Development of Bidirectional Amharic-Tigrinya Machine Translation using Recurrent Neural Networks

A Thesis Presented

By

Metages Ephrem

To

The Faculty of Informatics

of

St. Mary's University

In Partial Fulfillment of the

Requirements for the Degree of Master of Science

In Computer Science

January 2024

ACCEPTANCE

Development of Bidirectional Amharic-Tigrinya Machine Translation using Recurrent Neural Networks

By

Metages Ephrem

Accepted by the Faculty of Informatics, St. Mary's University, In Partial Fulfilment of the Requirements for the Degree of Master Of Science In Computer Science.

Thesis Examination Committee:

_____ _____ _____

Internal Examiner

{Full Name, Signature, and Date}

Dr. Million Meshesha  March 01, 2024

External Examiner

{Full Name, Signature, and Date}

_____ _____ _____

Dean, Faculty of Informatics

{Full Name, Signature, and Date}

January 2024

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Metages Ephrem Degneh

Full Name of Student

_____

Signature

Addis Ababa

Ethiopia

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

Tessfu Geteye (PhD)

Full Name of Advisor

Addis Ababa

Ethiopia

January 2024

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Almighty God who gave me the opportunity, strength, and wisdom to achieve whatever I have achieved so far.

Next, I am greatly indebted to several people who assisted me in completing this thesis. My most profound thanks go to my advisor Tessfu Geteye (PhD) for his support and guidance during the running of this thesis.

The completion of this thesis would not have been possible without the guidance and support of my family. I would like to thank my wife; whose love and guidance are with me in whatever I pursue

Table of Contents

List of Acronyms

AI	Artificial Intelligence
BIGRU	Bidirectional Gated Recurrent Unit
BILSTM	Bidirectional Long Short-Term Memory
BLEU	Bilingual Evaluation Understudy
CPU	Central Processing Unit
GPU	Graphical Processing Unit
GRU	Gated Recurrent Unit
HMT	Hybrid Machine Translation
LSTM	Long Short-Term Memory
METEOR	Metric for Evaluation of Translation with Explicit Ordering
MT	Machine Translation
NIST	National Institute of standard and technology
NLP	Natural Language Processing
NMT	Neural Machine translation
OOV	Out of vocabulary
RBMT	Rule-Based Machine translation
RNN	Recurrent Neural Network
SMT	Statistical Machine translation
TMMA	Tigray Mass Media Agency

List of Figures

List of Tables

Abstract

Machine translation employs Artificial Intelligence (AI) to autonomously convert text from one language to another, eliminating the need for human intervention. Contemporary machine translation transcends basic word-to-word conversion, aiming to convey the overall meaning of the source language text in the target language. It comprehensively analyzes all textual elements, discerning the intricate relationships between words. The advantages of machine translation include automated translation assistance, cost-effectiveness, rapid processing, and scalability.

Even though there has been a lot of movement in developing machine translation using Neural Machine Translation (NMT) there is only little research conducted for Ethiopian language pairs. This research aims to answer which Recurrent neural network (RNN) is best fitted for a bidirectional Amharic-Tigrinya machine translation depending on their Bilingual Evaluation understudy (BLEU) score.

The evolution of machine translation has progressed through rule-based, statistical, hybrid, and neural network approaches. Among neural network models, RNNs play a significant role, offering a diverse array of models. In this study, the researcher utilized a dataset consisting of 34,350 parallel Amharic and Tigrinya sentences, employing an 80/20 split for training and testing, respectively. The investigation aimed to identify the most suitable model for Amharic-Tigrinya and vice versa machine translation among options such as Long Short Term Memory (LSTM), LSTM with attention, Bidirectional Long Short Term Memory (BILSTM), BILSTM with attention, Gated Recurrent Unit (GRU), GRU with attention, Bidirectional Gated Recurrent Unit (BIGRU), and BIGRU with attention.

