WOLLEGA UNIVERSITY
INSTITUTE OF TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
DEPARTMENT OF COMPUTER SCIENCE
PROGRAM OF STUDY: MASTERS (REGULAR)

GEEZ TO AFAAN OROMOO MACHINE TRANSLATION USING RECURRENT NEURAL NETWORK

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science

M.Sc. Thesis
BY:

Meseret Fetene Biru

Advisor: Dr. M. Kumarasamy

March, 2022

Nekemte, Ethiopia

Meseret Fetene

Email: meseretfetene2010@gmail.com

Phone Number: 0980 180 800

WOLLEGA UNIVERSITY

INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

DEPARTMENT OF COMPUTER SCIENCE

PROGRAM OF STUDY: - MASTERS (REGULAR)

GEEZ TO AFAAN OROMOO MACHINE TRANSLATION USING RECURRENT NEURAL NETWORK

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science

M.Sc. Thesis
BY:

Meseret Fetene Biru (ID Number WU1205485)

Advisor: Dr. M. Kumarasamy

March, 2022

Nekemte, Ethiopia

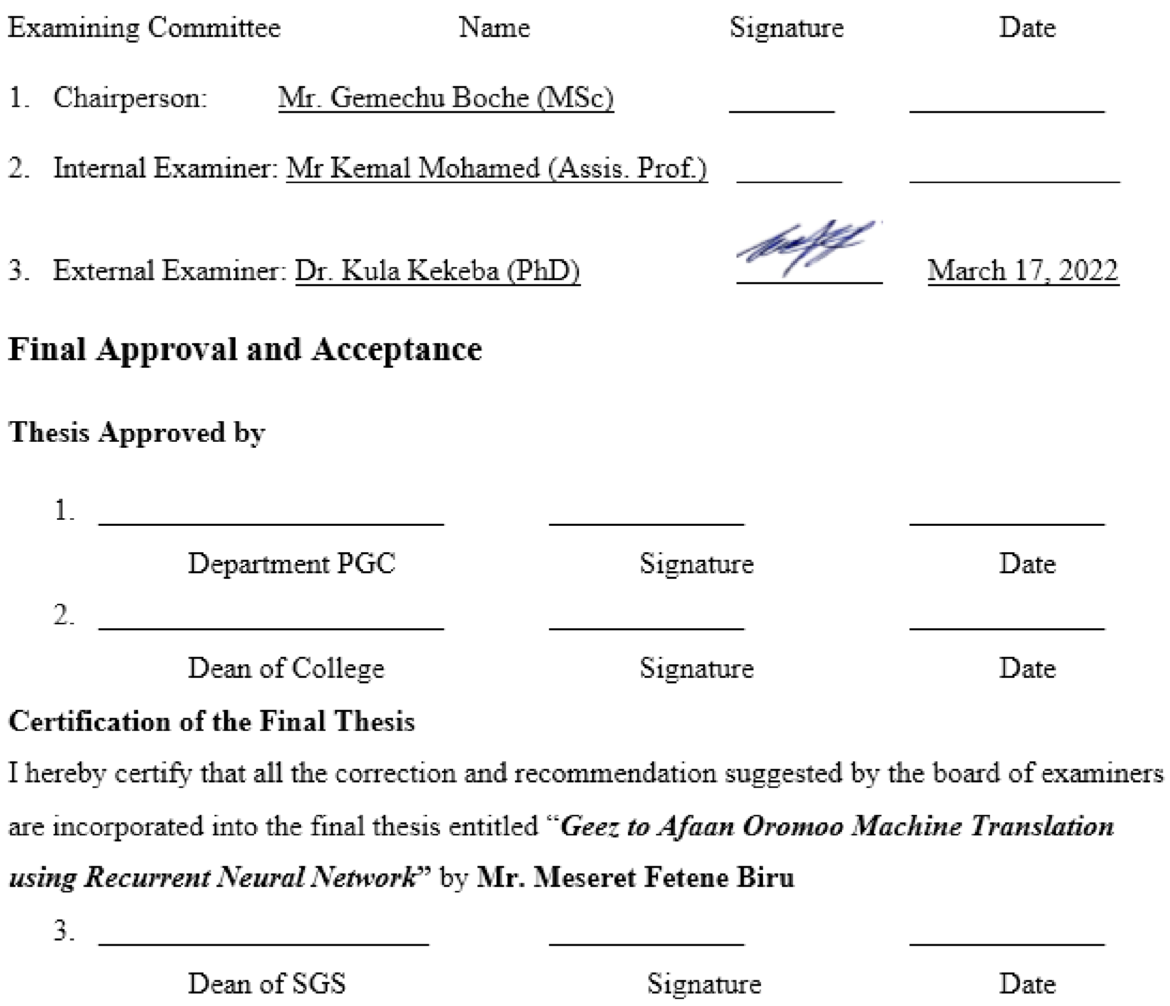
WOLLEGAUNIVERSITY

SCHOOL OF GRADUATE STUDIES

P.O. Box: 395, Nekemte, Ethiopia.

Approval Sheet for Submitting Final Thesis

As members of the Board of Examining of the Final MSc. thesis open defense, we certify that we have read and evaluated the thesis prepared by **Mr. Meseret Fetene Biru** under the title "***Geez to Afaan Oromoo Machine Translation using Recurrent Neural Network"*** and recommend that the thesis be accepted as fulfilling the thesis requirement for the **Degree of Master of Science** in **Computer Science.**



Examining Committee	Name	Signature	Date
1. Chairperson:	Mr. Gemechu Boche (MSc)	_______	_____________
2. Internal Examiner:	Mr Kemal Mohamed (Assis. Prof.)	_______	______________
3. External Examiner:	Dr. Kula Kekeba (PhD)	_________	March 17, 2022

Final Approval and Acceptance

Thesis Approved by

1. _______________________ _____________ _____________
Department PGC Signature Date

2. _______________________ _____________ _____________
Dean of College Signature Date

Certification of the Final Thesis

I hereby certify that all the correction and recommendation suggested by the board of examiners are incorporated into the final thesis entitled "***Geez to Afaan Oromoo Machine Translation using Recurrent Neural Network***" by **Mr. Meseret Fetene Biru**

3. ______________________ _____________ _____________
Dean of SGS Signature Date

Statement of the Author

I **Mr. Meseret Fetene Biru** hereby declare and affirm that the thesis entitled "***Geez to Afaan Oromoo Machine Translation using Recurrent Neural Network"*** is my own work carried out under the supervision of Dr. M. Kumarasamy. I have followed all the ethical principles of scholarship in the preparation, data collection, data analysis and completion of this thesis. All scholarly matter that is included in the thesis has been given recognition through citation. I have adequately cited and referenced all the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and I have not misrepresented, fabricated, or falsified any idea / data / fact / source in my submission. This thesis is submitted in partial fulfillment of the requirement for a degree from the Post Graduate Studies at Wollega University. I further declare that this thesis has not been submitted to any other institution anywhere for the award of any academic degree, diploma or certificate.

I understand that any violation of the above will be cause for disciplinary action by the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name: Meseret Fetene Biru Signature: ____________ Date: ______________

College: Engineering & Technology

Department: Computer Science

Declaration

This is to certify that this thesis entitled "**Geez to Afaan Oromoo Machine Translation using Recurrent Neural Network**" was accepted in partial fulfillment of the requirements for the award of Degree of Master in Computer Science by the school of graduate studies of Wollega University through the college of Engineering and Technology done by **Meseret Fetene Biru** is a genuine work carried out by him under my guidance.

Dr. M. Kumarasamy _____________ _____________

Advisor **signature** **Date**

Dedication

I dedicate this research work to my brother **Bedassa Fetene** who passed away with sudden accident. My gentle brother although you haven't seen my today's success, this is your vision and all my successes were the fruit of your labour since you are the reason for my education starting from joining school up-to I became employee and hence you have inerasable memory in my heart forever.

Acknowledgments

Above all, I would like to give thanks for Almighty God for giving me the ability, strength, knowledge, and opportunity to keep it up and accomplish this research study adequately. Without **His** blessings, this achievement was not thinkable.

Next, I would like to express my deepest gratitude to my advisor **Dr. M. Kumarasamy** for his guidance, quick response and supportive comment and advice.

I also wish to express my sincere gratitude to some of academic staff of college of Engineering and Technology **academic staff,** especially department of **IT** and **Computer Science** instructors and Wollega University **ICT** professionals for their guidance and support that contributed to the successful completion of this research.

My sincere thanks also go to my spiritual fathers, **Margeta Befikadu** and **Abba Gebre Sillassie**, your praise, love of education, precious advices and motivation always push me forward. Your spirit made me what I am today and it will be with me forever. Your support by providing me holly books for the source of datasets and cross-checking parallel corpora correctness with each other, I used for my model training and testing was your contribution.

My deepest gratitude and appreciation also go to my wife's sister (sister in-law) Ms. **Alemi Dessalegn**, my brothers **Ketema Fetene** and **Belay Ararso**, my sisters **Gebune Fetene** and **Maritu Fetene** who supported me economically by sponsoring me education fee and moral support. My brothers and sisters, this work is made possible with your confidence to support me economically, so thanks very much. I would also like to thank all my friends and classmates for their encouragement to complete my work

Last and most importantly, I would like to extend my heartfelt gratitude to all my family members my wife W/ro **Jemanesh Dessalegn**, my Son **Yohannes Meseret** and my daughter **Chaltu Meseret** for their love, encouragements, moral support and doing all the best, what they can do for me.

List of Abbreviations

AI – Artificial Intelligence

ANN - Artificial Neural Networks

API - Application Programming Interface

BLEU – Bilingual Evaluation Understudy

CBMT - Corpus-based Machine Translation

CL - Computational Linguistics

CNN- Convolutional Neural Network

DL – Deep Learning

DNN – Deep Neural network

EBMT - Example-Based Machine Translation

EDA – Explanatory Data Analysis

ECC – Ethiopian Catholic Church

EOTC - Ethiopian Orthodox Tewahedo Church

FFN – Feed Forward Network

GRU – Gated Recurrent Unit

IPW – International Phonetic Writing

LSTM – Long Short-Term Memory

MT – Machine Translation

NMT – Neural Machine Translation

NN –Neural Network

OOV- Out-Of-Vocabulary

RBMT - Rule-Based Machine Translation

RNN – Recurrent Neural Network

RQ – Research Quesition

seq2seq – sequence-to-sequence

SMT – Statistical Machine Translation

SOV – Subject-Object-Verb

SVO – Subject-Verb-Object

VSO – Verb-Subject-Object

TABLE OF CONTENTS

.

List of Tables

List of Figures

Abstract

Machine translation is a sub field of natural language processing that investigates the use of computer software to translate text or speech from one natural language to another. Since the world in which we are living today is occupied by massive languages, about six thousand languages worldwide, the speakers of these much languages need to interact with each other for different global issues. In today's era where the global population communicates easily with any angle of the world using different communication platforms the need for translation between different languages is vital. This communication gap is solved by using an expert translator. The use of manual translation is expensive and inconvenient. Many researches were done to resolve this problem using machine translation techniques for some of resourced languages. However, the researches done on our local languages are very low. The intension of this thesis is to design and implement Geez-Afaan Oromoo neural machine translation, based on the encoder-decoder Recurrent Neural Network approach. Geez is classical South Semitic language which was used in many inscriptions including religious history, philosophy, medical and other since the early 4th century. Today Geez remains only as a spoken language and the liturgy language of the Ethiopian Orthodox Tewahedo Church and Ethiopian Catholic Church in our country. Where, Afaan Oromoo is the most spoken language in Ethiopia and the official working language of Oromia regional state, and it is primary school language in Oromia, Finfinnee and Dirree Dawaa administrations. Afaan Oromoo is the fourth most widely spoken African language after Arabic, Hausa and Swahili. The machine translation of Geez document into Afaan Oromoo will be of paramount importance in order to enable Afaan Oromoo user to easily access the invaluable indigenous knowledge decoded in Geez language. To train the model, two experiments were conducted using two different RNN algorithms. The first experiment is conducted by using GRU to translate Geez to Afaan Oromoo and has a BLEU score of 73.75%. The second experiment is carried out by using LSTM and has a BLEU score of 77.55%. The result shows that the LSTM approach is slightly better than the GRU approach.

Keywords: Machine Translation, Recurrent Neural Network, Gated Recurrent Unit,

Long Short-Term Memory, Geez, Afaan Oromoo

Chapter One

1. Introduction

1.1. Background of the Study

Natural language processing (NLP) is a field of artificial intelligence in which computers analyze, understand, and derive meaning from human language in a smart and useful way. The intension of NLP is to perform tasks such as automatic text summarization, machine translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation. Machine Translation (MT) is a sub field of NLP that investigates the use of computer software to translate text or speech from one natural language to another. The overall process of invention, innovation, and diffusion of technology related to language translation drive the increasing rate of the MT industry rapidly (Ibrahim Gashaw and H L Shashirekha, 2020). A well-trained neural network leads the system towards its goal, which is to generate more efficient translation system that is capable in providing good accuracy. Today many applications such as Google Translate and Microsoft Translator are available online for many languages translations. Machine Translation, in recent years, has become a great concern in relation to NLP. The advances in technology, emergency of the internet, increasing digital data collections, the technical facility and the continuing interest of Interlingua resources sharing have necessitated the development of MT (Dawit Mulugeta , 2015).

Machine translation plays an important role in benefiting linguists, sociologists, computer scientists, etc. by processing natural language to translate it into some other natural language. This demand has grown exponentially over past couple of years, considering the enormous exchange of information between different regions with different regional languages. Machine Translation poses numerous challenges, some of which are: Not all words in one language have equivalent word in another language, two given languages may have completely different scripts and structures, and one word in source language may have many dialects or can have more than one meaning. Owing to these challenges, along with many others, MT has been active area of research for more than five decades (Ankush Garg and

Mayank Agarwal, 2018). Numerous methods have been proposed in the past which either aim at improving the quality of the translations generated by them, or study the robustness of these systems by measuring their performance on many different languages.

Machine Translation has different advantages (Tadesse Kassa, 2018). The first one is machine translation can save the time. Individuals are not expected to spend hours poring over dictionaries to translate the words. Instead, software can translate the content quickly and provide quality output to the user immediately. Machine translation can generate thousands of words within a minute. The second advantage of machine translation is that it is comparatively cheap. Initially, it might look like an unnecessary investment but in the long run it is a very small cost considering the return on investment it provides. This is because the use of the expertise of a professional translator, he/she will charge on a per page basis which is going to be extremely costly while this will be cheap in the case of MT. Third advantage is confidentiality that makes machine translation favorable. Giving sensitive data to a translator might be risky while with machine translation information is protected. Finally, a machine translation usually translates text with which it is trained. The same is true for professional, so there is no such major concern while a professional translator specializes in one field.

Machine translation approaches includes rule based, corpus based, hybrid and Deep Learning (DL). Rule-Based Machine Translation (RBMT), also known as Knowledge-Based MT, is a general term that describes MT systems based on linguistic information about source and target languages. Corpus-based Machine Translation (CBMT) Approach, also referred as data driven MT, is an alternative approach for machine translation to overcome the problem of knowledge acquisition problem of rule-based machine translation. Corpus Based MT uses a bilingual parallel corpus to obtain knowledge for new incoming translation. Statistical techniques are applied to create models whose parameters are derived from the analysis of bilingual text corpora. Example-based machine translation (EBMT) is one of the examples of CBMT, characterized by its use of bilingual dictionary with parallel texts as its main knowledge, in which translation by correlation is the main idea (Tadesse Kassa, 2018).

Deep learning Neural Machine Translation (NMT) approach is a recent approach of MT that produces high-quality translation results based on a massive amount of aligned parallel text corpora in both the source and target languages. It allows computational models that are composed of multiple processing layers to learn representations of data with various levels of abstraction. These methods have improved the state-of-the-art research in language translation. Neural Machine Translation is one of the deep learning end-to-end learning approaches to MT that uses a large Artificial Neural Networks (ANN) to predict the likelihood of a sequence of words, typically modeling entire sentences in a single integrated model. The advantage of this approach is that a single system can be trained directly on the source and target text no longer requiring the pipeline of specialized systems used in statistical MT (Ibrahim Gashaw and H L Shashirekha, 2020). Many companies, such as Google, Facebook, and Microsoft, are already using NMT technology. NMT has recently shown promising results on multiple language pairs. Nowadays, it is widely used to solve translation problems for many language pairs. However, much of the research on this area has focused on European languages despite these languages being very rich in resources.

Neural machine translation, is the use of neural network models to learn a statistical model for machine translation. Unlike the traditional phrase-based translation system which consists of many small sub-components that are tuned separately, neural machine translation attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation. As such, neural machine translation systems are said to be end-to-end systems as only one model is required for the translation.

Neural network is a new technique, widely used in different machine learning applications. It enables the system to learn like a human and to improve the efficiency with training. Neural network attracts researchers for using it in machine translation. The main idea behind this study is to develop a system that works as translator; with the help of history and past experiences a trained neural network that translates the sentences without using large database of rules.

1.2. Motivation

Ethiopia has eighty-six indigenous languages spoken within the country. Afaan Oromoo is the most widely spoken language in Ethiopia. About half of the populations in the country speak Afaan Oromo while 33.8% of the populations are native speakers. Geez on the other hand is liturgy language for Ethiopian Orthodox Tewahedo Church (EOTC), Ethiopian

Catholic Church, Eritrean orthodox and Beta Israel. There are a very huge number of resources available in Geez Language that range from religious to the philosophical, medical and other disciplines (Dawit Mulugeta , 2015). There is some proverb of *Nelson Mandela* which says “if you tell to the person with language that he/she can understand it goes to mind; if you tell to the person with his/her language it goes to heart”. That is why someone needs to know encoded information in another language to understand what the other language is saying. For example, on Facebook posts someone can click see Translation and understand the message of post or comments in the language that is not familiar to him/her. Due to a large number of speakers of Afaan Oromoo that are EOTC religious followers, need of translations from Geez to Afaan Oromo is highly increasing. So Afaan Oromoo native speakers need to get liturgy from EOTC in Afaan Oromoo rather than in Geez language. From the Ethiopian census data of 2007, it is reported that **41.3%** *of* **Oromo** population follow the EOTC religious. Since the liturgy or church service is provided using Geez language this much of believers need to learn about their religious in their own language. As it can be perceived easily the reason behind the decrease in the number of EOTC members is the barrier to the language. This barrier came from lack of translation from Geez to their mother tongue language Afaan Oromoo. This motivated the researcher to study and investigate the design and implementation of Geez to Afaan Oromo machine translation.

1.3. Statement of the Problem

Today people across the world are joining together to discuss different global issues in order to improve their environments. For example, different countries participate on academic and industrial research conference meetings. At the time of their meeting, they can discuss different issues such as education quality, health issues, human rights, democracy and the like. Also, within a country different ethnic group can interact for different issues such as socio-economic, religious, regional and local borders, marketing and business and so on. Due to this different people with different languages need translator to communicate.

Geez language is Ethiopic historical language used for a long year in different perspectives, such as official writing language of our country for about more than one thousand years. Currently this language has no native speakers and no ethnic group this language as natural language for interaction. The barriers to this indigenous language are being covered only by

EOTC church scholars who learn Geez for many years to communicate using Geez fluently. So, to overcome this barrier to the access of Geez language could be solved by conducting different researches on this indigenous language and facilitate the means in which the existing generation could re-learn this language for finding different source of languages encoded in ancient time. Machine translation is one of the means to support these issues.

Geez is an ancient language and many manuscripts are already archived by Ethiopian Orthodox Church as well as by the National Archival Agency. Geez had been known as being used in Ethiopia since the 4th century and as a spoken language close to a thousand years and had been serving as official written language practically up to the end of 19th century. (Tadesse Kassa, 2018)

Some attempts are made by EOTC and individuals to translate manually some of the religious manuscripts, law and philosophical works. The problems observed in manual translation are time taking, resource intensive, and linguistic knowledge of the language is mandatory. Although machine translation has its own challenges, such as difference in scripts like Geez and Afaan Oromoo, which needs further processing to make datasets machine readable, languages morphologically richness, the same pronunciations different characters in Geez such ad ሀ፥ሐ፥ኀ፥ኸ/Ha' sound, dialects or more than one meaning in the target language *etc.*, it can improve performance and reduce cost. Though there is advancement in applying MT for different European languages pairs, it is still in its infant stage for our local languages.

Machine translation models for different language pairs have been developed by using different methodologies and approaches used in the area of the study. Most of the studies have been done on language pairs of English and other foreign language such as Spanish, Chinese, Arabic, French, Japanese and most of major languages spoken in India and some of the studies are done on foreign language pairs which are morphologically related to each other (Jabessa Daba, 2013).

However, there has been a little work on machine translation for languages that are spoken in Ethiopia. Some of the studies are carried out on Amharic to Arabic language pair, Geez to Amharic, Amharic to Afaan Oromoo, English to Afaan Oromoo, and as far as the knowledge of the researcher there is no experiment on Geez to Afaan Oromo language pair. The experiment which was conducted on Geez to Amharic language pair was done by using statistical approach and hybrid approach methodologies. This study is proposed to translate Geez sentences into Afaan Oromoo using RNN approach machine translation.

1.4. Research Questions

Based on the statement of the problem given above, this study attempted to answers the following basic research questions:

- What is the similarity and difference between Geez and Afaan Oromoo writing system and orders of word classes (subject-verb agreement) in their sentence?
- How can vanishing gradient problem of recurrent neural network can be solved to improve the performance of neural machine translation?
- Which layers of recurrent neural network machine translation Long Short-Term Memory or Gated Recurrent Unit can perform well?
- To what extent or BLEU accuracy score does recurrent neural network approach perform machine translation from Geez to Afaan Oromoo?

1.5. Objectives of the Study

1.5.1. General Objective

The general objective of this research is to design and implement Geez to Afaan Oromoo machine translation using recurrent neural network machine translation approach.

1.5.2. Specific Objective

To achieve the general objective, the study addresses the following specific objectives:

i. To review literatures about machine translation, related works on machine translation and overview of Geez and Afaan Oromoo languages.

ii. To collect, clean and prepare Geez-Afaan Oromoo parallel sentences dataset used for implementing neural machine translation model.

iii. To design, train and test Geez to Afaan Oromo machine translation using RNN algorithms with LSTM and GRU.

iv. To evaluate the accuracy of the Geez-Afaan Oromoo neural machine translation model using BLEU score metrics.

1.6. Scope and limitation of the Study

1.6.1. Scope

This research study focuses on implementation of machine translation from Geez to Afan Oromoo languages using RNN based algorithm. Recurrent Neural Network is one the NMT family which is deep learning theme and suitable for shorter sentences datasets. Recently researchers and companies prefer NMT due to promising accuracy. Neural Machine Translation technology can be applied to any language pair including languages that are brand new or understood by few. They can also be fine-tuned accordingly to particular styles and types of a languages. Neural Machine Translations are primarily dependent on the training data used to train the neural network as it learns to mimic the data it has been trained with. The intention of the study will be to design and implement textual translation model only from Geez to Afaan Oromoo languages. It is focused on unidirectional text to text machine translation from source language namely Geez simple sentences to the target language sentences namely Afaan Oromoo.

1.6.2. Limitations of the study

The machine translation in this research work doesn’t include bidirectional machine translation, meaning the source language is Geez and the target language is Afaan Oromoo. Hence the trained model is expected to translate sentences from Geez to Afaan Oromoo only, but not Afaan Oromoo to Geez. It does not include speech to speech, speech to text or text to speech translation.

There are different limitations faced during the process of conducting this research. The first and the most challenge was the lack of public online parallel corpus for the training and testing. The limitation comes from the absence of sufficient amount digitally available documents in Geez. On the other hand, because of unavailability of machine translation research done on Geez to Afaan Oromoo languages it is difficult to compare the findings of the study with previous studies as a baseline, for this language pair.

1.7. Significance of the Study

Even though this research is mainly for academic exercises, it will have paramount contribution for the organization and individuals interested to investigate in similar area. The results of the study are expected to produce machine translation trained model from Geez to Afaan Oromoo. There are different indigenous documents in Geez language. Some of these documents are translated manually. Much information available in Geez language is neither translated manually nor with machine translation system into Afaan Oromoo. Therefore, if machine translation model is developed and extended to the commercial level to translate Geez language documents into Afaan Oromoo it is very important to retrieve relevant documents such as history, philosophy, medical, religious and other documents.

1.8. Organization of the Thesis

This paper deals with Geez to Afaan Oromoo recurrent neural network machine translation. It contains six chapters. This chapter discusses introduction part which incorporate background of the study, motivation, problem statement, research questions, objective, scope and limitation as well as significance of the study. The remaining part of the thesis is organized as follows. The second chapter presents literature review which briefly discusses about an overview of machine translation, the works that directly or indirectly related to this thesis work and, the Geez and Afaan Oromoo languages writing system, word classes, and orders in a sentence. The Third Chapter discusses the methodology of the study. It describes research design, data collection and preparation, training and testing the model**,** tools and programming language and machine translation evaluation. The fourth chapter discusses about recurrent neural network machine translation model of Geez-Afaan Oromoo languages. It describes proposed model architecture, data preparation for training and testing and GRU and LSTM algorithms. Chapter five is about model training and testing experiment conducted and the results and discussions. The chapter briefly explain the experimentation on machine translation model training and testing and test result of the system performance. The last chapter discusses conclusion and recommendation of the research work.

