages.
- To apply preprocessing such as cleaning, normalization and tokenization.
- To execute one hot encoding representation and word embedding technique.
- To design a Yemsa and Amharic encoder-decoder MT model.
- To conduct experiments with different DL models.
- To compare accuracy of each model for the Yemsa and Amharic MT models.

1.6. Significance of the Study

The role of this research can be used to design and develop machine translation model from Yemsa to Amharic, improving efficiency as compared to manual translation. Being able to translate and understand Yemsa benefits in getting resources that are written in Amharic, and vice versa, it is possible to address information and solve language barriers between individuals to read and understand several. It contributes for future researcher's and development regarding Yemsa to Amharic language pair used as an additional component in the area of natural language processing specifically in Information retrieval, text processing and speech processing. To encourage the locality to share information and to learn the language in order to improve the local languages, and to encourage others who do not have the opportunity or interest to know the language well to read and write whatever they like. It enables to access information and interaction easily and fills the communication gap between peoples using the two languages. It is essential to work on MT in order to help them translate data from the source to their target language instead of to manually translating their daily news between languages.

1.7. Scope and Limitation

To propose, this undertaking is an experiment for Yemsa to Amharic sentence. It does not include, speech to speech translation, text to speech to text translation, speech translation and document-level translation. It focuses on sentence-to-sentence translation from a source to the target language.

The limitation of this study is the scarcity of existing digital data in Yemsa and the absence of a publicly available corpus in both Yemsa and Amharic, the studies addressing Yemsa to Amharic machine translation had to create their own parallel corpus. They accomplished this by using various sources such as educational books, religious texts like the Bible, and other relevant documents. However, it's important to note that the limited availability of Yemsa texts posed a significant challenge in constructing a comprehensive parallel corpus.

1.8. Thesis Organization

There are seven chapters, including the current research thesis. Chapter Two presents a literature review on machine translation for non-Ethiopian and Ethiopian languages, a summary of the literature, an overview of Yemsa and Amharic languages, and the morphology of language similarities and differences. In Chapter3, research methodology was briefly discussed data preparation, the proposed architecture, the tools, software tools, hardware environments, and evaluation methods. Chapter four describes the research design for Yemsa to Amharic machine translation. Chapter five discusses experimentation, which briefly discusses the experimental setups, the number of experiments we conducted, and the parameters we considered during the experiments we conducted in the research. Chapter 6: Present Results and Discussions: presents the outcomes of every experiment and discusses the results. In the last chapter, we give our conclusion, recommendations and future work.

2. CHAPTER TWO: LITRATURE REVIEW

2.1. Overview

In this chapter, a concise overview of machine translation, major approaches to MT and evaluation methods for MT are discussed. Additionally, a detailed overview of the Yemsa and Amharic languages is provided. Additionally, a review of related work on the methodology and implementation of machine translation in both foreign and Ethiopian languages is provided.

2.2. Overview of Machine Translation

Machine translation means is a contemporary translation technique that uses computer support and is a branch of computational linguistics. This academic and professional area associated with both translation and Computer Science explores the functions of computer software that can translate texts in foreign languages to be readable and comprehensible [1]. MT is already widely used in many areas due to its high efficiency, high translation quality and low cost [24]. The most popular type of MT application is text translation such as web page, translation, scientific literature translation, E-commerce translation etc. Below are some typical applications of text translation. In order for non-native speakers to comprehend foreign-language literature that they have to assess, MT performs basic substitutions with translated key words. Since the beginning of research study in the field of NLP, thework on MT has become the main concern to replace the need for experts to translate between different languages.

MT is a domain in the area within computerized linguistics that looks into Software is utilized to convert text or speech from one human language to another, maintaining the natural language context [25]. The proliferation of computers and the Internet has brought the world closer together. Using MT, communication of persons shares their knowledge, culture, tradition, history and religious and philosophical writings in one language to another. The accessibility of documents written in one human language to another language is greatly improved by machine translation [26].

2.3. Approaches to Machine Translation

Machine translation is among the research topics in computational linguistics. Several methodologies have been available to automate the translation system. However, the goal has been restored the original information's significance in the translated verse [27]. In different studies MT is simply substitution of word in one languageto word in other but not assure good translation result.in MT system there is no human involvement but the machine is performing the process of conversation. In general, the process of translation has two levels known as Metaphrase and Paraphrase. Metaphrase word-to-word translation. Paraphrase relates to dynamic equivalence, that means the translated text would contain the gist of the original text but may not necessarily contain the word-to-word translation [28]. The methods in MT are mainly divided into rule-based (involving direct, transfer, and interlingua) translations, statistical translations, example-based translations, neural translations, and hybrid approaches. Unlike the traditional phrase-based translation system, which consists of many small sub-components that are tuned separately, NMT attempts to build and train a large neural network that reads a sentence and generates outputs. Furthermore, in the field of MT, DNN is a newly emerging approach that has proven to achieve excellent performance [29].

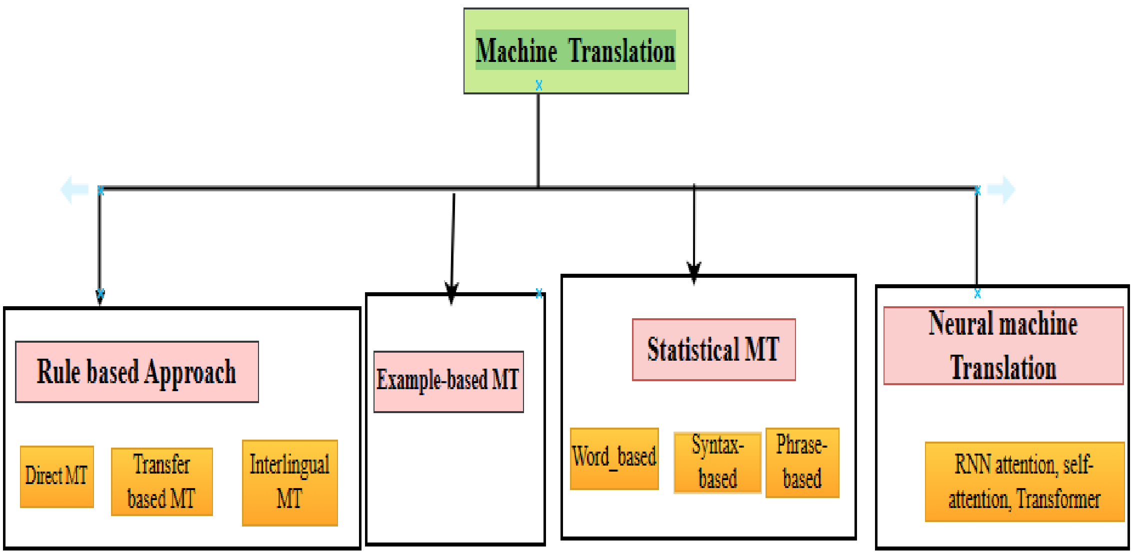


Figure 2.1: Approach of MT

2.4. Evaluation of Machine Translation

In machine translation systems, there are two types: human and automatic MT metrics [30]. Human evaluation is done manually by a linguist, but it is time- consuming and expensive. In MT automatic evaluation uses automatic metrics such as BLEU, NIST, METEOR, etc., so automated metering has emerged to solve the requirement for an objective, quick, affordable and quick assessment of machinetranslation system output rather than the human evolution way. Considered by the language, content type and use case MT technique and ones MT system evaluated using the most relevant automatic metric. However, many machine translation efforts today are assessed using the BLEU metric; thus, in order to compare with them, we also utilize it for our study. The most appropriate uses automated evaluation metrics depend on the language, use case, content types and MT approach. But nowadays, many MT works are evaluated by the BLUE metric that we described in the related work and to be comparable with them. Additionally, the studies conducted linguistic rater side-by-side evaluations, comparing and evaluating the number of translations predicted by the proposed model.

2.5. Overview of Yemsa Language

Yemsa is the language of the Yem people of the former Kingdom of Yamma [31]. It is a member of the western branch of the Omotic group of languages family, sharing most lexical similarities with Dawuro, Konta, Wolaita, Kaficho, Shekicho, Sheko, Bench, and Cushitic languages such as Afaan Oromo. The historical content of the language according to the history of the nation and the ancient great Enrayiya state and people, the language of Yemsa, namely Farsi, Latin, Hebrew, Arabic, Geez and Greek, was one of the ancient languages [2]. It is widely spoken in the district and has been mainly used as a spoken language at home, church, school, the market, etc. Although the language covers most Yem languages such as Amharic, Afaan Oromo and other are also spoken in some villages and boundaries of the regions of distinct. Since 2012, Yemsa has been used in education in KG and primary schools from grades 1–6 and broadcast on media from Wolkite FM 89.2 and Saja Min Media. Yemsa is characterized by an SOV word order and a pervasive head-final syntax in general. It can be marked in a variety of ways be it by morphology or tone. While the category refers to natural gender in nouns and adjectives with feminine as default gender is partly grammatical in verbal agreement.

Where a person is not marked in a given verb form, the different person's pattern either with the masculine or feminine 3rd person forms [33]. Dependentclauses precede main clauses. In morphology, suffixes are widely used. The Yemsa uses the Latin writing system, which utilizes a total of 34 letters made up of 22 single, 7 paired and 5 vowel consonants [33].

2.6. Morphology of Yemsa Language

In many languages, words play important grammatical roles, built out of smaller elements called phonemes. The way of linguistics study in which these words are built out of the stock of segmental morphemes is known as morphology [34]. Morphology is divided into two: the first is Inflectional morphology. The second is derivational morphology. The former is concerned with studies of morphemes that showgrammatical roles while the latter is concerned with how new words are from existing words [35].

Inflectional morphology of Yemsa including number, person, definiteness and case systems and derivation of nouns through affixation and compounding was discussed in detail below. Word formation in Yemsa is formed with affixation, reduplication and tone changes. Nominal can be derived from adjectives with the morpheme {-ba} as in: [Keesh-u] Adjectives, [keshu-ba] nouns**.** Nominal is identified including agentive experience, patient, result, manner, gerundive that are derived from verbs roots as shown in the example below.

Table 2.1: Nominal derived from verb and nouns

Verb	***Gloss***	***Noun***	***Gloss***	***Nominal Type***
Dig	To fear	Duginna	Coward	Experince
wost	Work(v)	Wostinna	Worker	agentive
maj	Dress(v)	Maj-a	Dress(n)	Patient
Kabu	Stand(v)	Kabuwa	Stand up	To develop
tej	To oath	Teja-a	oath	Result

The Yemsa language has derivational verbs from another verb such as stems namely: intensive causative and passive are identified. The causative verb is derived by suffix {-

i(S)} while passive from {-t} as passive former like wor 'kill' versus wor-t be killed. Adjectives are also derived from verbs from {-a} as in diff – a 'tall', tum-a 'full' and kn-a 'dirty'. However, nouns are derived means of compounding of verbs. For instance, offers the following compound nouns:

Table 2.2: Nouns derived by compounding

Compound noun	***Gloss***	***Constituents***
Umda-da	mountain	Noun-noun
Aba-kusu	thumb	Noun-noun
Harfak	Raise up	Verb-verb
Bustruki	Sleep fastly	Verb-verb

According to the studies, she is treated as a derivation of adverbs. Most frequency adverb in Yemsa is derived by reduplication as shown below**:** Example: - Golo golo-plainly, Wona wona- daily. The internal structure of the Yemsa includes noun phrases, including nominal heads, specifiers, complements, and adjuncts. Besides identified definite marker -s demonstratives den or jjet 'that' han 'these subject pronouns and possessive adjectives and indefinite pronouns such as owo many akama much; atto some, etc., anddefiniteness marking quantifiers such as numerals, measure phrases, and classifier phrases. A few adjectival phrases, such as foro mija ("white cow") and garo fizo (small got) are analyzed as adjuncts. Some studies described locative, temporal, and clausal genitives. In many languages, words play important grammatical roles. They are built out of smaller elements called phonemes. The branch of linguistics that studies the waysin which the words are built out of the stock of segmental morphemes is known as morphology. Morphology is divided into two categories: Inflectional morphology and derivational morphology.

***Inflection of noun*:** Language adds affixes to show grammatical relationships. The affixes could be added as prefix, suffix or shown through internal modification. Nouns and noun phrases are often marked for number. The most common distinction is between singular and plural [36]. In Yemsa plural nouns are marked with {kito} and occasionally with {- aki?o}. The environmental in which the variant {aki?o} is used not clear. The plural marks surface only when the noun to which number is added is definite. The following sentences demonstrate the absence of plural marker because the noun in each sentence is not defined. Bar Mija waag-e /She Cows buy-per/'She bouth t Cows',Bar Fizo waag-e/She Goat buy-per/'She bought Goats'. Example above which describe nouns "Cows" and "Goats" are plural in the English gloss but the nouns mija and fizo in Yemsa are not marked for plural. The of the language does not allow the plural marker to surface unless the noun is definite. The usual plural marker of Yemsa is {-kito}. In Yemsa a few, nouns are inflected for gender, Example: Faz →'Horse' Faza-a → 'mare', kan→ 'Dog'.

***Inflection of Verb*:** The inflection of verbs: person, gender, number, tense, mood and negation. In Yemsa, verbs are inflected for person as an agreement phenomenon. Consider the following examples Ta mija waag -e –n, I cow buy -per 1s/ 'I bought a cow',Inno mija waag -e –ni → 'We cow buy' -per 2pl

Masculine is expressed with -e whereas the femine with -a consider the following examples. The inflection of gender, however is not always over as it overlaps with aspect. And verbs of Yemsa are inflected for number. The following examples demonstrate this: Assu-s ham-I Man- def go- per / 'The man went'→Naa- s- di, boy - def -sat/'The boy sat down'

Inflection of Adjectives: In Yemsa, adjectives form their plural by reduplication.Ex Kawuna → 'short', Kawuna→ 'short ones', Ufur → 'Old', Ufur ufur→ 'old ones'

***Derivation of Nouns*:** Derivation of nouns is a number of nouns that are formed from verbs. The following lists demonstrate verbal noun derivation in Yemsa.

Table 2.3: Noun derived from verbs

Root words	***Derived nominal***	***Gloss***
Kass-	kasso	'To play'
Waag-	waago	'To buy'
Mams-	mamsu	'To ask'

***Derivation of verb*:** Derivation of verbs includes passive, accusative adjective, intensive and reciprocal. In Yemsa, a passive verb is formed with suffix {-te}. Examples Mes -te be broken', wor-te 'be killed'

In direct causative verbs show that someone directly causes someone to do an action.

In Yemsa direct causative verbs are formed with morphemes {(-S)i}. consider the following examples: Naars-si/naati/ 'make someone to taste' 2. Maa-si 'make somone to eat'. Though there is no single morpheme that derives a verb noun, there is a tendency for the derivational affix to be a back vowel u, -a or -o. Example; kassso maar 'To play is good.

2.7. Overview of Amharic Language

Amharic is a widely spoken language all over the country and serving as the working language of the government of Ethiopia [37]. The Amharic alphabet is called Fidel that developed from the Ge'ez abugida. The structure order of Amharic is SOV [38]. The modern Amharic writing system is known as hohiyat
(ሆሄያት). It is conveniently written in a seven-column tabular format [39]. In
contrast to certain other Semitic languages, writing alphabetically from left to right. There are 34 consonants in it, making 7 * 34 = 238 syllable patterns/fidels/. It has exhibited inflectional patterns that are rich in both verbs and nominals.

Amharic verbs have even more complex inflectional and derivational process than other word categories, and they comprise the majority of words in the language [39]. Nominals in Amharic nominal is inflected for different grammatical features that is gender, number, case and definiteness, but also verbs are inflected for person, gender, number, tense, mood and aspect [41]. Each letter represents a consonantfused with its corresponding vowel. The vowels are melded with the consonant form through diacritical markings. Amharic language follows a structure where a group of consonants comprises the fundamental meaning of a word, which is referred to as the root word. The root word can be expanded and modified by adding different affixes or combining it with other related root words to convey additional ideas and nuances. These vowel patterns produce derived words when combined with affixes including prefixes, infixes and suffixes. Such derivational processes make this language more morphologically complex. Additionally, Prepositions, conjunctions, negations and other syntactic words are attached to an orthographic word, creating a wide variety of word forms.

2.7.1. Amharic Punctuations

Identifying punctuation marks is essential for understanding word boundaries in natural language processing [42]. Punctuation marks are symbols used in sentences and phrases to clarify their meaning. There are unique punctuation marks for Amharic, and the most frequently used are listed below

':' (hulet netib) is used to separate words. Currently, this symbol is not as common, instead of using white spaces to separate words in electronic or paper-based writings, Amharic uses them.

:: (arat netib) is the ending symbol for a sentence

፣ (netela serez) used to break up sentences, separate lists, indicate sentence and comparative and sequential lists of words, phrases, or numbers.

? (tiyake milket) Indicates a clause or phrase that asks a question

! (kal agano) Positioned at the end of a sentence or word to emphasize emotions

."”(timirt tiks) used for quotation marks, direct speeches, or to draw attention to a word

2.7.2. Amharic Morphology

In Amharic, morphemes can be categorized as either free or bound. Free morphemes are capable of functioning as standalone words, while bound morphemes cannot [43]. A verb is derived from an Amharic base, which is a collection of consonants. However, Stems are, therefore, formed by intercalating the vowels among root consonants. A group of free or bound consonants or consonant-vowels which also known as stem.; a free stem can stand alone as a word, whereas a bound stem has a

bound morpheme attached to it. The word "በልጅነት" is an example of a morpheme. It can be broken down into three distinct morphemes using morphological analysis: the prefix "በ-," the root "ልጅ," and the suffix "-ነት." A word is made up of a group of phonemes, or sounds, and can be as simple as one morpheme or have multiple of them [43].There are two ways to create words from morphemes in Amharic. By derivation and inflection, these are.

Inflection: The combination of a word and a morpheme, known as inflectional morphology, typically provides a word that has the same syntactic function and is in the same class as the original stem. By designating a word category for gender, number, case, definiteness, aspect, politeness, one can achieve inflection [45]. Based on [48] states that Amharic word classes belong to five classes,

these are noun (ስም), verb (ግስ), adjective (ቅጽል), preposition (መስተዋደድ) and adverb

(ተውሳከ ግስ). since the Amharic is highly inflectional, a given root of a language word can be found in different forms. From these classes, highly inflected parts in Amharic are discussed in the following.

Nouns (ስም): A noun is a name that indicates a person, place, thing, animal, feelings or concept [46]. Amharic nouns are marked according to their gender, case, number, and definiteness; this causes the noun to become infected and acquire an affix.Therefore, a noun is in the form of stem + {Gender Marker Suffix, number Marker Suffix, Case Marker Suffix, Definiteness Marker Suffix}. It can be done by adding morphemes to a word category to indicate whether it is singular or plural (and vowel changes) or repetition of words as shown in Table 2.4[47].

Table 2.4: Amharic Noun Plural Formation

Noun in singular	***Morpheme***	***Plural Form***
ወንድም	-ኦች	ወንድሞች
ሰው	-ዎች	ሰዎች
ጻደቅ	-ን/-ኦች	ጻድቃን/ጻድቃኖች
ገዳም	-ት/ኦች	ገዳማት/ገዳሞች
ዘበነ	እነ-	እነዘበነ
ቅጠል	Plural formation repletion	ቅጠል-ቅጠል[ቅጠላቅጠል]

Amharic nouns are intended to belong to a word category that is definite. This can be accomplished by appending morphemes or vowels according to the nouns number, gender, or ending as shown in Table 2.5[48]. Amharic nouns are designed for a word category of gender as shown in Table 2.6 [49] and It is possible to do it by attaching the morphemes -ኢት. For example, ላም-ኢት[ላምት] በግኢት[በጊት] አሮጌ-ኢት[አሮጊት]. Amharic nouns are marked for a word category of caseand it can be in both objective and possessive case. Objective case can be achieved by affixation of morpheme -ን. For example: አህያ-ን [አህያን].

Depending on the person, number, gender, and/or ending of the noun, morphemes or vowels can be appended to create the possessive case (personal pronouns by prefixing የ, የ-ለማ➔ [የለማ].