The research initially fine-tuned hyper-parameters, including the number of units, layers, and epochs for LSTM and GRU. Once optimal hyper-parameters were determined, they were applied to the respective models, and the results were analyzed based on BLEU scores. Among the models considered, BIGRU with attention emerged as the most effective for Amharic-Tigrinya and vice versa machine translation, as evidenced by its superior BLEU score performance. For Amharic-Tigrinya machine translation scoring a loss of 0.0775, accuracy of 0.9786, and BLEU score of 3.3415. To conclude, this research has systematically investigated the experimental setup, hyper-parameter tuning, and model construction processes, providing a comprehensive overview of Amharic-Tigrinya NMT. Each chapter contributes to a nuanced understanding of the specific challenges posed by this linguistic context. The evaluation of various RNN models underscores the significance of attention mechanisms in improving BLEU scores, offering crucial contributions to the domain of machine translation. Notably, the BIGRU model with attention emerges as the top performer, achieving the highest BLEU score of 3.3415, thereby substantiating its efficacy in enhancing translation accuracy for Amharic-Tigrinya language pairs.

Keywords: Machine Translation, Neural Network, Long Short Term Memory, Gated Recurrent Unit, BLEU, Amharic, and Tigrinya

Chapter One

Introduction

1.1. Background of the study

Language is a system of conventional spoken, using a sign or written symbols using which human beings, as members of a social group and participants in its culture, express themselves [1].

Amharic is the official language of Ethiopia. It is a Semitic language family like Tigrinya but Amharic has the largest number of speakers after Arabic [2]. “Amharic language serves as the working language of Ethiopia and is also the working language of several of the states within the Ethiopian federal system [5]. With 31,800,000 mother-tongue speakers as of 2018, plus another 25,100,000 second-language speakers [6]”. Having that many speakers in one specific language tends to give more attention to literature, educational books, journals, magazines, and so on to the language that has more speakers than the other. Both languages are Semitic family languages, and both are written from left to right using the Ge’ez script.

Tigrinya also spelled Tigrigna is an Ethio-Semitic language commonly spoken in Eritrea and northern Ethiopia's Tigre Region [3]. “There is a strong influence of Geʿez on Tigrinya literature, especially with terms relating to Christian life, Biblical names, and so on [4]”. Translation. For centuries people have used a variety of techniques to communicate with people who spoke a different language from them whether by sign or using a third person as a language translator. Since then translation has evolved into the digitalized era using computers to translate from one language to another.

Natural language processing (NLP) is the system of computational strategies for the evaluation and synthesis of natural language and speech. “Natural Language Processing is a theoretically prompted variety of computational strategies for studying and representing happening texts at one or greater ranges of linguistic evaluation for the motive of accomplishing human-like language processing for a variety of responsibilities or applications [25]”. NLPs face troubles with sarcasm due to the fact the phrases generally

used for explicit irony or sarcasm, will be effective or poor in definition however they're used to create the other effect. “AI primarily based totally on NLP cannot differentiate between the poor and effective meanings of phrases and terms supposed for sarcasm [26].” Nowadays there has been some remarkable success in integrating NLP with neural network machine translation.

Machine translation (MT) is a sub-field of computational linguistics or NLP that investigates the use of software to translate text or speech from one natural language to another [7]. Machine translation is the task of translating a sentence in a source language to a different target language. Approaches for machine translation can range from rule-based to statistical to neural-based. RBMT as its name suggests the translation is based on rules that are programmed by the programmer. On the other hand, SMT is a machine translation paradigm that generates translations primarily based totally on a probabilistic version of the translation process, the parameters of which might be anticipated from the parallel text [8]. The neural-based network is a technique in artificial intelligence that teaches computer systems to system information in a manner this is stimulated via way of means of the human brain. It is a kind of machine studying system, called deep learning, which makes use of interconnected nodes or neurons in a layered structure that resembles the human brain.

RNNs have demonstrated remarkable effectiveness in machine translation, showcasing their capacity to capture sequential dependencies within language data. The recurrent architecture of RNNs enables them to model context and dependencies across different positions in a sentence, a critical feature for tasks like translation. In a seminal work from Cho et al. [30] presented research illustrating the superior performance of RNNs compared to traditional phrase-based models in machine translation. Notably, their ability to understand and leverage contextual information has been pivotal in improving translation accuracy and fluency. “Additionally, the incorporation of attention mechanisms, as proposed by Bahdanau et al. [32] has further augmented RNNs' capabilities by enabling them to selectively focus on specific segments of the input sequence during translation.” These findings collectively underscore the effectiveness of recurrent neural networks in advancing the state-of-the-art in machine translation.

This proposal paper is prepared regarding how the recurrent neural network would react in Amharic to Tigrinya and vice versa machine translation. Although there have been many machine translations for a variety of different languages there are only a few when regarding the Ethiopian local language. There have been some Amharic to Tigrinya machine translations but there is little work if not any regarding neural network machine translation in the Local language. This research aims to find which RNN model is best suited for Bidirectional Amharic-Tigrinya machine translation.