Chapter Two

2. Literature Review

This chapter discusses about overview of machine translation, machine translation evolution and available approaches to machine translation. Related work concerning machine translation has been also discussed. This chapter also include highlights on Geez and Afaan Oromoo languages writing system, word classes, and orders in a sentence.

2.1.Machine Translation Overview

Machine Translation is a sub-field of computational linguistics that aims to automatically translate text or speech from one language to another using a computing device. Earlier research focused on rule-based systems, which gave way to example-based systems in the 1980s. Statistical machine translation gained prominence starting late 1980s, and different word-based and phrase-based techniques requiring little to no linguistic information were introduced (Ankush Garg and Mayank Agarwal, 2018).

With the advent of deep neural networks in 2012, application of these neural networks in machine translation systems became a major area of research. (Shuoheng Yang, et al, 2020) Recently, researchers announced achieving human parity on automatic Chinese to English news translation using neural machine translation. While early machine translation systems were primarily used to translate scientific and technical documents, contemporary applications are varied. These include various online translation systems for exchange of bilingual information, teaching systems, and many others.

2.2.History of Machine Translation

Machine translation has a long history. The origin of this field could be traced back to the 17th century. In 1629, Ren’s Descartes came up with a universal language that expressed the same meaning in different languages and shared one symbol.

The particular experiment of MT started at about the 1950s, when the primary analyst in the field, Yehoshua Bar-Hillel, started his research at MIT 1951 and sorted out the principal

International Meeting on Machine Translation in 1952. From that point forward, MT has encountered three essential waves in its turn of events, the Rule-based Machine Translation, the Statistical Machine Translation, and the Neural Machine Translation. (Shuoheng Yang, et al, 2020)

Although there are some disputes about who first had the idea of translating automatically between human languages, the actual development of Machine Translation System can be traced back to an influential paper written in July 1949 by Warren Weaver - a director at the Rockefeller Foundation. (Dawit Mulugeta , 2015)

Computer scientists began trying to solve the problem of MT in the 1950s. The first published machine translation experiment was performed by the Georgetown University and IBM. It involved automatic translation of more than 60 Russian sentences into English. The system had only 6 grammar rules and 250 lexical items in its vocabulary. It was by no means a fully featured system. The sentences for translation were selected carefully, as the idea of the experiment was to attract governmental and public interest and funding the project by showing the possibilities of MT.

Many problems of MT had come to light right after, and, consequently, for a long time, MT was present only as a research area in computational linguistics. Overtime, different approaches for MT were defined and gained maturity for practical use today. The history of the development of MT approaches is given in **Figure 1** below.

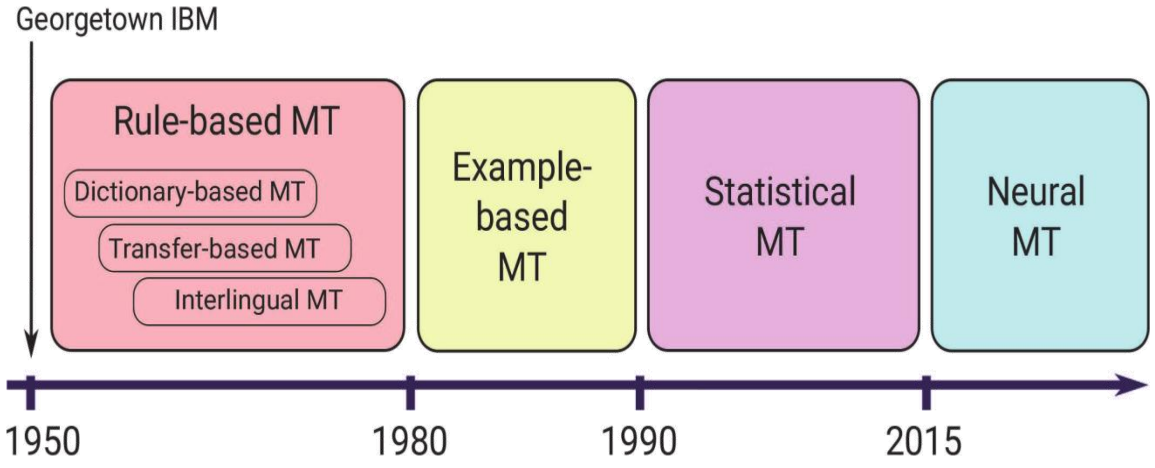


Figure 1: Timeline of Machine Translation Evolution (Mirjam Sepesy et al, 2018)

2.3.Approaches to Machine Translation

Different researches efforts have been done to explore the possibility of automatic translation of one language to other language. The different method of machine translation has discussed below.

i. Rule-based machine translation

Rule Based Machine Translation has much to do with the morphological, syntactic and semantic information about the source and target language (Jabessa Daba, 2013). The first approaches for MT were based on linguistic rules that were used to parse the source sentence and create the intermediate representation, from which the target language sentence was created. Such approaches are appropriate to translate between closely related languages. The rule-based machine translation methods include dictionary-based MT, transfer-based MT, and inter-lingual MT.

Dictionary-based MT uses entries in a language dictionary to find words equivalent in the target language. Using a dictionary as the sole information source for translation means that the words will be translated as they are translated in a dictionary. As this is, in many cases, not correct, grammatical rules are applied afterwards.

Transfer-based MT belongs to the next generation of machine translation. The source sentence is transformed into an intermediate, less language-specific structure. This structure is then transferred into a similar structure of the target language, and, finally, the sentence is generated in the target language. The transfer uses morphological, syntactic, and/or semantic information about the source and target languages.

The machine translation using interlingua works with the idea of changing the concept in the source language into inter-mediatory neutral language, known as Interlingua, in order to later target sentence will be derived by selecting other translation technique like dictionary-based modeling. This Interlingua is neutral representation and related to the structure of both languages. This Interlingua based translation needs to drive the inter-mediatory representation between languages of having similar structure and difficult for the languages of having different structures.

ii. Example-based machine translation

Example-Based Machine Translation is based on the idea of analogy. It is grounded upon a search for analogous examples of sentence pairs in the source and target languages (Biruk Abel, 2018). EBMT belongs to corpus-based approaches because examples are extracted from large collections of bilingual corpora. Given the source sentence, sentences with similar sub-sentential components are extracted from the source side of the bilingual corpus, and their translations to the target language are then used to construct the complete translation of the sentence.

iii. Statistical machine translation

Statistical Machine Translation (SMT) is a data driven approach which uses parallel aligned corpora and treat translation as a mathematical reasoning problem, in that every sentence in the target language is a translation with probability from the source language (Gelan Tullu, 2020). Statistical Machine Translation is based on statistical methods. It also belongs to corpus-based approaches; as statistical methods are applied on large bilingual corpora. Building a SMT system does not require linguistic knowledge (Jabessa Daba, 2013). Statistical MT utilizes statistical models generated from the analysis of texts, being either monolingual or bilingual. It is called training data. If more training data are available, better and larger MT systems can be built. Statistical MT systems are computationally expensive to build and store. Statistical MT can be adapted easily to a specific domain if enough bilingual and/or monolingual data from that domain are available.

iv. Hybrid machine translation

While statistical methods still dominate research work in MT, most commercial MT systems were, from the beginning, only rule-based (Biruk Abel, 2018). Recently, boundaries between the two approaches have narrowed, and hybrid approaches emerged, which try to gain benefit from both of them. Hybrid systems, guided by RBMT, use SMT to identify the set of appropriate translation candidates and/or to combine partial translations into the final sentence in the target language.

v. Neural machine translation

Recently, neural machine translation (NMT) has gained popularity in the field of machine translation. The conventional encoder-decoder NMT proposed by Cho (2014) uses two recurrent neural networks (RNN): one is an encoder, which encodes a source sequence into a fixed-length vector, and the other is a decoder, which decodes the vector into a target sequence (Yukio Matsumura, et al., 2017).

Neural Machine Translation emerged as a successor of SMT. It has made rapid progress in recent years, and it is paving its way into the translation industry as well. Neural Machine Translation is a deep learning-based approach to MT that uses a large Neural Network (NN) based on vector representations of words. If compared with SMT, there is no separate language model, translation model, or reordering model, but just a single sequence model, which predicts one word at a time (Zhixing Tana et al., 2020). The prediction is conditioned on the source sentence and the already produced sequence in the target language.

The prediction power of neural MT is more promising than that of SMT, as neural networks share statistical evidence between similar words. The input words are passed through the layers of the encoder to its last layer, the context vector, updating it for every input word. The context layer is then passed through the decoder layers to output words, and it is again updated for each output word. The encoder-decoder RNN architecture with attention is currently the state of the art for machine translation.

Although effective, the NMT systems still suffer some issues, such as scaling to larger vocabularies of words and the slow speed of training the models (Dereje Saifu, 2019). In addition, large corpus is needed to train NMT systems with performance comparable to SMT.

Neural machine translation is a way to do machine translation with a *single neural network*. The NN architecture is called encoder-decoder architecture and it involves *two* RNN layer. Neural machine translation was a revolution in the field of MT, mainly because it uses single NN architecture. The main aim of this modeling technique is to have a function *f* that takes in a word in a language X and output the corresponding word in the other language Y. This architecture is called seq2seq, and involves two RNNs: an encoder and a decoder (Shuoheng

Yang, et al, 2020). The encoder RNN will produce an encoding of the source sentence, while the decoder RNN is a language model that generates the target sentence, conditioned on the encoding. The following figure shows diagrammatical representation *of RNN-based translation model* encoder-decoder architecture.

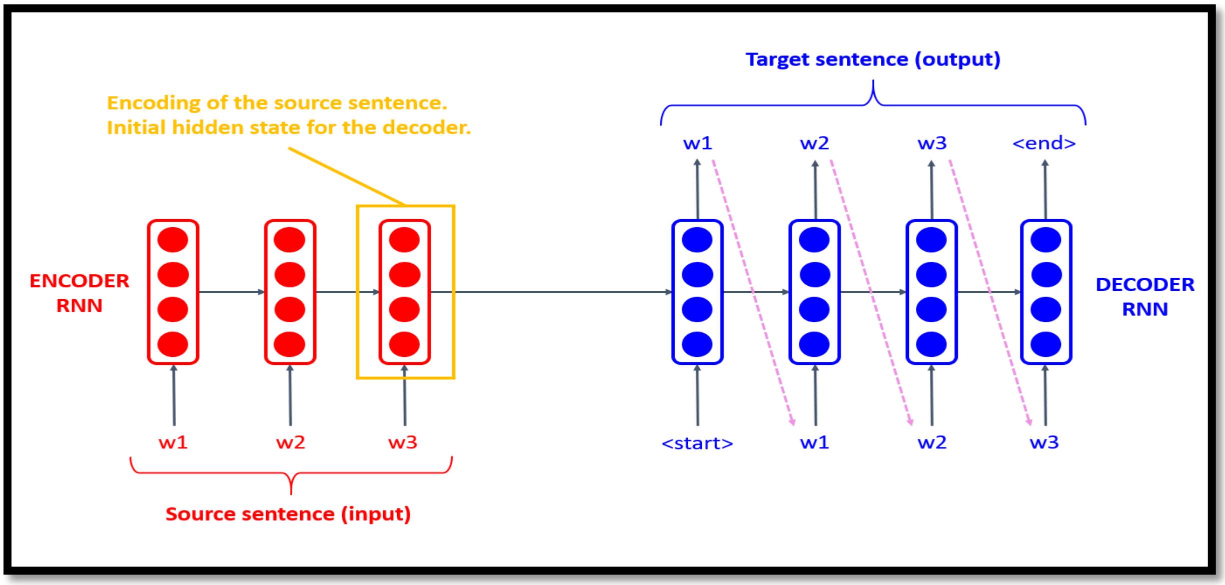


Figure 2: RNN-based translation model encoder-decoder architecture (Arthur et al., 2016)

2.4. Related Work

Different researches were proposed for text translation and speech recognition for technological supported languages like English to other non-Ethiopian languages (Dereje Saifu, 2019). Under resourced languages like Afaan Oromoo are not yet being translated on Google Translate. Few Ethiopian scholars focused on machine translation research areas of Afaan Oromoo and other languages pair.

In 2009, Sisay Adugna conducted research that attempts to apply statistical machine translation approach so as to design English-to-Afaan Oromo machine translation system. Monolingual and Parallel corpus used for the experiment was collected form governmental and a non-government organization document which exists on the web such as Constitution of FDRE, Universal Declaration of Human Right, proclamations of the Council of Oromia Regional State, religious documents, and other documents as these are the already translated and available documents. Then the corpus divided into 90% of it for training and 10% for testing the MT system. The corpus used for the experiment were preprocessed using Perl script which includes tasks like apostrophe, sentence aligning, tokenization, lowercasing and truncating long sentences that take the alignment to be out of optimality were done by those scripts. The size of the monolingual which is Afaan Oromo 62,300 sentences and bilingual corpus of 20,000 were used for conducting the experiment of which 90% and 10% used for training and testing the MT system respectively. The experimentation of statistical machine translation of English to Afaan Oromo was conducted and a score of 17.74% was found. Although Afaan Oromo is among resource-scarce languages of the world, the result of this experiment shows that the amount of data available can be used as a good starting point to build machine translation system from English to Afaan Oromo. The researcher also recommends a lot to do on translation between the two languages so as to enhance translation accuracy make real (Sisay Adugna, 2009).

Another research was conducted by Jabesa Daba in 2013 for partial fulfillment of degree of MSC in computer science from Addis Ababa University, with purpose of using hybrid approach to develop a bidirectional English-Afaan Oromo Machine translation system. He conducted the experiment with previously work done by Sisay (Sisay Adugna, 2009) which is having BLEU score of 17.74% not satisfactory and due to unidirectional problems, that is English to Afaan Oromo. The researcher uses Hybrid approach which is the combination of corpus-based approach and rule-based approach requires the availability of bilingual parallel corpus. Parallel corpus collected from different domain including the Holy Bible, the Constitution of FDRE, and the Criminal Code of FDRE, international conventions, Magalata Oromia and a bulletin from Oromia health bureau. After the corpus collected it passes through preprocessing activities such as tokenization, True-casing and cleaning were used. For the experiment purpose freely available software like IRSTLM toolkit, GIZA++, and Moses for the statistical part and Python programming language for the rule part were used. A total of 3000 English–Afaan Oromoo parallel sentences for training and testing the system was used in two experiments namely Experiment I and Experiment II. From the total of 3000 parallel sentences, 2, 900 parallel sentences were used for training whereas the rest were used for testing the system. Statistical and Hybrid approach were used for Experiment I and Experiment II respectively. The result of experiment I, the BLEU score methodology recorded result shows 32.39% for English to Afaan Oromo translation and 41.50% for Afaan Oromo to English translation. The result of experiment II BLEU score methodology shows that 37.41% for English to Afaan Oromo translation and 52.02% for Afaan Oromo to English translation. As mentioned by the researcher the reason for the difference between both the records in the two experiments were that there is a difference between feminine and masculine representation in English and Afaan Oromo languages (Jabessa Daba, 2013).

The other thesis was conducted by Yitayew Solomon in 2017 for partial fulfillment of the degree of MSc in Information Science from Addis Ababa University, with the purpose of using statistical machine translation approach, exploring an optimal alignment for bidirectional English-Afaan Oromoo MT Systems. For the researcher to have such an objective was, the research done by (Sisay Adugna, 2009)and (Jabessa Daba, 2013) score poor performance of BLEU score is 17% and 37% respectively, this is due to the alignment quality of the prepared data due to the unavailability of well-prepared corpus for the MT task for English to Afaan Oromo SMT and experimental research approach were used. FDRE criminal code, FDRE constitution; Megeleta Oromia, Holy Bible and simple sentences were used as data set or corpus for the experiments. To build the translation model, 6400 parallel sentences and 19300 and 12200 sentences, to build language model for both English and Afaan Oromo languages were used respectively. Randomly, for training 90% and 10 % testing of corpus size were used. Six experiments were done by the researcher to select the optimal alignment quality for English to Afaan Oromo where, Experiment I and II for word level alignment, Experiment III and IV for phase level alignment and experiment V and VI for sentence level alignment. Word level alignment when the max phrase length is 4 and min is 1which record 21% and 42% BLEU score from English-Afaan Oromo and from Afaan Oromo-English respectively. Phrase level alignment when the max phrase length is 16 and min is 4 which record 27% and 47% BLEU score from English-Afaan Oromo and from Afaan Oromo-English respectively. Sentence level alignment when the max phrase length is 30 and min is 20 which record 18% and 35% BLEU score from English-Afaan Oromo and from Afaan Oromo-English respectively. An optimal alignment is phrase level alignment when the max phrase length is 16 and min is 4 which record 27% and 47% BLEU score from English-Afaan Oromo and from Afaan Oromo-English respectively. (Yitayew Solomon, 2017)

Dawit Mulugeta has investigated Geez to Amharic automatic machine translation using satirical machine translation. The data used for the research experiment were found from both online and manually prepared. The data collected were in HTML, MS-word, MS-Publisher and MS-Excel format. To make this entire format suitable for the experiment, the researcher merges all documents to Ms.-Word format and align to verse/sentence level, cleaned for noisy characters and converted to plain text in UTF-8 format. Even if inherently data in both languages were verse level aligned, but the researcher align sentences manually which is misaligned at verse and sentence level. Language expert also used for cross checking of the correct alignment of the corpus. The dataset used by the researcher was biblical data. Regarding the organization of the data, out of the bilingual data, 90% for training and 10% for testing were used for experiment. Moses decoder, IRSTLM, GIZA++ and BLEU were used to build translation model, language model, Word alignment and evaluation of the Geez to Amharic MT system respectively. The Parallel corpus used for the experiment was sentence level aligned. The average translation accuracy of BLEU score was 8.26 (Dawit Mulugeta , 2015).

Tadese kassa 2018 also conducted the machine translation experiment on morpheme-based bi-directional Geez –Amharic machine translation. Their experimental results showed a better performance of 15.14% and 16.15%BLEU scores using morpheme-based from Geez to Amharic and from Amharic to Geez translation, respectively. As compared to word level translation there is on the average 6.77% and 7.73% improvement from Geez-Amharic and Amharic-Geez respectively (Tadesse Kassa, 2018).

Biruk Abel in 2018 conducted experiment on Geez-Amharic machine translation using hybrid approach by combining RBMT and SMT. Their proposed system is composed of two main components a Rule Based Geez Corpus Preprocessor and a Baseline SMT. The Rule Based Preprocessor takes the manually Part of Speech (POS) tagged Geez corpus and produces another corpus that contains reordered Geez sentences having similar structure with that of Amharic sentences. This component contains set of activities that process each Geez sentence in the input corpus one by one to determine POS pattern and subsequently apply the corresponding reordering rule. It first reads all sentences from the input file and iterates through all sentences and it first determines POS pattern and applies the corresponding reordering rule. After each sentence is processed the output corpus along with the Amharic corpus will be supplied as an input to the Baseline SMT. Then using the input corpora, the actual translation of Geez sentence to Amharic sentences will be performed by the Decoder of the Baseline SMT by using the Language model of Amharic and Translation model and improve performance baseline BLEU score from 72% to 76%.

As it can be seen from the above Geez to Amharic machine translation researchers used early approaches to machine translation (Biruk Abel, 2018). The following themes of studies discuss the implementations of NMT in the area of deep learning approaches.

In 2016 (Arthur et al., 2016) researched an approach that enhances NMT systems, with discrete probability-based translation lexicons. Translation of low-frequency words is encoded efficiently by this lexicon which is derived from either automatic learning as SMT hand-made lexicon or as a combination of both. NMT models calculate the next target word’s probability for English-Japanese language pair. Their experiments showed improvement of 2.02 BLEU score and faster convergence time. For the bias method, their system showed less effectiveness for the hand-made lexicon because it did not cover sufficient target-domain words OOV problem for fluent sentence translation. The linear interpolation method performed less than the bias method because of the constant interpolation coefficient that was fixed for every context.

(Junczys-Dowmunt et al., 2016) proposed an integration of attention-based NMT with phrase based SMT decoder. This work was submitted to the world MT 2016 shared operation, on news translation. They re-implemented the inference step of the NMT model with two phrase-based decoding algorithm- stack decoding and cube pruning. They created a baseline with Moses along with some additional feature functions. They launched a C++ compute unified device architecture format (version) of the inference steps for the neural models, which implemented the framework of Moses log-linear model as several illustration of the same element. They described their proposed algorithm for integrating graphical processing unit (GPU) based soft attention neural translation models into Moses. They ran the system for Russian-English dataset and it outperformed the best pure neural system submitted in the conference for MT, 2016 by 1.1 BLEU points. In terms of translation speed handling of OOV problem, their system was slower even than with a basic phrase-based SMT, operating with 24 central processing unit (CPU) threads.

In 2019 Afarso Birhanu conducted encoder-decoder deep learning experiment on a bi-directional text-based machine translation for English and Afan Oromo languages pair. In their experimental study, they have implemented the two designed systems and trained the systems in similar way to get correct comparison of their performance. Their accuracy BLEU score result in RNN architecture was 22.79 and 21.67 for English-Afaan Oromoo and afaan Oromoo-English respectively, while in CNN architecture 24.37 and 23.18 for English-Afaan Oromoo and afaan Oromoo-English respectively. They compared and reported that the CNN architecture performance is better for long sentences and RNN architecture performs well for shorter sentences (Afarso Birhanu, 2019).

Another deep learning machine translation research was conducted by Dereje Saifu in 2019 on the thesis topic hybrid artificial neural machine translation using deep learning techniques English-to-Afaan Oromoo, presented in partial fulfillment for the MSc from Adama Science and Technology University. The main aim of their research is to design a model that significantly reduces translation problems shown in the neural machine translation system. The proposed model was implemented by classifying the model into three layers namely; encoder layer, hybrid layer and decoder layer. Their experiment result shows that the proposed method significantly improved the translation quality of the state-of-the-art neural machine translation system on English-to-Afaan Oromoo translation tasks prepared dataset. The problems; out-of-vocabulary lead to unknown word output, lack of translation problem for long sentences, lack of decoding mechanism, which badly hurt the translation quality in the neural machine translation system is reduced by their proposed hybrid neural machine translation model. The translation accuracy was 85.5 BLEU Score. (Dereje Saifu, 2019)

This research is proposed to implement Geez to Afaan Oromoo machine translation using neural machine translation encoder-decoder RNN with GRU and LSTM algorithms to exercise the current approach to machine translation and improve performance. As the researcher's knowledge there is no Geez-Afaan Oromoo machine translation was developed before. Therefore, the research gap for this thesis is unavailability of the machine translation between the Geez and Afaan Oromo languages pair.

The related work researches were summarized in the following table 1. The summary shows research title, author with the year of the study, methods they used and the research gap authors recommended to be improved by other researchers.

No	Author/ Year	Research Topic	Technique/methods	Results(accuracy)	Gap
1	Sisay Adugna 2009	English – Afaan Oromoo Machine Translation: An Experiment Using Statistical Approach	Machine translation system using statistical approach	17.74% BLEU score	Only one reference translation is provided for evaluation of the system
2	Jabesa Daba 2013	Bidirectional English – Afaan Oromo Machine Translation Using Hybrid Approach	The research work is implemented using a hybrid of rule based and statistical approaches.	BLEU score of 32.39% for English-A/Oromo and 41.50% for A/Oromo-English translation. using a hybrid approach, the result obtained has a BLEU score of 37.41% for English-A/Oromo and 52.02% for A/Oromo-English translation	The rules which are developed and used in the system are only used for syntax reordering.
3	Dawit Mulugeta 2015	Geez to Amharic automatic machine translation using satirical machine translation	The IRSTLM and GIZA++ language modeling toolkit is used to train language model for this research.	The average translation accuracy of BLEU score was 8.26	This system does not perform well due to the limited size of the corpus.
4	Tadese kassa 2018	Morpheme-Based Bi-directional Ge'ez- Amharic Machine Translation	Conducted the machine translation experiment on morpheme-based bi-directional Geez – Amharic machine translation.	Better performance of 15.14% and 16.15% BLEU scores using morpheme-based from Geez to Amharic and from Amharic to Geez translation, respectively.	They focused only on morpheme and word as a translation unit

5	Biruk Abel 2018	Geez to Amharic Hybrid Machine Translation	Proposed system is composed of two main components a Rule Based Geez Corpus Preprocessor and a Baseline SMT.	76% BLEU score	Proposed system cannot support more complex sentences and apply more reordering rules
6	Afarso Birhanu 2019	Bi-Directional English-Afan Oromo Machine Translation Using Convolutional Neural Network	Conducted encoder-decoder deep learning experiment on a bi-directional text-based machine translation for English and Afan Oromo languages pair.	BLEU score result in RNN architecture was 22.79 and 21.67 for English-A/Oromoo and A/Oromoo-English respectively, while in CNN architecture 24.37 and 23.18 for English-A/ Oromoo and A/Oromoo-English respectively.	Not tested the systems on GPU based computer to minimize the required time for training the systems.
7	Dereje Saifu 2019	Hybrid Artificial Neural Machine Translation using Deep Learning Techniques English-to-Afaan Oromoo	The problems; out-of-vocabulary lead to unknown word output, lack of translation problem for long sentences, lack of decoding mechanism, which badly hurt the translation quality in the neural machine translation system is reduced by our proposed hybrid neural machine translation model.	The translation accuracy scale was 85.5 BLEU Score	Their RNN encoder-decoder is a word-based translation system. good at capturing local word reordering, idiom and translation.