Table 2.5: Amharic Nouns with Gender Marks

Subjective case	***Person***	***Number***	***Gender***	***Possessive case***
ልጅ	First	Singular Plural		ልጅ-ኤ-ልጄ ልጅ-ኣችን ልጃችን
	Second	Singular	Masculine Feminine	ልጅ-ህ-ልጅህ ልጅ-ሽ-ልጅሽ
		Plural		ልጅ-ኣችሁ-ልጃችሁ
	Third	Singular	Masculine Feminine	ልጅ-ኡ-ልጁ ልጅ-ዋ-ልጅዋ
		Plural		ልጅ-ኣቸው-ልጃቸው
በግ	First	Singular Plural		በግ-ኤ-በጌ በግ-ኣችን - በጋችን
	Second	Singular	Masculine Feminine	በግ-ህ-በግህ በግ-ሽ-በግሽ
		Plural		በግ-ኣችሁ-በጋችሁ
	Third	Singular	Masculine Feminine	በግ-ኡ-በጉ በግ-ዋ-በግዋ
		Plural		በግ-ኣቸው-በጋቸው

Verb (ግስ) ፡- Generally, Amharic verbs are derived from roots and express person, number, voice (passive or active), tense, gender, and other information through a combination of prefixes and suffixes [50]. While prefixes and suffixes are used to inflect verbs to show intention, verbs in the passive voice are designated bya suffix that varies depending on the person and number [51]. The firstone appears at the conclusion of an Amharic sentence, and the second one has a suffix that indicates the sentence's subject attached [52]. እሷ ምሳዋን በላ-ች

/she ate her lunch/ The word that is underlined is a verb, and the suffix "ች"" indicates that the sentence's subject, who is a woman of the feminine gender, is she (እሷ). Since Hebrew verbs are marked with person, case, gender, number, tense, aspect, mood, and other information, they generally exhibit a high degree of inflection. For instance, "አልሰበረንም" denotes the following: the negation አል.......ም, the subject እሱ (singular, masculine third person), the object እኛን (plural first person), and the past tense ሰበረ.

Adjectives (ቅጽል): Adjectives in Amharic change nouns or pronouns by characterizing, defining, or expressing quantity. While adjectives discuss an object's behavior or qualities, such as its shape, size, color, type, or property, nouns describe an objects nature. Adjectives are inflected for gender, number, and case just like nouns are. Some Amharic adjective categories can indicate numbers by repeating the second consonant and using the fourth vowel form in between the repeated consonants, according to reference[53]. For instance, the singular form ቀይ(red) become ቀያይ

'Qeyay' (ቀይኣይ) in plural form of the verb Adjectives invariably precede the nouns or pronouns they modify. However, not every word that appears before a noun can be an adjective. For example: - (ይህ መጽሓፍ) "This book" In this example ይህ/ "This" precede the noun መጽሓፍ "book" but this doesn't mean (ይህ) "this" is an adjective, it is a pronoun. some of the morphemes that are used to inflect given adjectives are, '-ኦ/o','-ኢት/it', -'ኦች/oc' and 'ት-/t'.

Table 2.6: Amharic Adjectives Inflection

Singular Form	***Plural Form***	***Prefix***	***Suffix***
በር	በሮች		...ኦች
ምድርዊ	ምድርያዊያን		..ያን
ማን	እነማን	እነ...	

***Derivational morphology*: -**is a branch of morphology that studies how morphemes are formed into words by means of compounding or affixation. An affix is a morpheme that is attached to a word's stem or base form to change its meaning or form a new word [54]. Nouns: - By inserting vowels between consonants, such as ጥ- ቅ- ም➔ ጥቅም, Verbal roots can be used to form Amharic nouns. It can be produced by suffixing bound morphemes with adjectives. (Derived noun) ➔ ደግነት (morpheme)-

ደግ (adjective). It can be derived from stems by bound morphemes suffixed or prefixed. For example, ጠቀም-ኤታ➔ It can be obtained by appending the bound morpheme -ታ to verbs that resemble stems. For example, ደስ-ታ➔ ደስታ. It is also obtained by suffixing bound morphemes to nouns. For example, ኃይልኧ-ኛ➔ ኃይለኛ. Compound Words (sometimes by affixing the vowels ኧ and ኦ). For this instance, Noun + [ኧ] + Noun.ቤት+[ኧ]+መንግስት➔ ቤተ-መንግስት.

Verb: verb is a word derived from roots that refers to an action, occurrence, or state of being or condition and forming the main part of the sentence [55][56] Amharic verbs take subject markers as a suffix like -ሁ /- hu/ for subject 'I', -ህ /-h/ for subject 'You', ች /-c/ for subject 'She' and so on, to agreewith the subject of the sentence [57]. Verbal Roots can be used to create Amharic verbs by adding the vowel -ኧ- to produce CኧC1C1ኧ C-, e.g., ስ-ብ-ር

➔ስኧብብኧር- [ሰበር] and repeating penultimate consonants and affixing the vowels -ኧ and -ኣ- to produce CኧC1ኣC1C1ኧC-, e.g, ፍ-ል-ግ➔ፍኧልኣልልኧግ [ፈላለግ]. By appending morphemes, verbal stems አ-, ተ-, አስ e.g., መጠቅ- (Verbal Stem) + አ- (morpheme)➔ አመጠቅ.

Adjectives*:* Amharic adjectives define, characterize, or express quantity, changing nouns or pronouns in the process. Always appear prior to the nouns or pronouns they modify. However, not every word that appears before a noun can always be an adjective. For example: - ይህ ቤት "this house" in this example ይህ "this" precedes the noun ቤት "house" but this doesn't mean ይህ "this" is an adjective, it is a pronoun. An Amharic adjective can be created by inserting vowels in between consonants from verbal roots (Baye Emam, 2003). For example, ድ-ር-ቅ by utilizing the vowel, ድኧርኧቅ it produces the adjective ደረቅ. Nouns with bound morphemes suffixed (ኧኛ፣ ኣማ፣ኣም፣ ኣዊ). stems through bound morpheme suffixation (ኣ፣ኡ፣ኢ,ታ).

Affixation: An affix is a morpheme that is attached to words stem or base form to change its meaning or generate a new term [58]. Affixes can be prefix, suffix, or infix in Amharic. Prefixes are morphemes that are added to words at the beginning, whereas suffixes are added to words at the end to create derivatives. Infixes are inserted in the body of a word causing a change in meaning, this is readily apparent in the reciprocal and repetitive aspects of an Amharic root word.

2.8. Related Work

There are many studies conducted on machine translation approaches, techniques and implementations that have been documented. In Ethiopia, several machine translations between monolingual and bilingual ways available and have been tried to be developed and documented as research work. This section discusses previous works related to our study in the MT domain. We classified the section into three sections as Machine for Ethiopian language pairs, machine translation between foreign language and Ethiopian language pair and Machine translation for foreign language pairs and we will discuss all in detail in the next subsections.

2.8.1. Machine Translation for Ethiopian Languages Pairs

In the study [55], Ge'ez-Amharic bi-directional NMT using deep learning methods conducted. The researcher wants to show off deep learning models abilities on MT tasks for those morphologically complicated languages. The author described the statistical MT approach's lack of suitable alignment quality as the reason the sentences are misaligned and the impact on the performance of the translation process. The research prepared a dataset from the Bible that used previous research, praise of Mary, Mass, and other Ethiopian Orthodox Church faith books. The researcher used different tools, such as edraw Max to draw figures for studies and a browser for searching. The researcher had done the experiments and evaluated the text-based Ge'ez-Amharic in bi-directional mode in two major deep learning models. The first is to sequence with attention and Transformer model: the researcher collected a corpus of about 20,746 parallel sentences for the experiment. 13,787 parallel corpuses were collected from previous researchers, and 6958 parallels were prepared by the researcher.

In the work [16], Bi-directional Amharic to Kistanigna MT using the DL approach. The aim of the research was to design Amharic and Kistanigna in a bi-directional MT system. The researcher collected parallel corpuses from the dictionary of kistanigna to Amharic and Holy Bible parts such as Saint Matiwos, Saint Marikos, and Saint Lukas Wongel. He was done by encode-decode architecture and training by LSTM, LSTM with attention, BI-LSTM, CNN with attention and Transformer models. The researcher used Morffessor tools able to segmentthe data and adjust the pair of sentences. To meet his goal, efficiency is considered in terms of training time, memory usage, and BLEU score.

Finally, the result of his studies was measured by measuring BLEU score metrics of 21.31 from Amharic to Kistanigna and 22.4 from Kistanigna to Amharic using the Transformer model. Then the researcher recommends that the transformer model was an efficient mode for his studies, and the limitation of the studies is the unavailability of enough datasets to conduct an extensive experiment**.**

In the work [17], bi-directional English-Awngi MT using deep learning. The main goal of the researcher is to design and implement English-Awngi using bi- directional deep learning. The reason behind this is that no research was developed between English and Awngi. The researcher collated parallel corpora from educational materials, religious books, and social and mass media documents. The researcher used 8000 pairs of sentences split by an 80/20 mechanism for training and testing. The researcher followed different preprocessing techniques and used the following tools: a Core i7 CPU laptop, Python programming and the models selected: the LSTM, CNN and transformer models after training the selected deep learning model. The results achieved after training were BLEU scores of 24.94 and 22.34 from English to Awngi and Awngi to English, respectively. The researcher recommended MT for other languages using the transformer model.

2.8.2. Machine Translation for Foreign Language Pairs

In the paper [29], German-English MT with Gated recurrent unit RNN. The aim of the researchers' framework is to serve as a pilot way of translating strings from German news media into English sentences, build applications, and pave the way for further work in the area. They develop a framework that can be useful in developing mobile applications (apps) for quick translation, where efficiency is crucial, and get datasets from the WMT2021. And used Python Sklearn train-test-split method; the corpus will be shuffled and split, with 80% of the data for training and 20% for testing, to add robustness to the framework. They completed two experiments by categorizing RNN model with different hyper-parameters. Model I had a final training accuracy of 0.655 and a final validation accuracy of 0.653, and model had a final validation accuracy of 0.645 and a final validation accuracy of 0.649. Model I have a final training loss of 2.78 and a final validation loss of 2.85. Model II has a final validation loss of 4.66 and a final validation loss of 5.55.

In the work [57] conducted their study on the MT system using DL for English to Urdu [64]**.** The source of the corpus contains words from English-Urdu parallel corpus from news and sentences that are frequently used in day-to-day life. They investigated the English to Urdu MT model, which is LSTM using encoder-decoder, using different parameters for training and the creation of a new parallel corpus. They evaluated the model using several metrics.

In the study [58], English to French Machine Translation using Seq to Seq model with NN conducted the input sequence was mapped to a vector with a fixed dimensionality using a multilayered LSTM, and the target sequence was decoded from the vector using another deep LSTM. They used WMT '14 English to French dataset and trained their models on a subset of twelve million sentences consisting of 348M French words and 304M English words, which is a clean selected subset from. They used the common BLEU score metric to assess their models. 34.81 BLEU score is c. The researcher found that neural network design performed better when compared to phrase-based statistical machine translation. They discovered that switching the word orders related several short-term dependencies in all source sentences, but not in target sentences, which greatly enhanced the LSTM's performance. between the source and target sentences and thereby reducing the complexity of the optimization barrier.

Based on a review of related works, there is no research that has been carried out on MT between Yemsa and Amharic. As a result, this study targeted the design of a Yemsa-Amharic machine translation. In this investigation, an effort is made to adapt Yemsa DL models to translate the sentence. Because deep learning models are the SOTA technique for translation, it is confinable advantage of using an encoder-decoder language model to simplify complicated tasks such as separate language modeling, translation modeling, tuning, and decoding. And also, DL has an advantage compared to previous approaches to improving translation. As researchers review different literature, they select RNN such as LSTM, LSTM+ attention, GRU and Transformers models. Then to evaluate the model performance, they used a BLEU to conduct these experiments, used to evaluate the model translation accuracy by contrasting its output with reference phrases that have been translated by humans.

2.8.3. Machine Translation for Foreign and Local Languages Pairs

In the work [59], English for Afaan Oromo Bi-directional MT using CNN by standing from the problem of the study of machine translation from English and Afaan Oromo that was studied by the SMT approach, that result was not effective. The researcher reviews different literature that studies foreign languages, like MT between English and other European languages. She suggested that a neural network-based approach has better learning ability because it captures and applies the features of humanlearning behavior and the previous studies done for translating the English to Afaan Oromo don't show satisfactory accuracy. She implemented three systems: the first system was a statistically word-based approach that refers to a baseline; the second system was an RNN approach that was a competitive model and the third system was CNN. For Afaan Oromo to English and English to Afaan Oromo languages, the result of CNN BLEU score 3.86 from English to Afaan Oromo and BLEU score 3.32, the training performance in a translation process from English to Afaan Oromo was better. The convolutional neural network approach is quicker to train than the RNN approach, according to the researcher's conclusions. The CNN approach result showed an improvement of BLEU score 1.58 on translations from English to Afaan Oromo and BLEU score 1.51 on translation from Afaan Oromo to English compared to the recurrent neural network approach.

In the work [60], English-to-Dawurootsuwa MT model using RNN, theresearcher was developing a unidirectional English-to-Dawurootsuwa MT model by using the NN technique. He was done designing the algorithm to recognize and analyze patterns in sequences of data and predicted the output text based on the inputted text data. The researcher used RNN, GRU and LSTM to contain a neuron and each neuron is replaced by the cells containing control gates. That is used for hidden cells that control gates to maintain the flow of sequence in an inaccurate order and a fully densely connected model. He used parallel corpus, a total of 20,345 pairs of corpora collected from different sources and divided into 90% by 10% for training and testing. The researcher conducted several hyper parameters, which are 22 inputs and 27 output nodes and implemented Python programming using the Keras library and the Adam algorithm. Totally, they achieved four results using the BLEU score and Arithmetic Mean Value (manual evaluation) techniques. In the RNN model, the BLEU score is 0.5187, the learning rate is 0.002, and the AMV result is 0.60914.

In the work [61], conducted on Context based MT with RNN for English and Amharic translation. The aim of researcher is to combine a RNN machine translation and context-based machine translator for translating English and Amharic text. He statedthat NMT mimics human neural networks in training and decoding the way of the phraseand sentences, the system has contained two: encoding and decoding with attention algorithm. In the encoding phase, each word is changed to a vector, which was modified to another vector based on words next to it in able to make the system context mechanism. The author used collected dataset from New Testament bibles, python programming language for implementation and its NLTK library and TensorFlow to build up the system. The researcher used a BLEU score method to evaluate the performance of the model. Their model outperforms the previous model by attaining a Bilingual evaluation metrics score obtained 2.805% results by using combinational approach.

In the work [62], on Attention-based NMT from English-Wolaita. The target of the researcher is to develop a NMT for English-Wolaita by using attention mechanisms.She has been trained on a parallel corpus covering religious and frequently usedsentences or phrases that can be used in day-to-day communication. She collected a parallel corpus totaling 27351 sentences. The researcher was prepared, and the model was trained and tested using the 80/20 ratio. And the data were preprocessed in a suitable format to be used in NMT. For building the proposed English-Wolaita NM system model, an LSTM encoder and LSTM decoder architecture with an attention mechanism has been proposed in the Seq-to-Seq concept. Overall, the researcher was evaluating the efficiency of the proposed studies, and to measure the performance by the blue score for testing the efficiency of the attention mechanism, the researcher tried to compare the model with the non-attention model and the attention mechanism. Lastly, she proved that the attention mechanism has a better translation and has achieved a BLEU score of 5.16 and 88.65% accuracy for English-Wolaita.

Table 2.7: Summery of Related work

Author	***Title***	***Tool***	***Data***	***Result***	***Gap***
Workineh Wogaso [1]	Attention-based Amharic to Wolaita Neural machine translation	Python, Keras library NumPy library, Colab tospeed up GPU, mosses	3000 sentences of parallelfrom religious books and 6280 datasets	NMT (BLEU score 0.5960 without attention 0.6258 With attention).	Only Recurrent neural networks and works time step not parallelism during Low Bleu score value
Amdework Asefa [55]	Bi-directional Ge'ez-Amharic NMT: deep learning Approach	Colab Jupyter Notebook using python programing Version 3.7 GPU RAM Keras library	Tensorflow datasets such as wmt13_translate/fr-en"46 with random 80% Training, 10% test, 10%validation dataset split.	BLUE score of 22.9 and 29.7	Hyperparameter is not clear state Implementation is not based on methods.
Mengistu Kinfe [16]	Amharic-Kistanigna Bi-directional MT using deep learning.	Python for coding with keras, PyTorch Libraries, Notepad++	From bibles and 3500 parallel sentences from Dictionary of Both languages.	21.31 and 22.40 BLEU Score	Not leverage word2vec He used dictionary but not show in the experiment.

Elias Asefa [60]	Developing English to Dawurootsuwa machine translation model using RNN	Python programing, Adam algorithm, keras toolkit, TensorFlow, Arithmetic Mean Value and BLEU score	Holy Bible, from cultural museum and educational office. Secondary data like books, articles, publications, unpublished, web	0.55 BLEU score, 0.60914 AMV	One direction only Not work on embedding techniques Conduct on RNN types
Agerie Belete [17]	Bi-directional English-Awngi Machine Translation Using Deep learning	Python programing usingGoogle Collaboratory by Jupiter notebookkeras	From Awngi app, Amhara Media Corporation, Proverbs and Educational document	BLEU score 25.78 & 24.94	Dataset not train by different train test ratio Not used CBOW and

2.1.1. Summary of Related Work

There are different kinds of research done on MT using different approaches for local and foreign language pairs. However, there has been no research conducted on machine translation between Yemsa and Amharic using any MT techniques previously. As, we review several studies done on other language pairs strongly endorsed by the NMT such as deep learning algorithms rather than other traditional machine translation approach. The aim of this work is to apply how to implement Yemsa to Amharic machine translation encoder-decoder architecture. Depending on the selected LSTM, LSTM withattention, BI-LSTM, GRU and transformer model, and we have done the experiments using those selected models because, in general, as we have seen in the above-related works, most of the research done for different language pairs in recent years has been done using this algorithm. To evaluate the performance of the model, we choose BLEU score, which evaluates translation accuracy by contrasting system output with reference sentences that have been translated by humans. We reduced the complexity of the work; we used preprocessing techniques and optimization methods for enhanced translation using either a BLEU score or training speed utilizing a deep learning MT model.

3. CHAPTER THREE: MATERIALS AND METHODS

3.1. Introduction

This chapter describes the proposed approaches, methods, tools, and techniques in detail used in studies for the Yemsa to Amharic machine translation model and vice versa. This includes the research approach used in the show and reflects the proposed method of the translation model, material, tools, and technique used in the thesis. In these studies, we followed pre-processed research methods. And in order to achieve the general and specific objectives of this study, we followed different steps. These are data collection, a literature review from different journals, methods, the preparation of a parallel corpus, preprocessing, training and testing of the proposed framework, the development of a prototype and validation of the model. The details of the techniques and algorithms reviewed and followed to build the translation model will also be reviewed. The linguistic behavior of the Yemsa and Amharic languages will also be investigated and identified.

3.2. Proposed Approach

The general research approach followed in this research will be discussed under list

- Collect parallel corpora from various sources for the Yemsa and Amharic
- Review several pieces of literature on previous studies on machine translation.
- Align the dataset using software tools.
- Using several data preprocessing techniques, such as data cleaning, normalization, tokenization, word embedding, and padding.
- The preprocessed data is converted into a vector.
- Design and build a model to translate input into the target language.
- Train the proposed model by using the prepared dataset.
- Train the model by utilizing different hyper parameters.
- Evaluate learning parameters such as train rate, embedding dimension, batch size, number of epochs and percentages for train and test.
- Select the best model for the Yemsa-Amharic machine translation.
- Lastly Thesis report writing

3.3. Literature Review

In this study, a detailed literature review on machine translation is carried out across different language pairs. By searching different related literature (text document, Internet, Books) Journals articles, conference and other literature related to MT is also done using the principles, methods, techniques, tools and materials they used. Additionally, the various implementation strategies will be carefully examined and the syntactic relations between Amharic and Yemsa will be examined.

3.4. Parallel Corpus

The success of DL heavily relies depending on the data's quality used, as it is a data-intensive process that requires vast amounts of data. It is powerful enough to achieve near-human-quality translation results with proper tuning and to enhance models over time by improving their general performance after their initial deployment has ended. Since there are no parallel corpora available and limited resources to prepare parallel corpora for Yemsa and Amharic, we used existing and publicly available documents, which are the New Testament Holy Bible and religious documents that include Yemsa and Amharic. Other sources of data were gathered from the education domain, including Yemsa fereta and others. To start our translation process, the data was checking syntax, semantic and correct of sense sentence in both languages, to prepare the data in the form of UTF-8 format. We prepared parallel corpus. Parallel corpus on both translation units has been created. We collected a total of 6000 parallel sentences to implement the experiments. Morfessor word segmentation tool is used to prepare data; we have those corpora from different sources such as Holy bible, Media Agency and Education domain. Such data is prepared by notepad and notepad++ saves txt files.

3.5. Material and Tools

3.5.1. Hardware Tools

We used the selected Hardware tools such as HP laptop with Processor: Intel(R) Core (TM) i5-7320M CPU @ 2.90GHz, 8GB RAM, 1T hard disk, screen size: 14 inches, Flash disk 32Gb, Wi-Fi Divce:4G lite.