1.2. Statement of the problem

In an increasingly globalized world, the need for effective communication across diverse languages is paramount. This research addresses the challenge of facilitating communication without shared language proficiency, emphasizing the necessity of developing such tools in multilingual countries like Ethiopia, where over 80 languages are spoken. While previous studies have explored machine translation between languages, this research stands out by leveraging neural network technology, aiming for heightened fluency and accuracy. The study focuses on Amharic to Tigrinya translation, a pertinent area considering the linguistic diversity in Ethiopia. The researcher contends that this approach can potentially alleviate issues such as the lack of technological language-learning methods, deficient communication skills in various languages, ineffective cross-language communication, and the dearth of fluency and accuracy in local language machine translation.

1.3. Research Question

The research main question is which model from LSTM, LSTM with attention, BILSTM, BILSTM with attention, GRU, GRU with attention, BIGRU, and BIGRU with attention is the best model?

1.4. Motivation

There are high quality and quantity of machine translation research regarding high-resourced languages, but when we come to low-resourced languages first of all there is not a lot of research to refer from, secondly when there is research on machine translation for low-level languages the overall performance of the system, accuracy, and fluency seems

to be much lower than the high-resourced languages. As it happens Ethiopian languages are grouped under low-resourced languages. This research aims to mitigate the gap of the accuracy and overall performance between high-resourced languages and low-resourced languages. Not only that but also this research aims to be a reference for further studies regarding neural network machine translation for Amharic and Tigrinya.

1.5. Objectives

1.5.1. General objective

The general objective of this research is to develop a Bidirectional Amharic-Tigrinya machine translation using recurrent neural network algorithms.

1.5.2. Specific objectives

The specific objectives of this research are as follows:

- To collect and prepare parallel text corpus for Amharic and Tigrinya
- To preprocess the parallel text corpus to make ready for model training and testing
- To design and develop recurrent neural network models for Amharic-Tigrigna and vice versa translation

1.6. Scope and limitation of the study

The research mainly focuses on the designing of machine translation system for Amharic to Tigrinya and vice versa by using recurrent neural network. Due to shortage of resources and budget the study only focuses on text to text translation for Amharic to Tigrinya and vice versa. So, the research does not focuses on other languages, speech, sign, or optical related machine translation.

1.7. Methodologies

Research methodology is the precise tactics or strategies used to identify, select, process, and examine data approximately a topic. In a studies paper, the methodology segment permits the reader to seriously compare a study's typical validity and reliability.

1.7.1. Literature review

A very thorough literature review needs to be conducted so that familiarity with the research topic and understanding of the current research in the field before carrying out a new investigation. The task of the literature review is to build an argument, not a library [22]. Conducting a research review also tells us what has been already done and identifies areas of controversy and contested claims.

1.7.2. Data collection and organization

After reviewing then the research will move to the next phase, which is data collection, since the research is on machine translation, we needs a parallel corpus. “Data collection is the process of sampling signals that measure real-world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer [23].” The collection will be normalized so that the result is more accurate.

The research uses parallel text corpus on Amharic and Tigrinya. To implement the proposed model Parallel corpuses are collected from apocalyptic and Holly bible books for training and testing purpose. The research will use 39,000 parallel sentences making the total of our research sentences 78,000. For preprocessing our data we plan to train and test our data 80% and 20% respectively.

1.7.3. Data preprocessing

In the context of preparing data for RNN machine translation, a meticulous preprocessing strategy is essential to enhance model performance and facilitate effective learning. Common preprocessing steps involve tokenization, text cleaning, lowercasing, padding, numerical encoding, handling out-of-vocabulary words, sequence alignment, splitting data into training and validation sets, creating batches, and incorporating embedding layers. These steps collectively contribute to shaping a well-structured dataset that aligns with the requirements of RNNs, ensuring optimal training and translation outcomes. As emphasized by Sutskever, Vinyals, and Le [35], proper data preprocessing is crucial in the development of neural machine translation models to address challenges related to sequence alignment, input representation, and overall model convergence. It lays

the foundation for successful training and subsequent deployment of RNN models in translation tasks [35].