Table 1: Summary of related works with their gap

2.5.Geez and Afaan Oromoo Languages Overview

Geez is a family of Afro-Asiatic Phylum language having its own scripts. These characters have their own meaning and pictorial representation that have been in use since ancient times. Geez, sometimes called Ethiopic language, is an ancient South Semitic language of Ethiopia and Eritrea in the Horn of Africa later became the official language of the Kingdom of Aksum (Dawit Mulugeta , 2015). Geez is still the liturgical language of the EOTC which is attested in inscription since the early 4th century. Geez has probably died out as a spoken language close to 13th century, but remained the primary written language of Ethiopia up to the 20th century. The literature includes religious texts, as well as secular writings.

Afaan Oromo is one of the languages of the Low land East Cushitic within the Cushitic family of the Afro-Asiatic Phylum (Ibrahim Bedane, 2015). It is also one of the major languages spoken in Ethiopia. Afaan Oromoo is the fourth most widely spoken African language after Arabic, Hausa and Swahili. Like other African and Ethiopian languages, Afaan Oromo has a very rich morphology. Latin based alphabet known as *Qubee* has been adopted and became an official script of Afaan Oromo since 1991 (Hamiid M., 1996). The language is widely used in Ethiopia and neighboring countries like Kenya and Somalia. Currently, Afaan Oromo is an official language of Oromia Regional State and used as an instructional media for primary and junior secondary schools of the region.

2.6.Geez Writing Systems

Writing is a method of representing language in visual or tactile form. Writing systems use sets of symbols to represent the sounds of natural languages. (Tadesse Kassa, 2018). Geez language has its own scripts known as fidal/ hoheyat (**ፊደል/ ሆሄያት**).

2.6.1. Geez Characters

Before the first Patriarch for Ethiopian Aba Frimentatos, Geez was written from right to left but now it is written from left to right (Memhir Abebe Betemariam, 2020). **አቡጊዳ** and **ሀሁ** were the two types of Geez scripts called previous and current (the table below shows this).

a	ግዕዝ	ካዕብ	ሣልስ	ራብዕ	ኃምስ	ሳድስ	ሳብዕ
፩	አ	ቡ	ጊ	ዳ	ሄ	ዉ	ዞ
፪	በ	ጉ	ዲ	ሃ	ዌ	ዝ	ሆ
፫	ገ	ዱ	ሂ	ዋ	ዜ	ሕ	ኆ
፬	ደ	ሁ	ዊ	ዛ	ሔ	ኅ	ጦ
፭	ሀ	ዉ	ዚ	ሓ	ኄ	ጥ	ዮ
፮	ወ	ዙ	ሒ	ኃ	ጤ	ይ	ኮ
፯	ዘ	ሑ	ኂ	ጣ	ዬ	ክ	ሎ
፰	ሐ	ኁ	ጢ	ያ	ኬ	ል	ሞ
፱	ኀ	ጡ	ዪ	ካ	ሌ	ም	ኖ
፲	ጠ	ዩ	ኪ	ላ	ሜ	ን	ሦ
፲፩	የ	ኩ	ለ	ማ	ኔ	ሥ	ዖ
፲፪	ከ	ሉ	ሚ	ና	ሤ	ዕ	ፎ
፲፫	ለ	ሙ	ኒ	ሣ	ዔ	ፍ	ጾ
፲፬	መ	ኑ	ሢ	ዓ	ፌ	ጽ	ፆ
፲፭	ነ	ሡ	ዒ	ፋ	ጼ	ፅ	ቆ
፲፮	ሠ	ዑ	ፊ	ጻ	ፄ	ቅ	ሮ
፲፯	ዐ	ፉ	ጺ	ፃ	ቄ	ር	ሶ
፲፰	ፈ	ጹ	ፂ	ቃ	ሬ	ስ	ቶ
፲፱	ጸ	ፁ	ቂ	ራ	ሴ	ት	ጶ
፳	ፀ	ቁ	ሪ	ሳ	ቴ	ጵ	ፖ
፳፩	ቀ	ሩ	ሲ	ታ	ጴ	ፕ	ኦ
፳፪	ረ	ሱ	ቲ	ጳ	ፔ	እ	ቦ
፳፫	ሰ	ቱ	ጲ	ፓ	ኤ	ብ	ጎ
፳፬	ተ	ጱ	ፒ	ኣ	ቤ	ግ	ዶ
፳፭	ጰ	ፑ	ኢ	ባ	ጌ	ድ	ሆ
፳፮	ፐ	ኡ	ቢ	ጋ	ዴ	ህ	ዎ

b	ግዕዝ	ካዕብ	ሣልስ	ራብዕ	ኃምስ	ሳድስ	ሳብዕ
፩	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
፪	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ
፫	ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ
፬	መ	ሙ	ሚ	ማ	ሜ	ም	ሞ
፭	ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ
፮	ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ
፯	ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ
፰	ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ
፱	በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ
፲	ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ
፲፩	ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ
፲፪	ነ	ኑ	ኒ	ና	ኔ	ን	ኖ
፲፫	አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ
፲፬	ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ
፲፭	ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ
፲፮	ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ
፲፯	ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
፲፰	የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ
፲፱	ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ
፳	ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ
፳፩	ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ
፳፪	ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ
፳፫	ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ
፳፬	ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ
፳፭	ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ
፳፮	ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ

c	ሏ	ሗ	ሟ	ሧ	ሯ	ሷ	ቋ	ቧ	ቷ	ኗ	ኧ
	ኳ	ቛ	ዟ	ዷ	ጓ	ጧ	ጷ	ጿ	ፇ	ፏ	ፗ

Table 2: a) Ancient Geez Scripts b) Current Geez Scripts c) Derived Geez Scripts

2.6.2. Geez Numerals (ኍልቈ ግዕዝ /አሐዝ)

Geez language also has its own numerals for designating numbers. These numbers are used to write Ethiopian unique yearly calendar. Table 2 below shows the Geez numerals.

Geez Numerals	Name in Geez	Corresponding Arabic Numeral	Name in English	Name in Afaan Oromoo
፩	አሐዱ	1	One	Tokko
፪	ክልኤቱ	2	Two	Lama
፫	ሠለስቱ	3	Three	Sadii
፬	አርባዕቱ	4	Four	Afur
፭	ሐምስቱ	5	Five	Shan
፮	ስድስቱ	6	Six	Ja'a
፯	ሰብዓቱ	7	Seven	Torba
፰	ስመንቱ	8	Eight	Saddeet
፱	ተሰዓቱ	9	Nine	Sagal
፲	አሰርቱ	10	Ten	Kudhan
፲፩	አሰርቱ ወአሐዱ	11	Eleven	Kudha Tokko
፲፱	አሰርቱ ወተሰዓቱ	19	Nineteen	Kudha Sagal
፳	ዕሥራ	20	Twenty	Digdama
፴	ሠላሳ	30	Thirty	Soddoma
፵	አርብዓ	40	Forty	Afurtama
፶	ኀምሳ	50	Fifty	Shantama
፷	ስድሳ	60	Sixty	Jaatama
፸	ሰብዓ	70	Seventy	Torbaatama
፹	ሰማንያ	80	Eighty	Saddeettama
፺	ተስዓ	90	Ninety	Sagaltama
፩፻	አሐዱ ምእት	100	One Hundred	Dhibba Tokko
፪፻	ክልኤቱ ምእት	200	Two Hundred	Dhibba Lama
፱፻	ተሰዐቱ ምእት	900	Nine Hundred	Dhibba Sagal
፲፻	አሰርቱ ምእት	1000	One Thousand	Kuma Tokko
፳፻	ዕሥራ ምእት	2000	Two Thousand	Kuma Lama
፺፻	ተስዓ ምእት	9000	Nine Thousand	Kuma Sagal
፩፻፻	አሐዱ እልፍ	10,000	Ten Thousand	Kuma Kudhan
፪፻፻	ክልኤቱ እልፍ	20,000	Twenty Thousand	Kuma Digdama
፲፻፻	አሰርቱ እልፍ/አእላፍ	100,000	One Hundred Thousand	Kuma Dhibba Tokko
፻፻፻	አእላፉት	1,000,000	One Million	Miiliyoona Tokko
፲፻፻፻	ትእልፊት	10,000,000	Ten Million	Miiliyoona Kudhan
፻፻፻፻	ትእልፊታት	100,000,000	One Hundred Million	Miiliyoona Dhibba Tokko
፲፻፻፻፻	ምእልፊት	1,000,000,000	One Billion	Biiliyoona Tokko

Table 3: Geez Numerals (ኍልቈ ግዕዝ/አሐዝ)

2.6.3. Geez Month's Numbers Nomenclature

Geez language has the name for counting the dates of months between the start and end date or between 1 and 30. This issue was summarized in the following table (Table 3).

Geez Months Numbers	Name of Month's Date number in Geez	Name of Month's Date number in English	Name of Month's Date number in Afaan Oromoo
፩	አመ አሚሩ ለሰኔ	On 1st June	Gaafa Waxabajjii 1
፪	አመ ሰኑዩ ለሰኔ	On 2nd June	Gaafa Waxabajjii 2
፫	አመ ሠሉሱ ለሰኔ	On 3rd June	Gaafa Waxabajjii 3
፬	አመ ረቡኡ ለሰኔ	On 4th June	Gaafa Waxabajjii 4
፭	አመ ኀሙሱ ለሰኔ	On 5th June	Gaafa Waxabajjii 5
፮	አመ ሰዱሱ ለሰኔ	On 6th June	Gaafa Waxabajjii 6
፯	አመ ሰቡዑ ለሰኔ	On 7th June	Gaafa Waxabajjii 7
፰	አመ ሰሙኑ ለሰኔ	On 8th June	Gaafa Waxabajjii 8
፱	አመ ተሱዑ ለሰኔ	On 9th June	Gaafa Waxabajjii 9
፲	አመ ዐሱሩ ለሰኔ	On 10th June	Gaafa Waxabajjii 10
፲፩	አመ ዐሱሩ ወአሚሩ ለሰኔ	On 11th June	Gaafa Waxabajjii 11
፲፪	አመ ዐሱሩ ወሰኑዩ ለሰኔ	On 12th June	Gaafa Waxabajjii 12
፲፫	አመ ዐሱሩ ወሠሉሱ ለሰኔ	On 13th June	Gaafa Waxabajjii 13
፲፬	አመ ዐሱሩ ወረቡዑ ለሰኔ	On 14th June	Gaafa Waxabajjii 14
፲፭	አመ ዐሱሩ ወኀሙሱ ለሰኔ	On 15th June	Gaafa Waxabajjii 15
፲፮	አመ ዐሱሩ ወሰዱሱ ለሰኔ	On 16th June	Gaafa Waxabajjii 16
፲፯	አመ ዐሱሩ ወሰቡዑ ለሰኔ	On 17th June	Gaafa Waxabajjii 17
፲፰	አመ ዐሱሩ ወሰሙኑ ለሰኔ	On 18th June	Gaafa Waxabajjii 18
፲፱	አመ ዐሱሩ ወተሱዑ ለሰኔ	On 19th June	Gaafa Waxabajjii 19
፳	አመ ዕሥራሁ ለሰኔ	On 20th June	Gaafa Waxabajjii 20
፳፩	አመ ዕሥራ ወአሚሩ ለሰኔ	On 21st June	Gaafa Waxabajjii 21
፳፪	አመ ዕሥራ ወሰኑዩ ለሰኔ	On 22nd June	Gaafa Waxabajjii 22
፳፫	አመ ዕሥራ ወሠሉሱ ለሰኔ	On 23rd June	Gaafa Waxabajjii 23
፳፬	አመ ዕሥራ ወረቡዑ ለሰኔ	On 24th June	Gaafa Waxabajjii 24
፳፭	አመ ዕሥራ ወኀሙሱ ለሰኔ	On 25th June	Gaafa Waxabajjii 25
፳፮	አመ ዕሥራ ወሰዱሱ ለሰኔ	On 26th June	Gaafa Waxabajjii 26
፳፯	አመ ዕሥራ ወሰቡዑ ለሰኔ	On 27th June	Gaafa Waxabajjii 27
፳፰	አመ ዕሥራ ወሰሙኑ ለሰኔ	On 28th June	Gaafa Waxabajjii 28
፳፱	አመ ዕሥራ ወተሱዑ ለሰኔ	On 29th June	Gaafa Waxabajjii 29
፴	አመ ሠላሳሁ ለሰኔ	On 30th June	Gaafa Waxabajjii 30

Table 4: Geez Month's Numbers Nomenclature

2.6.4. Geez Ordinal Numbers (ኍልቈ መዐርግ)

Geez use ordinal numbers like any other language to indicate the rank of inequalities. This is described in the following table. The ordinal numbers of Geez follow the following pattern (Memhir Abebe Betemariam, 2020).

Geez	Engilish	Afaan Oromoo
ቀዳማይ	First (1st)	Tokkoffaa (1ffaa)
ካልአይ	Second (2nd)	Lammaffaa (2ffaa)
ሣልሳይ	Third (3rd)	Sadaffaa (3ffaa)
ራብዓይ	Fourth (4th)	Arfaffaa (4ffaa)
ኃምሳይ	Fifth (5th)	Shanaffaa (5ffaa)
ሳድሳይ	Sixth (6th)	Ja’affaa (6ffaa)
ሳብዓይ	Seventh (7th)	Torbaffaa (7ffaa)
ሳምናይ	Eighth (8th)	Saddettaffaa (8ffaa)
ታስዓይ	Ninth (9th)	Saglaffaa (9ffaa)
ዓሥራይ	Tenth (10th)	Kurnaffaa (10ffaa)

Table 5: Geez Ordinal Numbers (ኍልቈ መዐርግ)

2.6.5. Geez Punctuation Marks

There are different punctuation marks in Geez language. There is No question mark in Geez, instead interrogative sentences use either three dots (፧) or Geez full stop (።). The following punctuation marks were used in Geez writing.

- ❖ Ge'ez Comma (፣) – used to list items
- ❖ Ge'ez Semicolon (፤) – to combine two sentences as one sentence
- ❖ Ge'ez Wordspace (፡) – Now a days replaced by white space between words.
- ❖ Ge'ez Preface Colon (፦) - To discuss something with additional concepts.
- ❖ Ge'ez Colon (፥) – shows something is going to be listed.
- ❖ Ge'ez Full Stop (።) – mention the idea written was ended or may require an answer.
- ❖ Ge'ez 3-Dot Question Mark (፧) – used at the end of interrogative statement.

2.6.6. Word Class in Geez (ግዕዝ ክፍላተ ቃላት)

There are seven word classes in Geez (Memhir Abebe Betemariam, 2020): noun (**ስም**), verb (**ግስ**), adjective (**ቅጽል**), preposition (**መስተዋድድ**), article (**መስተፃምር**), Adverb (**ተውሳከ ግስ**), and pronoun (**ተውላጠ ስም**).

i. Noun (ስም)

Noun is a name that represents a person, places, animal, thing, feeling and idea. In Geez there are different types of nouns in general concert and abstract, common and proper, collective and countable and uncountable noun. Noun is a word in Geez that is used to identify or address an object. All objects having a definite and indefinite volume are called by a name. Examples include: **ሰብእ** (person), **እንስሳ** (animal), **ኅብስት** (bread), **ቅብዕ** (butter), **ሀገር** (country), **ለሊት** (night), **እድ** (hand), **ርእስ** (head) (Abba Teklehaymanot Weldu, 2018)

ii. Verb (ግስ)

Verb is a word used to describe an action, state, or occurrence, and forming the main part of the predicate of a sentence (Zeradawit Weldehanna, 2015). It is a word Geez that indicates an action that has been done. Verbs can be categorized in to two; root verbs and derived verbs (Like Hiruyan Balay Mekonnen, 2012). Root verbs are Geez verbs that are used as a base for other verb forms and those verbs that follow them use the same derivation rules as their modal verb. There are eight root verbs in Geez **ቀተለ** (ajjeese), **ቀደሰ** (galateeffate), **ገብረ** (hojjete/uume), **አእመረ** (beeke), **ባረከ** (eebbise), **ሤመ** (muude), **ብህለ** (jedhe), **ቆመ** (dhaabbate). All verbs in geez start their derivation in past tense form (ቀዳማይ አንቀጽ) (Like Hiruyan Balay Mekonnen, 2012). As an example Derivation of root verb **ቀተለ** (ajjeese) is shown below (Abba Teklehaymanot Weldu, 2018).

- **ቀተለ** - killed - ajjeese
- **ይቀትል** - kill - ni ajjeesa
- **ይቅትል** – will kill - haa ajjeesu
- **ይቅትል** – to kill - akka ajjeesuuf
- **ቀቲል** – killing - ajjeechaa
- **ቅቱል** - killer - ajjeesaa
- **ቅትለት** - being killed, kill each other - ajjeefamuu, walajjeesuu *etc.*

iii. Adjective/ቅጽል

Adjectives are words or constructions used to qualify nouns. Adjective is a word in geez that is used to convey additional information about a noun. It gives detail like state of being, physical appearance, distance (near or far), structure and color of the noun under consideration (Zeradawit Weldehanna, 2015). The underlined words below are examples of adjectives.

- ሐጺር ወልድ - short boy - gurbaa gabaabaa
- ጸሊም ልብስ - black cloth - hoccuu gurraacha
- ጸዓዳ ብዕራይ - white ox - sangaa adii
- ነዊኅ ብእሲ - tall person - namicha dheeraa

iv. Preposition (መስተዋድድ)

Prepositions in Geez language are divided in to two (Zeradawit Weldehanna, 2015).

a. Prepositions that fall onto nouns to indicate start and end, direction, comparison, time, place etc. examples are **እም** (irraa), **ኀበ** (gara), **ከመ** (akka), **በ** (tiin), **ለ** (f), **አሜ** (yeroo), **ውስተ** (keessa), **አፍኣ** (ala).

- **ኀበ ቤተክርስቲያን** - to church – gara mana kiristaanaa
- **ከመ እግዝአብሔር** - like God – akka Waaqayyoo
- **እም ገሊላ** - from Galila – Galiilaa irraa

b. Prepositions having (of, ‘s) meaning for example **ዘ ፣ እንተ ፣ እለ** (kan) those that bear the meaning only after (**ከመ/መጠነ** - akka) and those used as a conjunction (**እስመ ፣ አምጣነ ፣ አኮኑ ፣ ወ** - fi/and)

- **ዮም ዘመጸአ ብእሲ** - person who came today - namicha har’a dhufe
- **ትማልም እንተ መጽአት ብእሲት** -woman who came yesterday - dubartii kaleessa dhufte
- **ከመ ሰማዕከ ንግረኒ** - tell me just after you hear – akka dhageesseen natti himi
- **አኮኑ ይመጽእ መምህር** - come, teacher is coming - Barsiisaan dhufaa jiraa koottu

v. Article (መስተፃምር)

Article is a word that joins sentences together such as **ወ** (fi), **አው፣ሚመ** (yookiin), **ዓዲ** (dabalataan), **ሂ** (akkasumas), **ባህቱ**(ta’us garuu), **አላ**(malee), **እንበይነዝ፣ በእንተዝ** (kanaaf), **እምዝ** (iyyu), **እምዴኅረዝ** (kanaan booda) etc. Are examples of articles in Geez (Like Hiruyan Balay Mekonnen, 2012).

vi. Adverb (ተውሳከ ግስ)

As adjectives (**ቅጽል**) give additional meaning to nouns, adverbs (**ተውሳከ ግስ**) are words that give additional meaning to verbs. They give information like how, why, where, when, by who etc. about verbs (Tadesse Kassa, 2018). The types of adverbs in Geez are described below.

- ❖ Adverb of manner (**ኩነታዊ**) for example **ፍጡን** - ariitiin (quickly)
- ❖ Adverb of place (**መካናዊ**) for example **ዝየ** -- as (here)
- ❖ Adverb of time (**ጊዛያዊ**) for example **ዮም** -- har’a (today)
- ❖ Adverb of frequency (**የደጊመ ጊዜ**) for example **ኰለሄ** -- yeroo hunda (usually)
- ❖ Adverb of certainty for example **እሙነ** -- **በርግጠኝነት** (surely)
- ❖ Adverb of degree for example **ጽዴቀ** -- **በአግባቡ** (fairly)
- ❖ Interrogative for example **ማእዜ** -- yoom (when)
- ❖ Relative for example **በጽባሕ** -- ganamaan (in the morning)

vii. Pronoun (መራሕያን/ተውላጠ ስም)

A **pronoun** is a word that substitutes for a noun or noun phrase. Pronouns (**መራሕያን**) can be used instead of noun, verb to be and adjectives. There are different types of pronouns namely personal, demonstrative, interrogative, indefinite, and possessive pronoun.

a. Personal Pronoun (ምድብ ተውላጠ ስም)

In Geez pronouns can be classified as singular and plural, masculine and feminine, and near and far. These are shown in the table 5 below:

Category	**Pronoun (ተውላጠ ስም)**			Gender			Type	
	ግዕዝ	A/Oromoo	English	Male	Female	Common	plural	Singular
1st Person/ Ramaddii 1ffaa	**አነ**	Ana	I			✓		✓
	ንሕነ	Nuyi	We			✓	✓	
2nd Person / Ramaddii 2ffaa (**ካልአይ መደብ**)	**አንተ**	Si'i	You	✓				✓
	አንቲ				✓			✓
	አንትሙ	Isin	Yours	✓			✓	
	አንትን				✓		✓	
3rd Person/ Ramaddii 3ffaa (**ሣልሳይ መደብ**)	**ውእቱ**	Isa	He	✓				✓
	ይእቲ	Ishee	She		✓			✓
	ውእቶሙ	Isaan	They			✓	✓	
	ውእቶን	Isin (kabajaaf)				✓	✓	✓

Table 6: Personal Pronoun (ምድብ ተውላጠ ስም)

We cannot talk about grammar without pronoun, because a pronoun tells about category of person (1st, 2nd and 3rd), near or far and gender (Masculine and Feminine) As you see from the Table 5 in Afaan Oromoo '***isin***' is used to express both Masculine and Feminine pronouns in 2nd person pronoun and '***isin*** ' used to express our respect to those are older than the speaker in 3rd pronoun, due to this it is called respect pronoun.

In Geez, for each Gender in 2nd and 3rd personal pronoun plural form each have Masculine and feminine form. In Afaan Oromoo the plural form of **'*si'i*'**, **'*isa*'** and **'*ishee*'** are **'*isin*'** and **'*isaan*'** respectively for both Masculine and feminine. Pronoun in Geez can be used being

Subject in leading the sentence. Let’s see example with verb **አእመረ**(beeke) (Memhir Abebe Betemariam, 2020).

- **አነ አአምር** - I know - Ani nan beeka
- **ንሕነ ነአምር** - We know - Nuyi ni beekna
- **አንተ ተአምር** - You know (Masculine) - Ati ni beekta (dhiiraaf)
- **አንቲ ተአምሪ** - You know (Feminine) - Ati ni beekta (dubaraaf)
- **አንትሙ ተአምሩ** - You know (Masculine) - isin ni beektu (dhiirotaaf)
- **አንትን ተአምራ** - You know (Feminine) - isin ni beektu (dubartootaaf)
- **ዊእቱ፡ የአምር** - He knows - Inni ni beeka
- **ይእቲ፡ ተአምር** - She know - Isheen ni beekti
- **ዊእቶሙ የአምሩ** - They know (Masculine) - Isaan ni beeku (dhiirotaaf)
- **ዊእቶን የአምራ** - They know (Feminine) - Isaan ni beeku (dubartootaaf)

b. Demonstrative’s pronouns (አመልካች)

A demonstrative pronoun stands in for a person, place or thing and can function as a subject, an object or an object of the preposition. It is used before a verb of the sentence not before a noun. In Geez the following demonstrative pronouns exist.

Near					Far				
Gende	Singular		Plural		Gende	Singular		Plural	
	Geez	A/O	Geez	A/O		Geez	A/O	Geez	A/O
Male	**ዝንቱ፡/ ዝ**	Kana/ isa kana dha	**እሎ/እሉ/እሎንቱ፡**	Isaan kana/ Isaan kana dha	Male	**ዝኩ/ ውእቱ፡/ ዝክቱ፡/ ዝስኩ**	Sana/ Sana dha	**ውእቶሙ /እልኩ/ እልክቱ፡**	Isaan sana/ Isaan sana dha
Female	**ዛቲ/ዛ**	Ishee kana /ishee kanadha	**እሎን/ እላ/ እላንቱ፡**		Female	**ይእቲ/ እንታክ/ እንትኩ**	Ishee kana/ Ishee kana dha	**ውእቶን/ እልኮን/ እልክቶን**	

Table 7: Demonstrative pronouns in Geez

Example, **ዝንቱ ውእቱ ወነጌለ መንግስተ ሰማያት።** Wangeelli Mootummaa Waaqayyoo kana dha.

c. Possessives (አገናዛቢዎች)

Possessive pronouns are words that demonstrate ownership or possession. Possessive pronouns show that something belongs to someone or something. In Geez and Afaan Oromoo the following are possessive markers or suffixes. The table below show possessive suffixes of Geez with example.