3.5.2. Software Tools

Software tools are used to implement the research because they can help researchers carry out their work more efficiently and accurately. The software tools used will depend about the subject of study and the type of research being conducted. Before selecting the tools, we considered some criteria that are helpful in selecting the appropriate software tools with their corresponding libraries. The other criteria are to select tools with enough learning materials such as free video tutorials and existing experience and the other is to select tools that must be used on machines with limited resources (like CPU only). We used Python as a programming language with Tensorflow and Keras libraries environment to implement the neural layers. These tools fulfill all the consideration criteria and they are used in Python, which is familiar to us." if you were working with a particular corpus format, the functions to read and write the format could be kept together"[63]

Google Colab: is based in the cloud platform that offers a free environment for Jupiter notebooks for running Python code. It is a Google has provided that users to run Python code using Google's computational resources, including GPUs and TPUs, without the need for any local hardware or software installation. And Integration with Google Drive: Google Colab and Google Drive are integrated, enabling to access notebooks and data from anywhere. While Google Colab is popular platform for running python code online and provides basic python environment, some commonly used libraries like Tensor flow, Pandas, Numpy etc.

Tensorflow: An open-source tool framework is used for data movement and programming that is adaptable throughout a number of tasks, like as, including deep learning and machine learning [64]. The Google Brain group created it and is now widely used for developing and training machine learning models. The architecture of Tensorflow works for preprocessing of the data, model training and testing. Tensor Flow also used for graph framework for graphical representation of the series of computations during the training. It provides two distributions one for CPU- based and GPU based computation.

Keras**:** is designed to be modular, user-friendly, and extendable, facilitating use for developers to build and experiment with deep learning models. It has variety of already constructed layer like optimizers, activation functions, and loss functions, as well as tools for data preprocessing and model evaluation. Keras is also integrated with Tensorflow, allowing developers to take advantage of the Tensorflow computation graph and hardware acceleration features [65].

Diagrams.net: is a no charge online software tool to draw system architecture, flowcharts, and visual representations of information. It includes drag-and-drop capabilities and various formatting options to create a professional-looking diagram. Diagrams created in the past can still be loaded and edited in the current version of the application. It also provides various integrations with popular platforms, such as Google Drive.

Table 3.1: Tools and material

Tools and material	***uses***
Anaconda with integrated Python	To implement and run model code
HP PC: 14-inch, 2.50 GHz, Intel Corei5 CPU- 8G, 8GB RAM, 1 Tera HDD	To process multiple tasks of the research and as storage
External storage such as Flash disk	To store data/files
Microsoft office word	To write documentation and
Microsoft PowerPoint	Prepare presentation PowerPoint
Mendeley	For reference Citation
Notepad++	To prepare dataset and text processing
Pen and paper	To take a note and printing
Google Colab	To compile and execute Python code
Google Drive	To upload dataset

In addition, edraw max 8.9, draw.io, and the online diagrams tool will be used to draw diagrams of these studies. And web browsers like Google Chrome, Opera and Micro soft Edge will also be used for searching source. Overall, we used different tools in order to build the proposed model, like word alignment and decoding. Language modeling (LM) is an effort to capture natural language regularities in order to enhance the functionality of various natural language applications.

3.6. Data Pre-processing

The training and testing corpus for our work was obtained from different sources that provide parallel text (Yemsa and Amharic) in the religious sector, specifically bible texts, from primary Grade student text and proven. The input sentence must be transformed during preprocessing to a shape that is appropriate for the morphological analysis. Tokenization, normalization, vectorization is all included in the preprocessing stage.

3.6.1. Data Cleaning

Data cleaning, on the other hand, involves locating and fixing faults or inconsistencies in the data. This may include taking out duplicates making spelling corrections, removing irrelevant or noisy data, and resolving inconsistencies between different data sources. It is important to remove any noise or inconsistencies present in the dataset. It ensures that the input data is of high quality and reduces the opportunity for the model to learn from misleading or irrelevant information. To achieve our objective, we followed several pre-processing techniques in these studies such as removing unnecessary punctuation, correcting spelling errors, eliminating duplicate sentences, removing digits and removing special characters.

In Yemsa language, there is some hard sound when speaking, for example, Hai'o. This type of punctuation is removed, and for other special characters or different characters in Amharic language, the extra space will be removed those types of steps were done in Amharic, but in Yemsa, we were not able to obtain the same meaning as the acronym when studied. To perform the experiments, we tried to collect manually all the parallel sentences collected from different sources. We try to clean on notepad++ in order to avoid mis-aligned sentences, remove mistranslated sentences and miss-spalled words as well as we remove duplicate sentences from the file. In order to avoid those errors, we try to clean on notepad++.

The process of parallel corpus cleaning involves a number of tasks, such as removing words that have a digit, shortening sentences that are too long, removing digits, superfluous spaces, unwanted punctuation marks, and special characters. Parallel data cleaning is an important step in machine translation, as it helps to ensure that the input data is of high quality and suitable for training a translation model. Sentence alignment is the process of identifying which sentences in the source language correspond to which sentences in the target language. This is important because the training data must be aligned in order for the translation model to learn how to translate accurately. Language identification is the process of identifying the language of each sentence in the parallel corpus. This is useful for the translation model must know which language it is translating from and which language it is translating to.

Algorithms 3.1: Algorithms for cleaning a dataset

Define: regular_expression_for_filtering

Remove special character

For pair in lines

Define: clean_pair

For line in pair

Normalize Unicode characters

Tokenize on white space

Remove punctuation from each token

Remove non-printable chars form each token

Remove tokens with numbers in them

Store as string

Append to clean_pair

Return array (cleaned)

3.6.2. Data Tokenization

Tokenization is vital in machine translation. It breaks sentences into words. These smaller units are also fragments [66]. It also identifying and separating words, punctuations and other elements of the text into discrete units that can be processed by the MT model. This is crucial because in order to provide an accurate translation, it needs to comprehend the grammar and meaning of each word in the text. For this thesis we used python programming language to implement Experimental tasks. The amount of vocabulary, maximum duration of sequence and the representation of words with unique numbers have all been determined. For efficient model training, the field additionally adds two more tokens, the "start of sequence" (sos) and "end of sequence" (eos) tokens.

Algorithm 3.2: Algorithms for tokenization a dataset

Input: Parallel sentences

Output: Tokenized sentences

Let text pairs be an empty list of strings

While input parallel sentences ends with "\n":

Split each sentence into two if a delimiter is found

Tokenize both Yemsa and Amharic sentences into words

Put both tokenized sentences in text pairs by appending them

End of while

Return text pairs…… from (Workineh Wogaso, 2020).

The above algorithms provided perform tokenization on parallel sentences. First it initializes an empty list called "text pairs" to store the tokenized sentences. The code enters a loop that continues given that the input parallel sentences end with a newline character ("\n"). Inside the loop, each sentence is split into two parts if a delimiter is found. This step assumes that the sentences are separated by a specific delimiter. Both the Yemsa and Amharic sentences are tokenized into words. Tokenization is the method of dividing a sentence or text into individual words or tokens. The tokenized Yemsa and Amharic sentences are appended to the "text pairs" list. Once the loop ends, the function returns the "text pairs" list containing the tokenized sentences.

We explored and accessed the complete parallel corpus of data in order to give each word a unique number. Therefore, the process of assigning a special number to a word begins as soon as that word is viewed and, the words were represented by the unique number. Depending on the size of the vocabulary in each language, the most recent term is assigned the last number. These steps improve the preparation of data for training by converting the source dataset to integer number representation form. Data that has been converted to integer form is referred to as tokenized data. As a result, the vocabulary y size is set and the row data is tokenized for data pre-processing. The last vocabulary has an index of 1, whereas the first vocabulary has an index of 0. For the Yemsa and Amharic languages in our total word corpus, we have a total of 11690 and 14226 different vocabularies. ኢትዮጵያ ተፈጥሮ ያደላት ሀገር ናት tokenized as "ኢትዮጵያ", "ተፈጥሮ", "ያደላት", "ሀገር,"ናት". However, in order to know the boundary of end sentences we have used to indicate the beginning of a new sentence (SoS) and to indicate the end of the sentences (Eos) like "", "ኢትዮጵያ", "ተፈጥሮ", "ያደላት","ሀገር,"ናት".

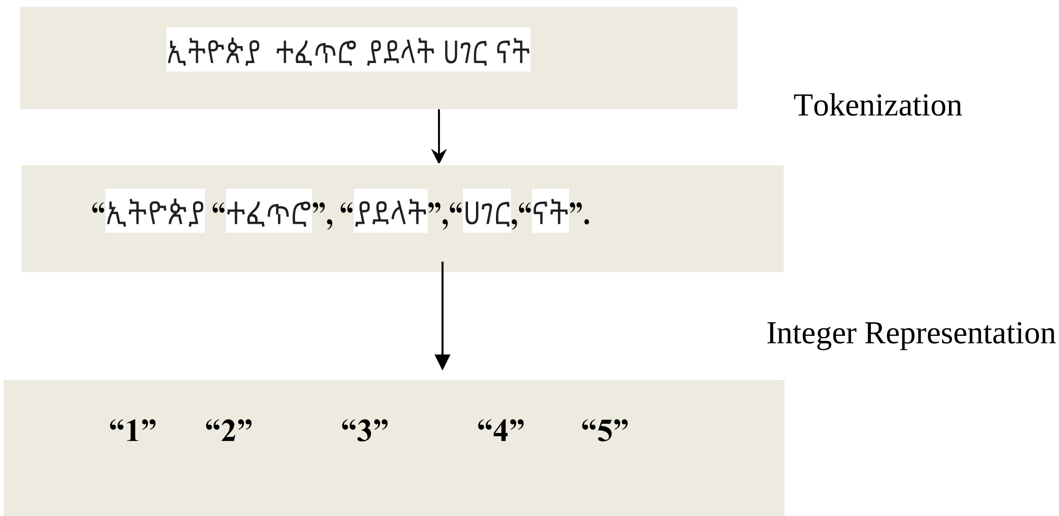


Figure 3.1: Tokenization and integer representation

A list of sentences cannot be understood by the machine directly. Instead, these sentences must be broken up into a list of words, and these lists of words must then be represented in integer form. Each word in the phrase is represented in this integer form representation by an individual integer number that was created during the vocabulary creation process. Once the text has been tokenized, the resulting tokens can be further processed such as removing punctuation, unnecessary symbol. These transformed tokens are then used as input for various NLP models to extract meaningful information and perform the desired analysis. Tokenization is the action that causes this. In our work, the division of the sentence into words is done using whitespace.

3.6.3. Data Normalization

Normalization is one the data preprocessing that comes after tokenization. We have applied normalization for Amharic sentences. Those algorithms adopted by[67]. Two categories of normalization problems that come up with the Amharic language were developed by the author. The first one is a replacement made using a representative alphabet from a set of similar alphabets. For example [ሐሓኃሃኀኻ] is replaced with 'ሀ'. We have shown this character replacement more in the algorithm 3.3. After that, the labeling process involves performing word normalization using Labialized Amharic characters such as ጠጥቶዋል or ጠጥቱኣል to ጠጥቷል.

Algorithm 3.3: Normalizing labialized characters

Define normalization list (n) with value (v) # e.g., ሃኀሐሓኻ=>ሀ, ኁሑኹ=>ሁ…ሠ=>ሰ...

Open and read Amharic corpora # this is input

While reading line do

For c in line #read each character from each line

if c is in n # if c is in a normalization list

Then replace c with normalized value (v) # e.g., if c is ሃ, ኀ or ሐ then replace c with ሀ v
end if end while

close file

3.7. One Hot Vector Representation

One technique of transforming each unique word into individual numbers is known as one-hot representation. A vector with a length equal to the number of categories in the given dataset each category consisting of a single unique word is produced by this technique [66]. After changing the dataset to integer form, it has to be changed in to two-dimensional longer vector which is, one-hot vector representation in order to make suitable machine translation. In this technique, each word in the vocabulary has a distinct binary vector allocated to it with a value of 1 in the position corresponding to the word and 0 in all other positions. For each word in the sentences in this one-hot representation, a distinct vector representation is used. Integer-based vocabulary is not directly processed by the neural network. It must switch to vector representation, also known as a single hot vector representation, in order for the neural network to operate the vocabulary. We expressed the vocabulary by 1 for each location in the word list, otherwise by 0. The neural network may use the binary vectors as input and they are simple to process using normal matrix operations. To fix the length of the sentences during one hot vector representation, we utilized zero padding to the right. In this case the maximum length of the sequence is 22 and 16. So, we have used zero at right for the remaining index. Sequence lengths for both the input and outputs are variable in machine translation.

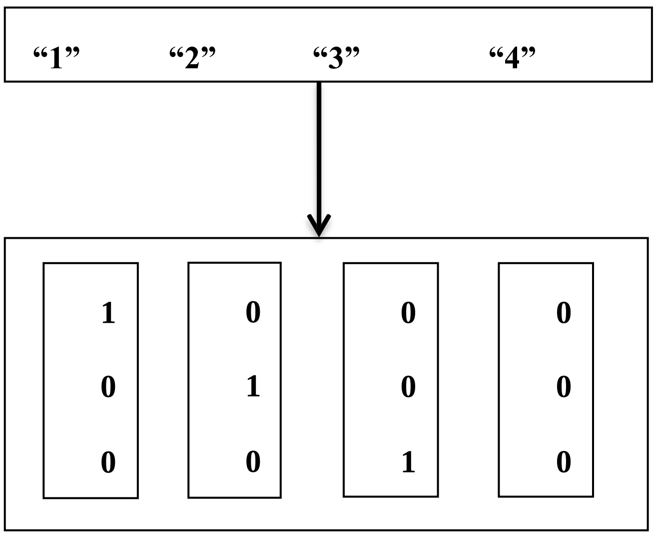


Figure 3.2: One hot vector representation

3.8. Padding

Padding is used to make sure that all input and output sequences are equal in length. It is inserting zeros to until the series reaches the appropriate length, at the conclusion. During one hot vector representation we have used zero to the right to set the length of the sentences fixed, which is known as zero padding. We used padding of sequence for unique words and to compare in the input part and the output part [17]. If Yemsa come at first columns, it is called input sequence and the output sequence is Amharic; in most of them, the length of Yemsa is longer than Amharic. So, we have used padding, which involves adding special tokens (usually zeros) to the end of the sequence until it reaches the desired length. Once the padding is completed and the vector size is standardized, we have given such vectors to the neural network in the embedding layers.

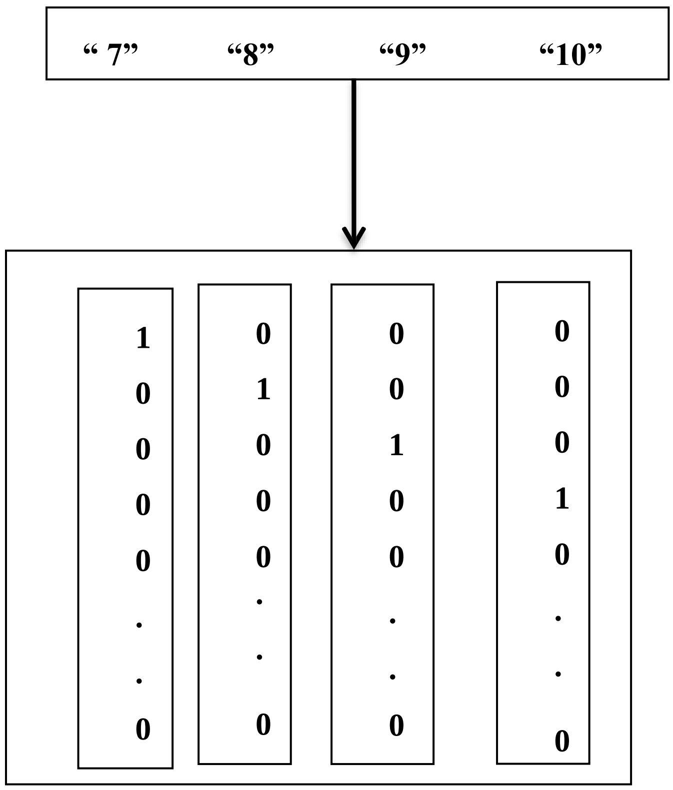


Figure 3.3: Padding

3.9. Word Embedding

There may be some issues with training waste instead of learning when zero values are encountered when we had directly passed one hot vector of input to the neural network. It is a way of representing words on a vector space where the words having the same meaning have similar vector representations. Additionally, the decoder accepts the encoder output via the context vector and produces the target sentences in the end. The task is processed either directly or via word embedding by the encoder-decoder model [68]. We made use of word embedding to address this issue, which converts a higher dimensional to a lower dimensional vectorization. Word embedding changes the visual display of text in words to actual numbers between -1 and 1. In our thesis we used embedding dimension and layer keras embedding to generate word embedding with corresponding vector and predication-based word embedding models are word2vec and Glove.

Word2vec uses either CBOW or skip-gram models to learn word vectors from text corpora. Word2vec is a predictive model to efficiently create word embedding by using neural network. It was first introduced by Mikolov et al. from Google in 2013[68]. which is the most popular word embedding model. It uses shallow neural networks to calculate a word embedding based on the context of the words. The objectivefunction of Word2Vec causes the words that have a similar context to have similar embedding. Embedding is an important step in machine translation because it allows usto represent words and phrases in a way that can be processed by neural networks. Thereare different ways to generate machine translation embedding. By using word embedding, the dimensionality of the input can be reduced, making the computations more efficient and allowing the neural network to learn more meaningful representationsfrom the data. One common approach is to pretrain a neural network on A large amountof text using techniques such as word2vec. This allows the network to learn word embedding that the correlations between words in statistics in the corpus.

Table 3.2: dimensional embedding vector

ኢትዮጵያ	0.2	0.4	1.2
ተፈጥሮ	0.1	0.3	0.5
ያደላት	0.7	0.9	0.8
ሀገር	0.6	2.1	1.5
ናት	0.9	2.2	2.0

3.10. Encoder

After padding and word embedding was done, the produced tensor will be provided as an input to the encoder. Encoder is one component of NMT architecture that accepts a single element as an input, then gathers data about the element and propagates it [69]. It only reads one elements or word at time, hence when reading with n wordsor an input sequence of length L, it will take L time steps.

3.11. Decoder

Decoder is responsible to Convert a sentence in the source language into a sentence in the target language. It decodes what it has received and produces the equivalent meaning of what has been encoded. Decoder generates a sentence at a time and repeats the search until it encounters the end of the string indicator which shows the last word of the sentence. It translates the vector representation back to a target sequence with variable length. Encoder-decoder networks use one network to encode or turn into an internal representation of the input, and then another network uses that input to turn it into the result of [71].

3.12. Attention Mechanism

The neural network finds it challenging to handle long sentence as a result, especially if they are longer than the sentences in the training corpus. Because a neural network must be able to fit all the data from a source sentence into a fixed-length vector, this encoder-decoder approach may have issues [64]. Data sequences from the whole source language are encoded by the encoder into a single context vector and then passedto the decoder to produce the output sequence. As a result, the decoder can only access the output context vector of the encoder. Because of this, one realistic solution to this problem is the attention mechanism, which has recently become more popular in neural network training. Unlike encoder-decoder architectures without attention mechanisms, which initialize the decoder hidden state only once using the source representation, architecture for encoding and decoding with care mechanisms predicts a target sentence on the context vectors associated with the source language position and the previously generated target words.

3.13. Optimization Algorithms

There are different optimization algorithms; those are GD, RMS prop, and Adam [73]. We utilized Adam optimization because it is better to increase the performance of the translation without decreasing the accuracy of our model.

3.14. Performance Measurement Methods

3.14.1. Accuracy

The accuracy metric is used to monitor the model's performance during training. It gives an indication of how successfully the model can categorize or predict the right output given the input data. We also used this measurement of accuracy to assess the quality of the translation in the translation process. It is computed computationally by adding the errors for each sample throughout the training set. Each batch of data is used to calculate training loss, which is then plotted as a curve.

3.14.2. Loss Function

The loss function is a crucial element in the training process when solving a translation or recreation problem and it is a crucial element in a machine translation system [73]. This is used to quantify the discrepancy between the expected and actual values, where the robustness of the model grows as the value of the loss functiondecreases.

3.14.3. Cross-Entropy

Cross- entropy is the default loss function to use when classifying in binary problems. It is intended for use binary classification where the target values ranging from 0 to 1. It is the suggested loss function in terms of mathematics for the maximum probability inference framework. In order to predict class 1, it will compute a score that encapsulates the average difference between the actual and anticipated probability distributions. The value of perfect cross entropy is 0 and the score is minimized loss. It can be specified as the loss function in Keras by specifying binary cross-entropy at compiling the mathematical computation behind [72] cross-entropy is tough, but fortunately, it is easy to understand and implement it.