1.7.4. Model development techniques

After gathering the corpus, the research will then move on to selecting which translation and language model is more suitable for the research. Language modeling is the use of various statistical and probabilistic techniques to determine the probability of a given sequence of words occurring in a sentence [24]. Choosing the right neural network translation model helps with the accuracy of this study.

This thesis will be conducted using a recurrent neural network model. The reason why we chose this model is that RNN is a type of neural network model that is particularly well-suited for processing sequential data, such as time series data or text. In an RNN, the hidden state of the network is influenced not only by the input at the current time step but also by the hidden state at previous time steps. This allows the network to maintain a "memory" of the input sequence and use it to make predictions or generate output. Since there are various models we can use we will try to compare some of them; LSTM, LSTM with attention, BILSTM, BILSTM with attention, GRU, GRU with attention, BIGRU, and BIGRU with attention. These are all RNN architectures that are commonly used for various NLP tasks such as machine translation, sentiment analysis, and named entity recognition. Here are there descriptions;

- LSTM: This model is chosen because of its use of a memory cell that allow information to be stored over long period of time and selectively forgotten or retrieved as needed[29].
- LSTM with attention: This model is chosen because it is a variant of LSTM that uses an attention mechanism to selectively focus on important parts of the input sequence when making predictions [32].
- BILSTM: This model is also chosen because it is a variant of LSTM that processes input sequences in both forward and backward directions. This allows the model to capture both past and future context when making predictions [31].

- BILSTM with attention: This model also chosen because it is also a variant of BILSTM model with an extension of attention based model which helps the model to be effective for speech recognition and machine translation tasks [34].
- GRU This model is chosen like LSTM this model also uses a memory cell, but they have fewer gates than LSTMs, making them faster to train and more computationally efficient [30].
- GRU with attention: This model is chosen because the attention mechanism helps the model to focus on important parts of the input sequence, and GRU provides a gating mechanism that helps in preventing the model from forgetting previous information [32].
- BIGRU: This model is chosen because it is a variant of GRU model that processes input sequences in both forward and backward directions. This allows the model to capture both past and future context when making predictions [32].
- Bi-GRU with attention: This model is chosen because it is a variant of GRU models that allows the model to capture information from both past and future contexts [33].

1.7.5. Model Evaluation

For evaluation, the most common way is a human evaluation which is conducted manually. Even though the manual method is the common way it can become hard if the data get bigger. In that case, there is also an automatic evaluation system like BLEU. The reason this study chose BLEU is because of BLEU for neural network machine translation include its simplicity, low computational cost, and the fact that it correlates well with human evaluations. In a study by Sutskever, BLEU was used to evaluate the performance of a neural machine translation system, and the authors found that the BLEU scores were strongly correlated with human judgments of translation quality [35]. Additionally, BLEU has been widely adopted as a standard evaluation metric for machine translation competitions, such as the Workshop on SMT, further highlighting its utility in the field.

1.8. Significance of the study

This research aims to create a recurrent neural network machine translation system that can translate text files from Amharic to Tigrinya and vice-versa so that pieces of knowledge can be shared between the two languages by translating language-written books, magazines, memos, emails, text messages, and so on to the other language. The research also has the purpose of determining the performance of Amharic to Tigrinya Recurrent neural network machine translation and laying a foundation for future research regarding neural networks. Furthermore, the research also aims to create cultural relations and friendships between the two language-speaking people since they can understand and communicate with each other using their language. Not only that but the research also aims to evaluate and determine how neural network translation improves the accuracy of Amharic to Tigrinya Recurrent neural network machine translation.

The aim of this study is to create a machine learning system that translates text files from Amharic to Tigrinya in order for knowledge to be shared between the two languages by translating language-written books, magazines, memos, emails, text messages, and so on to the other language. In doing the above we will be able to communicate with one another easily, and we will be able to learn another language than we already know. Not only that children and student books can be translated easily to have the same education curriculum.

The research is also important in contributing to good communication mediums between Amharic and Tigrinya language speakers. Any documented file can easily be translated and can be displayed in the desired language between Tigrinya and Amharic. Not only that but also books, journals, research papers, news, advertisement, and so on can be translated into the desired language so that sharing of some knowledge between the two language speakers.