Category	Singular			Plural		
	Geez	A/O	English	Geez	A/O	English
1st person	ዚአየ	Kan koo kiyya	Mine	ዚአነ	Kan keenya	Ours
2nd Person	ዚአከ	Kan kee/ dhiiraaf	Yours	ዚአክሙ	Kan keessan (Dhiirotaaf)	Yours (Male)
	ዚአኪ	Kan kee/ dubaraaf	Yours	ዚአክን	Kan keessan (Dubartootaaf)	Yours (Female)
3rd Person	ዚአሁ	Kan isaa	His	ዚአሆሙ	Kansaanii (Dhiirotaaf)	Their (Male)
	ዚአሃ	Kan ishee	Her	ዚአሆን	Kansaanii (Dubartootaaf)	Their (Female)

Table 8: possessive pronoun in Geez

As it had shown in table 8 above possessive pronoun in Geez has gender (Masculine and Feminine) different name which common in Afaan Oromoo and in English Languages. For example, in 2nd possessive pronoun ዚአከ and ዚአኪ respectively stands for Male and Female but in Afaan Oromoo both of it takes '***kankee' (Yours).***

2.6.7. Sentences Structure in Geez

A Geez sentence is a sequence of words of Geez language that is used by its speakers to express their thought/idea in spoken or written form. There are three ways to form a sentence in Geez as follows SVO (Subject-Verb-Object), SOV (Subject Object Verb) and VSO (Verb-Subject-Object). Example:

- ❖ ኤልያስ መሀረ ትምህርት - Eeliyaas barannoo barsiise - Elias taught lesson. (SVO)

- ❖ **ቀተለ ሙሴ አርዌ** - Muuseen bofa ajjeese - Muse killed snake. (VSO)
- ❖ **አብርሃም ሠዉዓ ብግዕ** - Abrahaam hoolaa qale – Abraham slaughtered sheep. (SVO)

The syntactic structure is formed by combining different words in sequence. As it can be seen from the above examples, the order of words in Geez and Afaan Oromoo is somewhere different. This issue is one of the challenges in machine translation.

2.7.Afaan Oromoo Writing Systems

2.7.1. Afaan Oromoo Characters

The writing system of Afaan Oromo is called Qubee, a Latin alphabet (Ibrahim Bedane, 2015)Afaan Oromo has 33 total characters, of which 5 vowels, 7 double consonants and 21 consonant phonemes, i.e., sounds that make a difference in word meaning. Afaan Oromo vowels are represented by the letters, a, e, i, o, and u, or long vowels: aa, ee, ii, oo, and uu. The length of the vowel makes a difference in word meaning. For example: Lafa (ground) and Laafaa (soft). The complete list of Afaan Oromo alphabet is listed in Table 8 below.

Number	**Capital**	**Small**	**IPW**	**Type**	**Long**	**Short**
1	A	a	/a/	Vowel	AA/aa	A /a
2	B	b	/b/	Consonant	-	-
3	C	c	/č’/	Consonant	-	-
4	D	d	/d/	Consonant	-	-
5	E	e	/e/	Vowel	EE/ee	E/e
6	F	f	/f/	Consonant	-	-
7	G	g	/g/	Consonant	-	-
8	H	h	/h/	Consonant	-	-
9	I	i	/i/	Vowel	II/ii	I/i
10	J	j	/ğ/	Consonant	-	-
11	K	k	/k/	Consonant	-	-
12	L	l	/l/	Consonant	-	-
13	M	m	/m/	Consonant	-	-
14	N	n	/n/	Consonant	-	-
15	O	o	/o/	Vowel	OO/oo	O/o
16	P	p	/p/	Consonant	-	-
17	Q	q	/q/	Consonant	-	-
18	R	r	/r/	Consonant	-	-
19	S	s	/s/	Consonant	-	-
20	T	t	/t/	Consonant	-	-
21	U	u	/u/	Vowel	UU/uu	U/u
22	V	v	/v/	Consonant	-	-
23	W	w	/w/	Consonant	-	-
24	X	x	/t’/	Consonant	-	-

25	Y	y	/y/	Consonant	-	-
26	Z	z	/z/	Consonant	-	-
27	CH	ch	/č/	Double consonant	-	-
28	DH	dh	/đ/	Double consonant	-	-
29	NY	ny	/ň/	Double consonant	-	-
30	PH	ph	/p’/	Double consonant	-	-
31	SH	sh	/š/	Double consonant		
32	TS	ts	/s’/	Double consonant		
33	ZH /ž	zh	/ž	Double consonant		

Table 9: Afaan Oromo alphabet

2.7.2. Afaan Oromoo Numeral (Lakkoofsota Afaan oromoo)

Numerals include words that refer to order or quantity of something. Most of the time the position of numerals follows the category with which they form a syntactic unit (Abebe Medeksa, 2016). Afaan Oromo numerals can be cardinal or ordinal. The following are the cardinal numerals in Afaan Oromo: ‘tokko’ (one), ‘lama’ (two,) ‘kudhan’ (ten), ‘dhibba tokko’ (one hundred), ’kuma lama’ (two thousand), *etc*. In Afaan oromoo Arabic numbers are used to figure out countable things. Numbers come after the noun they modify, for example “four” is “Afur” , “three Orange” is “Burtukana sadii”, just as “two birr” is “qarshii lama” and “three hundred” is dhibba sadi. Also, in Afaan Oromoo ordinal numbers are used to express the rank of something (Getachew Emiru, 2016). This Ordinal numbers are formed by adding the suffix -ffaa or -affaa to the number. For example, “tokkoffaa” is “ first”, “lammaffaa” is “second”, “sadaffaa” is “third” and continued in such a way.

2.7.3. Afaan Oromo Punctuation Marks

The most commonly used punctuation marks in Afaan Oromo are the following:

- Period (**.**) - is placed at the end of declarative sentences, statements thought to be complete and after many abbreviations.
- Question mark (**?**) - is used to indicate a direct question when placed at the end of a sentence. For example: Isa ni jaalattaa? (Do you love him?)
- Exclamation mark (**!**) - is used at the end of command and exclamatory sentences.
- Comma (**,**) - is used to show a separation of ideas or elements within the structure of a sentence. For example: Gabaa deemee buna, ashaboo, mimmixaa fi mi’eessituu bite.
- Colon (**:**) **-** is used to separate and introduce lists, clauses, and quotations, along with several conventional uses.

- Semi colon **(;) -** is used to connect independent clauses. It shows a closer relationship between the clauses than a period would show.
- An apostrophe mark (**'**) - in Afaan Oromo apostrophe is used to represent a glitch called hudhaa sound. It is used to write the word in which most of the time two vowels appeared together.

2.7.4. Afaan Oromoo Word Classes

Word classification is a classification of word based on word semantic or semantic coherence rather than synthetic or meaning of the word in a sentence. This classification depends on the word 's contribution and meaning in a sentence. Let's discuss some of the basic Afaan Oromoo word classes, which are standard for most Linguists. These include: noun (Maqaa), pronoun (Bamaqaa), adjective (Ibsamaqaa), verb (Xumura), adverb (IbsaXumura), conjunction (Walqabsiistota), preposition (Durduuba) and Interjection (Rajjeeffannoo) (Getachew Emiru, 2016).

i. Afaan Oromoo Nouns (Maqaa)

Afaan Oromo nouns are words used to represent name or identify any of categories of things, people, places or ideas or a particular one of these entities. Whatever exists, we assume, can be named, and that name is a noun (Abebe Medeksa, 2016). Many (but not all) Oromo nouns inflect for gender (masculine, feminine), while all inflect for number (singular - specific vs. non-specific, plural) and case (nominative, accusative, dative, genitive, instrumental, locative, ablative, and vocative)

As in other languages like English, nouns in Afaan Oromoo have different types or classes. There are proper and common nouns, collective nouns, basic noun, derived noun and concrete and abstract nouns. Proper nouns (maqaa dhunfaa) are nouns that represent a unique entity like a specific person, place, river, building and country. In Afaan Oromoo language Proper nouns are used in capitalized. For examples;

- Adama , Nekemte and sebeta
- Awash, Dabana and Gibe

The first example is when noun used to represent a place and on the second nouns represent river. Common nouns are nouns which describe an entire group of entities. Common nouns are those nouns that have similar characters and grouped under the same entity. The group is a single unit, but it has more than one member. For instance: the word “nama “(people) can represent male, female, child and young. Also there are words such as “mana”, “barsiisaa”, “saree” and “muka” are used as a common noun. Common nouns in Afaan Oromoo sentence can be used as follows:

- ❖ Boonaan <u>baratаа</u> cimaadha - Bona is the clever <u>student</u>
- ❖ Tolaan <u>muka</u> jala jira. – Tola is under the <u>tree</u>

The underlined word is representing the common nouns. The common nouns in Afaan Oromoo can use affixes to show the plurality of nouns and this behavior of adding affix make it different from personal and collective nouns.

Collective nouns are nouns when a group of different entities described by one entity. For example, the word forest “bosona” and food “nyaata” are used as collective noun in Afaan Oromoo. Concrete nouns refer to their ability to register on your five senses. If you can see, hear, smell, taste, or feel the item, it's a concrete noun. Abstract nouns on the other hand refer to abstract objects such as ideas or concepts.

Nouns in Afaan Oromo are treated as either male or female, though there are typically no gender markers in the words themselves. Gender can be shown through a demonstrative pronoun, a definite article, a gender-specific adjective, or the verb form (if the noun is a subject). The notable exceptions are those nouns derived from verbs, where the masculine noun adds an -aa suffix and the feminine noun adds a -tuu suffix to the verb root. For example the English word teacher for masculine “ barsiisaa” and for feminine “barsiistuu”. Also for English word student to masculine we use ―barataa and for feminine we use “barattuu”.

ii. Afaan Oromoo pronouns (Bamaqaa)

In grammar, a pronoun is defined as a word or phrase that may be substituted for a noun or noun phrase, which once replaced, is known as the pronoun ‘s antecedent. It can do everything that nouns can do. This is used in place of noun in most cases when a noun is prestated. A pronoun can act as a subject, direct object, indirect object, object of the preposition, and more.

Without pronouns, we would have to keep on repeating nouns, and that would make our speech and writing redundant, not to mention cumbersome. Most pronouns are very short words. In the following sentence we can see a noun and their corresponding pronoun in Afaan Oromoo language

- Calaan mana barumsa deeme. Inni mana barumsa deeme.
- Toltuun naato-qabettiidha. Isheen naato-qabettiidha.

The underlined words show that when the pronoun represent or used instead of nouns or as a noun. In Afaan Oromo, there are different categories of pronoun, which includes Personal pronoun (bamaqaa Ramaddii), demonstrative pronoun (bamaqaa akeeektuu), double pronoun (bamaqaa mirree), Possessive pronoun (bamaqoota qabeenyaa), Reflexive pronoun (bamaqaa ulfinaa), reciprocal pronouns (bamaqaa Waliyyoo), relative pronoun (bamaqaa firomsee), Interrogative pronoun (Bamaqoota iyyafannoo) and indefinite pronoun (bamaqaa waalleyyuu) (Getachew Emiru, 2016)

Oromo uses plural pronouns (isin and isaan) also as the respect/formal pronouns. Mostly, one uses the respect form when talking to/about older and respected members of the community. In many areas of Oromia, ati is rarely used (and considered rude) and only the respect form of "you", isin, is used. The personal pronouns as subjects and direct objects are listed below along with possessive markers.

Category	Subject Pronouns		Direct Object Pronouns		Possessive Pronouns	
	English	A/Oromoo	English	A/Oromoo	English	A/Oromoo
1st singular	I	Ani, Ana	Me	Na	My, Mine	Koo, kiyya, kan kooti,
1st plural	We	Nuti	Us	Nu	Our, Ours	Keenya, kan keenya
2nd singular	You(sng)	Ati Si'i	You	Si	Your, Yours	Kee, kan keeti
2nd	You(pl)	Isin	You(pl)	Isin	Your,	Keessan,

plural		Sin			Yours	kan keessani
3rd singular	He, it	Isa, Inni	Him, it	Isa	His, its	Kan isaa
	She	Isheen/ishii	Her	Ishee	Her, Hers	Kan ishee
3rd plural	They	Isaan	Them	Isaani	Their, Theirs	Kan isaanii

Table 10: Afaan Oromoo common Pronouns

iii. Afaan Oromoo Adjectives (Ibsa Maqaa)

In Afaan Oromo adjectives (addeessaa) come after the nouns they qualify (Gelan Tullu, 2020). For example: in the following adjectival phrases, uffata adii (white cloth) and muka gabaabaa (short stick), adii (white) and gabaabaa (short) are adjectives that qualifies the nouns uffata and muka respectively. In the following examples, the highlighted words are adjectives:

- Isaan mana **bareedaa** kessa jiraatu - They live in a **beautiful** house.
- Asaantiin har’a uffata **gurraacha** uffachaa jirti - Asanti is wearing **black** cloth today.

In Afaan Oromoo language adjectives are categorized into basic adjectives (maqibsa bu‘uuraa), demonstratives adjectives (maqibsa akeektuu), possessive adjectives(maqibsa qabeenyaa), interrogative adjectives(maqibsa iyyafannoo) and Manner adjectives

iv. Afaan Oromoo verb (Xummura Afaan Oromoo)

A verb (xummura) is a word that expresses an action, state of being in or relationship between two things. In Afaan Oromo verbs mostly appear at the end of a sentence (Abebe Medeksa, 2016). Consider the following example:

- Guutaan kaleessa deeme. - Guta went yesterday.
- Caalaan kubbaa dhiite. - Chala kicked a ball.

In this example, the underlined words “deeme”(went) and “dhiite”(kicked) are verbs of the sentence. Most Afaan Oromoo verbs are in their infinitive form, for example, deemuu (to go). The verb stem “deem-” is the infinitive form ‘deemuu’ with the final ‘–uu’ dropped. Afaan Oromo verbs can be categorized into main (transitive or intransitive) and auxiliary verbs (Abebe Medeksa, 2016). Transitive verbs are main verbs which transfer message to complements or objects. Consider the following examples:

Kumaan bishaan waraabe. - Kuma fetched water.

Kumaan ulee cabse. - Kuma broke a stick.

Each of the verbs, waraabe (fetched) and cabse (broke) in these sentences have objects that complete the verbs' actions. Intransitive verbs are main verbs which do not take object or complement in a sentence. For example: in the sentence, Ijoolleen rafan (Children slept), it is impossible for an object to follow the verb rafan (slept).

Auxiliary verbs support the main verbs used in a sentence, add functional or grammatical meaning to the clauses in which they appear. For example:

- ❖ Tolaan kaleessa ganama fiigaa ture. - Tola was running yesterday morning.
- ❖ Taphni ijoolleef faayidaa baay'ee qaba. - playing has many advantages for children.

In the above sentences the words 'ture' and 'qaba' are auxiliary verbs. The following are Afaan Oromoo auxiliary verbs 'dha', 'ta`e', 'qaba', 'ture', 'jira', etc. Afaan Oromoo verbs take subject markers such as '-e', '-ine', '-ite' and '-ani' for subjects I, we, she and they respectively to agree with the subject of the sentences, as shown in the following examples:

- ❖ Ani isan gors**e.** - I advised him.
- ❖ **Nu`i** isa gors**ine.** - We advised him.
- ❖ Isheen isa gors**ite.** - She advised him.
- ❖ Isaan isa gors**ani.** - They advised him.

v. Afaan Oromoo Adverb (Ibsa xummuraa Afaan Oromoo)

Adverbs are words which are used to modify verbs. In Afaan Oromoo adverbs come before the verb they modify. Afaan Oromoo adverbs are categorized as adverbs of time, place and manner (condition) (Gelan Tullu, 2020)

a. **Adverbs of time** show the time the action takes place. The following are the words that can be used as adverbs of time in Afaan Oromoo language. 'amma' (now), 'boru' (tomorrow), 'kaleessa' (yesterday), 'yoom' (when) ,'har'a' (today), 'galgala' (tonight) etc. Consider the following example.

- ❖ Boonsaan **kalessa** dhufe. - Bonsa came **yesterday**.
- ❖ Qananiisaan **boru** ni fiiga. - Kenenisa will run **tomorrow**.

In these examples the word ‘kaleessa’(yesterday) and ‘boru’(tomorrow) are adverbs of time. Mostly adverbs of time answer the question of when the action takes place.

b. Adverbs of place show the place where the action takes place. The following are the words that can be used as adverb of place in Afaan Oromoo. ‘as’ (here), ‘achi’ (there), ‘gadi’ (below), ‘gubbaa’ (above), ‘jidduu’ (middle), ’irra’ (on), *etc*. Consider the following example,

- ❖ Tolaan **mana** jira. - Tola is at **home**.
- ❖ Inni konkolaataa **irra** jira. - He is **on** the car.

c. Adverb of manner show how the action of the sentence is done (Abebe Medeksa, 2016). The following are Afaan Oromoo words that can be used as adverb of manner ‘ariitin’ (quickly), ‘suuta’ (slowly), ‘akka gaarii’ (well) *etc*. Consider the following example, the word ‘ariitin’(quickly),‘baay’ee’(very) and ‘suuta’ (slowly), are adverbs of manner.

- ❖ Inni **ariitin** fiiga. - He is running **quickly**.
- ❖ Abdiin **baay’ee** cimaa dha. - Abdi is **very** clever
- ❖ Gammachiis **suuta** deemuu jaallata. - Gamachis likes to walk **slowly**

vi. Afaan Oromoo Preposition (Durduube Afaan Oromoo)

A preposition links a noun to an action or to another noun. It links nouns to other parts of the sentence. Afaan Oromoo prepositions divided into two categories: true prepositions and postpositions, with true prepositions coming before the noun and postpositions coming after the noun, they relate to (Getachew Emiru, 2016).

- ❖ Keeniyaan Itoophiyaarraa **gara** kibbaatti argamti. - Kenya is located **to** the south of Ethiopia.
- ❖ Roobeeraan Boqonnaa**rra** jira - Robera is **on** vacation.

From the above examples, we can notice that the postpositions **itti**, **irra**, and **irraa** most often occur as suffixes, *-tti*, *-rra*, and *-rraa*, on the nouns they relate to. In Afaan Oromoo the use of postpositions is preferred and occurs with a higher frequency than the use of Prepositions.

vii. Afaan Oromoo Conjunction (Wal-qabsiistuu Afaan Oromoo)

Whereas prepositions link nouns to other parts of the sentence, conjunctions usually link more complete thoughts together. A word that can be used to join or connect two phrases, clauses or sentences is known as a conjunction. In Afaan Oromoo Conjunction can be divided into coordinating and subordinating conjunctions. Coordinating conjunctions are used to connect two independent clauses. Mostly, these conjunctions are used when the speaker needs to lay emphasis on the two sentences equally.

- ❖ Nyaatan barbaada sababiinsaa nan beela'e. - I want food because I am hungry.
- ❖ Shaayiin jaalladha garuu bunan caalaa jaalladha. - I like tea, but I like coffee more.

Some common Afaan Oromo conjunctions are: fi (and), garuu/immoo (but), yookin-for declaratives, moo-for questions (or), haa ta’u malee (however), *etc*. Afaan Oromo Subordinating conjunctions are yoo (if), akka waan (as if), sababiin isaa (because), kanaafuu (so, therefore), akka (so that, in order to), ta’us (though), tu’ullee (even though), yeennaa (when), hamma (until), erga (after), dursa (before), *etc*.

viii. Afaan Oromoo Introjections (Raajii Afaan Oromoo)

Interjections are words we can use to express our feeling, emotions for suddenly happening of situations. Afaan Oromoo also has own introjections words as other language Amharic and English. Those words are ishoo for showing happiness *wayyoo* for sadness *ah* for silent event or situation happened (Getachew Emiru, 2016).

- ❖ Ishoo! Baga gammadde. This means wow! Congratulation! The word “**ishoo”** is used to express pleasure of something or for showing happiness and is used as introjections word.

2.7.5. Sentences Structure in Afaan Oromoo

In Afaan Oromo, the sentence structure is subject-object-verb (SOV), where the subject comes first, followed by the object and the verb comes at the end of the given sentence (Workine Tesema & Duresa Tamirat, 2017). Afan Oromo and Geez are sometimes similar and other time different in sentence structuring. Afan Oromo uses SOV language pattern while Geez uses SOV, SVO and VSO. For example,

- ❖ Mooneerraan bilisa bahe - *Monerra* has got freedom “Monerraa“is a subject, “bilisa” is an object and “bahe” is a verb. Therefore, it has SOV structure.
- ❖ Caalaan nyaata nyaate – chala ate meal, “caalaan” is the subject, “nyaata” is the object and “nyaate” is the verb of the sentence.

Chapter Three

3. Research Methodology

Research methodology is a way to systematically plan for solving the research problem. It may be understood as a science of studying how research is done scientifically. The advantage of knowing the methodology of the study before doing the experiment is to reason out what, how and why the methods or the techniques are selected for the experiment to know the risks for conducting the research in detail.

In order to successfully complete this research, the following steps will be followed. As a first step, datasets available in Geez language from religious books will be collected. Since there is no online Geez – Afaan Oromo parallel sentences available, the parallel corpora will be prepared manually to produce the standard parallel corpora required for machine translation. Researcher will prepare 8000 Geez – Afaan Oromoo parallel sentences for this study. After datasets preparation experiment will be conducted. All necessary components will be installed on the computer to train the model. Then finally the model will be trained and tested using 80% (6400 sentences) of data for training and 20% (1600 sentences) of data for testing.

3.1.Research Design

Research design is a master plan specifying the methods and procedures for collection and analyzing the needed information (Prabhat Pandey & Meenu Mishra Pandey, 2015). This research will follow the experimental research design. The python programming language will be used to conduct the experiment, using lines of codes for recurrent neural network machine translation. The proposed model will be trained using RNN architecture. Training dataset and testing dataset should pass through preprocessing before being imported to the model for translation. In preprocessing stage dirty data should be cleaned by removing punctuations and digits, convert all characters to lowercase and activities like normalization, tokenization and removal of stop words takes place. Word embedding to feed the model with aligned parallel Geez-Afaan Oromoo sentences will be prepared by using One-Hot encoding. Then the model will be trained with RNN special networks GRU and LSTM libraries to reduce vanishing gradients. The model will predict translation after training by using test datasets and the performance of the model will be measured using BLEU score metrics. The conceptual NMT model architecture was shown in the following figure.

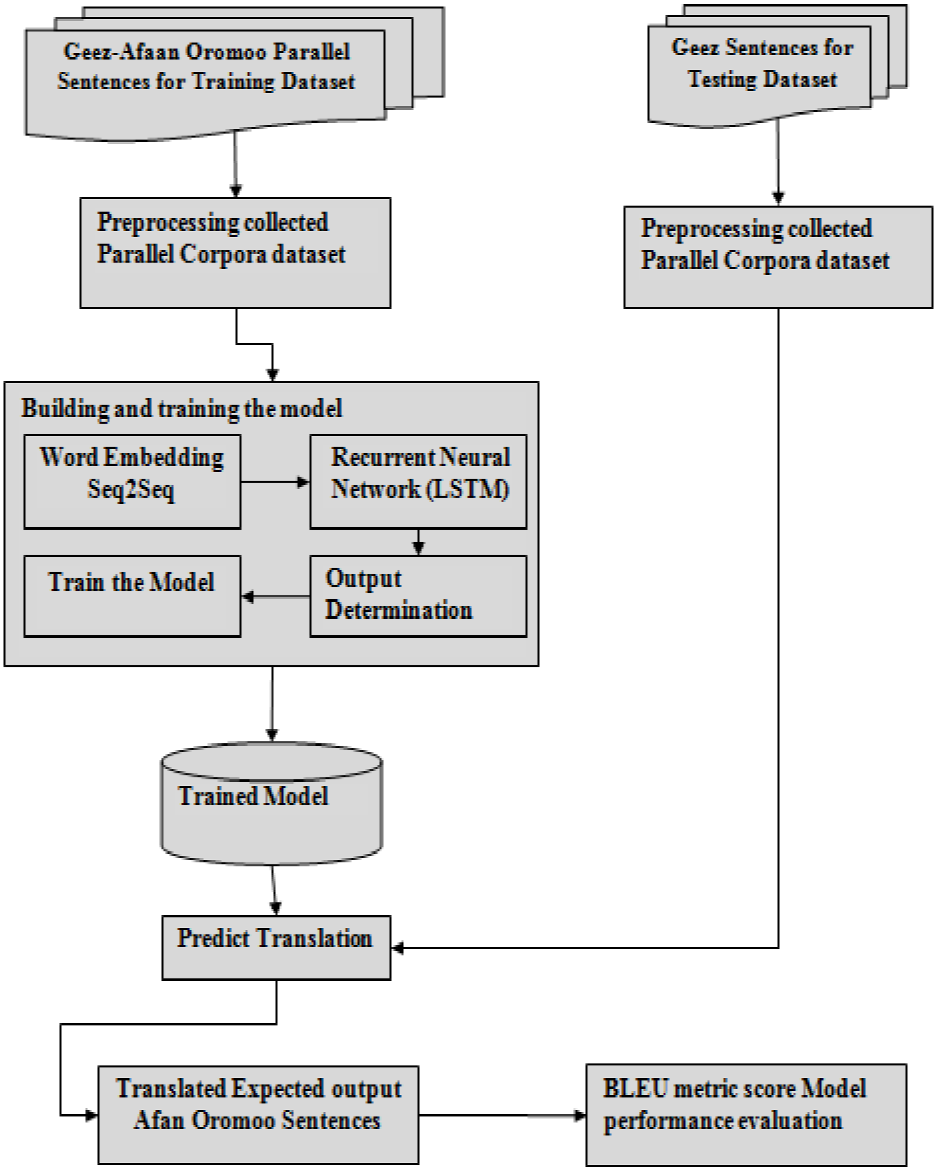

Geez-Afaan Oromoo Parallel Sentences for Training Dataset
Geez Sentences for Testing Dataset
Preprocessing collected Parallel Corpora dataset
Preprocessing collected Parallel Corpora dataset
Building and training the model
Word Embedding Seq2Seq
Recurrent Neural Network (LSTM)
Train the Model
Output Determination
Trained Model
Predict Translation
Translated Expected output Afan Oromoo Sentences
BLEU metric score Model performance evaluation

Figure 3: Architecture of the neural network MT Model (Girma Moges, 2020)

3.2.Data Collection and preparation

A well collected, sized, and defined text data used to provide or show practical morphological comportment of a language. The collection of text data is, therefore, an essential component in developing machine translation model. Because of this we collected the dataset for this research from different religious books and some of digitally available Geez sentences properly matching them with their Afaan Oromoo meaning with help of Geez experts or EOTC religion scholars. Thus, the corpus used in this research were collected from religious books such as Psalms, Anaphora book, Kidan, Liton, Wudase Mariam, Anketse Birhan, Malka Mariam and Malka Iyyasus. Since there is no public online soft copy of these documents, parallel corpora required for this research has been prepared manually. Psalms, Anaphora/Kitaaba Qiddaasee, Kidan, Liton and Wudase Mariam books in hard copy was available in Geez – Afaan oromoo. Malka Mariam and Malka Iyyasus will be translated manually from Geez-Amharic. Eight thousand (8000) Geez and Afaan Oromoo parallel corpora simple sentences dataset were collected and prepared as listed in Table 11 below.