3.15. Fine Tuning

Overfitting can occur when a model becomes too specialized to the training data and performs poorly on new, unseen data. i.e, they model the training data too well and performed poorly on unseen data. One way to overcome the issue here is to apply regularization. And it adjusts the weights and biases during training to minimize the loss function and improve accuracy. It is widely used for training neural networks efficiently [73].

Learning Rate: The step size at which the model's weights are changed during training is determined by the learning rate. It is used in the training of neural network model and adjusting the weight of the network that assign the range between positive real values (0.0 to 1.0). It also controls the speed of convergence and the sensitivity to changes in the loss function. The learning rate influences the size of the step update parameter that is used to approach your loss functions' minimum point [71]. Different learning rates can have a significant impact on the model performance, so finding an appropriate learning rate is important.

Batch Size: The batch size determines how many training examples there are examples processed in the iteration before updating the model’s eights. A change in batch size can affect the training dynamics, memory requirements and convergence speed. Smaller batch sizes can lead to more stochastic updates, while larger batch sizes can provide more stable gradients but may require more memory.

Number of Epochs: it describes a whole run of the training dataset. The number of epochs determines how many times the model will iterate over the training data. Too few epochs may result in underfitting, while too many epochs may lead to overfitting. It is important to find the right balance to achieve optimal performance.

3.15.1. Dropout

Dropout is the way effective way to regularize deep NNs by introducing stochasticity in neuron activations [71]. During network optimization, dropout randomly deactivates neurons by setting their values to zero with a certain rate.This technique enhances the network's ability to generalize and reduces the risk of over fitting. As a result, a more generic network is developed, which performs better on fresh data and is less prone to overfitting [72].

3.15. 2. Early Stopping

As explained earlier, an over fitting model performs well on training data, perform poorly on skipped data according to [69], it is a widely used method to prevent poor ability to generalize when training a model that is overly expressive through gradient-based optimization. During the training process, the networks performance on unseen data is assessed using a separate by test set. This test set serves as a selection that is typical of the test set, containing data that the model has not encountered before. By evaluating the network performance on this test set. According to Geoff Hinton "Early stopping beautiful free lunch". We should thus always monitor error on a testing set during training and stop (with some patience) if you're testing error does not improve enough.

3.16. Evaluation Metrics

BLEU scores are popular in NLP, including machine translation addressing this inconsistency between the training objective and evaluation metrics is well- researched area and finding a solution can have a significant impact on a wide variety of tasks popular today [68]. It calculates common elements of n-grams continuous sequences of match words between the system output and the reference translations. The core ideas behind BLEU score that a good translation should contain similar n-grams to the human reference translations. Performance evaluation is important after training models, there are various methods used in MT in relation to translation accuracy. BLEU computes the precision of n-grams including unigrams, bigrams, trigrams, etc. The effectiveness of MT is determined by assessing how closelyit aligns with human translation. When the machine-generated text closely resembles the human translation, it is considered to be of higher quality and more successful in achieving accurate translation [67]. Other automated metrics: In addition to BLEU, there are other metrics such as METEOR, NIST, ROUGE, and TER that are capture different aspects of translation quality. METEOR does consider multiple reference translations when evaluating the quality of a machine translation. NIST Emphasis on longer n-grams, lexical and surface-level features may not adequately capture the coherence, and fluency of the translation. ROUGE focuses on text summarization quality, while TER measures the edit distance between the machine translation and the reference translation.

4. CHAPTER FOUR: RESEARCH DESIGN

4.1. Introduction

This Chapter covers the architectural design of Yemsa to Amharic MT model and it's vice versa using deep learning models. The proposed studies are based on construction of an encoder and decoder structure. The encoder decoder architecture is a recent development that became the key technology of Google Translate in late 2016[65]. The Encoder-Decoder architecture also called the Sequence-to-Sequence model, is widely employed in application involving MT and other natural language processing. The first phase which is carried out by the architecture is preprocessing steps the input read from the given path/directory, then the after-preprocessing step is done, the next step is converting the preprocessed corpus convert to vectorization. In vectorization techniques we applied One-hot vector, Padding and word embedding were done. So, we have changed it to a lower dimension vector using Keras embedding. Keras is one python library. The model of the encoder decoder is a popular architecture for machine translation tasks. The model is consisting of two parts: The first one is that the encoder takes an input sequence (in this case, a Yemsa sentence) and encodes it into a fixed-length vector. The encoder processes the input sequence token by token, updating its hidden state at each time step.

The second decoder takes the encoded vector and generates a target sequence (in our case, an Amharic sentence) one token at a time. The decoder performs a time step each time as input the previous target sequence generated token from the decoder, as well as the previous hidden state of the decoder. The hidden state is updated depending on the previous state and the previous target token and is used to generate the next target token. This process continues until the decoder brings about end of sequence token, indicating that the decoder has generated the entire target sequence. The encoding and decoding process are trained on jointly using a Seq2Seq loss function such as cross-entropy loss. During training, the input sequence is supplied into the encoder and the decoder generates the target sequence token by token. The final encoder hidden layes context vector and target embedding sequence are accepted by the decoder model during training in order to predict the target output. Thus, the context vector passed through to the decoder to get an output sequence.

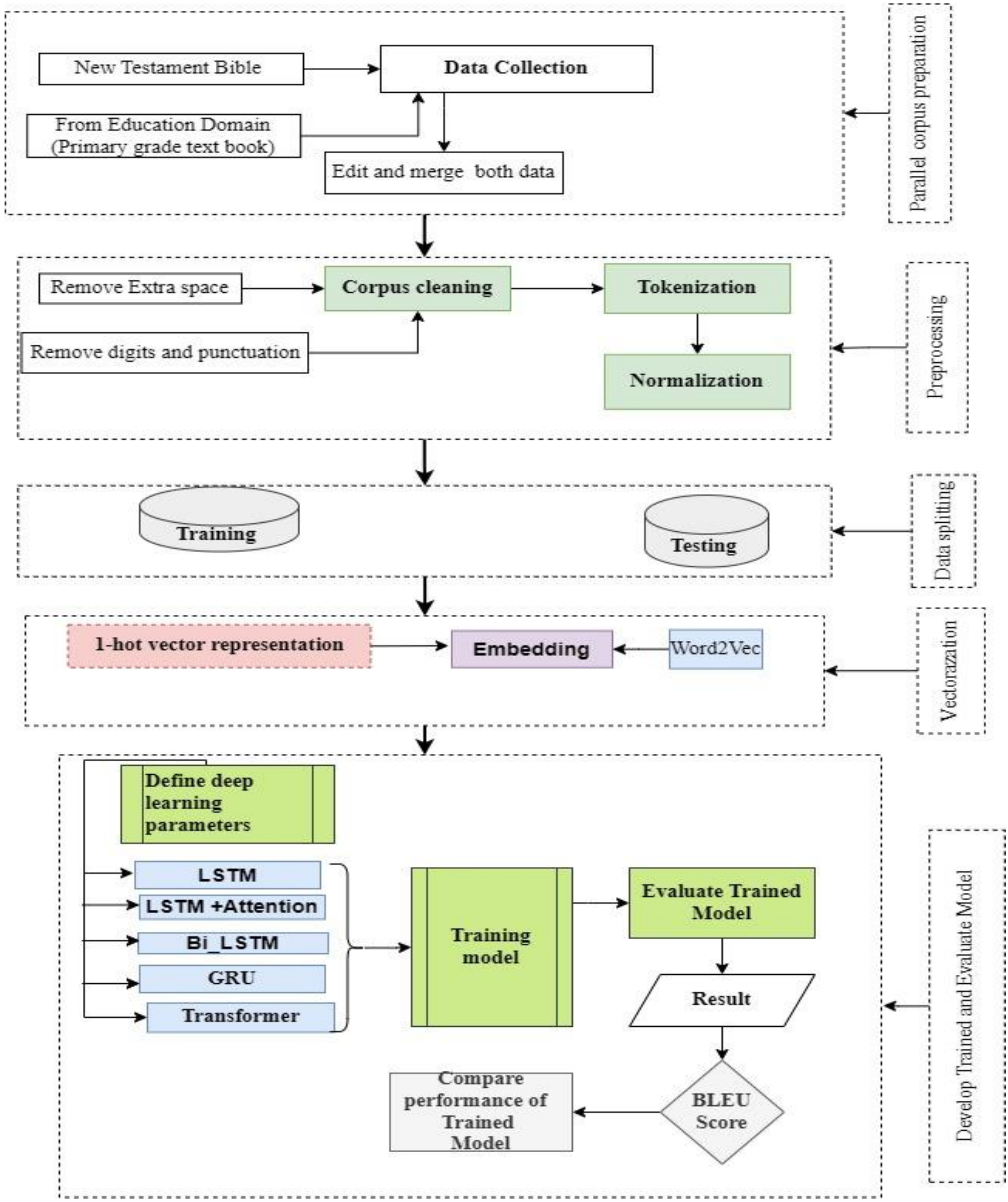


Figure 4.1: Architectural design of Yemsa to Amharic

4.2. Parallel corpus Preparation

We employ a number of religious documents as a dataset source and others for our study, as shown above in Figure 4.1. The dataset, or corpus, for both Yemsa and Amharic was compiled from hard copies of the Holy Bible and primary text books, as proven in both languages. We used a bottom-up technique to prepare the corpus, which included aligning each book verse level first, then merging the aligned books to their respective language, because parallel corpus was used equal numbers for both languages to for the purpose of effective translation.

4.3. Pre-processing

The initial stage of the pre-processing process is cleaning, tokenization, and normalization of the input corpus. Corpus cleaning: Once the corpus is constructed to guarantee the standard and consistency of the data, this means removing any noise, formatting artifacts, or irrelevant information from the sentences. The other preprocessing steps also includes tokenization, normalization, or any other necessary transformations to prepare the text for further analysis. Character normalization was used for data analysis in the studies.

4.4. Data Splitting

We have used these 6,000 pairs of sentences for the proposed study by classifying them into a training set and a testing set of data. Prior to splitting the dataset, we shuffled it because it was important to shuffle the data. The rearranged dataset was randomized in order of data points, which helps reduce bias and overfitting.

Table 4.1: Data splitting in percentages and purpose.

Dataset	***Training***	***Testing***
Total dataset in %	70%	30%
	75%	25%
	80%	20%
	90%	10%

4.5. Vectorization

The significance of vectorization is vital in machine translation by representing words or sentences as numerical vectors that machine learning models can process. There are several approaches to approaches for vectorization that are common and used in this thesis, including hot vector, word embedding, including word2vec, Glove, and Fast Text. Word embedding is used for dense vectors in a continuous vector space, capturing semantic relationships between words.

4.6. Model Selection

4.6.1. Long Short-term Memory Model

LSTM is a distinctive recurrent neural network model that solves the long sequence dependence problem in recurrent neural networks by adding memory units, input gates, output gates and forgetting gates and improves the ability of recurrent neural networks to process long sequence data [72]. The LSTM has a container that can takedata for an extended period of time. The cell is controlled by three gates: the input gate,the forget gate, and the output gate. In this case, after preprocessing and preparing the dataset, the next step is to create encoder and decoder models in sequence-to-sequence Architecture. The Seq2Seq model consists of two main components: an encoder LSTM and a decoder LSTM. The encoder LSTM takes in the input sequence (Yemsa sentence) and generates a fixed-length vector representation of the sequence, called the context vector.

After this context vector is produced into the decoder and generates output of the sequence one step at time. To train the model we used a dataset of Yemsa to Amharic sentence pairs. Each sentence in Yemsa is fed into the encoder LSTM, one token at a time and the final hidden state of the LSTM is used as the context vector. The decoder LSTM is then trained to generate the corresponding Amharic sentence, one token at a time, using the context vector and the previous output tokens as input. During training the model learns to minimize the cross-entropy loss between the predicted Amharic sentence and the ground truth Amharic sentence. The model is used for back propagation to optimize through the time, which, based on the gradient of the loss function, modifies the weights of the LSTM cells in the encoder and decoder. To build the model, First the Input and output sequence feed as x= (x1, x2, x3, ……… xn) and y= (y1, y2, y3,

……...yn), where x represents the input sequence and y represent output sequence. Second the embedding layer to changes the input and output sequences into dense vector representations.

At the third step the encoder uses LSTM layers to process the input sequence and generate a fixed-length vector representation of the sequence called the context vector. The encoder LSTM take in the embedded input sequence as input and generates a final hidden state h_n, which is used as the context vector.

Then the decoder LSTM to generate the output sequence and decoder LSTM take the embedded output sequence as input, from the context vector and the previous hidden state

and generate the current state and the output token

$$h_1, h_2, h_3, \dots \dots \dots h_n = LSTM(x_1, x_2, x_3, \dots \dots xn$$

In the given expression, x1, x2, x3, ..., xn represent the input sequence, where each x_i is an input at time step i and computes the hidden states h1, h2, h3, ..., h_n, where hn represents the hidden state at time step i.

$$h_t = (y_t - 1, h_t - 1, c$$

The model is used to update the hidden state h_t at time step, ty_t-1 represents the previous output at time step t-1. h_t-1 represents the previous hidden state at time step t-1. It contains information from the previous time step and captures the relevant context for the current time step.

$$y_t = softmax(W_s * h\_t$$

Hence, y_t-1 is previous output token, h_t-1 is the preceding hidden state, Ws is a weight of matrix, and LSTM is the Long Short-Term Memory cell. applying a dataset of input-output sequence pairs, train the model. Reduce the cross-entropy loss between the expected output sequence and the actual output sequence during training. Feed the input sequence into the encoder LSTM during inference to produce the context vector. Input the context vector and previous output tokens into the decoder LSTM, which will then construct the output, sequence one token at a time.

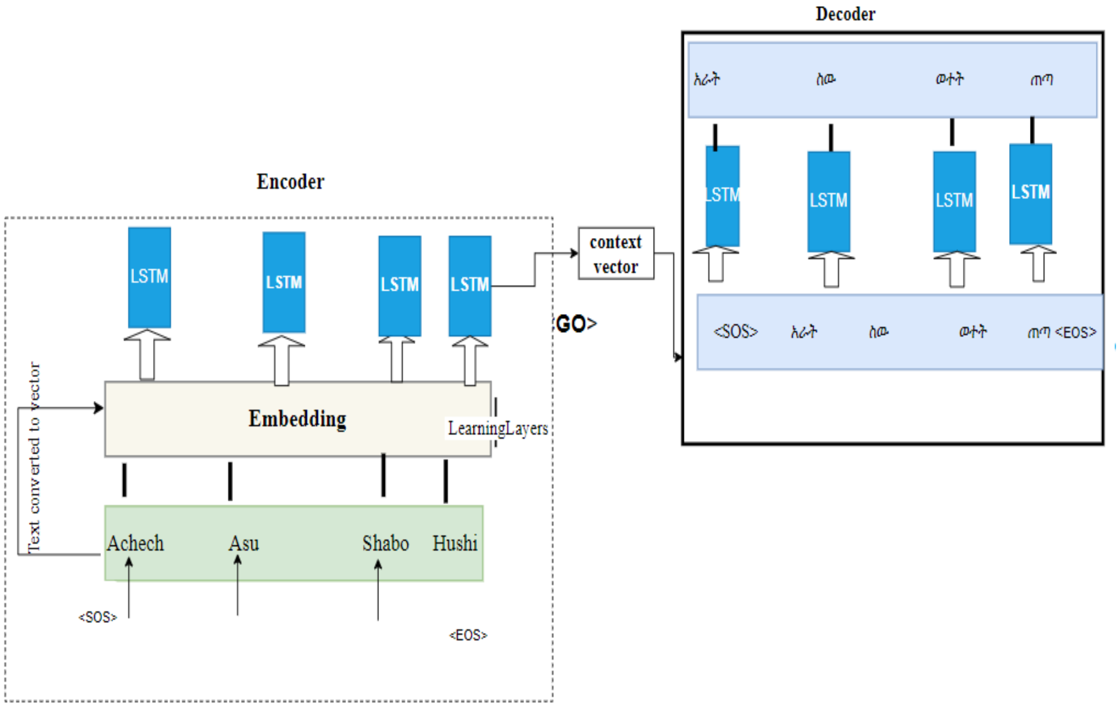


Figure 4.2: LSTM encoder decoder architecture

4.6.2. Bi-directional Long Short-term Memory Model

LSTM contains of some gates. There is an input gate for the input layer. here are forget gates and output gates in the output layer. The internal state in LSTM models only depends on prior time steps at each time step. However, in MT, the context from both the source and target languages is important. Bidirectional LSTMs address this by processing the input sequence in both ways forward and reverse directions the performance of a Bi-LSTM technique using only word embedding as input features is advanced, and adding additional capabilities could make the performance even better [73]. This allows the model to capture information from past and future time steps simultaneously.

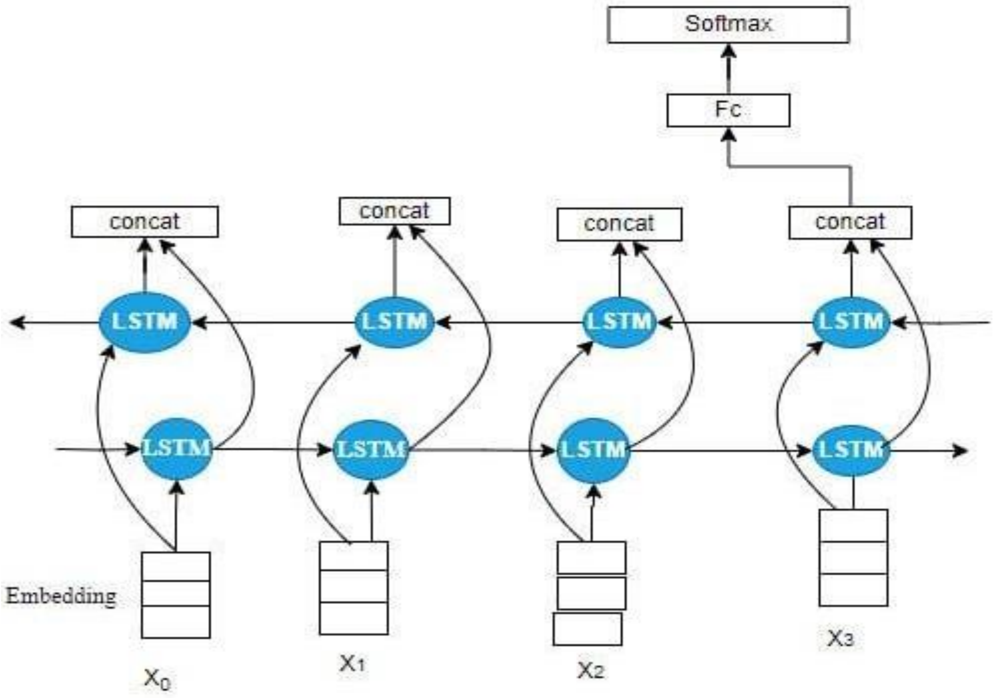


Figure 4.3: Bi-directional Long Short-term Memory Model Architecture

4.6.3. LSTM +Attention

The LSTM with attention architecture is an extension of the LSTM model. This method helps the network remember all the visited words in a sentence by supporting the storing of larger context vectors within the sequence [63]. An LSTM memory cell ispresent in the encoder. It receives an input as a sequence of elements at a single time stamp, processes those elements to gather data, and then propagates that data forward. One word is processed by the LSTM encoder for each time stamp. Therefore, if a sentence has n words or padding of sentence is of length L so it will require n number of time stamps to process it. The encoder creates a thought vector to convey the meaning of dispersed source language words. The input of attention mechanism here is all hidden state of the encoder and the current output of the decoder, for the first time we have used <sos> in our case. Attention mechanism is improving the performance of the model and allows the decoder long short-term memory to selectively attend to different the generated translations. When the model generates each output token, this can help it better capture long-range dependencies and enhance the quality of certain input sequence segments.