1.9. Organization of the thesis

This research comprises six chapters, with the initial chapter serving as an introduction to the research. The first chapter delineates the problem statement, motive,

objectives, scope, limitations, significance, research methodology, and evaluations. The second chapter conducts a comprehensive review of prior research to ascertain existing knowledge in the field. Moving to the third chapter, the research provides an overview of both the Amharic and Tigrinya languages, analyzing their relationships and differences. The fourth chapter proposes an architecture and methodology to guide the research. Subsequently, the fifth chapter details the implementation of the proposed model, conducting experiments and evaluations. Finally, the sixth chapter encapsulates the research's conclusions and recommends avenues for future exploration.

Chapter Two

Literature Review

2.1. Introduction

In this chapter, we review research conducted regarding this study to identify gaps in current research and try to mitigate, if not fill, the gap. In order to do that, first, we review how machine translation became what it is today by examining machine translation history from its roots. Then, we see how machine translation works and its approaches. After that, since the research is conducted using RNN, we explore some of the RNN approaches and determine which evaluation method is suitable for the research. To understand what has been done, the researcher conducts some related research by classifying them into three categories: machine translation for foreign languages, machine translation for foreign and local languages, and last but not least, machine translation for local languages. By reviewing the above, we try to identify a gap and attempt to solve the problem.

2.2. History of machine translation

The history of MT can be traced back to the early 1950s when researchers first began to explore the possibility of using computers to translate human languages. This was a time of great optimism and excitement about the potential of computers to automate tasks that had previously been thought impossible. However, progress in MT has been slow, with many challenges still remaining today.

One of the earliest examples of MT was the Georgetown-IBM experiment in 1954, "which attempted to translate Russian sentences into English using an IBM 701 computer. The results were not particularly impressive, with many errors and a limited range of vocabulary, but the experiment marked an important milestone in the development of MT [58]".

In the following years, researchers continued to explore exploring different MT approaches rule-based systems, and statistical models. One of the most influential rule-based MT systems was the SYSTRAN system developed in the 1960s, "which was used by the United States government to translate documents from Russian and Chinese [59]".

In the 1990s, SMT emerged as a new approach to MT, "which uses statistical models to learn the probabilities of different word and phrase combinations in source and target languages. This approach has been highly successful, with SMT systems achieving high levels of accuracy in a number of language pairs [37]".

More recently, NMT has become the dominant approach to MT, using deep learning techniques to train neural networks to translate between languages [32]." NMT has achieved even higher levels of accuracy than SMT and is now widely used in commercial applications, such as Google Translate and Microsoft Translator [44]."

Despite these advances, there are still many challenges in MT, including dealing with idiomatic expressions, ambiguous words, and complex syntax. However, ongoing research and development are likely to continue improving the accuracy and usability of MT systems in the years to come.

2.3. Background of machine translation

MT is the process of automatically translating text from one language to another using computers. The history of machine translation dates back to the 1940s when the first attempts were made to translate languages using electronic computers. "The field of MT has seen significant advancements in recent years, especially with the use of deep neural networks and other machine-learning techniques [37]." MT is an important area of research as it has many practical applications, such as facilitating international communication, aiding in language learning, and supporting multilingual information access. However, MT is still a challenging task due to the complexities of human languages, and there is ongoing research to improve its accuracy and efficiency. To face these challenges machine learning has developed a variety of approaches discussed in section 2.4 and the flow of machine translation is described in the figure below. As shown below first we give the source text as an input and after processing the source text it will be transformed into the corresponding target text as an output.

Figure 2.1: Flow of machine translation

2.4. Approaches of machine translation

There are a wide variety of machine translation approaches that have been conducted by many of scholars. Most of the scholars have defined machine translation according to its nature, scholars like Bahdanau and Koehn have contributed a lot for the study of machine translation. The approaches that are listed in the figure below are some of the approaches that have been researched by many scholars. The following approaches are discussed in the sub-sections below briefly.