Table 11: Source of dataset and size

No	Data Source	Number of parallel sentences
1	መጽሐፈ ቅዳሴ/Kitaaba Qiddaasee	3255
2	መዝሙረ ዳዊት/ Faarsaa daawit	2750
3	ውዳሴ ማርያም/ Galata Maaramii	730
4	መልክአ ማርያም / Malkaa Maaramii	320
5	መልክአ እየሱስ/ Malkaa Iyyasusii	290
6	አንቀጸ ብርሃን/ Baha Ifaa	270
7	ኪዳን/Kadhannaa Waadaa	200
8	ሊጦን/Kadhannaa Liixoonii	185
Total		**8000**

3.3.Training and Testing the Model

It is challenging task to train, validate, and test the model using deep learning approaches for those under resourced language like Geez and Afaan Oromoo. We use hold out method machine learning approach, from the prepared dataset (80% for training and 20% for model testing) with only prepared parallel corpora. The hold-out method for training the machine learning models is a technique that involves splitting the data into different sets: one set for training, and other sets for testing. The hold out method is used to check how well a machine learning model will perform on the new data. Once the Geez-Afaan Oromoo parallel corpus is prepared, then it passes through the training phase, predicting/translating phase and final testing step.

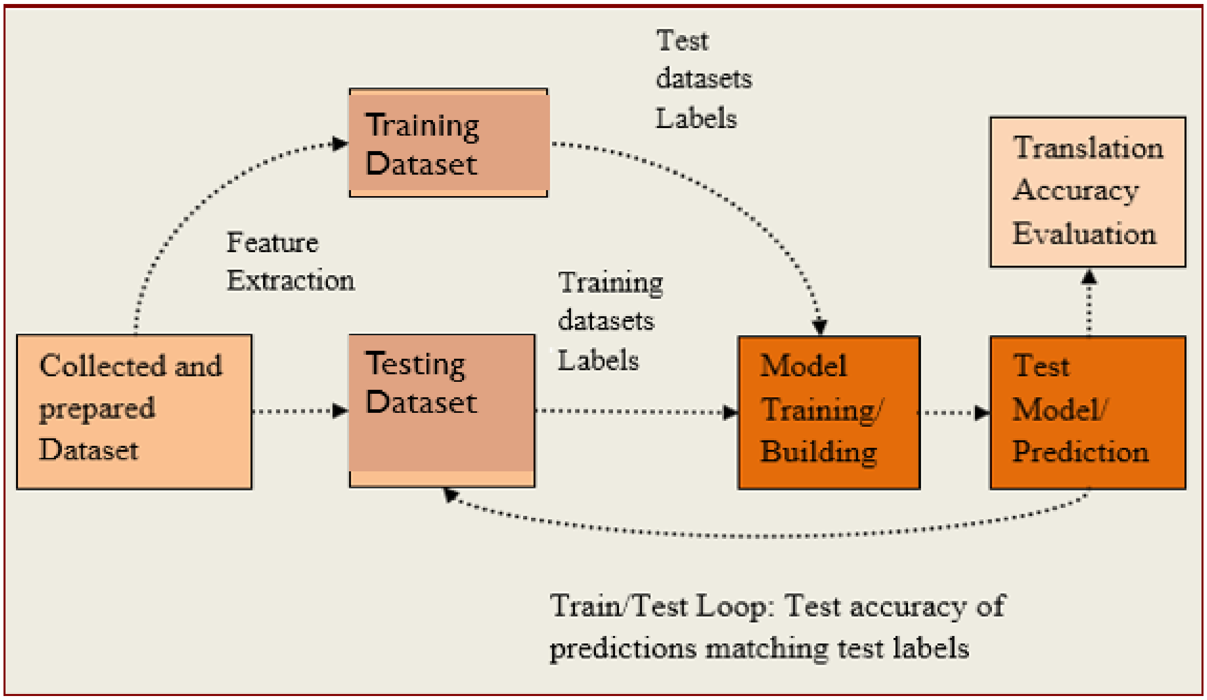


Figure 4: Model training, testing and evaluation work flow

3.4. Implementation Environment

3.4.1. Software (package)

- **Google Colab:** Google Colaboratory is a free online cloud-based Jupyter notebook environment that allows us to train our machine learning and deep learning models on CPUs, GPUs, and TPUs. The main reason to use Colab is that, it provides free online GPU and TPU runtime processors that are too much faster compared to CPU and allows more main memory RAM to execute very faster than traditional Jupyter notebook IDE. We choose to use these tools because most of the popular libraries come installed by default on Google Colab, like Pandas, NumPy, scikit-learn are all pre-installed. All the notebooks on Colab are stored on our Google Drive. The best thing about Colab is that our notebook is automatically saved after a certain period of time and we don't lose our progress. If we want, we can export and save our notebook in both .py and .ipynb formats.
- **Keras:** is an open source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the Tensorflow library.

- **Tensorflow:** a comfortable framework for deep learning, to construct the network layers, preprocessing the data, build the model, and train and estimate the model.
- **Pandas:** is mainly used for data analysis. Pandas allows importing data from various file formats such as CSV, JSON, SQL database tables or queries, and Microsoft Excel. We used it to import CSV cleaned dataset to the model training and testing.
- **Scikit-Learn**: A set of python modules for machine learning and data mining. This study uses it for feature extraction and training and testing model.
- **Numpy:** Array processing for number, strings, and objects. This study uses it for handling converting the text to numeric data and numeric data back to text for features and training and testing the model.
- **NLTK:** Natural language toolkit is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries and package. This study uses the framework for text preprocessing libraries.

3.4.2. Hardware

For model implementation and thesis construction the following hardware tools were used

- **Laptop Computer**
- Processor: Intel® Core (TM) i5-7200U CPU @ 2.50GHz 2.70 GHz
- Installed memory (RAM): 4.00GB (3.46 GB usable)
- System type: 64 –bit operating system, x64-based processor
- **Desktop Computer**
- Processor: Intel® Core™ i3-9100 CPU @ 3.60GHz 3.60 GHz
- Installed memory (RAM): 4.00GB (3.81 GB usable)
- System type: 64 –bit operating system, x64-based processor

3.4.3. Programming Language

Python programming language is best for implementation of the machine learning algorithm and MT approaches because it has better scientific and numerical libraries that offer a working environment of analysis of data in terms of number. This study uses the language to implement the proposed model construction because python is one of the most accessible programming languages available since it has simplified syntax and not complicated, which gives more emphasis on natural language. Due to its ease of learning and usage, python codes can be easily written and executed much faster than other programming languages.

1.5. Machine Translation Evaluation

Machine translation evaluation could be done by using manual or automatic evaluation methods. In order to evaluate the performance of machine translation in terms of translation accuracy, the machine translated sentence must be evaluated either by source and target language fluent expert or some automated measuring metrics. Human based evaluation is more expensive, because in order to evaluate translation accuracy we need to find for the expert of both languages. This expert compares the system output against given reference sentence and describes the result of comparison in percentage computation. The most challenging issues in conducting human evaluation of machine translation output are high costs and time consumption. Therefore, in order to solve this slowness of expert-based evaluation, the best choice is to use automatic measuring metrics. One of such automatic metrics is **B**i-**L**ingual **E**valuation **U**nderstudy (BLEU) that is used to measure translation accuracy by comparing the system's translation output against human translated reference sentences.

Moreover, in order to generalize the evaluation result of machine translation, the selection of correct testing technique is necessary. Therefore, the total dataset must be split into training set and testing set. This splitting of data into these two sets helps to train system on training dataset and to test the system on testing dataset in order to get convincing report evaluation result of translation accuracy. Therefore, the best technique of testing is Pareto principle (80/20) in which 80 percent of total data is used for training set, while 20 percent of total dataset is left for testing the system.

1.5.1. Bi-lingual Evaluation Understudy (BLEU) Score

Even though human-based evaluation is good to measure translation quality of machine translation by considering different conditions like grammatical correctness or real intention of speakers, it requires much time as expert of the languages compares each system's output sentence against reference sentence and computes percentage accuracy. The BLEU algorithm evaluates the precision score of a candidate machine translation against a reference human translation. The reference human translation is assumed to be a model example of a translation, and we use n-gram matches as our metric for how similar a candidate translation is to it.

The proper way to evaluate the translation accuracy is measuring the BLEU scores of the translation. Therefore, BLEU score is automatic evaluation technique that calculates n-grams precision by working on total words of translated sentence compared against words of reference sentence. BLEU score does not take any consideration for the grammatical correctness of the textual data. Sometimes, the output of system may be the repeated copy of a single word. In such case the unnecessary repeated word may generate incorrect matching probability. In order to minimize such problem, it is necessary to use modified n-gram method which limits the number of repeated words if a single word is occurred more than the number of its occurrence in reference sentence. The BLEU score is a number between zero and one that measures the similarity of the machine-translated text to a set of high-quality reference translations (Sameen Maruf, 2019). The higher the BLEU score (those close to one) the more the translation resembles with the translation of a human translation. It has been shown that BLEU scores correlate well with human judgment of translation quality. In this thesis BLEU score metrics was used to evaluate the performance of the model.

Chapter Four

2. Designing Geez to Afaan Oromoo Neural Machine Translation

Neural Machine Translation is a fully-automated translation technology that uses neural networks. Neural Machine Translation provides more accurate translation by accounting the context in which a word is used, rather than just translating each individual word on its own. This chapter discusses the designing and implementation of neural machine translation from Geez to Afaan Oromoo based on the encoder-decoder language modeling by using RNN based architecture. Architecture of the neural machine translation model from Geez to Afaan Oromoo was described in this Chapter.

4.1. Architecture of Geez to Afaan Oromoo Neural Machine Translation

The proposed model uses the RNN encoder-decoder technique on the basis of the NMT architecture. The proposed solution experimented on Geez-Afaan Oromoo prepared parallel datasets that translates Geez written text into Afaan Oromo text using deep learning approach. The RNN encoder-decoder is trained on a parallel corpora dataset and performs an end-to-end translation.

In recent time, the encoder-decoder end-to-end language modeling is becoming attractive language modeling for machine translation task. It is a language modeling which is based on deep learning algorithm. According to the work of different researchers, in the early emergence of neural machine translation the recurrent neural network was used to generate word co-occurrence probabilities for statistical machine translation. Through time, different researchers continued to work on its gradual development to develop the pure encoder-decoder recurrent neural network-based machine translation (Bahdanau, 2015).

Neural Machine Translation is MT approach that applies a large ANN toward predicting the likelihood of a sequence of words, often in the form of whole sentences. Unlike SMT, which consumes more memory and time, NMT trains its parts end-to-end to maximize performance. Neural machine translation systems are quickly moving to the forefront of machine translation, recently outcompeting traditional forms of translation systems.

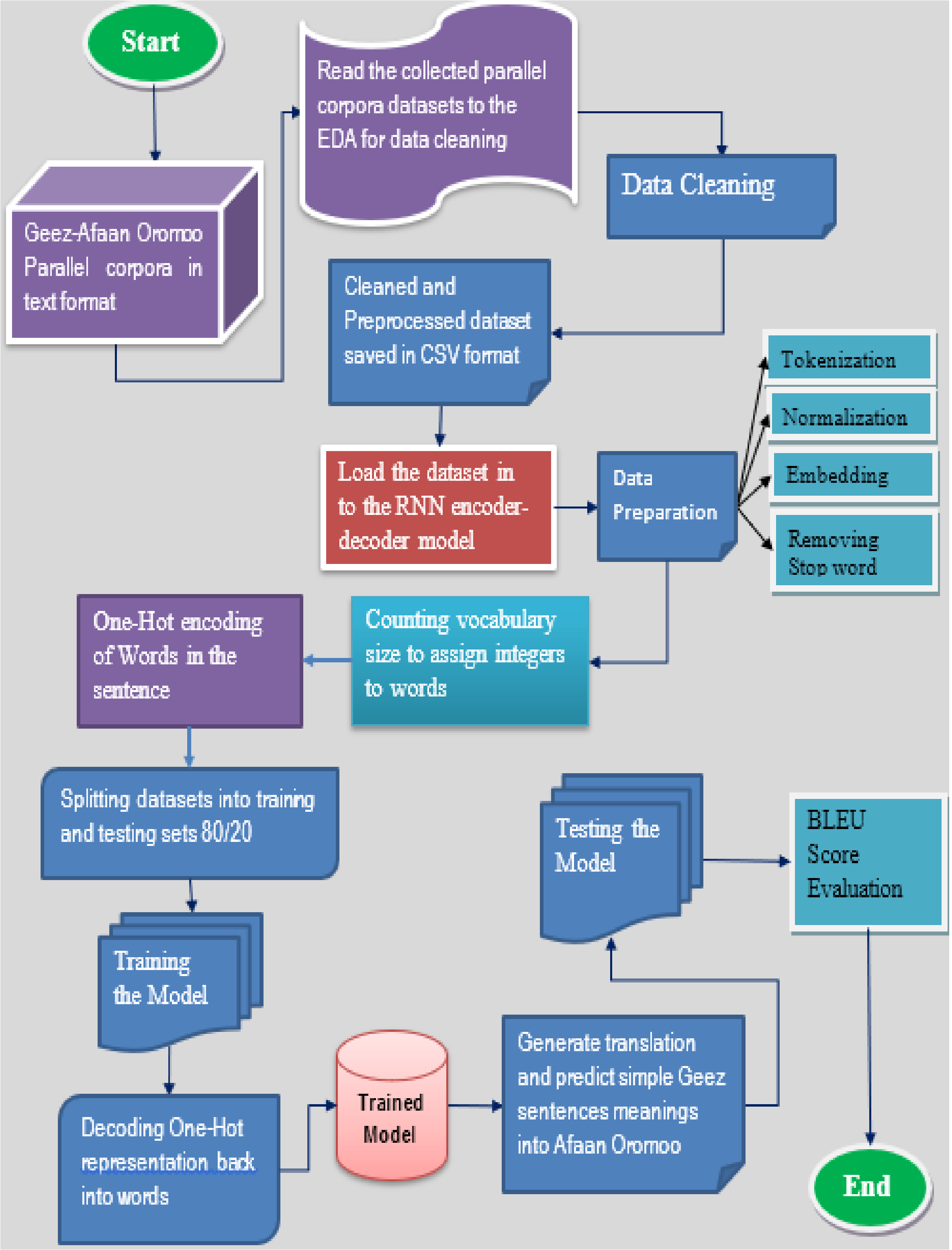

Start
Read the collected parallel corpora datasets to the EDA for data cleaning
Geez-Afaan Oromoo Parallel corpora in text format
Data Cleaning
Cleaned and Preprocessed dataset saved in CSV format
Tokenization
Normalization
Embedding
Removing Stop word
Load the dataset in to the RNN encoder-decoder model
Data Preparation
One-Hot encoding of Words in the sentence
Counting vocabulary size to assign integers to words
Splitting datasets into training and testing sets 80/20
Testing the Model
BLEU Score Evaluation
Training the Model
Trained Model
Generate translation and predict simple Geez sentences meanings into Afaan Oromoo
Decoding One-Hot representation back into words
End

Figure 5: Geez to Afaan Oromoo Encoder-Decoder RNN MT Model Architecture

The designing phase of machine translation system takes a consideration of two-step processes. These are training phase and testing phase. In encoder-decoder language modeling the training phase is followed by testing phase. The availability of these two phases is related to that the human translator need to learn the structures of language pairs before starting translate from one natural language to another. In similar manner, machine translation system must take training on the structure of both source and target languages like that of human translator learns one language through practicing. Therefore, like human translator, machine also learns the structure of language pair by repeatedly looking into the structures of the languages from datasets during training phase. So, the design of neural network-based systems at training and testing phase was shown in Figure 5 above. After designing training time components, we continued on the designing of components required at testing time to test the systems on testing datasets after training systems. This testing step is the step in which the system must translate the unseen text of the source language (Geez) to target language (Afaan Oromoo) from the experience of the trained model of the system gained at training time.

From figure 5 Geez to Afaan Oromoo neural machine translation starts its process having prepared Geez and Afaan Oromoo languages parallel corpora datasets collected for the purpose of this research. The collected datasets have cleaned and processed to have machine readable format. After cleaned and processed the Geez script converted into UTF-8 the format unreadable for human being represented by question marks but it is readable internally for machine. For example, ??? ??? ?? ??? ??? ???? ???? ???? – lapheen koo waan gaarii baase ani hojiikoo mootiitti nan dubbadha (instead of **ጐሥዐ ልብየ ቃለ ሠናየ አነ ኣየድዕ ግብርየ ለንጉሥ** - lapheen koo waan gaarii baase ani hojiikoo mootiitti nan dubbadha). This is because of Geez scripts are not ASCII standard; it is Unicode characters which is often readable for computer and later after load into the model for training made readable for human.

After data cleaning and processing the cleaned data loaded into the encoder-decoder RNN model for training and testing the model. In machine learning the datasets for training and testing were like the fuel for the engine. Without the datasets it is impossible to train the model. Especially in deep learning theme sufficient amount of dataset is very important to improve the accuracy and performance of the model, because as much as the machine trained on the datasets the probability to learn well can be increased.

After feeding the cleaned dataset to the encoder layer the quantity of vocabulary size should be known and one-hot encoding will be done. This is because, our collected datasets are in text format, we have designed supporting components that can change data from text format to vector form representation. The purpose of this conversion is that neural networks require vector form representation to work on data. So, the original text format must be converted into a one-hot form representation style by use of unique word id and zero-padding. This representation style uses a fixed length of sentence. Therefore, to set the length of sentences to the fixed length, zero is added to short sentences. So, adding this at the end of each short sentence is known as zero-padding. This conversion process has been done in reverse direction after getting output from the system to form the translated output target language sentence.

Then datasets will be divided into training and testing with a ratio of 80/20, mainly 80% of datasets (6400 in our case since the total parallel sentences were 8000 and 20% (1600) for testing) purpose. Recently the best ratio to split the dataset into training and testing is 80/20. This is because it is a flexible dataset split rule that leaves insufficient data in test set. The model uses 4/1 strategy, meaning train on four samples and reserve one sample for testing by randomly accepting five parallel corpus and make one of it unseen for training the model. This makes not jump many steps and manage action easily. When the training completes the prediction start to test the performance of the trained model with unseen dataset reserved for model testing. Depending on the test result the average BLEU score metrics calculated and the translation and evaluation process end.

4.2. Datasets preprocessing

The first step in data processing and analysis is preprocessing, in which text data before using for the application purpose must be preprocessed and makes it ready for usage for further analysis and interpretation. Unstructured data are data in different formats and representations. So, such data has to be normalized, tokenized, padded and some words which have no contribution to the content building in the document have to be removed.

A. Normalization

Before text data is used in training NLP models, it's pre-processed to a suitable form. Text normalization is often an essential step in text pre-processing. Text normalization simplifies the modelling process and can improve the model's performance. In order to carry out processing on natural language text, we need to perform normalization that mainly involves eliminating punctuation, converting the entire text into lowercase or uppercase, converting numbers into words, expanding abbreviations and contractions, canonicalization of text, and so on.

B. Tokenization

Tokenization is a step that splits longer strings of text into smaller pieces or tokens. Larger chunks of text data can be tokenized into a list of sentences, sentences can be tokenized into list of words, etc. The tokenizer function contains the number of words in the document as a parameter and it fits with the training and the test data. Those tokens are also converted to vector representation to be understood by the deep learning algorithm. Thus, we have a number of text documents and those documents are tokenized in to a number of word tokens to be represented by a vector with unique numerical representation.

C. Word-embedding

For neural network-based machine translation, the cleaned parallel corpus has to be changed into the format which is suitable to train machine the structures of both languages from the collected parallel corpus. In the process of language modeling; word embedding plays a great role in adjusting the data into a form suitable for the machine to learn from. The machine cannot directly work on a sequence of sentences. Rather, these sentences must be split into a sequence of words, and then these sequences of words must be changed into integer form representation.

D. Stop word removal

There are many words in a given text document that are used for grammatical formation and connecting parts of a sentence rather than describing the intent of the text document. Stop-words are words that have not content building contributions and words that occur most frequently in many text data but are not relevant or have no impact to create translation among text documents (Moges Girma, 2020). For example, in Geez **ዘ፡ ወ፡ ከ፡ ውእቱ፡** and in Afaan Oromoo **fi, dha, yoo, gara** are stop words. Stop words come redundantly in the corpus but not counted in the dictionary many times.

4.3. Designing Encoder-Decoder for RNN Model

The encoder of RNN based system is designed based on GRU and LSTM architecture which is proposed in this research. The GRU uses gate unit to control the flow of information. It uses the current input and its previous output, which can be considered as the current internal state of the network, to give current output, and LSTM designed to avoid long term dependency problems.

The decoder part in encoder-decoder RNN language modeling is the component which is dedicated to searching of the word of the target language that matches the contextual vector received from encoder layer. In order to perform this task, the decoder must first train on sentences of the source target languages to get the structural information of the target language.

4.4. Building Recurrent Neural Network (RNN) Model

A recurrent neural network is a type of ANN that is used when users want to perform predictive operations on sequential or time-series based data. These Deep learning layers are commonly used for ordinal or temporal problems such as Natural Language Processing, Neural Machine Translation, automated image captioning tasks and likewise. Below at left one is a representation of standard RNN and the right one is a representation of Feed-Forward Network.

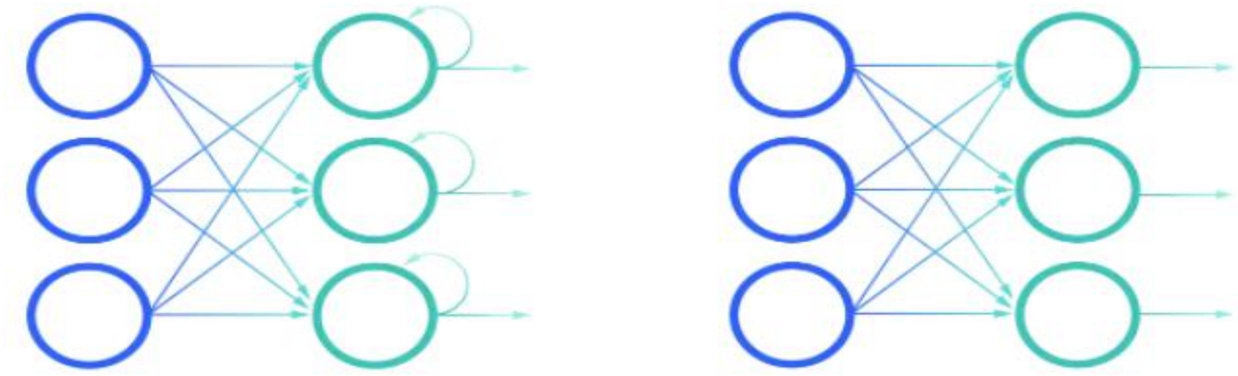


Figure 6: a) Standard RNN b) Feed-Forward Network **(Gaurav Singhal, 2020)**

If you look at the figure 6, you will notice that structure of Feed Forward Neural Network and recurrent neural network remain same except feedback between nodes. These feedbacks, whether from output to input or self- neuron will refine the data. There is another notable difference between RNN and Feed Forward Neural Network. In RNN output of the previous state will be fed as the input of next state (time step). This is not the case with feed forward network which deals with fixed length input and fixed length output. This makes RNN suitable for task where we need to predict the next character/word using the knowledge of previous sentences or characters/words.

Let’s take a look at a single unit of RNN architecture. Where it takes input from the previous step and current state Xt and incorporated with Tanh as an activation function, here we can explicitly change the activation function.