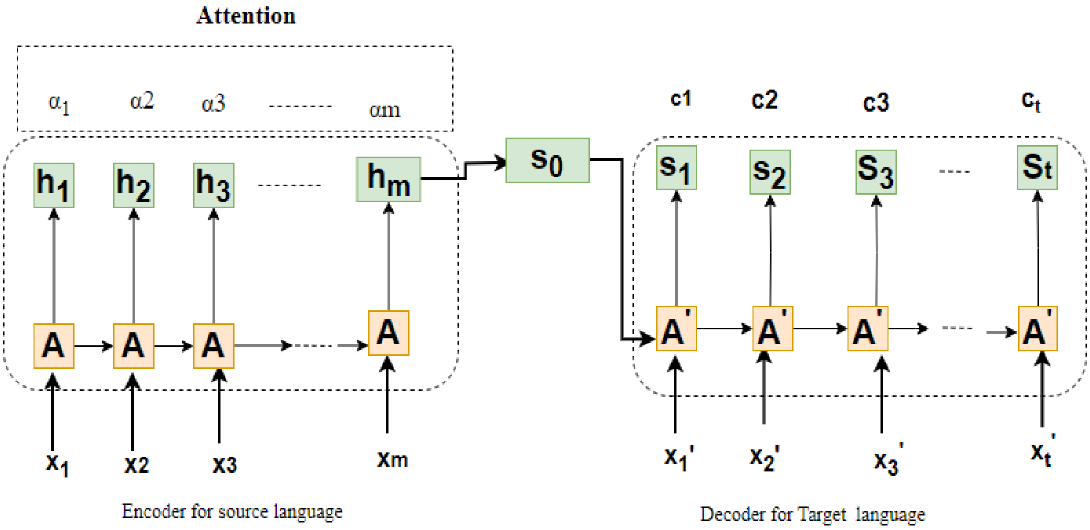


Figure 4.4 LSTM with Attention Architecture

When we use encoder-decoder architecture, the all-input sequence is summarized into a single vector as output from the encoder. This summarized vector has been given to the decoder to decode it. Continuing in the decoder, with access limited to the last layer of the encoder, representing the input sequence with a single vector would be inefficient. This issue may even get worse if the size of vocabulary is huge. To address this issue, we applied an attention mechanism for our encoder decoder model. In the way of attention-based encoder-decoder models as well as lastly generated words.

4.6.4. Gated Recurrent Unit Model

Gated recurrent unit is a modification to the recurrent neural network hidden layer that makes it much better at capturing long range connections. There is a vanishing gradient in recurrent neural networks. To address this problem GRU model is done by computing the activations at time t of recurrent neural network. It's the activation function applied to the parameter activations in the previous time set and the current input. The decoder generates the output sequence based on the input context and the last hidden state, starting with a unique token to mark the beginning of output production, which is appended to the end of the input; there’s also one at the end of the output an Amharic sentence is passed to it. Then, all layers of GRU run one after the other, following up with a SoftMax on the final layer’s output to generate the first output word. Then, it passes through word into the first layer and again repeats the generation. This is how we get the GRUs to act as a neural network model.

Decoder outputs used for word prediction and generate a new hidden decoder state and a new output word prediction and it is similar to long short-term memory but no output gate [94]. The higher length of the sequence in the output sentence data is set as the number of GRUs used. The GRU combined information from a new input source annotation vector. In this case, we would be producing the translation processes source representations, then the attention network feeds into and to generate discriminative context vectors. Gated attention is viewed as the last decoder state as the history and original source representations of current input. Because GRU is able to manage the information flow between the history and current inputs through its reset and update gates. Both models are straightforward but effective in learning and decoding. After all of the above steps, The Encoder outputs a context vector as its final output, incorporating current input data, previous output, and future output. This context vector is then utilized in the attention layer or passed directly to the decoder.

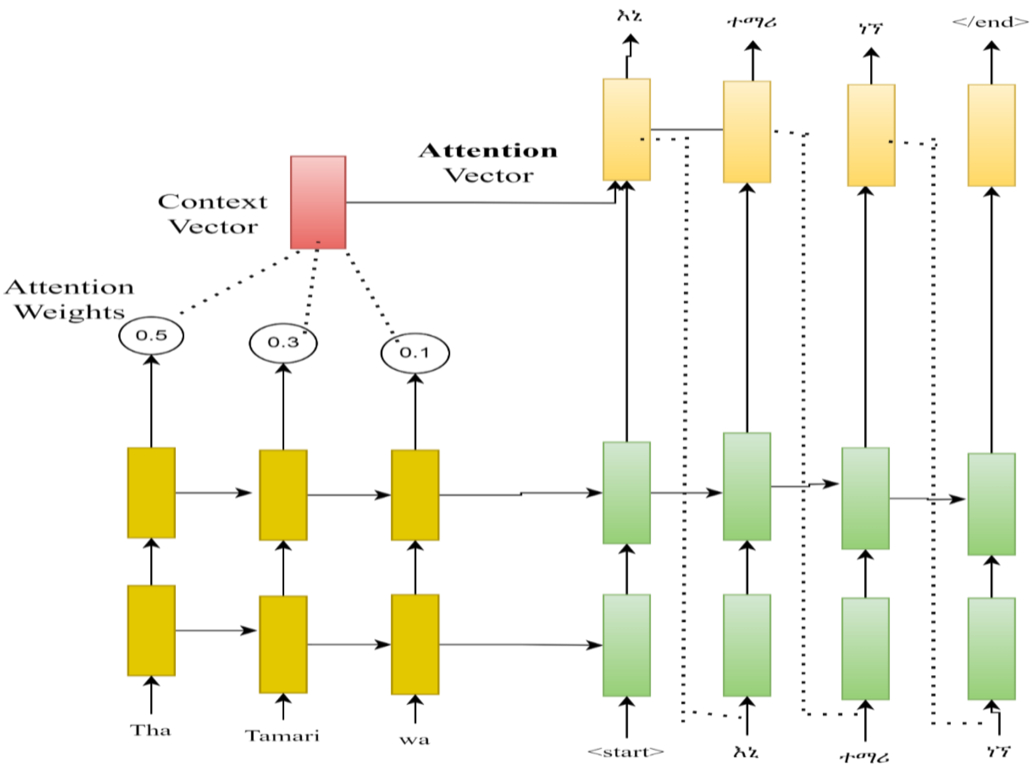


Figure 4.5: Workflow GRU

4.6.5. Transformer Model

The transformer model was invented in 2017; it consists of encoder and decoder Architecture like RNN but difference in working. Recurrent neural network has slowed in training and input data passed sequentially or serially. In transformer model input data passed in parallel. Consider in our case Yemsa to Amharic in RNN network we passed Yemsa sentences one word after the other and the hidden state depends on the previous words that the word embedding generated one word at time. In the transformer model there is no concept of time. Let’s see and discuss how this setup of the encoder and the decoder stack works and how it works as figure 4.6 below

- ***Input Embedding***: Since computers cannot understand words, they must instead interpret them as numbers, vectors and metrics. The aim is to map each word to a place in space where words with similar meaning are physically closer to one another; this is known as embedding space. We used pre trained space to save time and embedding space map words to vectors.
- ***Positional Encoder:*** in the process of embedding space words maps to vectors. However, a word may have different meanings in different sentences. This is where positional encoder comes in; it's a vector that has information on distance between words and sentence.
- ***Attention:*** what part of the input should we focus on? We translate from Yemsa to Amharic applying self-respecting self-attention. We can answer how relevant is the ith word in the Yemsa sentence to the other word in the same Yemsa sentence.
- ***Feed-forward:*** is a simple feed forward network that applies every one of the attention vectors that are used in transforming the attention vector into digestible by the next encoder block or decoder block.
- ***Decoder:*** As we know the above LSTM decoder is to generate output sequences. Sublayers in this architecture function similarly to encoder layers, but the tasks assigned to each multi-headed attention layer vary. It takes the Keys and Values matrix from the encoder stack's output and produces its own matrix from the layer underneath it.
- ***Decoder multi-Head Attention:*** The multi-head attention block is the main

innovation behind transformers. It contains three vectors from each of the encoder's input vectors; it is named query vector, key vector and value vector. These vectors are trained and updated during the training process.

- ***Add and normalization:*** from the multi head attention block, adds them all together, and then applies layer normalization to the outcome to normalize it. If you have heard of batch normalization, layer normalization is similar but instead of normalizing the input features rather than the input across the batch dimensions**.**

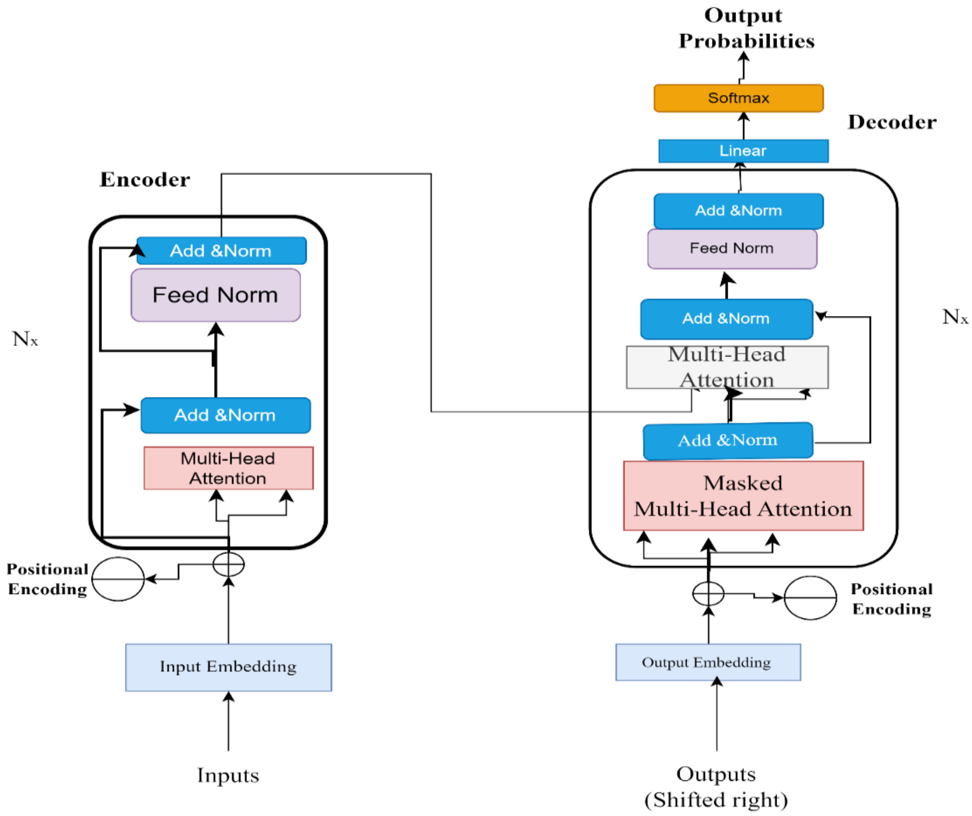


Figure 4.6: The work flow Transformer model (Vaswani et al, 2017)

5. CHAPTER FIVE: EXPERIMENTATION

5.1. Introduction

The designed architecture is discussed in the chapter before chapter. Thus, this section provides detail on corpus source, experimental implementation, detail Explanation how dataset preprocessed and how it used, Variety each conducted model experiment, parameter selection of the thesis was done It also discusses how the preprocessing affect training was suggested based on the selected model.

5.2. Data Collection and Preparation

We have collected 6000 pairs of sentences in two columns (one for Yemsa and other for Amharic) separated by a space tab as a sample in Appendix I. The corpus we have prepared are from religious books such as saint Marikos, saint Matiwos, Saint Lukas, saint Yohannies wengel and from primary grade text book. The data set used from each portion is specified under the table.

Table 5.1: Source and size of the data collected for experiment

Data source	***Number of Parallel Sentence***
Saint Marikos	1000
Saint Matiwos	800
Saint Lukas	1450
Saint Yohannies	750
From Education Grade 1-6	2000

5.3. System Environment

To implement our proposed studies model programming language selection and preparing the required environment is necessary. Python is one of the programming languages used to create the system. Python is supporting a set of a freely available library. We used Python 3.8 with Jupyter notebook for coding, Keras library, Tensorflow 2.12 library and numpy library which is freely available. We have chosen google Colab as a baseline for implementation and as main integrated development environment.

Google Colab has enabled GPUs containing 32GB RAM with cloud resources provided by google, it is the only tool that can increase training time hundreds of times more quickly than a CPU. We used the desktop with and HP Laptop for preparation of corpus and preprocessing, for writing the research report. The Google Colab with GPU is used for the major laboratory environment for implementation of training the proposed model of the studies. To start our experimental jobs first storing the dataset and save as txt files then upload to Google drive, then open Google Collaboratory notebook mount drive to Collaboratory after mounting, find to obtain the path of dataset, second each preprocessing step is discussed below.

5.4. Data Cleaning

The first step in data preprocessing is corpus cleaning. It is the first step of data pre-processing. We begin the cleaning process by loading the dataset by their path using the. open ()' function. Each line of text files contains a single pair of sentences separated by a tab, the first in Yemsa and the second in Amharic. Using the read ()' function, we split the loaded text into new lines and sentences.

```
def preprocess_Amha_sentence(sent):
    '''Function to preprocess Amharic sentence'''
    sent = re.sub("'", '', sent) # remove the quotation marks if any
    sent = re.sub("[२३०८९५७१४६]", "", sent) # remove the digits
    sent = sent.strip()
    sent = re.sub(" +", " ", sent) # remove extra spaces
    #sent = '<start> ' + sent + ' <end>' # add <start> and <end> tokens
    return sent
```

```
[ ] # Generate pairs of cleaned Yemsa and Amharic sentences
import re
sent_pairs = []
for line in lines:
    sent_pair = []
    yemsa = line.rstrip().split('\t')[0]
    Amha = line.rstrip().split('\t')[0]
    yemsa = preprocess_yemsa_sentence(yemsa)
    sent_pair.append(yemsa)
    Amha = preprocess_Amha_sentence(Amha)
    sent_pair.append(Amha)
    sent_pairs.append(sent_pair)
```

Figure 5.1: sample code to show cleaned data

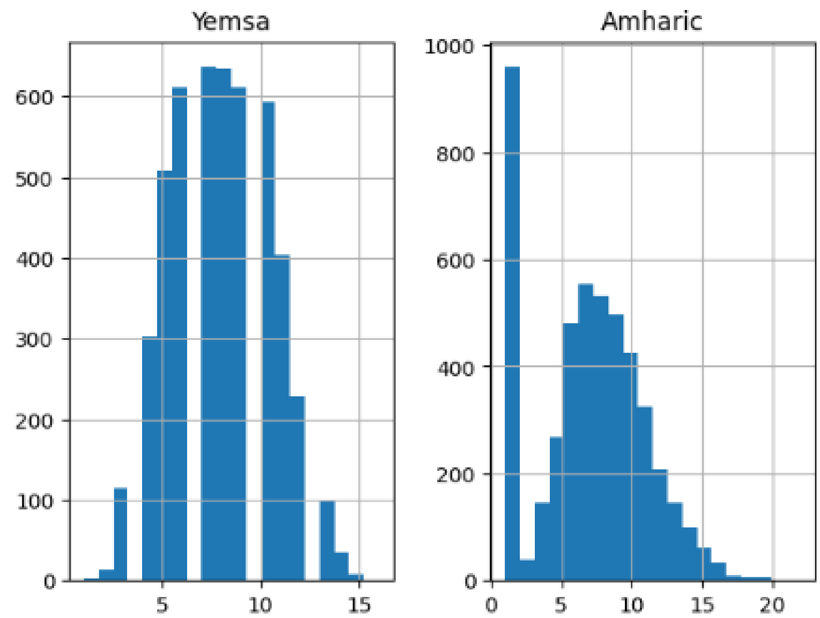
5.5. Tokenization

The process of transforming plain text into token sequence is called tokenization. In that process, two major operations are carried out in order. As required for modeling, we mapped words to integers using the Keras library's tokenized class. As required for modeling, we mapped words to integers using the Keras library's tokenized class. We separate tokenizes for both Yemsa sequences and the Amharic sequences. A function named create tokenizer () to train a tokenized on a list of phrases for tokenization, we utilized the built-in function tf. keras.preprocessing.text and the delimiter white space (' ').The Tokenizer(filters=' ') library is derived from the Tokenizer class.

```
#tokenize the input sentences(Yemsa sentencelanguage)
input_tokenizer = Tokenizer(num_words=MAX_NUM_WORDSi)
input_tokenizer.fit_on_texts(input_sentences)
input_integer_seq = input_tokenizer.texts_to_sequences(input_sentences)
print(input_integer_seq)

word2idx_inputs = input_tokenizer.word_index
print('Total unique words in the input: %s' % len(word2idx_inputs))

max_input_len = max(len(sen) for sen in input_integer_seq)
print("Length of longest sentence in input: %g" % max_input_len)

x_tokenizer = Tokenizer(num_words=MAX_NUM_WORDSi)
x_tokenizer.fit_on_texts(X_test)
x_integer_seq = x_tokenizer.texts_to_sequences(X_test)
print(x_integer_seq)

word2idx_x = x_tokenizer.word_index
print('Total unique words in the input: %s' % len(word2idx_x))

max_x_len = max(len(sen) for sen in x_integer_seq)
print("Length of longest sentence in input: %g" % max_x_len)
```

Figure 5.2: Sample code for tokenization

The longest sentence is initially determined using a Python function, after which each input and output sequence is encoded as an integer and padded to the longest possible phrase. This is as a result of using one hot encoder for the output sequences and a word embedding for the input sequences. The function named encode sequences () had performed these operations and returned the result. One-hot encoding is required for the output sequence. This is because the model had to predict probability that obtaining their corresponding word in the vocabulary as output. At the end we prepare the data for training and testing.




```
[[2, 298, 4010, 1231, 1628, 122, 55, 501, 21], [351, 352, 13, 3, 1232, 2361], [6, 212, 454, 1233
Total unique words in the input: 11690
Length of longest sentence in input: 16
[[2, 193, 1192, 1193, 417, 3, 418, 13, 94, 1194], [127, 1, 30, 1195, 1196, 419, 4, 34, 633, 1197
Total unique words in the input: 4408
Length of longest sentence in input: 16
[[2, 1], [2, 1], [2, 57, 408, 1875, 5, 14, 1876, 1], [2, 3744, 31, 409, 858, 3745, 3746, 1], [2,
Total unique words in the output: 12491
Length of longest sentence in the output: 22
encoder_input_sequences.shape: (4797, 16)
encoder_input_sequences.shape: (1200, 16)
decoder_input_sequences.shape: (4797, 22)
decoder_output_sequences.shape: (4797, 22)
```

Figure 5.3: Shows the total dataset read for training

The above sample code present that it creates a data frame called length with columns Yemsa and Amharic and It initializes an instance of the tokenizer class called input tokenizer and fits it to the input sentences. At first, it converts the input sentences to integer sequences using the texts_to_sequences method. Secondly, it assigns the mapping of words to indexes from input_tokenizer to the variable word2idx_inputs and prints the total number of unique words in the input. Thirdly, it computes the maximum input length among all the input sequences and prints it. Finally, it initializes another instance of the Tokenizer class called x_tokenizer and fits it on the X_test data. It converts X_test to integer sequences using the texts_to_sequences method and gives them to the next method, which means creating layers.

```
from numpy import array
from numpy import asarray
from numpy import zeros

num_words = min(MAX_NUM_WORDSi, len(word2idx_inputs) + 1)

embedding_layer = Embedding(num_words, EMBEDDING_SIZE,input_length=max_input_len)
decoder_targets_one_hot = np.zeros((
        len(input_sentences),
        max_out_len,
        num_words_output
    ),
    dtype='float32'
)
decoder_targets_one_hot.shape

for i, d in enumerate(decoder_output_sequences):
    for t, word in enumerate(d):
        decoder_targets_one_hot[i, t, word] = 1

encoder_x = Input(shape=(max_x_len,))

y= embedding_layer(encoder_x)
```

Figure 5.4: Creating Embedding Layer and One-Hot Encoder

The sample code shows that it creates an embedding layer using the embedding class, specifying the number of words, the embedding size, and the input length. And also creates a umpy multidimensional array called **decoder_targets_one_ho**t of shape (number of input sentences, maximum output length, and number of words in the output) and initializes it with zeros. It iterates over the decoder_output_sequences and assigns a value of 1 at the corresponding position in decoder_targets_one_hot for each word in the sequence, using one-hot encoding. Finally, this code creates an embedding layer for the selected models and generates one-hot encoded targets for the decoder based on the provided output sequences.

```
encoder_inputs = Input(shape=(max_input_len,))
x = embedding_layer(encoder_inputs)
print("truuuue")
encoder = LSTM(LSTM_NODES, return_state=True, dropout=0.001)
encoder_outputs, h, c = encoder(x)
encoder_states = [h, c]
encoder_outputsx, hx, cx = encoder(y)
encoder_statesx = [hx, cx]
decoder_inputs = Input(shape=(max_out_len,))
decoder_embedding = Embedding(num_words_output, LSTM_NODES)
decoder_inputs_x = decoder_embedding(decoder_inputs)
decoder_lstm = LSTM(LSTM_NODES, return_sequences=True, return_state=True,dropout=0.2)
decoder_outputs, _, _ = decoder_lstm(decoder_inputs_x, initial_state=encoder_states)
decoder_dense = Dense(num_words_output, activation='softmax')
decoder_outputs = decoder_dense(decoder_outputs)
```

Figure 5.5: Building encoder decoder and compile model

The function for above to codes presents an encoder with an input shape, applies an embedding layer to the inputs, and passes them through an LSTM layer to obtain the encoder outputs and final states. For the decoder, it defines an input layer, an embedding layer, and an LSTM layer with return sequences. The decoder LSTM takes the decoder inputs and the encoder states as the initial state. The decoder LSTM outputs are passed through a dense layer with a SoftMax activation to generate the final decoder outputs, representing the probability distribution over the output vocabulary. The given code snippet defines a sequence-to-sequence model using Keras. It uses categorical cross-entropy loss and the Adam optimizer to construct the model. The Adam optimizer is used to compile the model. The optimizer is configured with default parameters, and the loss function is set to categorical cross-entropy while accuracy is chosen as the evaluation metric. To monitor the training process, an early stopping to protect overfitting.