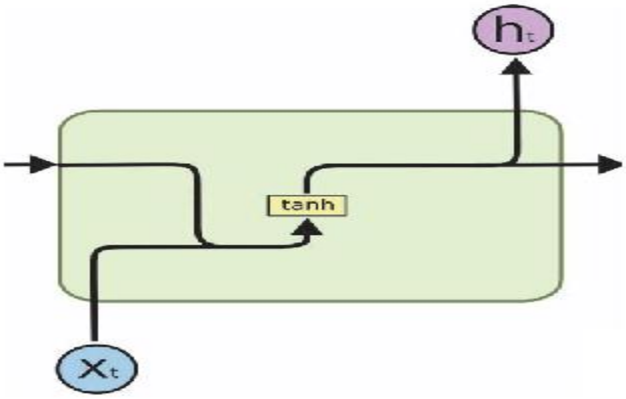


Figure 7: A single unit of RNN architecture

Sometimes we only need to look at recent information to perform a present task. But this is not the case we face all the time. When a standard RNN network is exposed to long sequences or phrases it tends to lose the information because it cannot store the long sequences and as the methodology is concerned it focuses only on the latest information available at the node. This problem is commonly referred to as Vanishing gradients.

Recurrent Neural Networks enable you to model time-dependent and sequential data problems, such as stock market prediction, machine translation, and text generation. You will find, however, RNN is hard to train because of the gradient problem. RNNs suffer from the problem of vanishing gradients. The gradients carry information used in the RNN, and when the gradient becomes too small, the parameter updates become insignificant. This makes the learning of long data sequences difficult. Due to this network does not learn the effect of earlier inputs and thus causing the short-term memory problem.

To overcome this problem specialized versions of RNN are created like LSTM, GRU, Time Distributed layer, ConvLSTM2D layer. In this study we used LSTM and GRU special families of RNN neural network MT model training and testing.

4.4.1. Long Short-Term Memory

Long Short-Term Memory in short LSTM is a special kind of RNN capable of learning long term sequences. They were introduced by Schmidhuber *and Hochreiter* in 1997. It is explicitly designed to avoid long term dependency problems, remembering the long sequences for a long period of time is its way of working.

The popularity of LSTM is due to the Getting mechanism involved with each LSTM cell. In a normal RNN cell, the input at the time stamp and hidden state from the previous time step is passed through the activation layer to obtain a new state. Whereas in LSTM the process is slightly complex, as you can see in the above architecture at each time it takes input from three different states like the current input state, the short-term memory from the previous cell and lastly the long-term memory.

These cells use the gates to regulate the information to be kept or discarded at loop operation before passing on the long term and short-term information to the next cell. We can imagine these gates as Filters that remove unwanted selected and irrelevant information. They are a total of three gates that LSTM uses as *Input Gate*, *Forget Gate*, and *Output Gate*.

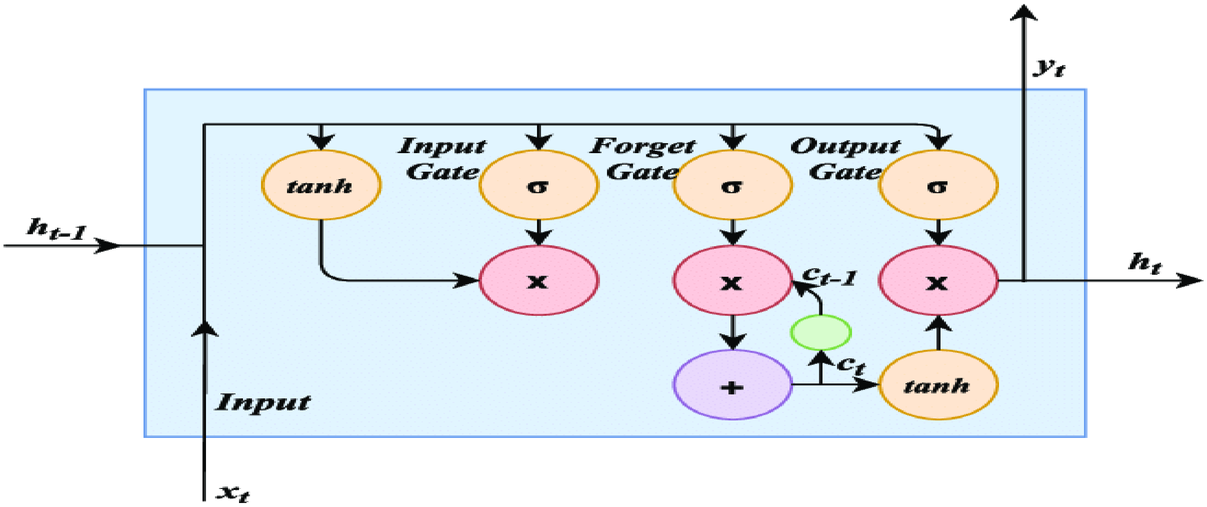


Figure 8: LSTM Input Gate, Forget Gate, and Output Gate **(Gaurav Singhal, 2020)**

Input Gate

The input gate decides what information will be stored in long term memory. It only works with the information from the current input and short-term memory from the previous step. At this gate, it filters out the information from variables that are not useful.

Forget Gate

The forget gate decides which information from long term memory be kept or discarded and this is done by multiplying the incoming long-term memory by a forget vector generated by the current input and incoming short memory.

Output Gate

The output gate will take the current input, the previous short-term memory and newly computed long-term memory to produce new short-term memory which will be passed on to the cell in the next time step. The output of the current time step can also be drawn from this hidden state.

4.4.2. Gated Recurrent Unit

The workflow of the Gated Recurrent Unit, in short GRU, is the same as the RNN but the difference is in the operation and gates associated with each GRU unit. To solve the problem faced by standard RNN, GRU incorporates the two gate operating mechanisms called Update gate and Reset gate.

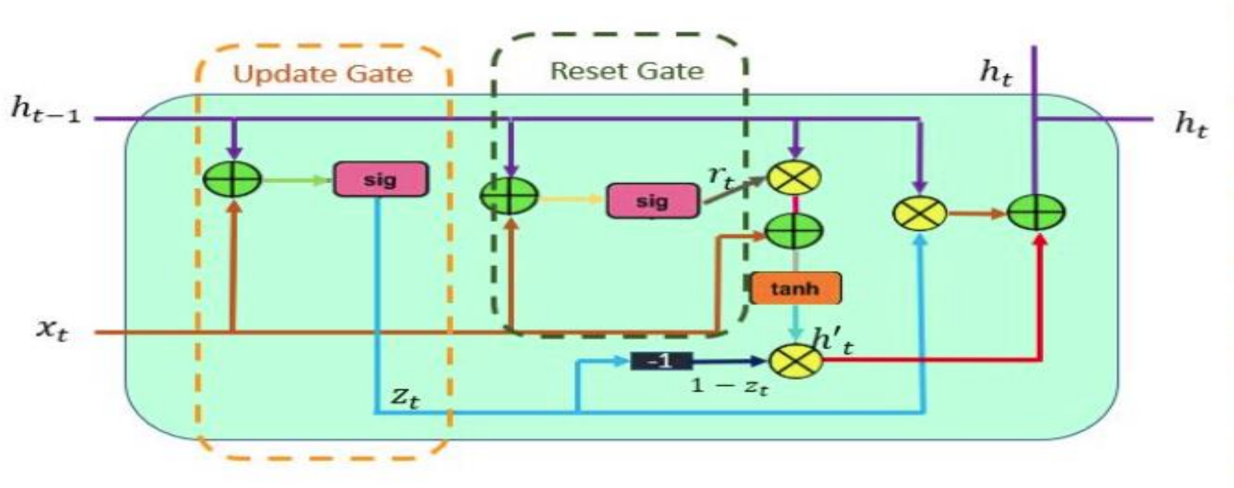


Figure 9: GRU Update gate and Reset gate **(Gaurav Singhal, 2020)**

Update gate

The update gate is responsible for determining the amount of previous information that needs to pass along the next state. This is really powerful because the model can decide to copy all the information from the past and eliminate the risk of vanishing gradient.

Reset gate

The reset gate is used from the model to decide how much of the past information is needed to neglect; in short, it decides whether the previous cell state is important or not. First, the reset gate comes into action it stores relevant information from the past time step into new memory content. Then it multiplies the input vector and hidden state with their weights. Next, it calculates element-wise multiplication between the reset gate and previously hidden state multiple. After summing up the above steps the non-linear activation function is applied and the next sequence is generated.

4.4.3. Comparison of GRU and LSTM

Both GRU and LSTM are the special neural networks of RNN encoder-decoder architecture. The few differencing points are as follows: The GRU has two gates, LSTM has three gates GRU does not possess any internal memory, they don’t have an output gate that is present in

LSTM. In LSTM the input gate and output gate are coupled by an update gate and in GRU reset gate is applied directly to the previous hidden state. In LSTM the responsibility of reset gate is taken by the two gates i.e., input and output. From working of both layers i.e., LSTM and GRU, GRU uses less training parameter and therefore uses less memory and executes faster than LSTM whereas LSTM is more accurate on a larger dataset. One can choose LSTM if you are dealing with large sequences and accuracy is concerned, GRU is used when you have less memory consumption and want faster results.

GRU is a variant of LSTM. Although the structure of GRU is simpler than that of LSTM, the effect is not decreased. GRU model has only two door functions: update door and reset door. Update gate is used to control the degree to which the state information of the previous time is brought into the current state.

Chapter Five

5. Results and Discussion

5.1 Introduction

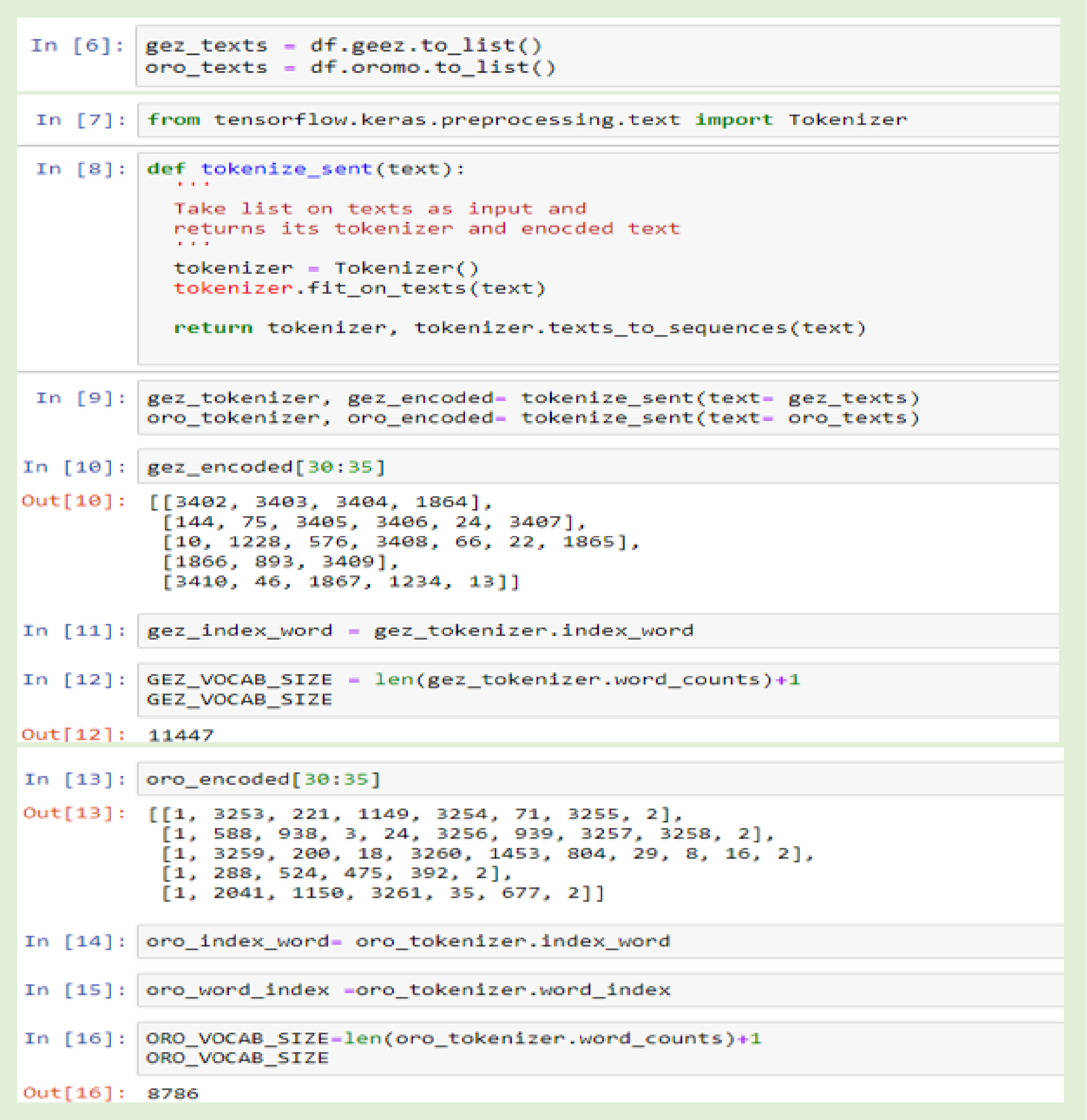
In this chapter experimentations, procedures, and evaluation methods are described in detail. Thus, to make experimentation, the tools and programming languages are described and the evaluation metrics and methods as well as the result of the evaluation are explained concerning the data that is used for experimentation and the metrics or approaches used for evaluation.

5.2. Dataset preparation for training

Neural machine translation approach requires huge bilingual parallel corpus to translate with high accuracy. For this research work, parallel documents of Geez and Afaan Oromo that are collected from different EOTC books are used. In order to perform our experimental study, we have collected datasets total of 8000 sentences of parallel corpus from different religious books. The datasets were collected by aligning this data of parallel corpus along its translation with spreadsheet and later converted to text format by copying the datasets from spreadsheet to the notepad, Geez and Afaan Oromoo sentences being separated with tab. The spreadsheet is important during manual dataset collection (writing datasets from the hardcopy) to avoid redundancy of sentences.

Next, this aligned data is stored in text format to feed into model. we have classified the data elements into training and testing sets. The training set consists of 6400 sentences of parallel corpus which is 80% of total collected data, while the test set consists of 1600 sentences which is 20% of total datasets. In order to feed the system with parallel corpus, the system opens the total textual formatted datasets and splits the data into set of sentences for both training and testing data sets. After splitting all elements of datasets into sequences of sentences for both sets, then again splits sentences into sequences of words. Therefore, these sequences of words in sentences can be fed into the system by use of integer form representation, which is known as tokenization. In order to form integer form representation, each word in the data is assigned with one unique integer number as soon as visited. So, the maximum number that used as word ID is also used to define the size of vocabulary of the dataset for one language. Therefore, our dataset consists of 11447 vocabulary size for Geez language and 8786 for Afan Oromo language.

The following figure shows lines of codes and its sample outputs that list sentences into words/tokenize, identify vocabulary size and assign unique integer ID for list of words that can easily understandable by the machine.



```
In [6]:  gez_texts = df.geez.to_list()
         oro_texts = df.oromo.to_list()

In [7]:  from tensorflow.keras.preprocessing.text import Tokenizer

In [8]:  def tokenize_sent(text):
             '''
             Take list on texts as input and
             returns its tokenizer and enocded text
             '''
             tokenizer = Tokenizer()
             tokenizer.fit_on_texts(text)

             return tokenizer, tokenizer.texts_to_sequences(text)

In [9]:  gez_tokenizer, gez_encoded= tokenize_sent(text= gez_texts)
         oro_tokenizer, oro_encoded= tokenize_sent(text= oro_texts)

In [10]: gez_encoded[30:35]

Out[10]: [[3402, 3403, 3404, 1864],
          [144, 75, 3405, 3406, 24, 3407],
          [10, 1228, 576, 3408, 66, 22, 1865],
          [1866, 893, 3409],
          [3410, 46, 1867, 1234, 13]]

In [11]: gez_index_word = gez_tokenizer.index_word

In [12]: GEZ_VOCAB_SIZE = len(gez_tokenizer.word_counts)+1
         GEZ_VOCAB_SIZE

Out[12]: 11447

In [13]: oro_encoded[30:35]

Out[13]: [[1, 3253, 221, 1149, 3254, 71, 3255, 2],
          [1, 588, 938, 3, 24, 3256, 939, 3257, 3258, 2],
          [1, 3259, 200, 18, 3260, 1453, 804, 29, 8, 16, 2],
          [1, 288, 524, 475, 392, 2],
          [1, 2041, 1150, 3261, 35, 677, 2]]

In [14]: oro_index_word= oro_tokenizer.index_word

In [15]: oro_word_index =oro_tokenizer.word_index

In [16]: ORO_VOCAB_SIZE=len(oro_tokenizer.word_counts)+1
         ORO_VOCAB_SIZE

Out[16]: 8786
```

Figure 10: Tokenization, Encoding with integers and Vocabulary size

5.3. Tools and programming language

A. Tools

There are different development tools of deep learning for natural language processing that is open source and easily adapted. We train the model on Colaboratory that support GPU. Since the training process requires high computation capacity, running on CPU is not sufficient to complete experimentation. TensorFlow is an end to end open-source platform for deep learning and it builds and trains deep learning models by using the high-level Keras API. Keras and TensorFlow are used for preprocessing and the construction of the model for training and validation.

B. Programing Language

Python is an object-oriented, an interpreter, and high-level programming language with dynamic semantics. Python programming is used for the experimentation from Geez to Afaan oromoo text translation using RNN approaches with Google Colab.

C. Experimental setup

Deep learning experimentations require high processor and GPU supported computing. In this study, we used a computer with a memory capacity of 4 GB RAM, 2.50 GHz processor, and 64-bit Windows 10 operating system. Since the local machine's RAM and CPU were low, we used Colab which support GPU to minimize time required for training the model.

5.4. BLEU Evaluation Metrics

BLEU is the most popular and commonly used precession (that is, it considers the number of n-gram matches as a fraction of the number of total n-grams in the output sentence) oriented metric for measuring the translation quality of a machine translation system. It considers not only single word matches between the output and the reference sentence, but also n-gram matches, up to some maximum n. This allows it to reward sentences where translated word order is closer to the local word order in the reference. It is the most commonly used form of evaluation in machine translation.

Translation quality is the correspondence between a machine translation with that of a human translator. A high-quality translation is the one which is closer to a professional human translation and BLEU"s main idea is the measurements of this closeness. BLEU score value falls in the range between 0 and 1, the higher the BLEU score (those close to one) the more the translation resembles with the translation of a human translation.

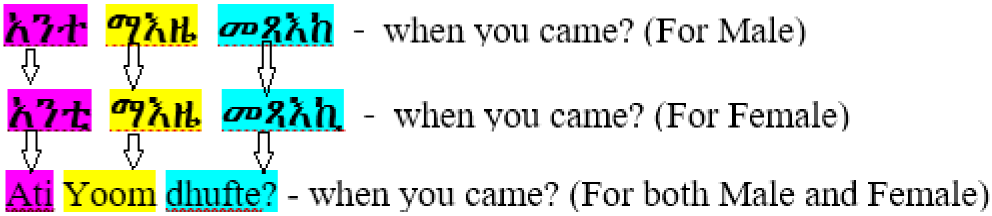
5.5. Experimental Results

In our experimental study, we have implemented the two designed models for our study and trained the systems in two ways to get correct comparison of their performance. We have used 100 sentences to test the model from 1600 test datasets provided for testing. After training and testing each system, we have measured the result of testing the system by using BLEU score metrics to see the difference in their scores. In order to report the result of our testing, we have used the average BLEU score result. Therefore, this average result of RNN based system using LSTM and GRU is summarized in the table below.

RNN special network layers	Geez to Afan Oromoo MT in BLEU score	Remark
GRU based model	**73.75%**	
LSTM based model	**77.55%**	

Table 12: Experimental result in BLEU score metrics

From Table 12 above when we compare the result of LSTM based model, with GRU based LSTM model shows better BLEU score results than that of GRU model. The BLEU score methodology is used to see the result of the translation process in both methods. The result recorded from the BLEU score methodology shows 73.75% for GRU and 77.55% for LSTM. The reason behind this promising accuracy is due to the language nature of Geez and Afaan Oromoo sentence structures or word class orders. As it is discussed in literature review Afaan Oromoo follow SOV while Geez language can also follow this sentence structure, though Geez follows SVO and VSO word orders in the sentence. For example,



አንተ ማእዜ መጻእከ - when you came? (For Male)

አንቲ ማእዜ መጻእኪ - when you came? (For Female)

Ati Yoom dhufte? - when you came? (For both Male and Female)

From the above example we can see that Geez and Afaan Oromo sometimes follow similar pattern (SOV), Geez and Afaan Oromo have sometimes similar word order in their sentence. When source and target languages follow similar word orders or sentence structure it leads to better accuracy of machine translation. So, this is one of our research questions and even though Geez and afaan Oromoo has different scripts for their writing system, their sentence structure may or may not vary. Because of this reason and capability of RNN encoder-decoder can work well on short sentences, some translation sample outputs are perfect translation. The following tables table 13 and 14 are sample output of GRU and LSTM architectures NMT model results respectively.

Input: ወኀደጋ ለደብተራ ሴሎም
Predicted translation: godoo Yooseef ni dhiise
Actual translation: godoo seeloom ni dhiise
BLEU Score: **0.90**
Input: ማርያም ድንግል ለነፍስየ ደብረ ምስካያ
Predicted translation: maariyaam dubroo maariyaam
Actual translation: dubroo maariyaam lubbuukoof tulluu baqannaa
BLEU Score: **0.56**
Input: ወአንተኒ ኦ እግዚእነ
Predicted translation: yaa gooftaa hundumaa si haa tau
Actual translation: yaa gooftaa keenyaa atis
BLEU Score: **0.39**
Input: እዎ እግዚኦ አምላክነ
Predicted translation: yaa gooftaa nutti murteessamuuf
Actual translation: dhugaadha waaqayyo gooftaa keenyaa
BLEU Score: **0.71**
Input: እግዚኦ እለ ዘንስእለከ
Predicted translation: yaa gooftaa hundumaa kan kae
Actual translation: yaa gooftaa nuti warri si kadhannu
BLEU Score: **0.56**
Input: እግዚአብሔር መክፈልትየ ለዓለም
Predicted translation: waaqayyo si abdatan ni jiraata
Actual translation: waaqayyo baraan carraakooti
BLEU Score: **0.54**
Input: እግዚኦ በመዐትከ ኢትቅሥፈኒ
Predicted translation: yaa gooftaa dhiifamnikees samiirradha
Actual translation: yaa gooftaa dheekkamsakeetiin na hin rukkuti in
BLEU Score: **0.56**
Input: አብ ወልድ መንፈስ ቅዱስ ይቄድሱ
Predicted translation: abbaa ilmi hafuurri qulqulluun ni raawwat u
Actual translation: abbaa ilmi hafuurri qulqulluun ni kabaju
BLEU Score: **0.39**
Input: ረስየነ ድልዋነ
Predicted translation: kan geenye nu taasisi
Actual translation: kan geenye nu taasisi
BLEU Score: **1.0**
Input: ንሥአተ ዘቅዱስከ ምሥጢር ጸግዎሙ
Predicted translation: icciitii qulqulluukee fudhachuu nuu kenni

Actual translation: icciitiikee qulqulluu fudhachuu kenniif
BLEU Score: **0.55**
Input: ወኵሎ አሚረ ይድሕርዎ
Predicted translation: guyyaa hunda olaanaa
Actual translation: guyyaa hunda isa galateeffatu
BLEU Score: **0.64**

Table 13: GRU based RNN machine translation sample output

Geez sentence: ወንብጻሕ ቅድመ ገጹ
Actual oromo Sentence: durasaatti haa dhiyaannu
Translated oromo Sentence: durasaatti haa tau
BLEU Score: **0.76**
Geez sentence: ብኪ አስተማሰሉ ቅዱሳን
Actual oromo Sentence: qulqulloonni sitti fakkeeffaman
Translated oromo Sentence: qulqulloonni sitti fakkeeffaman
BLEU Score: **1.0**
Geez sentence: ሊተሰ ተወክፈተኒ የማንከ
Actual oromo Sentence: anaanis mirgaankee na simatte
Translated oromo Sentence: anaanis mirgaankee na simatte
BLEU Score: **1.0**
Geez sentence: ኅደረ ዉስተ ከርሥኪ
Actual oromo Sentence: gadameessa kee keessa bule
Translated oromo Sentence: gadameessa kee keessa bule
BLEU Score: **0.56**
Geez sentence: ለናዝዞትየ እሙ ብጽሒ ፍጡነ
Actual oromo Sentence: haadha warra gaddanii dafii naaf qaqqabi
Translated oromo Sentence: haadha garuu yooseefiif kan barreeffame
BLEU Score: **0.56**
Geez sentence: በኵሉ አጽናፈ ምድር

Actual oromo Sentence: hanga daangaa lafa hundaa
Translated oromo Sentence: hanga daangaa lafa ni dhaalu
BLEU Score: **0.42**
Geez sentence: ተዘከርክዎ ለእግዚአብሔር ወተፈሣሕኩ
Actual oromo Sentence: waaqayyoon nan yaade nan gammades
Translated oromo Sentence: waaqayyoon nan barbaade naa jedha
BLEU Score: **0.76**
Geez sentence: አመ ሣልስት ዕለት
Actual oromo Sentence: guyyaa sadaffaatti
Translated oromo Sentence: guyyaa sadaffaatti
BLEU Score: **1.0**
Geez sentence: ባሕቲቱ ኣላ ንገብር
Actual oromo Sentence: ittiin hojjechuuf malee
Translated oromo Sentence: situ hojjeta
BLEU Score: **0.0**
Geez sentence: ማርያም ድንግል ጸጋዊተ ሰላም ወተስፋ
Actual oromo Sentence: dubroo maariyaam kennaa nagaa abdii
Translated oromo Sentence: dubroo maariyaam simboo kennaa kan uffatte
BLEU Score: **0.56**

Table 14: LSTM based RNN machine translation sample output

5.6. Discussion of Result

The main purpose of this study is implementing and designing Geez-Afaan Oromoo machine translation by conducting an experiment on Geez to afaan oromoo neural machine translation using encoder-decoder recurrent neural network with GRU and LSTM for better performance. The experiments are conducted by using two different RNN approaches. From the results of the experiments, we can see that the result recorded from a BLEU score shows that the LSTM is better than the GRU approach for Geez-Afaan Oromo machine translation.

We implemented proposed model using encoder-decoder RNN with LSTM and GRU machine translation algorithms. The bidirectional LSTM was used to encourage back propagation; hence the model checks the backward words incoming from decoder input in the hidden layer known as dense layer. The plot diagram from the training process is shown as follows in the next figure.

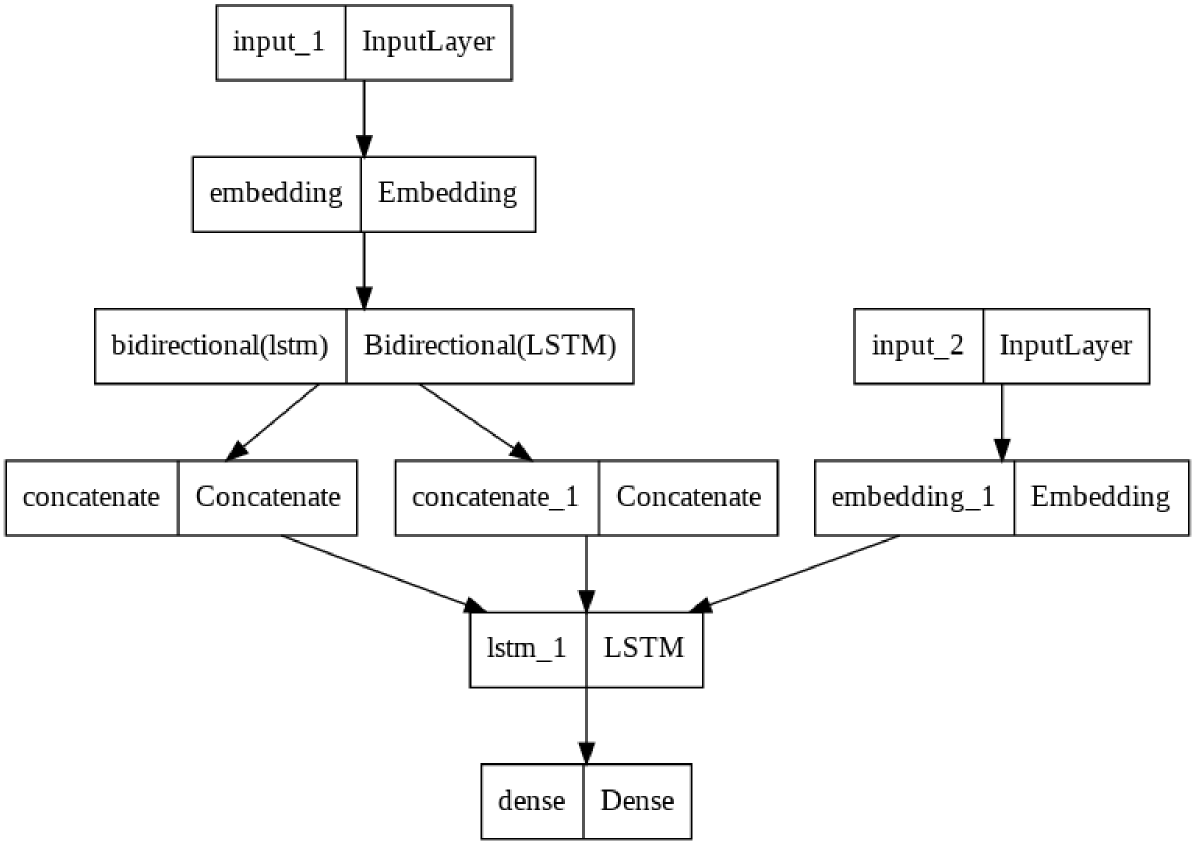


Figure 11: Model training flow of activities

From diagram it can be seen encoder input accepts prepared datasets and embedding takes place. After that bidirectional LSTM neural layers work on vector data. Later in the hidden layer it rearranges and concatenates words with their correct orders to form predicted sentence for target language.

The following python plot again shows, as the number of epochs increased training and validation loss decrease and their accuracy increase. We have trained our model for 30 epochs for five times the training time is very short because we used Google Colab connecting to GPU which minimize run-time and maximize training efficiency. When the model is trained on Jupyter notebook with CPU it is very time consuming and python plot graphs could not display or plotted.

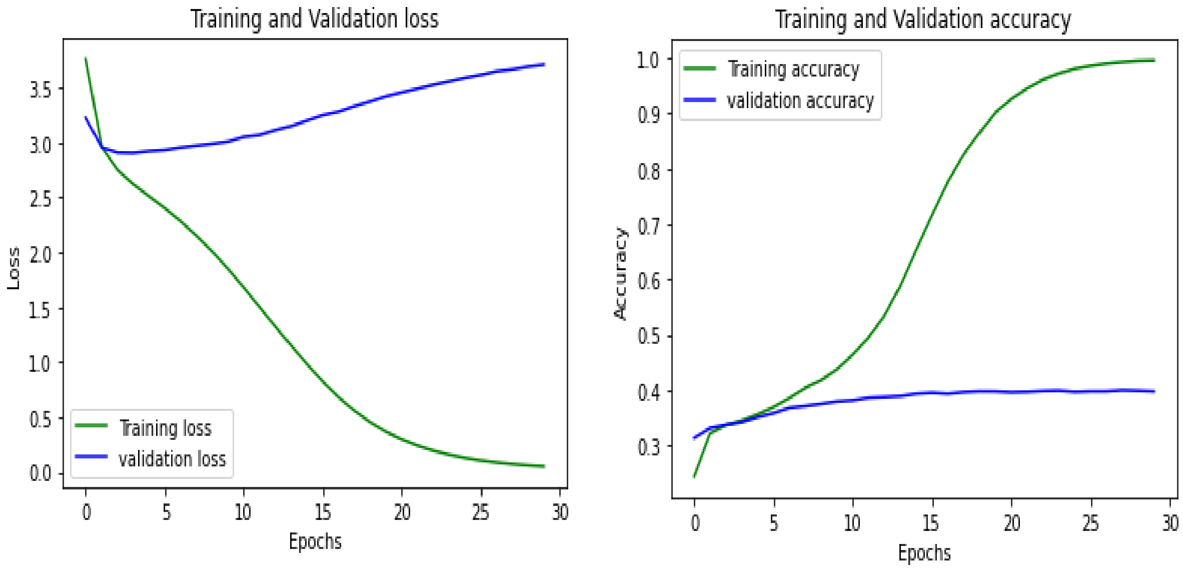


Figure 12: a) Training & Validation Loss b) Training & Validation Accuracy

The experiment was conducted by using RNN algorithms known as GRU and LSTM. Recurrent neural network is one of the NMT family. GRU and LSTM are special neural networks of RNN that were used to decrease the vanishing gradient problems. If these cells were absent the vanishing gradient can deadlock training and testing process or block it at all. Therefore, in our research question (RQ2) how vanishing gradient problems can be solved is answered by using these two special neural networks of RNN known as GRU and LSTM.

The other research questions were to compare the performance level of RNN special families GRU and LSTM and evaluating to what extent of BLEU score accuracy level RNN based NMT can perform well on Geez to Afaan Oromoo machine translation. So, as it can be seen from experimental results Table 12 the accuracy level of LSTM based Geez to Afaan Oromoo NMT slightly better than GRU based algorithms. The BLEU score of LSTM is rated to be **77.55%** and BLEU score of GRU based algorithm performance result was **73.75%**.

In general, the translation performance of this study is good as it is the first attempt on Geez to Afaan Oromoo machine translation. However, the obtained result is not sufficient enough concerning deep learning theme NMT, which is the state-of-the-art machine translation today. It is better to explore further research by incrementing parallel corpus dataset and using morpheme-based machine translation for Geez-Afaan Oromoo.

Chapter Six

6. Conclusions and Recommendations

This chapter is the conclusion driven from this research work and the recommendation for any individual researcher or organizations that are interested to work on the machine translation between Geez and Afaan Oromoo language pair (any other language pair) or directly or indirectly related tasks.

6.1. Conclusions

Machine Translation is an automated translation of text in one language, namely source language into another language, called target language performed by a computer. It provides text translations based on computer algorithms without human involvement. With Machine Translation, source text is easily and quickly translated into one or more target languages.

The main goal of our study is to design and implement automated machine translation from Geez to Afaan Oromoo language pair by using recurrent neural network algorithm in order to solve the problem of depending only on human translation. Therefore, this thesis work is done on the language pair with a scarce dataset consisting of 8,000 parallel corpora collected from different EOTC books.

The design and implementation process of Geez – Afaan Oromo machine translation involves collecting Geez – Afaan Oromo parallel corpus, corpus preparation which also involves data cleaning and preprocess, dividing the corpus as a training and test sets, training and testing, and evaluating its accuracy with BLEU score by using Python programming language.

This study is the first attempt to translate Geez sentences into Afaan Oromoo and we used simple sentences with 9, 13-word length for Geez and Afaan Oromo respectively. This is because of the parallel corpus preparation done manually, since there is no digitally available Geez documents. Encoder-decoder RNN are a promising result approach for shorter sentences machine translation and even we used special network layers of RNN called GRU and LSTM to solve RNN vanishing gradient problems and improve translation accuracy. The experiment result has the BLEU score accuracy result **73.75%** for GRU and **77.55%** for LSTM.

6.2. Recommendations

Natural language processing is now active research areas on MT, since machine translation is an important research area of NLP application, and the approach we have used is family of deep learning approach. This approach is currently being deployed at commercial level to produce human-like fluent translation. For example, today's Google translation which actively working with NMT is an amazing and very close to human translation.

We collected very small corpus due to the challenge Geez language documents are not available digitally to collect easily and align it with its Afaan Oromoo equivalent meanings. In addition, our domain for dataset collection is only religion document books. Therefore, we have the following recommendations which include the development of resources and future research directions on Geez and Afaan Oromoo language pair machine translation to extend the accuracy and quality of this research area.

- Improving this work using wide coverage of domain area which is not included in this study and increasing the size of training corpus those collected from different domain categories: politics, history, philosophy, medicine, governance and others using the national archival agency that kept ancient Geez documents available in our country.
- As the objective of our study is to implement machine translation only at level of text-to-text translation, we recommend that speech-to-speech machine translation between these language pair will be studied in future work to extend the research in speech translation and simplify the usage.
- The prepared data set for this research can be used for further MT researches to increment datasets upon this available scarce resource and other tasks like sentence classification, part of language tagging, Word sense disambiguation, segment words to their morphemes or similar activities.
- Making the translation bi-directional translation from Geez to Afaan Oromoo and from Afaan Oromoo to Geez to compare in which direction the model can perform with more accuracy.

References

Abba Teklehaymanot Weldu. (2018). *Kellaal Yegeez Kuwanka Memmarya Metsehaf.*

Abebe Medeksa. (2016). *Statistical Afaan Oromo Grammar Checker, A Thesis Submitted to The School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for The Degree of Master of Science in Information Science.*

Afarso Birhanu. (2019). *Bi-Directional English-Afan Oromo Machine Translation Using Convolutional Neural Network, A Thesis submitted to the school of graduate studies of Addis Ababa University in partial fulfillment of the requirement for the Degree of Master.*

Ankush Garg and Mayank Agarwal. (2018). *Machine Translation: Overview.*

Arthur et al. (2016). *Incorporating Discrete Translation Lexicons into Neural Machine Translation.*

Bahdanau, D. (2015). *Neural Machine Translation by Jointly Learning to Align and Translate.*

Biruk Abel. (2018). *Geez to Amharic Machine Translation, A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science.*

Dawit Mulugeta . (2015). *"Geez to Amharic Automatic Machine Translation: A StatisticalApproach" A Master Thesis, submitted to Addis Ababa University, Ethiopia.*

Dereje Saifu. (2019). *Hybrid Artificial Neural Machine Translation using Deep Learning Techniques English-to-Afaan Oromoo, A Thesis Submitted to the School of Electrical. Engineering and Computing: Presented in Partial Fulfillment for, the Degree of Masters of Science, ASTU.*

Gaurav Singhal. (2020). *LSTM versus GRU Units in RNN.*

Gelan Tullu. (2020). *Bidirectional Amharic-Afaan Oromo Machine Translation Using Hybrid Approach "A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science Computer Science" Addis Ababa, Ethiopia.*

Getachew Emiru. (2016). *Development of Part of Speech Tagger Using Hybrid Approach, A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Masters of Science in Information Science, Addis Ababa University.*

Girma Moges. (2020). *SEMANTIC-AWARE AMHARIC TEXT CLASSIFICATION USING DEEP LEARNING APPROACH* .

Hamiid M. (1996). *English-Oromo Dictionary, Sagalee Oromoo Publishing Inc, Atlanta.*

Ibrahim Bedane. (2015). *The Origin of Afaan Oromo: Mother Language* .

Ibrahim Gashaw and H L Shashirekha. (2020). *AMHARIC-ARABIC NEURAL MACHINE TRANSLATION OpenNMT.* Mangalore university.

Jabessa Daba. (2013). *"Bidirectional English – Afaan Oromo Machine Translation Using Hybrid Approach" A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Scienc.*

Junczys-Dowmunt et al. (2016). *Attention-based NMT Models as Feature Functions in Phrase-based SMT.*

Like Hiruyan Balay Mekonnen. (2012). *Hiyaw Lissaan sostegna ettim* .

Memhir Abebe Betemariam. (2020). *Geez Lahullumm Beteshale akkerareb kefl ande.*

Mirjam Sepesy et al. (2018). *Machine Translation and the Evaluation of Its Quality.*

Moges Girma. (2020). *Semantic-Aware Amharic Text classification using Deep learning Approach.*

Prabhat Pandey & Meenu Mishra Pandey. (2015). *Research Methodology, Tools and Techniques.*

Sameen Maruf. (2019). *Document-wide Neural Machine Translation, thesis submitted for the degree of Doctor of Philosophy at Monash University.*

Shuoheng Yang, et al. (2020). *A Survey of Deep Learning Techniques for Neural Machine Translation.*

Sisay Adugna. (2009). *English – Oromo Machine Translation: An Experiment Using a Statistical Approach.*

Tadesse Kassa. (2018). *(2018) “ Morpheme-Based Bi-directional Ge’ez -Amharic Machine Translation” A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Masterof Science in Information Science Addis Ababa University ,Ethiopia*

Workine Tesema & Duresa Tamirat. (2017). *Investigating Afan Oromo Language Structure and Developing Effective File Editing Tool as Plug-In Into Ms Word to Support Text Entry and Input Methods.*

Yitayew Solomon. (2017). *Optimal Alignment for Bi-directional Afaan Oromo-English Statistical Machine Translation.*

Yukio Matsumura, et al. (2017). *English-Japanese Neural Machine Translation with Encoder-Decoder model.*

Zeradawit Weldehanna. (2015). *Merho sewasew zelissane Geez beaddis Tibeb.*

Zhixing Tana et al. (2020). *Neural Machine Translation: A Review of Methods, Resources, and Tools.*

Appendix I: Sample Parallel Datasets

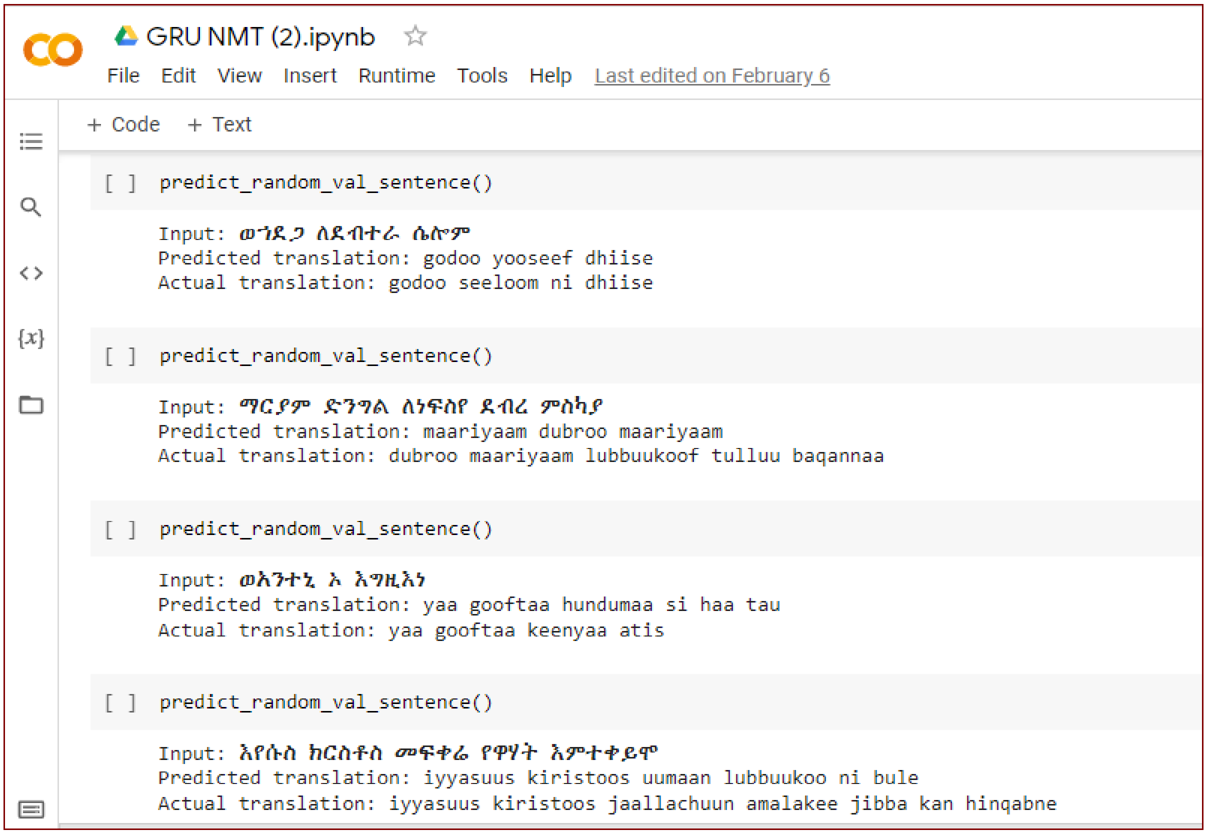
Geez	Afaan Oromo
ጐሥዐ ልብየ ቃለ ሠናየ አነ ኣየድዕ ግብርየ ለንጉሥ	lapheenkoo waan gaarii baase ani hojiikoo mootiitti nan dubbadha

ወአንግፈኒ እምጸላእትየ ወእምቀላየ ማይ	warra na jibbanii fi boolla gadi fagoorraa na oolchi
ወኵሎሙ እለ ፀውዱ ያበውኡ አምኃ ለግሩም	warri naannoosaa jiran hunduu raajichaaf harka fuudhii galchu
እግዚአብሔር አምላክነ	Waaqayyo Gooftaa keenya
ሰላም ለግንዘተ ሥጋኪ በእደ ሐዋርያት አርጋብ	harka duuka buootaan kan kafaname kafanama foonkeef nagaan haa ta’u
በአፈወ ዕፍረት ቅድው ዘአሳበ ሤጡ ዕፁብ	gatiinsaa guddaa kan tae urgooftuu miyaa’uun
ማርያም ድንግል ወለተ ኅሩየን ሕዝብ	dubroo maariyaam ilmoo ummata filatamanii
ረሰይኩኪ እግዝእትየ ህየንተ እም ወአብ	giiftiikoo gaarummaankee akka harmaa ta’ee na haa guddisu
ይሕጽነኒ ከመ ጥብ ፍቅርኪ ሐሊብ	jaalallikee akka aannan harmaa na haa jabeessu
ሰላም ለመቃብርኪ ለጌቴሴማኒ በመርሕባ	geeteeseemaanii kan jedhamtu awwaalakeef nagaan haa ta’u
እንተ ይእቲ ለኢየሩሳሌም ቅሩባ	isheenis iyyerusaalemiitti dhiyoo dha
ማርያም ድንግል ለሀገረ መንግሥት ርግባ	dubroo maariyaam gugee mootummaa Waaqayyoo
ትትሜጦ ወርቀ እምአፌር ወጸበለ አፈው እምሰባ	saabaarraa urgooftuu biyyee warqee keessaa argamu
ጌቴሴማኒ ለሥጋኪ ምስካባ	boqonnaa foonkee geeteeseemaanii
ሰላም ለመቃብርኪ ለኢየሩሳሌም በአድያማ	kutaa iyyerusaalemitti kan argamtu awwaalakeef nagaan haa tau
እንተ ይእቲ ጌተሴማን ስማ	isheenis maqaanshee geeteeseemaanii jedhama
ማርያም ሰንበት ዕረፍተ ጽዑራን እምፃማ	maariyaam boqonnaa sanbataa dadhabbiin kan dhiphataniif
አመ ኃደረ ላዕሌኪ ፀሐየ ቅዱሳን ዘራማ	aduun qulqulloota raamaa gaafa sirra bule
ኃይለ ልዑል ጸለሌኪ በመንክር ግርማ	humni olaanaa simboodhaan si haguuge
ሰላም ለፍልሰተ ሥጋኪ መካነ ፈሥሐ ምጡቀ	bakka gammachuu fi fagootti olbauu foonkeef nagaan haa tau
በአዕባነ ባሕርይ ዘተነድቀ	amala gamoo dhagaa dinqisiifatamu
ማርያም ድንግል አፀደ ወይንየ	dubroo maariyaam bakka biqiltuu wayiniiti
ንስቲተ ለዕበይኪ እንተ አቅረብኩ ግናየ	ulfaataa maqaakeef galata xiqqoo ani dhiyeesse
ኢይትኀደግ ዲበ ምድር ምሥጢ ሰማየ	lafarratti akka hin hafne samiitti ol naa baasi
ማርያም ደብተራ እንተ ኢትትከደኒ ሠቀ	dubroo maariyaam kan gaddaa fi uffatamu miti
ጽሒፈ ዉዳሴኪ እምኃለቀ ሰበ ኮነ ጥቀ	galatnikee barreeffamee kan dhumu miti

ዝናሙ ክረምት ቀለሙ ወሰማይ ረቀ	bokkaan gannaa qalam samiin waraqaa utuu ta’eellee
ሰላም ለፍልሰተ ሥጋኪ ወለተ ንጉስ ይሁዳ	yaa mucaa mootii yihudaa olbauu foonkeef nagaan haa tau
ወወለተ ሌዊ እኁኁ ዘሀብተ ክህነቱ አገዳ	mucaa obboleessasaa leewwii argama lubummaasaa
ማርያም ድንግል ምዕዝት ዘእምጵጌረዳ	dubroo maariyaam urgaankee daraaraa tsigeredaa caala
በጽሒፍ ኢይፈጽሞ ለስብሐትኪ እንግዳ	Bal’inni lafaa bifa gabateen yoo qophaa’ellee
እሙ ኮነ በቅድመየ ስፍሐ ምድር ሰሌዳ	bulee haaraa kan ta’e galatake uumamni barreessee hinxummuru
ሰላም ለፍልሰተ ሥጋኪ ደባትረ ብርሃን ኀበ ተተክሉ	dunkaanonni ifaa gara dhaabamanitti olba’uu foonkeef nagaan haa ta’u
ለገነተ ጵባሕ በማዕከሉ	karaa baha jannataa gidduutti
ውድስት አንቲ ወስብሕት በአፈ ኩሉ	qooqa hundumaan galateeffamtuu waan taateef
እምነ ጻድቃን ዘታሕቱ ወእምትጉሃን ዘላዕሉ	qulqulloota gad jiranii fi olaantota gubbaa jiraniin
ማርያም ለኪ ስብሐተ አደሉ	yaa maariyaam siif galanni dhiyaata
በዝንቱ ቃለ ማኅሌት ወበዝንቱ ይባቤ	jecha galataa kanaan sagalee yibbaabee kanas
ለዘይስእለኪ ብእሲ ጊዜ ረከቦ ምንዳቤ	namni si kadhatu yeroo gidiraan isa argatu
ብጽሒ ፍጡነ ትሰጥዊዮ ዘይቤ	dafii qaqqabiitii jabaadhu sii jira jedhiin
ማርያም ዕንቊየ ክርስቲ ሎቤ	kiristiiloobee kan jedhamu faaya miyaa dha
ወምዕዝት ምግባር እምከርቤ	naamusnikee urgaa qumbiirraa kan adda ta’e dha
ዘጸገየ ማኅፀንኪ አፈወ ነባቤ	gadameessikee daraaraa dubbatu argamsiiseera
ስብሐት ለኪ ማርያም በኍልቀ ኩሉ ሥዕርትየ	maariyaam lakkoofsa rifeensa matakoon sin galateeffadha
ስብሐት ለኪ ማርያም በኍልቀ ኩሉ አዕጽምትየ	maariyaam lakkoofsa lafeewwankoon sin galateeffadha