5.6. Parameter Selection

This section describes the parameters of our experimental study to train and test the selected model. To get the intended outcome, we applied various experiments on various parameters using our training data as well as testing. We started by selecting the dimension of embedding size. We chose latent dimensions of 512 and 1024, and we used embedding dimensions of 128 and 256. In order to select the best dimension for the training, it depends on the selected model. And next, we choose batch size 64. It indicates how many samples must be examined before the internal model is updated parameters.

We select a learning rate 0.01. It serves to minimize the loss of training. The other hyper parameter is dropout rate used to reduce the redundant or repeated nodes utilized to lower the training dropout rate, which is 0.2, by reducing the redundant or duplicated nodes. Then we have done experiments with epochs multiplied by 50. It can be described as a number of iterations to learn the model working through the whole dataset. On the other hand, epoch is described as how many times our model trains the whole dataset.

Table 5.2: Hyper-parameters and values of the proposed studies models

No	***Hyperparameters***	***Model Types***				
		GRU	***LSTM***	***Bi-LSTM***	***LSTM with Attention***	***Transformer***
1	Batch size	64, 32	64, 32	32, 64	32, 64	32, 64
2	Embedding Dim	256	256	256	64	64
3	Hidden Dim	256	256	256	64	64
4	Learning rate	0.01,0.001	0.01,0.01	0.001,0.01	0.01,0.01	0.01,0.01 ,0.1
5	Optimization	Adam optimizer	Adam optimizer	Adam optimizer	Adam optimizer	Adam optimizer
6	Loss function	Sigmoid	Cross entropy	Cross Entropy	Cat-cross Entropy	Cat-cross Entropy
7	Epoch	50	50	50	50	50
8	Units	1024	512	512	1024	1024
9	Num_ Head	-	-		-	8
10	Dropout rate	0.1	0.2	0.2	0.2	0.2

6. CHAPTER SIX: RESULT AND DISCUSSION

6.1. Introduction

This chapter discusses the experimental results of the proposed studies by showing the necessary experimentation results. Evaluating the performance by BLEU score metrics acquired through different neural network models and their comparisons. The outcome of the experiment is shown in different graphs and tabular forms.

6.2. Experimental Result

Experiments are conducted to measure the efficiency of selected model result through on the training by using different hyperparameter in the task. We used Colab to train our model. Colab or Collaboratory is a product of Google that helps users to write and execute python code using the browser and it is principally used for data analysis, machine learning. We carried out four DL algorithms as discussed that are LSTM, LSTM with attention, Bi-LSTM and GRU with attention and Transformer model.

Experiment 1: the first Experiment done by LSTM

In this scenario, we used the parameter that described the previous chapter, as seen by the tabular form of the first experiments with the LSTM model. We used batch size, embedding dimension, epoch, loss function, learning rate and dropout hyperparameter to build LSTM model. We have seen the output contain input layers, output layers, training time, accuracy and loss in graphical form after training once and we changed the value of batch size, embedding dimension, dense unit, learning rate, and epoch between 10 and 50 epochs.

The model trains by increasing when maximizing epoch's value, accuracy rate for training increases, and loss decreases. We used Adam for the optimization algorithm to update network weight about the learning rate with 0.01 after 50 epochs. We have also used SGD with the same epoch and learning rate, but Adam faster than that of SGD at training time; we got 0.20 and 95% accuracy for this model. In this model, there is an activation function at each neuron, which is for activating neurons. We got BLEU scores of 4.37 and 5.27 from the Yemsa to Amharic translation and Amharic to Yemsa respectively. The change in the size of parameters in the experiment, including the number of epochs, batch size, learning rate, and embedding size, affected the results of training and testing. The experiment concentrated on measuring the model's performance using the BLEU (Bilingual Evaluation Understudy) score, accuracy, loss and training time.

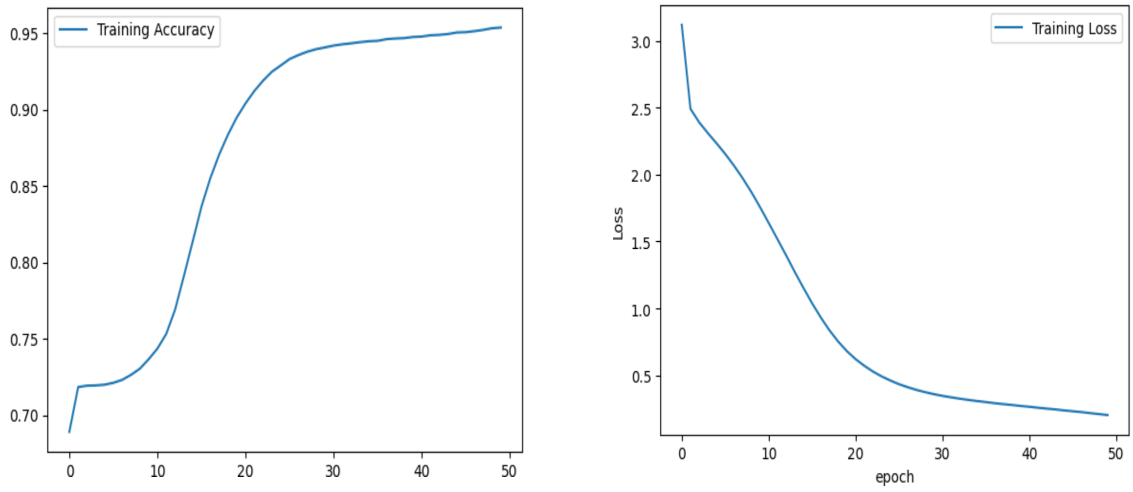


Figure 6.1: Accuracy and Loss

Experiment 2: LSTM with Attention

In this trial, we added attention mechanisms to shorten the LSTM model's training time and the effectiveness of the model with the same data size and hyperparameters. The same dataset and hyperparameters were used in this experiment as in the prior one. The training times that the analysis compares with the LSTM model training time with attention to the training time without attention in the prior experiment. Attention layer has been allowed the model to capturing the context and forget the ignore word**.** For this reason, encoder-decoders without attention mechanisms cannot handle large numbers of sentences. The other outcome of attention methods can reduce the amount of training time needed for the model to finish. And the effectiveness of the LSTM model with attention includes an accuracy of 98%, a loss of 0.11 from Yemsa to Amharic and the impact of attention on the metrics compared to previous experiments below in the figure. We see better accuracy, loss and training results compared to previous experiments. In this phase, the BLEU score from Yemsa to Amharic has taken 420 seconds, and from Amharic to Yemsa, it has taken 404 seconds, and the BLEU score has obtained 4.742 and 7.82.

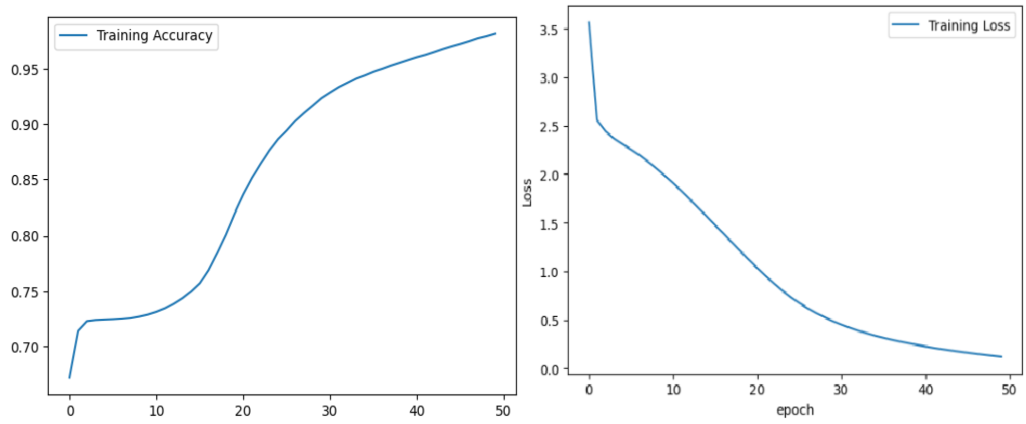


Figure 6.2: Snapshot of LSTM with Attention model Accuracy and Loss

Experiment 3: BI_LSTM Model

Our third experiment was done using bi-directional LSTM. It contains double LSTM cells on the encoder side, which take up a lot more memory than LSTM. Here we have used the same parameters as we did with LSTM. We obtained accuracy results of 98.5% and 98.89% and loss values of 0.18 and 0.14 from Yemsa to Amharic and Amharic to Yemsa. It took 490 seconds and 465 seconds to get BLEU 6.9 and 7.25 from Yemsa to Amharic and Amharic to Yemsa, respectively.

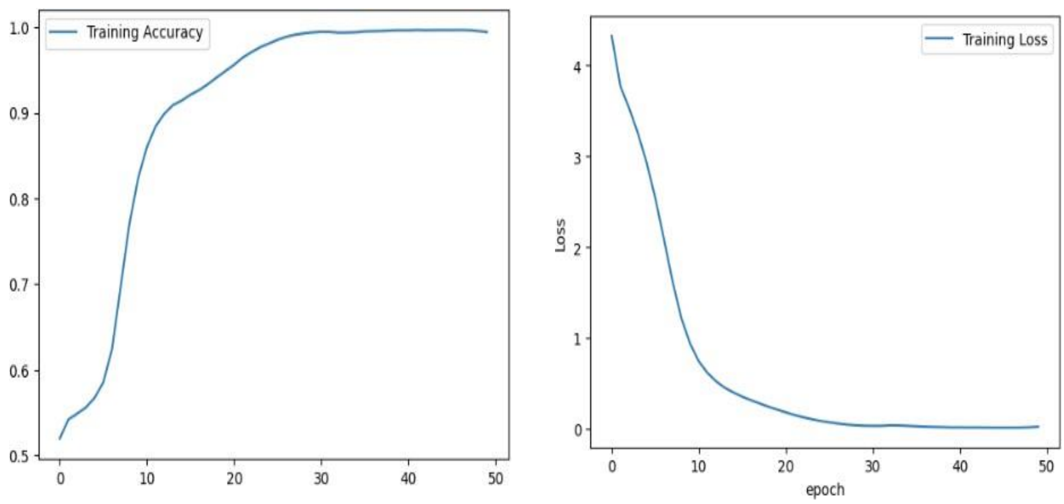


Figure 6.3: Snapshot of BI_LSTM model Accuracy and Loss

Experiment 4: GRU Model with attention

In this phase also, we conducted the GRU recurrent neural network model with attention, and we see the result as shown below in the figure. In this case, we used a similar number to others in general; the performance of a GRU model can be affected by several hyper parameters when changing, including number of epochs, batch size, number of layers, and learning rate. Typically, training the model for more epochs can improve its performance as it allows the model to learn from more. In the GRU model, we analyze the result of varying the number of epochs on the mode and explore how increasing the number of epochs influences the model's ability to learn from the training dataset, improve its performance, and decrease the value of loss.

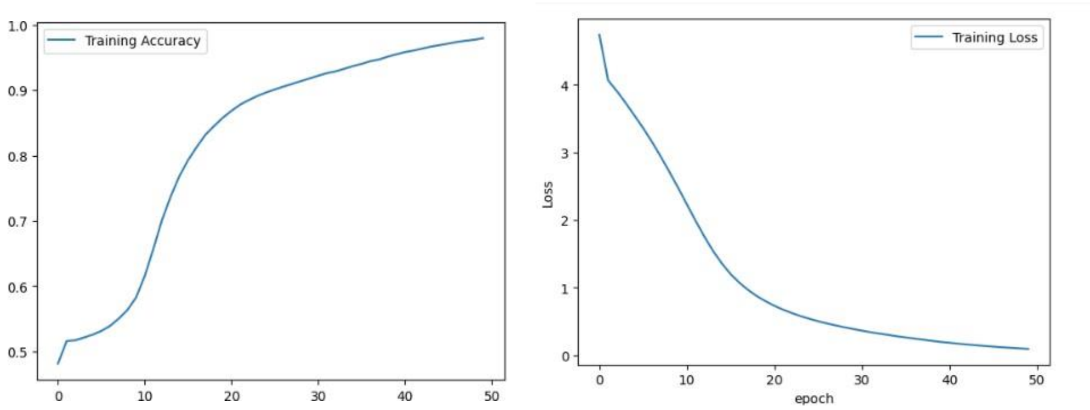


Figure 6.4: Snapshot of GRU model Accuracy and Loss

Experiment 5: Transformer Model

Our transformer model does not have an additional attention mechanism. Here we have fed both token embedding and position embedding vectors to our transformer model. because the transformer model accepts all inputs in parallel. We have used a single number of layers in our transformer model. Like the LSTM and LSTM + Attention models, our transformer models use only the last encoder context vector passed to the decoder directly without additional attention mechanisms because the transformer model has self-owned attention. The encoder-transformer model has feed-forward network and self-multi-head attention. The decoder model has masked multi-head focused and a feed-forward NN. And we have used the RLU feed-forward network in both the encoder and decoder models. The result that is learned from the data is shown below in figure.

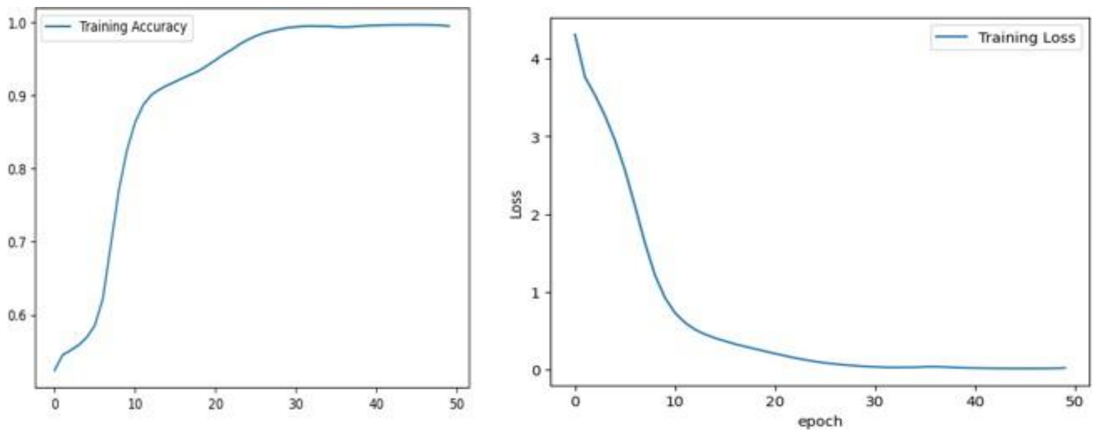


Figure 6.5: Training Accuracy and Loss for Transformer

As shown in the above charts, the embedding dimension batch size combination is determined by using the different epoch numbers. The number head for our transformer is 4 and number hidden layers are 2. To obtain the minimum loss and level, batch sizes 32 and 64, embedding dimension 256, and the number learning rate of 0.01 give us the smallest loss level value.

6.3. Changing Training and Testing Dataset Ratio

The result that is obtained from the proposed model experiment by using a different ratio for the training and testing split is presented in the following table by using the accuracy metrics and loss in the form of percentages for the train and test data.

Table 6.1: Result of an experiment for training and testing dataset ratio

Training	***Test***	***Accuracy***	***Loss***
70%	30%	97.0%	0.0587
75%	25%	98.8%	0.0446
80%	20%	99.4%	0.0468
90%	10%	97.9%	0.0939

Based on the above table 6.1, we see that the proposed model is giving a favorable result with different ways of splitting training and testing dataset ratios. From those experiments, using 80% for training and 20% for testing has a better performance than the others.

Table 6.2: Accuracy, time and Loss when using latent and embedding dimension

Embedding Dimension	***Latent Dimension***	***Training Loss***	***Training Accuracy***	***Time(sec)***
128	128	1.5000	0.740	253sec
256		1.3440	0.778	337sec
128	256	0.6000	0.895	153sec
256		0.1260	0.982	350sec
128	512	0.0190	0.976	317sec
256		0.1158	0.987	247sec
128	**1024**	**0.0113**	**0.995**	**240sec**
256		0.0432	0.993	258sec

From the above table, we have obtained the best fit for embedding and latent dimension using 128 and 1024, respectively, for our proposed model. Therefore, training accuracy is high when compared to other selected methods, training time is short, and training loss is minimal, as seen in the above figure. But the accuracy rate of latent dimension 1024 and embedding dimension 256 is relatively similar. The next step is choosing the dropout rate and learning rate. We used the above best experiments as input, including drop rates of 0.1, 0.2, and 0.5, learning rates of 0.1, 0.01, 0.001 and batch sizes of 32 using 50 epochs based on the related works [[1], 17]. As we haveseen in Table 6.2. The others main important aspects are dropout and learning rates. First, we selected a 0.1 learning rate and dropout; we got a 0.0468 losses and accuracy of 99.4% at training time. Second learning rate 0.001, dropout rate 0.2, embedding dimension 128, latent dimension 1024, and batch size 32 with 50 epochs, the result of loss is 0.0113, which is the minim value that has been shown in figure below.

Figure 6.6: Learning Rate and Loss Level for Training

Overall, to summarize the above experimental results for this study used different models in translating from Yemsa to Amharic and vice versa, evaluated by using the BLEU score metric. Firstly, in Experiment 1 an LSTM model was utilized, achieving a BLEU score of 4.37 for Yemsa to Amharic translation and 5.27 for Amharic to Yemsa translation. Secondly, experiment 2 introduced an LSTM model with attention mechanisms, resulting in improved translation quality. It obtained a BLEU score of 4.48 for the Yemsa to Amharic translation and a significantly higher score of 8.877 for the Amharic to Yemsa translation.

Thirdly, experiment 3 incorporated a bidirectional LSTM model, which further enhanced translation performance. It achieved a higher BLEU score of 6.9 for the Yemsa to Amharic translation and 7.25 for the Amharic to Yemsa translation. Fourthly, experiment 4 incorporated a GRU model, yielding a BLEU score of 4.742 for Yemsa to Amharic translation and 7.82 for Amharic to Yemsa translation. Lastly, experiment 5 introduced a Transformer model, which demonstrated the highest translation quality among all the experiments. It achieved an impressive BLEU score of 9.788 for the Yemsa to Amharic translation and an equally high score of 9.8 for the Amharic to Yemsa translation.

Overall, the results indicate that models with attention mechanisms and bidirectional structures (Experiments 2 and 3) outperformed the basic LSTM model (Experiment 1) in both translation directions. The GRU model also showed competitive performance. Notably, the Transformer model exhibited the highest translation quality, emphasizing its effectiveness in capturing complex language patterns.

Table 6.3: Experimental result of the proposed model

Experiment	***Model***	***BLEU score***	
		Yemsa _Amharic	***Amharic- Yemsa***
Experiment 1	LSTM	4.370	5.270
Experiment 2	LSTM with attention	4.480	8.877
Experiment 3	Bi-LSTM	6.900	7.250
Experiment 4	GRU	4.742	7.820
Experiment 5	Transformer	**9.788**	**9.80**

In analyzing the BLEU scores obtained from the experiments conducted based on the above table. The results of experiments highlight a clear increase in the BLEU scores as we move from the simpler models (LSTM, GRU) to the more complex ones (Bi-LSTM, LSTM with attention), culminating in the highest score with the Transformer model. The difference in BLEU scores clearly indicates the superiority of the Transformer architecture for Yemsa to Amharic machine translation model. The Transformer model showed a considerable improvement compared to the other models in terms of the BLEU scores. In Experiment 5, the Transformer model achieved a BLEU score of 9.788, which is notably higher than the scores obtained by the other models. When comparing it specifically to Experiment 3 using Bi-LSTM, the BLEU score of the Transformer model increased by approximately 2.888 points. Transformer model is mainly to reduce the number of trainable parameters, memory demand, as well as training time. The reason that the LSTM and GRU were lower was because they used a one-time step in the training.