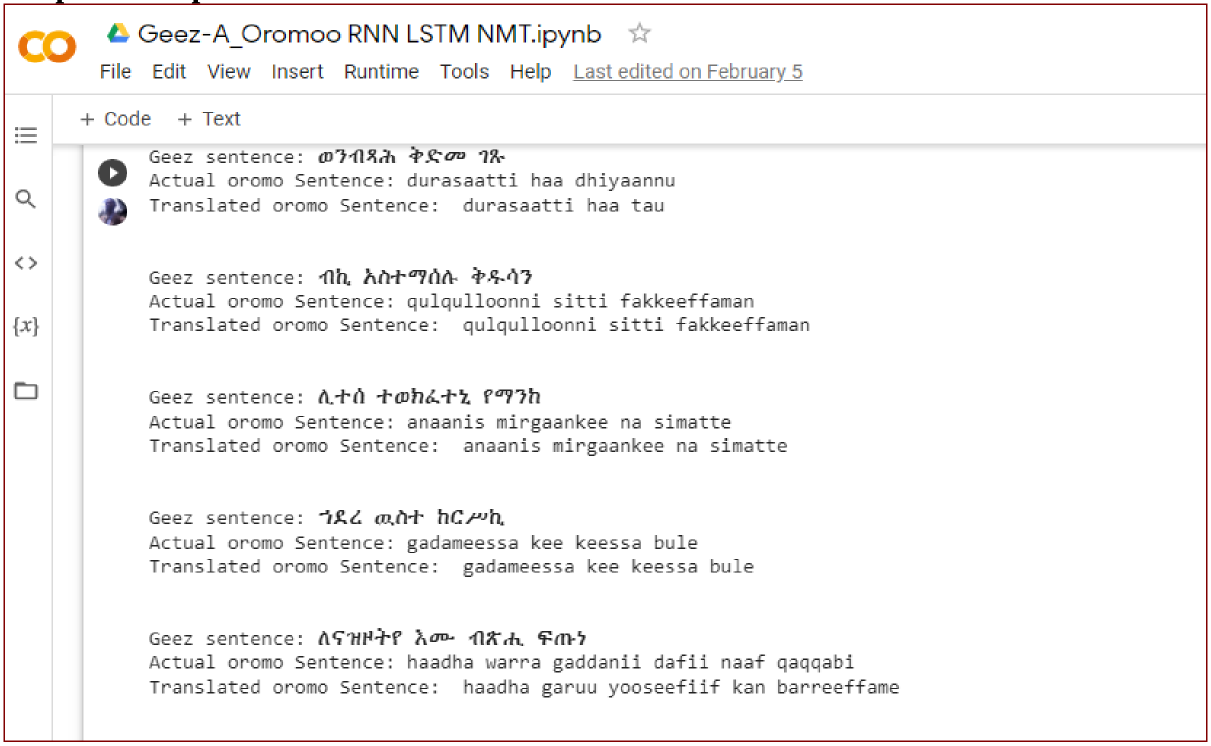
Appendix II: Sample output

A) Snapshot Outputs from GRU based RNN machine translation


GRU NMT (2).ipynb
File Edit View Insert Runtime Tools Help Last edited on February 6
+ Code + Text

```
[ ]  predict_random_val_sentence()

     Input: ወጎደጋ ለደብተራ ሴሎም
     Predicted translation: godoo yooseef dhiise
     Actual translation: godoo seeloom ni dhiise

[ ]  predict_random_val_sentence()

     Input: ማርያም ድንግል ለነፍስየ ደብረ ምስካይ
     Predicted translation: maariyaam dubroo maariyaam
     Actual translation: dubroo maariyaam lubbuukoof tulluu baqannaa

[ ]  predict_random_val_sentence()

     Input: ወአንተኒ ኦ እግዚእነ
     Predicted translation: yaa gooftaa hundumaa si haa tau
     Actual translation: yaa gooftaa keenyaa atis

[ ]  predict_random_val_sentence()

     Input: እየሱስ ክርስቶስ መፍቀሬ የዋሃት እምተቀይሞ
     Predicted translation: iyyasuus kiristoos uumaan lubbuukoo ni bule
     Actual translation: iyyasuus kiristoos jaallachuun amalakee jibba kan hinqabne
```

B) Snapshot Outputs from LSTM based RNN machine translation


Geez-A_Oromoo RNN LSTM NMT.ipynb
File Edit View Insert Runtime Tools Help Last edited on February 5
+ Code + Text

```
Geez sentence: ወንብጻሕ ቅድመ ገጹ
Actual oromo Sentence: durasaatti haa dhiyaannu
Translated oromo Sentence:  durasaatti haa tau


Geez sentence: ብኪ እስተማሰሉ ቅዱሳን
Actual oromo Sentence: qulqulloonni sitti fakkeeffaman
Translated oromo Sentence:  qulqulloonni sitti fakkeeffaman


Geez sentence: ሊተሰ ተወክፈተኒ የማንከ
Actual oromo Sentence: anaanis mirgaankee na simatte
Translated oromo Sentence:  anaanis mirgaankee na simatte


Geez sentence: ጎደረ ዉስተ ክርሥኪ
Actual oromo Sentence: gadameessa kee keessa bule
Translated oromo Sentence:  gadameessa kee keessa bule


Geez sentence: ለናዝዞትየ እሙ ብጹሒ ፍጡነ
Actual oromo Sentence: haadha warra gaddanii dafii naaf qaqqabi
Translated oromo Sentence:  haadha garuu yooseefiif kan barreeffame
```

Appendix III: GRU Algorithm of RNN Source Code

```
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import unicodedata
import re
import numpy as np
import time
import string
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
file_path = '/content/drive/MyDrive/gzorom.
lines = open(file_path, encoding='UTF-8').read().strip().split('\n')
lines[5000:5010]
print("total number of records: ",len(lines))
exclude = set(string.punctuation) # Set of all special characters
remove_digits = str.maketrans('', '', string.digits) # Set of all digits
def preprocess_gez_sentence(sent):
    '''Function to preprocess Geez sentence'''
    sent = sent.lower() # lower casing
    sent = re.sub("'", '', sent) # remove the quotation marks if any
    sent = ''.join(ch for ch in sent if ch not in exclude)
    sent = sent.translate(remove_digits) # remove the digits
    sent = sent.strip()
    sent = re.sub(" +", " ", sent) # remove extra spaces
    sent = '<start> ' + sent + ' <end>' # add <start> and <end> tokens
    return sent
def preprocess_oro_sentence(sent):
    '''Function to preprocess Afaan Oromoo sentence'''
    sent = re.sub("'", '', sent) # remove the quotation marks if any
    sent = ''.join(ch for ch in sent if ch not in exclude)
    #sent = re.sub("[२३०८१५७९४६]", "", 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
sent_pairs = []
for line in lines:
    sent_pair = []
    gez = line.rstrip().split('\t')[0]
    oro = line.rstrip().split('\t')[1]
    gez = preprocess_gez_sentence(gez)
    sent_pair.append(gez)
    oro = preprocess_oro_sentence(oro)
    sent_pair.append(oro)
    sent_pairs.append(sent_pair)
sent_pairs[5000:5010]
class LanguageIndex():
    def __init__(self, lang):
        self.lang = lang
        self.word2idx = {}
        self.idx2word = {}
        self.vocab = set()

        self.create_index()

    def create_index(self):
        for phrase in self.lang:
            self.vocab.update(phrase.split(' '))

        self.vocab = sorted(self.vocab)

        self.word2idx['<pad>'] = 0
        for index, word in enumerate(self.vocab):
            self.word2idx[word] = index + 1

        for word, index in self.word2idx.items():
            self.idx2word[index] = word
def max_length(tensor):
    return max(len(t) for t in tensor)
def load_dataset(pairs, num_examples):
    inp_lang = LanguageIndex(en for en, ma in pairs)
    targ_lang = LanguageIndex(ma for en, ma in pairs)

 input_tensor = [[inp_lang.word2idx[s] for s in en.split(' ')] for en, ma
in pairs]

arget_tensor = [[targ_lang.word2idx[s] for s in ma.split(' ')] for en, ma
in pairs]max_length_inp, max_length_tar = max_length(input_tensor),max_len
gth(target_tensor)
   input_tensor = tf.keras.preprocessing.sequence.pad_sequences(input_tens
or,
                                                    maxlen=max_length_inp,
                                                        padding='post')

 target_tensor = tf.keras.preprocessing.sequence.pad_sequences(target_tens
or,
                                                   maxlen=max_length_tar,
                                                            padding='post'
)

    return input_tensor, target_tensor, inp_lang, targ_lang, max_length_in
p, max_length_tar
input_tensor, target_tensor, inp_lang, targ_lang, max_length_inp, max_leng
th_targ = load_dataset(sent_pairs, len(lines))
# Creating training and validation sets using an 80-20 split
input_tensor_train, input_tensor_val, target_tensor_train, target_tensor_v
al = train_test_split(input_tensor, target_tensor, test_size=0.2, random_s
tate = 101)

# Show length
len(input_tensor_train), len(target_tensor_train), len(input_tensor_val),
len(target_tensor_val)
BUFFER_SIZE = len(input_tensor_train)
BATCH_SIZE = 64
N_BATCH = BUFFER_SIZE//BATCH_SIZE
embedding_dim = 256
```

```
units = 1024
vocab_inp_size = len(inp_lang.word2idx)
vocab_tar_size = len(targ_lang.word2idx)

dataset = tf.data.Dataset.from_tensor_slices((input_tensor_train, target_t
ensor_train)).shuffle(BUFFER_SIZE)
dataset = dataset.batch(BATCH_SIZE, drop_remainder=True)
def gru(units):

    return tf.keras.layers.GRU(units,
                                   return_sequences=True,
                                   return_state=True,
                                   recurrent_activation='sigmoid',
                                   recurrent_initializer='glorot_uniform')
class Encoder(tf.keras.Model):
    def __init__(self, vocab_size, embedding_dim, enc_units, batch_sz):
        super(Encoder, self).__init__()
        self.batch_sz = batch_sz
        self.enc_units = enc_units
        self.embedding = tf.keras.layers.Embedding(vocab_size, embedding_d
im)
        self.gru = gru(self.enc_units)

    def call(self, x, hidden):
        x = self.embedding(x)
        output, state = self.gru(x, initial_state = hidden)
        return output, state

    def initialize_hidden_state(self):
        return tf.zeros((self.batch_sz, self.enc_units))
class Decoder(tf.keras.Model):
    def __init__(self, vocab_size, embedding_dim, dec_units, batch_sz):
        super(Decoder, self).__init__()
        self.batch_sz = batch_sz
        self.dec_units = dec_units
        self.embedding = tf.keras.layers.Embedding(vocab_size, embedding_d
im)
        self.gru = gru(self.dec_units)
        self.fc = tf.keras.layers.Dense(vocab_size)

        # used for attention
        self.W1 = tf.keras.layers.Dense(self.dec_units)
        self.W2 = tf.keras.layers.Dense(self.dec_units)
        self.V = tf.keras.layers.Dense(1)

    def call(self, x, hidden, enc_output):

        hidden_with_time_axis = tf.expand_dims(hidden, 1)

        # score shape == (batch_size, max_length, 1)
        # we get 1 at the last axis because we are applying tanh(FC(EO) +
FC(H)) to self.V
        score = self.V(tf.nn.tanh(self.W1(enc_output) + self.W2(hidden_wit
h_time_axis)))

        # attention_weights shape == (batch_size, max_length, 1)
        attention_weights = tf.nn.softmax(score, axis=1)
        # context_vector shape after sum == (batch_size, hidden_size)
        context_vector = attention_weights * enc_output
        context_vector = tf.reduce_sum(context_vector, axis=1)

        # x shape after passing through embedding == (batch_size, 1, embed
ding_dim)
        x = self.embedding(x)

        # x shape after concatenation == (batch_size, 1, embedding_dim + h
idden_size)
        x = tf.concat([tf.expand_dims(context_vector, 1), x], axis=-1)

        # passing the concatenated vector to the GRU
        output, state = self.gru(x)

        # output shape == (batch_size * 1, hidden_size)
        output = tf.reshape(output, (-1, output.shape[2]))

        # output shape == (batch_size * 1, vocab)
        x = self.fc(output)

        return x, state, attention_weights

    def initialize_hidden_state(self):
        return tf.zeros((self.batch_sz, self.dec_units))
encoder = Encoder(vocab_inp_size, embedding_dim, units, BATCH_SIZE)
decoder = Decoder(vocab_tar_size, embedding_dim, units, BATCH_SIZE)
optimizer = tf.optimizers.Adam()

def loss_function(real, pred):
    mask = 1 - np.equal(real, 0)
    loss_ = tf.nn.sparse_softmax_cross_entropy_with_logits(labels=real, lo
gits=pred) * mask
    return tf.reduce_mean(loss_)
checkpoint_dir = './training_checkpoints'
checkpoint_prefix = os.path.join(checkpoint_dir, "ckpt")
checkpoint = tf.train.Checkpoint(optimizer=optimizer,
                                 encoder=encoder,
                                 decoder=decoder)
EPOCHS = 10
for epoch in range(EPOCHS):

    start = time.time()

    hidden = encoder.initialize_hidden_state()
    total_loss = 0

    for (batch, (inp, targ)) in enumerate(dataset):
        loss = 0

        with tf.GradientTape() as tape:
            enc_output, enc_hidden = encoder(inp, hidden)

            dec_hidden = enc_hidden
            dec_input = tf.expand_dims([targ_lang.word2idx['<start>']] * B
ATCH_SIZE, 1)

            # Teacher forcing - feeding the target as the next input
            for t in range(1, targ.shape[1]):
                # passing enc_output to the decoder
                predictions, dec_hidden, _ = decoder(dec_input, dec_hidden
, enc_output)

                loss += loss_function(targ[:, t], predictions)

                # using teacher forcing
                dec_input = tf.expand_dims(targ[:, t], 1)

        batch_loss = (loss / int(targ.shape[1]))

        total_loss += batch_loss

        variables = encoder.variables + decoder.variables

        gradients = tape.gradient(loss, variables)

        optimizer.apply_gradients(zip(gradients, variables))

        if batch % 100 == 0:
            print('Epoch {} Batch {} Loss {:.4f}'.format(epoch + 1,
                                            batch,
                                            batch_loss.numpy()))
    # saving (checkpoint) the model every epoch
    checkpoint.save(file_prefix = checkpoint_prefix)

    print('Epoch {} Loss {:.4f}'.format(epoch + 1,
                                        total_loss / N_BATCH))
    print('Time taken for 1 epoch {} sec\n'.format(time.time() - start))
checkpoint.restore(tf.train.latest_checkpoint(checkpoint_dir))
def evaluate(inputs, encoder, decoder, inp_lang, targ_lang, max_length_inp
, max_length_targ):

    attention_plot = np.zeros((max_length_targ, max_length_inp))
    sentence = ''
    for i in inputs[0]:
        if i == 0:
            break
        sentence = sentence + inp_lang.idx2word[i] + ' '
    sentence = sentence[:-1]

    inputs = tf.convert_to_tensor(inputs)

    result = ''

    hidden = [tf.zeros((1, units))]
    enc_out, enc_hidden = encoder(inputs, hidden)

    dec_hidden = enc_hidden
    dec_input = tf.expand_dims([targ_lang.word2idx['<start>']], 0)

    for t in range(max_length_targ):
        predictions, dec_hidden, attention_weights = decoder(dec_input, de
c_hidden, enc_out)

        # storing the attention weights to plot later on
        attention_weights = tf.reshape(attention_weights, (-1, ))
        attention_plot[t] = attention_weights.numpy()

        predicted_id = tf.argmax(predictions[0]).numpy()

        result += targ_lang.idx2word[predicted_id] + ' '

        if targ_lang.idx2word[predicted_id] == '<end>':
            return result, sentence, attention_plot

        # the predicted ID is fed back into the model
        dec_input = tf.expand_dims([predicted_id], 0)

    return result, sentence, attention_plot

def predict_random_val_sentence():
    actual_sent = ''
    k = np.random.randint(len(input_tensor_val))
    random_input = input_tensor_val[k]
    random_output = target_tensor_val[k]
    random_input = np.expand_dims(random_input,0)
    result, sentence, attention_plot = evaluate(random_input, encoder, dec
oder, inp_lang, targ_lang, max_length_inp, max_length_targ)
    print('Input: {}'.format(sentence[8:-6]))
    print('Predicted translation: {}'.format(result[:-6]))
    for i in random_output:
        if i == 0:
            break
        actual_sent = actual_sent + targ_lang.idx2word[i] + ' '
    actual_sent = actual_sent[8:-7]
    print('Actual translation: {}'.format(actual_sent))
    attention_plot = attention_plot[:len(result.split(' '))-
2, 1:len(sentence.split(' '))-1]
    sentence, result = sentence.split(' '), result.split(' ')
    sentence = sentence[1:-1]
    result = result[:-2]
```

Appendix IV: LSTM Algorithm of RNN Source Code

```
import pandas as pd
import numpy as np
import string
import matplotlib.pyplot as plt
%matplotlib inline
import tensorflow as tf
from sklearn.model_selection import train_test_split
import re
import os
from google.colab import drive
drive.mount('/content/drive')
df= pd.read_csv("/content/drive/MyDrive/gezoromcorpus.csv")
df.oromo = df.oromo.apply(lambda x: 'sos '+ x +' eos')
gez_texts = df.geez.to_list()
oro_texts = df.oromo.to_list()
from tensorflow.keras.preprocessing.text import Tokenizer
def tokenize_sent(text):
  '''
  Take list on texts as input and
  returns its tokenizer and enocded text
  '''
  tokenizer = Tokenizer()
  tokenizer.fit_on_texts(text)
  return tokenizer, tokenizer.texts_to_sequences(text)
gez_tokenizer, gez_encoded= tokenize_sent(text= gez_texts)
oro_tokenizer, oro_encoded= tokenize_sent(text= oro_texts)
gez_encoded[30:35]
gez_index_word = gez_tokenizer.index_word
GEZ_VOCAB_SIZE = len(gez_tokenizer.word_counts)+1
GEZ_VOCAB_SIZE
oro_encoded[30:35]
oro_index_word= oro_tokenizer.index_word
oro_word_index =oro_tokenizer.word_index
ORO_VOCAB_SIZE=len(oro_tokenizer.word_counts)+1
ORO_VOCAB_SIZE
max_gez_len = 0
for i in range(len(gez_encoded)):
  if len(gez_encoded[i]) > max_gez_len:
    max_gez_len= len(gez_encoded[i])
max_oro_len = 0
for i in range(len(oro_encoded)):
  if len(oro_encoded[i]) > max_oro_len:
    max_oro_len= len(oro_encoded[i])
print(max_gez_len)
max_oro_len
```

```
from tensorflow.keras.preprocessing.sequence import pad_sequences
gez_padded=pad_sequences(gez_encoded, maxlen=max_gez_len, padding='post')
oro_padded=pad_sequences(oro_encoded, maxlen=max_oro_len, padding='post')
gez_padded.shape
oro_padded.shape
gez_padded= np.array(gez_padded)
oro_padded= np.array(oro_padded)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(gez_padded, oro_padded
, test_size=0.2, random_state=0)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
from tensorflow.keras.layers import LSTM, Dropout, Dense, Embedding, Bidir
ectional, Add, Concatenate, Dropout
from tensorflow.keras import Input, Model
encoder_input = Input(shape=(max_gez_len, ))
encoder_embd = Embedding(GEZ_VOCAB_SIZE,512, mask_zero=True)(encoder_input
)
encoder_lstm = Bidirectional(LSTM(256, return_state=True))
encoder_output, forw_state_h, forw_state_c, back_state_h, back_state_c = e
ncoder_lstm(encoder_embd)
state_h_final = Concatenate()([forw_state_h, back_state_h])
state_c_final = Concatenate()([forw_state_c, back_state_c])

## Now take only states and create context vector
encoder_states= [state_h_final, state_c_final]
# Decoder
decoder_input = Input(shape=(max_oro_len -1,))
# For zero padding we have added +1 in Afaan Oromoo vocab size
decoder_embd = Embedding(ORO_VOCAB_SIZE, 512, mask_zero=True)
decoder_embedding= decoder_embd(decoder_input)
#We used bidirectional layer above so we have to double units of this lstm
decoder_lstm = LSTM(512, return_state=True,return_sequences=True )
# just take output of this decoder dont need self states
decoder_outputs, _, _= decoder_lstm(decoder_embedding, initial_state=encod
er_states)
# here this is going to predict so we can add dense layer here
# here we want to convert predicted numbers into probability so use softma
x
decoder_dense= Dense(ORO_VOCAB_SIZE, activation='softmax')
# We will again feed predicted output into decoder to predict its next wor
d
decoder_outputs = decoder_dense(decoder_outputs)

model5 = Model([encoder_input, decoder_input], decoder_outputs)


from tensorflow.keras.utils import plot_model
plot_model(model5)
```

```
model5.summary()


y_train


model5.summary()


model5.compile(optimizer='adam', loss='sparse_categorical_crossentropy', m
etrics=['accuracy'])


from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping

checkpoint = ModelCheckpoint("/content/drive/MyDrive/Gez-
Oro/RNN_Model/model_checkpoints/model5/", monitor='val_accuracy')

early_stopping = EarlyStopping(monitor='val_accuracy', patience=5)

callbacks_list = [checkpoint, early_stopping]


EPOCHS= 20


encoder_input_data = X_train
decoder_input_data = y_train[:,:-1]
decoder_target_data = y_train[:,1:]

# Testing
encoder_input_test = X_test
decoder_input_test = y_test[:,:-1]
decoder_target_test= y_test[:,1:]


history = model5.fit([encoder_input_data, decoder_input_data],decoder_targ
et_data,
                    epochs=EPOCHS,
                    batch_size=32,
                    validation_data = ([encoder_input_test, decoder_input_
test],decoder_target_test ),
                     callbacks= callbacks_list)


from matplotlib import pyplot
pyplot.plot(history.history['loss'], label='train')
pyplot.plot(history.history['val_loss'], label='test')
pyplot.legend()
pyplot.show()


loss_train = history.history['loss']
loss_val = history.history['val_loss']
epochs = range(0,20)
pyplot.plot(epochs, loss_train, 'g', label='Training loss')
pyplot.plot(epochs, loss_val, 'b', label='validation loss')
pyplot.title('Training and Validation loss')
pyplot.xlabel('Epochs')
pyplot.ylabel('Loss')
pyplot.legend()
pyplot.show()


loss_train = history.history['accuracy']
loss_val = history.history['val_accuracy']
epochs = range(0,20)
pyplot.plot(epochs, loss_train, 'g', label='Training accuracy')
pyplot.plot(epochs, loss_val, 'b', label='validation accuracy')
pyplot.title('Training and Validation accuracy')
pyplot.xlabel('Epochs')
pyplot.ylabel('Accuracy')
pyplot.legend()
pyplot.show()


model5.save_weights("/content/drive/MyDrive/Gez-
Oro/RNN_Model/saved_models/model5")
model5.load_weights("/content/drive/MyDrive/Gez-
Oro/RNN_Model/saved_models/model5")


from tensorflow.keras.layers import LSTM, Dropout, Dense, Embedding
from tensorflow.keras import Input, Model
encoder_model = Model(encoder_input, encoder_states)
decoder_state_input_h = Input(shape=(512,))
decoder_state_input_c= Input(shape=(512,))
decoder_states_input= [decoder_state_input_h, decoder_state_input_c]
dec_embd2 = decoder_embd(decoder_input)
decoder_output2,state_h2, state_c2 = decoder_lstm(dec_embd2, initial_state
=decoder_states_input)
deccoder_states2= [state_h2, state_c2]

decoder_output2 = decoder_dense(decoder_output2)

decoder_model = Model(
                      [decoder_input]+decoder_states_input,
                      [decoder_output2]+ deccoder_states2)
```

```
def get_predicted_sentence(input_seq):
    # Encode the input as state vectors.
    states_value = encoder_model.predict(input_seq)
        # Generate empty target sequence of length 1.
    target_seq = np.zeros((1,1))
       target_seq[0, 0] = oro_word_index['sos']
        stop_condition = False
    decoded_sentence = ''
        while not stop_condition:
        output_tokens, h, c = decoder_model.predict([target_seq] + states_
value)
        # Sample a token
        sampled_token_index = np.argmax(output_tokens[0, -1, :])
        if sampled_token_index==0:
          break
        else:
         # convert max index number to marathi word
         sampled_char = oro_index_word[sampled_token_index]
        # aapend it ti decoded sent
        decoded_sentence += ' '+sampled_char
          if (sampled_char == 'eos' or len(decoded_sentence) >= 72):
            stop_condition = True
                target_seq = np.zeros((1,1))
        target_seq[0, 0] = sampled_token_index
        states_value = [h, c]
        return decoded_sentence
def get_Oromo_sentence(sequence):
  sentence=""
  for i in sequence:
    if ((i != 0 and i != oro_word_index['sos']) and i != oro_word_index['e
os']):
      sentence = sentence + oro_index_word[i]+' '
  return sentence
def get_gez_sent(sequence):
    sentence =''
    for i in sequence:
      if(i!=0):
        sentence = sentence + gez_index_word[i]+' '
    return sentence
for i in range(16):
  print("GEEZ sentence:",get_gez_sent(X_train[i]))
  print("Actual OROMO Sentence:",get_Oromo_sentence(y_train[i]))
  print("Translated OROMO Sentence:",get_predicted_sentence(X_train[i].res
hape(1,9))[:-4])
  print("/’n")
```