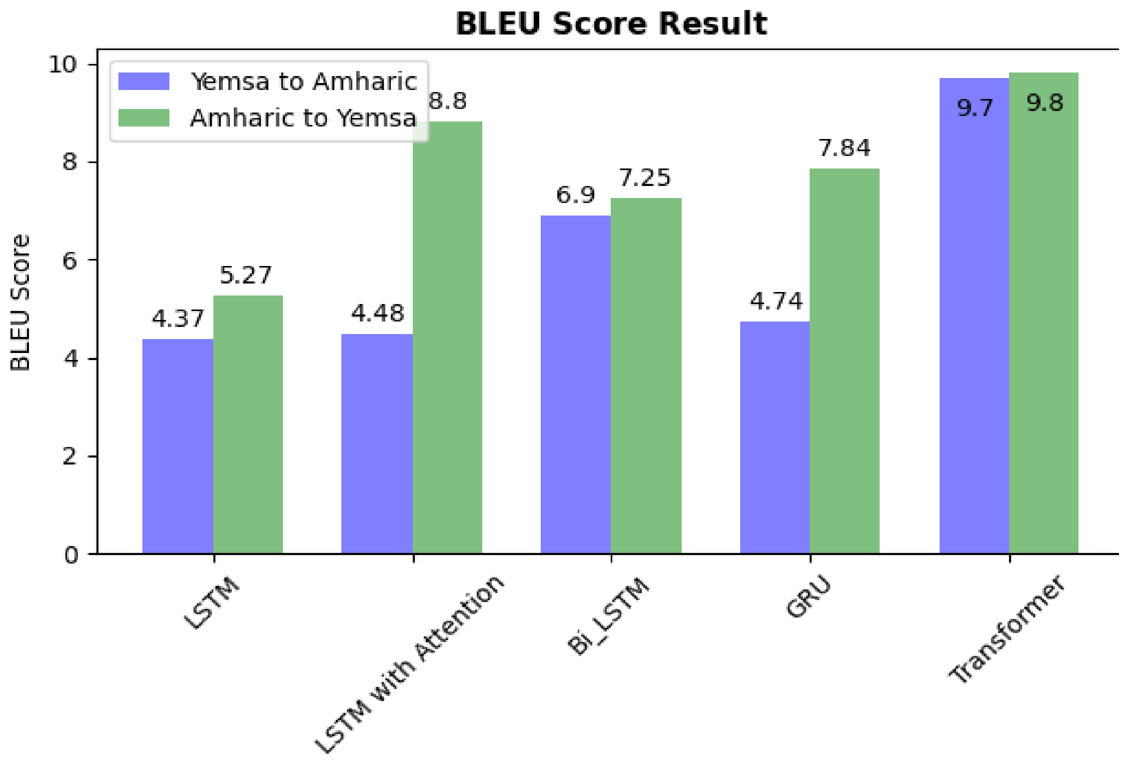


Figure 6.7: BLEU score Experimental result for conducted model

6.4. Linguist Evaluation

As we done in the experiment (Section 6.2), we used to measure BLEU score evaluation metric to evaluate the result that translated by the training model. Besides this automatic evaluation, we our result compared with manual translation are also applied in order to ensure whether the proposed model is acceptable or not. Most of our dataset is collected from religious books, so we have selected people who know about the bible in order to evaluate the machine translated sentences quality. Because the concept of grammar in the Bible is different from regular Yemsa text grammar, below we have shown the sample of translation result in our proposed model. Hence, we asked three Yemsa language experts 1 from culture and tourism office, 1 from our office and 1 from education office) selected purposively, to evaluate the translated Yemsa and Amharic texts and its ‘vice versa by proposed model. These scholars were selected purposively because they know Amharic and Yemsa languages very well and sufficient knowledge in conceptual for Yemsa and Amharic language. Those were evaluating results, and corrective actions. Below we have shown the sample of translation result in our proposed model.

Table 6.4: Sample Yemsa-Amharic translation result

input_sentence	***Actual target sentence***	***predicted_sentence***
Betelem hamfaat koonte naasin gasasutiwa	'ወደ', 'ቤተ', 'ልሔምም', 'እነርሱን', 'ሰድዶ	ወደ በቴ ክርስቲያን እነረሱን ስደዱ
Kiristoosnintugga'efees barwa; Abaasinnawa Naasinna mormefewa.	'አብን', 'ወልድን' , 'የሚክድ,' 'ይህ','የጌታ' ,'ተቃዋሚ', 'ነው'	አብንና ወልድን የሚክድ ይህ የክርስቶስ ተቃዋሚ ነው
Esus wolgire Nitto nibaasi chimaktu es ajajiison tichiwa.	ኢየሱስም','መልሰ','እንዲህ', 'አላቸው።', 'ስለ', 'ልባችሁ', 'ጥንካሬ','ይህችን','ትእዛዝ', 'ጻፈላችሁ'	ኢየሱስም መልሰ የማታምን ጠማማ ትውልድ ሆይእስከ መቼ ከእናንተ ጋር እኖራለሁ
Eelsabet Maaramini nagaason odefana kabaasik katbaassi di nawaas fille	'ኤልሳቤጥም', 'የእናቱን', 'ሰላምታ', 'ስምታ', 'ህፃኑም', 'በማኅፀንዋ', 'ውስጥ', 'ዘለለ'	ኤልሳቤጥም የማርያምን ሰላምታ በሰማች ጊዜ ፅንሱ በማኅፀንዋ ውስጥ ዘለለ
Asu shootunoron asu zagifenawza ees asusin	'በሰራችሁትን', 'ክፍል' ,'ስራ', 'በተወካያቹ' , 'አማካኝነት' , 'በክፍል', 'ውስጥ', 'አሰዩ'	የደረሳችሁበትን ሀሳብ በተወካያችሁ አማካኝነት ለክፍል ጓደኞቻችሁ በንግግር አቅርቡ

The above table sample translation result from Yemsa to Amharic translated sentence which is Actual sentence and predicted sentence that by evaluators, it contains input sentence, Actual target sentence and predicted sentence. As suggestion of evaluators the words that written in different spelling but they have the same meaning in concept manual evaluation method is better. But manual evaluation, depend on the levels of the evaluator knowledge. It can be challenging to achieve consistent results for a given translated sentence when using different evaluators.

Table 6.5: Sample of Amharic_Yemsa machine translation by proposed model

input_sentence	***Actual target sentence***	***predicted sentence***
እንደሚባለው ባህላዊ አለባበሶቻችንን ለተተኪው ትውልድ ለማስተዋወቅና ለማስተላለፍ ተገቢ ነው	'Kaamasi', 'kes', 'newa', 'keronebur'usbaase', 'wa', 'kuulbaase', 'oodde', 'fatasu	assinynyano inno mamsuni chowaason zuutira zagutak sholefeniwa yisete
እንዲሁም የሰው ልጅ ሊያገለግል ነፍሱንም ለብዙዎች ቤዛ ሊሰጥ እንጂ እንዲያገለግሉት አልመጣም።	Asuni', 'Naa', 'dey', 'oomiisik', 'wostonaknawa', 'showoni', 'meyanon', 'fatuk', 'kaabaason'	maler sinna boor'a ke'e ke'e baassin keph
ስለዚህ በሰንበት መልካም መሥራት ተፈቅዶአል አላቸው።	'Es', 'bari', ''boor'a'', 'beysani', 'wonaasik', ''ma'a'', 'wuza', 'zagu', 'aane', 'kaltowawa'	abantiis taatobaason nittok imanak ma'a safaraba sinna boor'a
ችሎታን ለማሳደግ የሚጠቅሙ ተግባራትን ዘርዝሩ	'tinera', 'wollesefat', 'imte', 'kaamasikitoni.'	biyyeset kaan kashen kaasos oombe meer
ነገር ግን ወራት ይመጣል ሙሽራውም ከእነርሱ ሲወሰድ ያንጊዜ በዚያ ወራት ይጦማሉ አላቸው	'Sinuntano', 'asu', 'baassostan', 'wuma', 'wona', 'yoona', 'esbarik', 'soomraa', 'shiphosonewa'	Sinuntano mari baassostan oor kesana wona yoona. Es kabaasik soomere shiiphosonewa

6.5. Discussion on the Result of the Study

In order to achieve this thesis goal of designing and implementing the Yemsa to Amharic machine translation and vice versa, different preprocessing steps were done. After that, encoder-decoder architecture was built with attention or without attention using LSTM, GRU and transformers. We obtained different result on each After reading all of the input sentences, the encoder reads all of the data into a single real-valued vector, which is then sent to the decoder, which uses the vector to produce the target translation. When increasing the length of a sentence the interdependency between the words at the start and sentences finish becomes somewhat loosely related. This architecture works quite well for short sentences, and this may make it difficult for the recurrent neural networks to apply to long and complex sentences. The context within a sentence is a result of the inter-dependency of nearby words in the given sequence. To identify a word by a specific index at the moment it is encountered in the data, each word in the sentence must be given a new identity number as it is visited. However, as the size of the dictionary rises as well as, so does the number of words used to represent them and the size of the word vector that is required becomes higher.

The use of an attention mechanism with a fundamental Encoder-Decoder architecture resolves these two basic problems. Based on the result of the experiment on LSTM with attention, the BLEU score shows the attention-based approach is better than the LSTM without attention for Yemsa-Amharic and Amharic-Yemsa MT models. Based on our experimental findings, better outcome is obtained Transformer model is used instead of LSTM. However, it is difficult to obtain a translation model that is more effective given the short amount of data employed in this study. Even with a small dataset, the highest possible BLEU score of 9.8 and accuracy of 99% is recorded by adjusting different hyperparameter while experimenting with this small corpus. From our experimental finding we have seen that deep learning has become a successful MT technique as a result of increased processing capacity. Sentences can be translated by built model with remarkable accuracy using the Encoder-Decoder architecture. Finally, consider all the experiments. A better BLEU score is achieved or documented when Amharic is used as the source language and Yemsa as the target language. This is because alignment quality is better when Amharic is used as the source language.

6.6. Summary

According to the results of the studies, the Transformer model performs better than LSTM, LSTM with attention, Bi-LSTM and GRU models in terms of accuracy in BLEU score and training time. The transformer model also shows better performance than other models selected in these studies when translating more complex sentences. In both directions of translation, the transformer model has a faster training time than the other models that were used in the experiment. In general, our investigations confirmed that our transformer model provided good translation accuracy in both directions between the language pairs.

6.7. Answering Research Questions

At the beginning of this work, we have formulated three research questions to be answered after the experiment; hence, here is their answer with the findings of the study.

Q1: How to develop a parallel corpus for Yemsa to Amharic Machine translation?

Based on the research methodology, we followed the criteria to prepare the corpus in several ways: first, by distributing letters to selected organizations; second, by asking language experts at the central Ethiopia region at Yem Zone education office; At the third stepIdentify Yemsa and Amharic native speakers: Find individuals who are fluent in both Yemsa and Amharic languages. These individuals will be crucial in providing translations and validating the quality of the corpus. Gather existing document: this can include books, articles, documents and others for Amharic and Yemsa. Choose a variety of texts that cover different domains, such as bibles, lower grade text books, proven. Align the source sentences in Yemsa with their corresponding translations in Amharic. This step ensures that each sentence pair is correctly aligned, allowing the machine translation model to learn the mapping between the two languages. Cleaning and Preprocessing: Clean the corpus by removing any irrelevant or noisy data, such as duplicates, formatting issues, or sentences with low quality translations. Preprocess the text by tokenizing, normalizing, and applying any necessary linguistic-specific preprocessing steps.

Q2: Which vectorization techniques are used in Yemsa to Amharic machine translation?

The techniques that were used in this study were one hot vector, word embedding such as Glove, word2vec using Skip gram, and CBOW vectorization techniques. Each of the techniques vary in the result obtain. Word embedding plays a pivotal role in machine translation models, revolutionizing the way we approach language understanding and enabling more accurate and context-aware translations. By leveraging the power of deep learning and neural networks, word embedding allows us to represent words as dense vectors in a high-dimensional space. This representation captures rich relationships between words in terms of semantics and syntax, enabling us to extract meaning and context from text with astonishing accuracy. Word2Vec and Glove are two groundbreaking word embedding techniques, have had a transformative impact on machine translation, reshaping our understanding and analysis of language in this domain. In word2vec there is Skip gram and CBOW Architectures.

In our studies focused on Yemsa to Amharic machine translation, we have discovered that the Continuous Bag-of-Words (CBOW) architecture of Word2Vec consistently outperforms other approaches. CBOW and Skip-gram are two word2vec variations of Word2Vec that provide different strategies for learning word embedding. CBOW, as its name suggests, aims to predict a target word based on the context words surrounding it. In the context of Yemsa to Amharic translation, CBOW treats the Yemsa context words as inputs and the target Amharic token as the output. By optimizing the prediction of the Amharic target word given the Yemsa context, the CBOW model learns to generate word embedding specific to the translation task.

The CBOW architecture has shown significant advantages in our studies for Yemsa to Amharic translation. It performs well when the local context of Yemsa words carries crucial information for accurate translation. CBOW effectively capture collocations, idiomatic expressions, and other local word relationships that are vital for maintaining context and fluency in the translation process. Additionally, CBOW focus on local context aligns well with the goal of Yemsa to Amharic machine translation and vice versa, which aims to generate translations explain the point clearly the meaning and context of the source language. By considering the immediate Yemsa context, CBOW embedding can capture the nuances of word usage and disambiguate words more effectively, leading to more accurate and context-aware translations in Amharic.

It's important to note that the choice between CBOW and Skip-gram may depend on the characteristics of the Yemsa to Amharic translation dataset. However, based on our studies, the CBOW architecture consistently delivers superior results for this specific translation task. Its ability to capture local context, handle rare words, and generate accurate translations makes it an excellent choice for enhancing the quality and fluency of Yemsa to Amharic machine translation.

Q3: Which deep learning model is better from Yemsa to Amharic machine translation? Based on section 6 Experiment works were discussed different experiments, LSTM, LSTM with attention, Bi-LSTM, GRU, Transformer applied for the proposed studies. In order to answer this question regarding the best deep learning model for Yemsa to Amharic machine translation, an experimental investigation was conducted using several popular models, including LSTM, LSTM with Attention, Bi-LSTM, GRU and Transformer. The way in which these models operate was evaluated based on the BLEU score, a commonly used metric for MT evaluation. Interestingly, the results revealed that the Transformer model outperformed the other models, achieving a significantly higher BLEU score. This indicates that the Transformer model is particularly well-suited for Yemsa to Amharic MT, suggesting its superiority over the other deep learning architectures in this specific context.

The high BLEU score obtained by the Transformer model can be attributed to its unique architecture and attention mechanism. Unlike recurrent neural network models such as LSTM, GRU and Bi-LSTM, the Transformer model relies on self-attention to capture long-range dependencies and build context-aware representations. This attention mechanism allows the model to focus on appropriate input sequence segments, enabling it to effectively capture and translate the syntactic and semantic structures present in Yemsa and Amharic languages. Additionally, the Transformer's parallel processing capability contributes to its efficiency, making it well-suited for handling the complexities of MT tasks. In the investigation comparing all models for Yemsa to Amharic MT, the Transformer model achieved a significantly higher BLEU score, indicating its superiority.

7. CHAPTER SEVEN: CONCLUSION AND RECOMMENDATION

7.1. Overview

This chapter finalizes the whole work and gives a general conclusion about the study contribution and findings of the work and suggests some feature works for the upcoming researchers.

7.2. Conclusion

We have seen a broad concept of machine translation and its approach, how the text is translated from linguistically speaking, since first chapter. The design and implementation of an Yemsa to Amharic MT model and vice versa is the main objective of this researchwe have used types of RNN models and Transformer model for translation purpose, this thesis work is done on a scarce resource language pair with a data set of 6,000 parallel corpuses collected from data sources such as religion books consisting Saint mathiwos, Saint Markos, Saint Lukas and saint Yohannes wengel and another educational domain from Grade 1-6 student textbook. Five major experiments were conducted and results were recorded for all translations with identification of loss and accuracy measured. We have conducted experiments using LSTM, LSTM with attention, Bi-LSTM, GRU and Transformer model.

Generally, when we compare automatic evaluations, the Transformer model achieves a higher result than the other models. Particularly, it improved the translation quality by 1.96 BLEU scores from the LSTM with attention model at the evaluation performance when we conducted an experiment on the transformers model. Lastly, when we compare the time efficiency of both directions of translation for the transformer model, it uses a higher time efficiency (shorter time) on translation from Amharic to Yemsa than Yemsa to Amharic translation. The reason for the longer time for translation from Yemsa to Amharic is that Yemsa has used a higher number of vocabulary words, which must pass through the encoder during the training period of Yemsa to Amharic translation. The state-of-the-art BLEU score achieved in the study was 99.4% accuracy with a loss of 0.0113, and BLEU scores of 9.7 and 9.8 from Yemsa to Amharic and Amharic to Yemsa translation respectively.

7.3. Research Output

Over the past decade, the performance of machine translation models has experienced significant advancements, with noticeable year-to-year improvements driven by the increasing size of available datasets and faster computing capabilities. Additionally, there are ever more translation systems being developed in both the commercial and educational sectors. We tried to find that deep learning models are good at modeling machine translation algorithms for under-resourced languages like Yemsa and Amharic. We contributed 6,000 datasets from the New Testament Bible and other educational materials from low grades. Our parallel corpus can be used as an input for other machine translation research and natural language processing applications. So, in order to identify the best approach, algorithms and modify some parts that of the trained model, we conducted the experiment and preparing a parallel corpus is an important point. We think this study, which is used as a reference for the next researcher and their user, has several values for the local language Yemsa.

7.4. Future Work

Based on the discussion of the results of the study, we recommend the following as future research directions: Since our Yemsa-Amharic MT is the first study. We have not previously prepared parallel corpora like Wordnet for Yemsa and Amharic language pairs. Our model needs a standardized parallel corpus. So, we have collected and prepared a standardized 6,000-parallel corpus

The errors occurred because there was no prepared algorithm for Yemsa language stemming and stop removal, and the error occurred because the model was performed for shorter sentences rather than long sentences. For longer sentences, the interdependence of words decreases compared with shorter sentences, resulting in an error. There is also apprediction error due to contextually related meaning. Due to a lack of training data, our model did not learn the language as we expected. In these few training data cases, the model was wrongly translated in some cases. Due to this error, our results becamelow.

- We suggest that further studies include a larger number of datasets by collecting or using different domain areas to improve translation quality.
- For future we recommend any interested use the prepared data set can be used for other NLP applications tasks like stemming, part of language tagging or similar activities.
- Recommend to increase the effectiveness of the deep learning model by adding longformer and TinyBERT with transformer model.
- As the goal of our study is to implement machine translation only for text-to-text translation, we suggest that additional studies conduct speech-to-text and vice versa machine translation between these language pairs.
- Also, we suggest that additional studies include an increased number of datasets to further increase the translation quality and also test the systems on GPU-based computers to further minimize the required training time of the systems.
- Finally, we suggest that the implemented system be used to investigate machine translation between other language pairs.

REFERENCES

[1] Workineh Wogaso Gaga, 2020. Attention-based Amharic-to-Wolaita Neural MachineTranslation, MSc Thesis, Addis Ababa University, Addis Ababa, Ethiopia.

[2] Niguse Zebene, 2018. Fereta books, All in one printing, Saja Town, Central Ethiopia Region,Ethiopia.

[3] S. Dereje, 2019. Hybrid Artificial Neural Machine Translation using Deep Learning Techniques English-to-Afaan Oromo, MSc Thesis, Addis Abeba science and Technology university, Addis Ababa, Ethiopia.

[4] Xv, A., & Chao, W, 2020. Russian-English Bidirectional Machine Translation System. Beihang University, Beijing 100191, China.

[5] The world Atlas of languages UNESCOWAL. https://en.wal.unesco.org/world-atlas-languages

[6] Demeke Gizachew, 2018. A sketch of Yemsa Grammar,1st ed, Yem Zone Culture and Tourismdepartment, Saja Town, Central Ethiopia Region, Ethiopia.

[7] Temesgen and Misganna., 2018. Yemsani Wolla. Yem zone Culture and Tourism department,Saja Town, Central Ethiopia Region, Ethiopia, 150 p.

[8] Misganna Kebede, 2020. Yemsani Wolla, 1st ed Central Ethiopia region and Yem Zone culture and tourism department, Saja Town, Central Ethiopia Region, Ethiopia, 156p.

[9] Bosen, G., & Anteneh, G. 2017. Tonality of Yemsa as a Cause for Linguistic Polysemy. InInternational Journal of Social Sciences Arts and Humanities.

[10] Aklilu Yilma Ralph Siebert Kati Siebert, 2002. Sociolinguistic Survey of the Omotic Languages Sheko and Yem. SIL International, 1(2): 18–26.Mequanint munye, 2020.

[11] Amharic English Bilingual web search Engine. LAP Lambert Academic Publishing, Theodor-Heuss-Ring 128p.

[12] Tadesse Kassa, 2018. Morpheme-Based Bi-directional Ge‟ez-Amharic Machine Translation,MSc Thesis. http://etd.aau.edu.et/handle/123456789/18688.

[13] Constitution of the Federal Democratic Republic of Ethiopia PREAMBLE, 1995. FederalDemocratic Republic of Ethiopia.

[14] Thamme, Chris, M T.,M,2021. Many-to-English Machine Translation Tools and data.

[15] Genet, W. (2021). Ge ez-Amharic machine translation using deep learning, thesis, bahir dar university, bahir dar, Ethiopia.

[16] Mengistu, Kinfe, 2022. Amharic-kistanigna Bi-directional machine translation using deeplearning., MSc Thesis, Bahidar university, Bahidar, Ethiopia.

[17] Agerie Belete, 2022. Bi-directional English-Awngi Machine Translation Using Deep learning. MSc Thesis, Bahirdar University, Bahirdar, Ethiopia.

[18] Aklilu Yilma Ralph Siebert Kati Siebert, 2002. Sociolinguistic Survey of the Omotic Languages Sheko and Yem. SIL International, 1(2): 18–26.

[19] Alex Graves, M. L. S. F. R. B. H. B. ´ & J. S, 2000. A novel connectionist system for unconstrained handwriting recognition. IEEE, 855p.

[20] An Intuitive Understanding of Word Embeddings. From Count Vectors to Word2Vec. Retrievedfrom Analytics Vidhya. embeddings-count-word2veec/ Last Accessed: June

[21] An, P., Wang, Z., & Zhang, C, 2022. Ensemble unsupervised autoencoders and Gaussian mixture model for cyberattack detection. Information Processing and Management, 59(2). https://doi.org/10.1016/j.ipm.2021.102844

[22] Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y, 2014. On the Properties of Neuralmachine translation: Encoder-Decoder Approaches.

[23] Pretrained Models. International Joint Conference on Natural Language Processing, pages 306–316, August 1st

[24] Goyal, P., Pandey, S., & Jain, K, 2020. Deep Learning for Natural Language Processing. In Deep Learning for Natural Language Processing. Apress.

[25] Eyassu, S., & Gambäck, B, 2005. Classifying Amharic News Text Using Self-OrganizingMaps. www.unicode.org/charts/PDF/U1200.pdf.

[26] Tadesse Kassa, 2018. Morpheme-Based Bi-directional Ge‟ez-Amharic Machine Translation,MSc Thesis. http://etd.aau.edu.et/handle/123456789/18688

[27] Zhang, S., Zheng, D., Hu, X., & Yang, M, 2021. Bidirectional Long Short-Term Memory Networks for Relation Classification, School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China, pp 75-80.

[28] Leah Marie Hamilton, Jacob Lahne, 2023. Natural language processing. Science.

[29] Li, G., Reagan, K., Saxen Varde, A. S., & Wilde, B, 2022. A Framework for German-EnglishMachine Translation with GRU RNN.

[30] M. T. Bame, 2020. Transformer Based Amharic Headline Generation Using Sub-word2Vec Representation, A Thesis Submitted to the School of computing in Partial Fulfilment for the Master of Science in Information Technology, Jimma University, Jimma, v.18(5), pp 371-410.

[31] Maren M., Lukas B., Christoph L.,2021. Early Stopping without a Validation Set, BernsteinCenter for Computational Neuroscience, 6 Jun 2021.

[32] Moses, K, 2020. Encoder-Decoder Seq2Seq Models, Clearly Explained!! A step-

by-stepguide to understanding Encoder-Decoder Sequence-to-Sequence models.
[33] Demeke Gizachew, 2018. A sketch of Yemsa Grammar,1st ed, Yem Zone Culture and Tourismdepartment, Saja Town, Central Ethiopia Region, Ethiopia
[34] Hockett, 2020. The morph as a minimal linguistic form. Morphology,30(2), 117–134.
[35] Fekede, 2018.Derivational morphology;https://sontario.pressbooks.pub/essentials /chapter/5-5-derivational-morphology
[36] Rorabaugh, A. K., Caíno-Lores, S., Johnston, T., & Taufer, M, 2022, High frequency accuracy and loss data of random neural networks trained on image datasets
[37] Kingma, Diederik P., and Jimmy Lei Ba. 2015. "A : A m s O." 1–15.
[38] P. Goyal, S. Pandey, and K.2020. Jain, Deep Learning for Natural Language Processing.Apress, https://doi: 10.1007/978-1-4842-3685-7.
[39] Tesfaye, S, 2020. A Hybrid approach for Machine Translation from Ge‟ez to Amhariclanguage, MSc Thesis, St. Mary‟s University, Addia Ababa, Ethiopia.
[40] Wolf, L. Reference Grammar of Amharic, 1st ed.; Otto Harrassowitz: Wiesbaden, Germany,2000. [Google Scholar].
[41] Takahash, E.G and D.P, 2022. Japanese-to-English Machine Translation Using RecurrentNeural Networks, Stanford University, 2-6
[42] Worku Keleme work, 2013. Automatic Amharic text news classification: Aneural networksapproach, Ethiop. J. Sci. & Technol. 6(2) 127-137, 2013.
[43] Britannica, 2012. Ethnic groups and languages.
[44] Thamme, Chris, M T., M,2021. Many-to-English Machine Translation Tools and data.
[45] Pretrained Models. International Joint Conference on Natural Language Processing, pages 306–316, August 1st
[46] Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J, 2019. Proceedings of the 40th AnnualMeetin Association for Computational Linguistics.
[47] Yasrab, R. (2018). ECRU: An encoder-decoder based recurrent neural network (RNN) for road-scene understanding. Journal of Imaging, 4(10).
[48] Hutchins, W. J. (1995). Machine Translation: A Brief History. Concise History of the Language Sciences, 431–445
[49] Baye Emam, 2003. የአማርኛ ሰዋስው. 2nd ed. Addis Ababa, 365p
[50] Baye Yimam, 1999. Root reduction and extensions in Amharic. Ethiopian Journal of Language and Literature, 31-55.

[51] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. http://arxiv.org/abs/1412.3555

[52] Getaun Amhare, 2010. ዘመናዊ የአማርኛ ሰዋስው በቀላል አቀራረብ

[53] Ass.prorf GEtahun Amare, "የአማርኛ ሰዋስው በቀላል አቀራረብ", 1989.

[54] Tigist Tensou Tessema. 2014, "Word Sequence Prediction for Amharic Language",Master‟sthesis, Addis Ababa University, Ethiopia,

[55] Amdework Asefa, 2021. A Bi-Directional Ge‟ez-Amharic Neural Machine Translation: A Deep Learning Approach. MSc Thesis, Debre Birhan University, Debre Birhan, Ethiopia.

[56] Takahashi S, Wada H, Tadenuma R, and Watanabe S.,2003 "English-Japanese MT", Readings in MT.

[57] Andrabi, S. A. B., & Wahid, A, 2022. Machine Translation System Using Deep Learning forEnglish to Urdu. Computational Intelligence and Neuroscience, 2022.

[58] Tian, T., Song, C., Ting, J., & Huang, H, 2021. A French-to-English Machine Translation Model Using Transformer Network. Procedia

[59] Arfaso, B, 2019. Bi-Directional English-Afan Oromo Machine Translation Using Convolutional Neural Network. MSc, Thesis, Addis Ababa University, Addis ababa.

[60] Elias Asefa, 2021. developing English to Dawurootsuwa machine translation model usingRNN. MSc Thesis, Addis Ababa university, Addis Ababa Ethiopia.

[61] Ashengo, Y. A., Aga, R. T., & Abebe, S. L, 2021. Context based machine translation with recurrent neural network for English–Amharic. 35(1), 19–36.

[62] Mekdes, M, 2023. Attention-based Neural Machine Translation from English-Wolayitta,MSc Thesis, St Mary‟s university, Addis Ababa, Ethiopia.

[63] Steven Bird, E. K. and E. L, 2014. Natural Language Processing with Python. Printed in theUnited States of America; jing, Cambridge, Fa Sebastopol, Taipei, Tokyo, 504p.

[64] Chollet, F, 2019. Deep learning with Python: Towards Data science

[65] Tessema, M, 2018. Design and Implementation of Amharic Search Engine Local WebContent View project Cloud-A Volunteer Computing as a Service.

[66] Zoumana Keita, 2021.Text data representation with one-hot encoding, Tf-Idf, Count vectors,Co-occurrence Vectors and Word2Vec. Towards Data Science, Feb 19, 2021

[67] Yang, S., Wang, Y., & Chu, X. 2020. A Survey of Deep Learning Techniques for Neural Machine Translation. http://arxiv.org/abs/2002.0752

[68] Yonghui Wu, M. S. Z. C. Q. V. L. and M. N, 2016. Google‟s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,

Cornell university.

[69] Gochel, D, 2003. An integrated approach to automatic complex sentence parsing or Amharictext, Addis Ababa University, Addis Ababa, Ethiopia.

[70] Shivkaran Singh, M. Anand Kumar and K.P. Soman, Attention based English to Punjabi neural MT, Journal of Intelligent & Fuzzy Systems, 201

[71] S. RUDER, 2019. algorithm, Sebstian ruder an overview of gradient optimization, Insight Centre for Data Analytics, NUI Galway Aylien Ltd., Dublin,14p.

[72] T. v. F. a. P. v. N, 2019.Machine Learning Crash Course Google developers.

[73] Wilson, J. (2020). What is the effect of learning rate in neural network,Retrieved.

8.APPENDIX

8.1. Appendix A: Sample of Parallel Corpus

Yemsa	*Amharic*
Kushunaasewa wochonaase asin biiti. Taptira ta sinnaason aruti. Taan biifetiisimato kitaas ashnewa megane barki aafawa.	እኔ ራሴ እንደ ሆንሁ እጆቼንና እግሮቼን እዩ፤ በእኔ እንደምታዩት መንፈስ ሥጋና አጥንት የለውምና እኔን ዳስሳችሁ እዩ አላቸው
Es baron makefaat kushubaasewa wochobaase asin baassok besi.	ይህንም ብሎ እጆቹንና እግሮቹን አሳያቸው
Baasso dey giraknawa diinqokna tuumefaat bar sinbaason aane amanosoto. Yesus dey ekka yi: Isa maastoni wuza faaroso? yire mamsi.	እነርሱም ከደስታ የተነሣ ገና ስላላመኑ ሲደነቁ ሳሉ። በዚህ አንዳች የሚበላ አላችሁን? አላቸው
Baasso dey kiichcho kurxummi imete.	እነርሱም ከተጠበሰ ዓሣ አንድ ቁራጭ፥ ከማር ወለላም ሰጡት
Yesus ekka dey yi: Nittoneen ane faana kabaasik Museni tumaassi raajjuni meeni tichaassinawa gaamni tichaassina taak ticche bar zuuttera sholsu yira nittok makenaas haniiswa yi.	እርሱም ከእናንተ ጋር ሳለሁ በሙሴ ሕግና በነቢያት በመዝሙራትም ስለ እኔ የተጻፈው ሁሉ ይፈጸም ዘንድ ይገባል ብዬ የነገርኋችሁ ቃሌ ይህ ነው አላቸው
Es kabaasik korto tichaason yaadatsonek nibbesiison gachi.	በዚያን ጊዜም መጻሕፍትን ያስተውሉ ዘንድ አእምሮአቸውን ከፈተላቸው
Ekka dey yi: Kiristoos shana ephphe toonamatonawa keezsinari wono dey kitun kabunamatona zeemma tichchera faar	እንዲህም አላቸው ክርስቶስ መከራ ይቀበላል በሦስተኛውም ቀን ከሙታን ይነሣል
Esiisimato dey Yerusalemun kabira dara zuuttambaase asik harminewa boor'ni fakine basa sunaasik awastonir. yistera tichche.	በስሙም ንስሐና የኃጢአት ስርየት ከኢየሩሳሌም ጀምሮ በአሕዛብ ሁሉ ይሰበካል ተብሎ እንዲሁ ተጽፎአል

Es barik nitto zaalwa.	እናንተም ለዚህ ምስክሮች ናችሁ
Ta abataas nittok imna abdiison nittok wosuna. Nitto dey denun hugna imtona kabaneen Yerusalem katamaassi footi.	እነሆም አባቴ የሰጠውን ተስፋ እኔ እልክላችኋለሁ፤ እናንተ ግን ከላይ ኃይል እስክትለብሱ ድረስ በኢየሩሳሌም ከተማ ቆዩ
Es seen orfo nawobaasakitonin ephphe Bitaniya hami. Kushubaason kabgire baassotin suusi.	እስከ ቢታንያም አወጣቸው እጆቹንም አንሥቶ ባረካቸው
Suusfet baassostan oor kesse den sama hami.	ሲባርካቸውም ከእነርሱ ተለየ ወደ ሰማይም ዐረገ
Ha'ooson galattet wonna wonna betemeqdesiissi aane tishisefe.	ዘወትርም እግዚአብሔርን እያመሰገኑና እየባረኩ በመቅደስ ኖሩ
Kushunaasewa wochonaase asin biiti. Taptira ta sinnaason aruti. Taan biifetiisimato kitaas ashnewa megane barki aafawa.	እኔ ራሴ እንደ ሆንሁ እጆቼንና እግሮቼን እዩ በእኔ እንደምታዩት መንፈስ ሥጋና አጥንት የለውምና እኔን ዳስሳችሁ እዩ አላቸው
Es baron makefaat kushubaasewa wochobaase asin baassok besi.	ይህንም ብሎ እጆቹንና እግሮቹን አሳያቸው
Dey mangu wosto woste asusakitonin isaasin hoi'taasta, hepsaasin aldaasta suutete	በዚያ እርሱን ክፉ አድራጊዎቹንም አንዱን በቀኝ ሁለተኛውንም በግራ ሰቀሉ
Esseen orfo Yesus Abataaso, zagsefe baron aane arsoto sinna boor'a feeshun uwa yi. Meya taar'oni meya dey maybaasta ixa ha'sere haa'esete.	ኢየሱስም አባት ሆይ የሚያደርጉትን አያውቁምና ይቅር በላቸው አለ ልብሱንም ተካፍለው ዕጣ ተጣጣሉበት
Esiisimato dey Ne Ayhudni taato sinfaatane teetneeson fu'owa yiset feeseter.	አንተስ የአሁድ ንጉሥ ከሆንህ፥ ራስህን አድን እያሉ ይዘብቱበት ነበር
Suuttena mesqeliisi teetta: Haniis Ayhudni taatowa yifa ticha tichire yeetisete.	ይህ የአይሁድ ንጉሥ ነው ተብሎ በግሪክና በሮማይስጥ በዕብራይስጥም ፊደል የተጻፈ ጽሕፈት ደግሞ በእርሱ ላይ ነበረ

8.2. **Appendix B:** Yemsa Language Orthographic

Aa ኣ	**Bc** በ	**Cc** ጬ	**CHch** ቸ	**Dd** ደ
Ee ኤ	**Ff** ፈ	**Gg** ገ	**Hh** ሃ	**Ii** ሃ
Jj ጀ	**Kk** ከ	**Ll** ለ	**Mm** መ	**Nm** ነ
NGng -	**NYny** ኘ	**Oo** ኦ	**Pp** ፐ	**PH** ጰ
Qq ቀ	**Rr** ረ	**R’r’** _	**Ss** ሰ	SHsh ሸ
Tt ጠ	**TSts** ጸ	**Uu** ኡ	Vv ቨ	**Ww** ወ
Xx ጠ	Yy የ	Zz ዘ	ZHzh ዠ	

Yemsa vowels

Aa-ኣ	**Ee-ኤ**	**Ii-ኢ**	**Oo-ኦ**	**Uu-ኡ**

8.3. Appendix C: Amharic Language Orthographic

	Ge'ez ä	*Ka'eb* u	*Salis* ī	*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
ạ	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
p̣	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

8.4. Appendix D: Required Python Libraries

```python
# import the required libraries:
import os, sys
from keras.models import Model
from keras.layers import Input, LSTM, GRU, Dense, Embedding,Bidirectional
from keras.preprocessing.text import Tokenizer
from keras.utils import pad_sequences
from keras.utils import to_categorical
from keras.optimizers import Adam
from tensorflow.keras.utils import plot_model
from keras.backend import learning_phase_scope
from keras.callbacks import EarlyStopping
from nltk.translate.bleu_score import sentence_bleu
import numpy as np
import pickle
import matplotlib.pyplot as plt
```

Appendix-Figure: I Snapshot of python Code for Import all required library

8.5. Appendix E: Sample Code for Normalization

```python
#Normalizing words with Labialized Amharic characters such as ተጥቁዋል or  ተጥቁአል to  ተጧል
input_sentence=re.sub('(ሉ[ዋአ])','ሏ',input_sentence)
input_sentence=re.sub('(ሙ[ዋአ])','ሟ',input_sentence)
input_sentence=re.sub('(ቱ[ዋአ])','ቷ',input_sentence)
input_sentence=re.sub('(ሩ[ዋአ])','ሯ',input_sentence)
input_sentence=re.sub('(ሱ[ዋአ])','ሷ',input_sentence)
input_sentence=re.sub('(ሹ[ዋአ])','ሿ',input_sentence)
input_sentence=re.sub('(ቁ[ዋአ])','ቋ',input_sentence)
input_sentence=re.sub('(ቡ[ዋአ])','ቧ',input_sentence)
input_sentence=re.sub('(ቹ[ዋአ])','ቿ',input_sentence)
input_sentence=re.sub('(ሁ[ዋአ])','ኋ',input_sentence)
input_sentence=re.sub('(ኑ[ዋአ])','ኗ',input_sentence)
input_sentence=re.sub('(ኙ[ዋአ])','ኟ',input_sentence)
input_sentence=re.sub('(ኩ[ዋአ])','ኳ',input_sentence)
input_sentence=re.sub('(ዙ[ዋአ])','ዟ',input_sentence)
input_sentence=re.sub('(ጉ[ዋአ])','ጓ',input_sentence)
input_sentence=re.sub('(ዱ[ዋአ])','ዷ',input_sentence)
input_sentence=re.sub('(ጡ[ዋአ])','ጧ',input_sentence)
input_sentence=re.sub('(ጩ[ዋአ])','ጯ',input_sentence)
input_sentence=re.sub('(ፁ[ዋአ])','ጿ',input_sentence)
input_sentence=re.sub('(ፉ[ዋአ])','ፏ',input_sentence)
```

Appendix-Figure: II Snapshot of python code for normalization

8.6. **Appendix F**: Python Code for BLEU Score for Testing Model

```python
i= 0
count = 0
total_score = 0
from nltk.translate.bleu_score import sentence_bleu
for i in range(1200):
  if i<len(X_test):
     input= X_test[i]
  else:
     continue
  input_seq = encoder_x_sequences[i:i+1]
  translation = translate_sentence(input_seq)
  output=y_test[i]
  output=output.split()
  translated =translation.split()
  blue_score = sentence_bleu(output, translated)
  count = count +1
  total_score = total_score + blue_score
  average_score=total_score/count
  print('Input Language : ', input)
  print('Actual translation : ', output)
  print('Predicted translation : ', translation)
print("Average BLEU score",average_score)
```

Appendix-Figure: III Snapshot code for BLEU Score

8.7. **Appendix G:** Sample output result for proposed model

```
Input Language :  እንዲዘምኑና እንዲሻሻሉ እየተደረገ ከትውልድ ወደ ትውልድ ሊተላለፉ ይገባል።
Actual translation :  ['Wollera', 'maksok', 'kabena', 'barik', 'galatba', 'kushoni.', '<eos>']
Predicted translation :  ek sinnitiista fa ke'eessi hamtiwa estak oonu barista duunoyna hanyani nawa tuptera
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 30ms/step
Input Language :  ሕሊናም መሐመድን ከአባቱ ጋር በዓይን አስተዋወቃቸው።
Actual translation :  ['Han', 'kaptasi', 'ko’ta,', 'arinyasikito', 'zagunak', 'chimana', 'wuzas', '<eos>']
Predicted translation :  bechchubaason dey ka’usik katusonak mamsuni
Average BLEU score 9.842134879001307e-234
```

```
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 32ms/step
Input Language :  እረፍት ሲያደርግ፣ ሲበላና ሲተኛ የዋለው በሬ አህያው ከእርሻ ሲመለስ ያገኘዋል
Actual translation :  ['Han', 'kaptasi', 'ko’ta,', 'arinyas', 'zagunak', 'chimana', 'wuzas', '<eos>']
Predicted translation :  isa isaas basaasta tujo ephete aaffaason maamak ukaat ki'et raaji innok makowa yisete
Average BLEU score 5.191550744154619e-234
```

Appendix-Figure IV: Snapshot