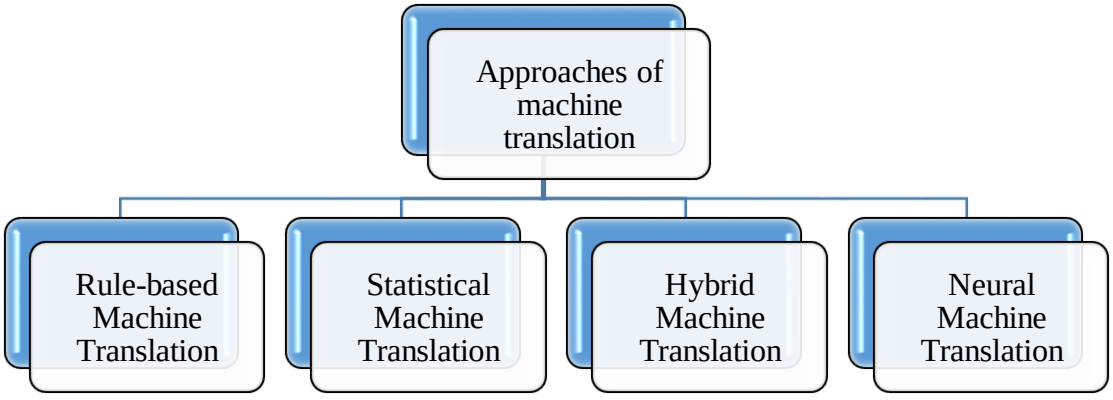

Figure 2.2: Approaches of machine translation

2.4.1. Rule-based machine translation (RBMT)

Rule-based machine translation is a type of machine translation that uses a set of linguistic rules and dictionaries to translate text from one language to another. RBMT relies on pre-defined rules to analyze the source text and generate the corresponding translated text [38]. These rules are typically created by linguistic experts and are designed to account for the grammatical and syntactical differences between the source and target languages as shown in figure 3. "RBMT has been used for decades and is known for producing high-quality translations, especially for technical or specialized texts [39]."

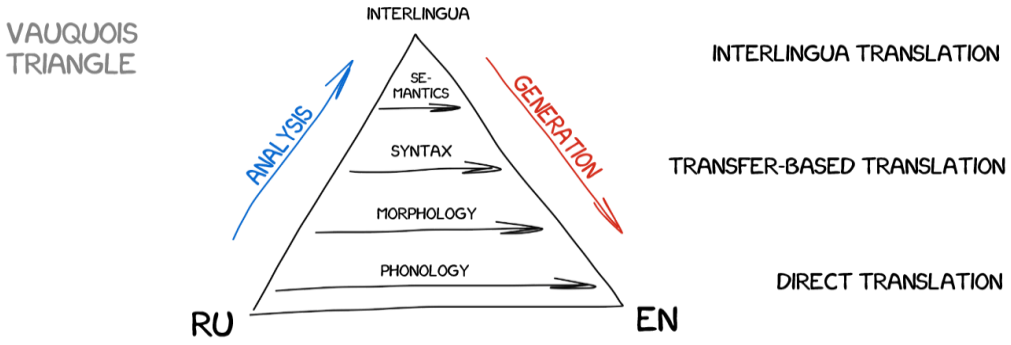

Figure 2.3: Vauquois triangle

The figure above was design by Jean Vauquois in 1962 and named Vauquois triangle. "The Vauquois triangle is a linguistic model that represents the relationships between the levels of linguistic analysis: syntax, semantics, and morphology [62]." The model proposes that there are three levels of analysis that interact with each other in a hierarchical manner. "The lowest level is morphology, which deals with the analysis of individual words and their inflections. The middle level is syntax, which deals with the analysis of the structure of sentences. The highest level is semantics, which deals with the analysis of meaning."

However, RBMT systems can be limited by the complexity and variability of language and may struggle with translating idiomatic expressions or colloquial language [40]. Despite these limitations, RBMT continues to be used in various applications, including legal translation and technical documentation. Some notable examples of RBMT

systems include SYSTRAN, Apertium, and Moses. The advantages and limitations of RBMT are listed below under Table 2.1.

Table 2.1: Advantages and Limitation of RBMT

Advantages	Limitations
Controlled output: RBMT systems produce translations that follow a set of predefined rules [60].	Limited coverage: RBMT systems rely on a set of predefined rules [37].
Linguistic accuracy: knowledge of the target language's grammar and syntax [61].	High development cost: Developing and maintaining a set of rules for RBMT can be expensive and time-consuming, [60]
Transparency: RBMT systems can be more transparent. [37].	Difficulty in scaling: to handle large volumes of data.

2.4.2. Statistical machine translation (SMT)

SMT is a type of machine translation that relies on statistical models to translate text from one language to another. SMT models learn from large parallel corpora, which are collections of texts in two or more languages that have been translated into each other as shown in the figure below. These models use probability distributions to estimate the likelihood of a translation given a source sentence and select the most likely translation based on these probabilities.

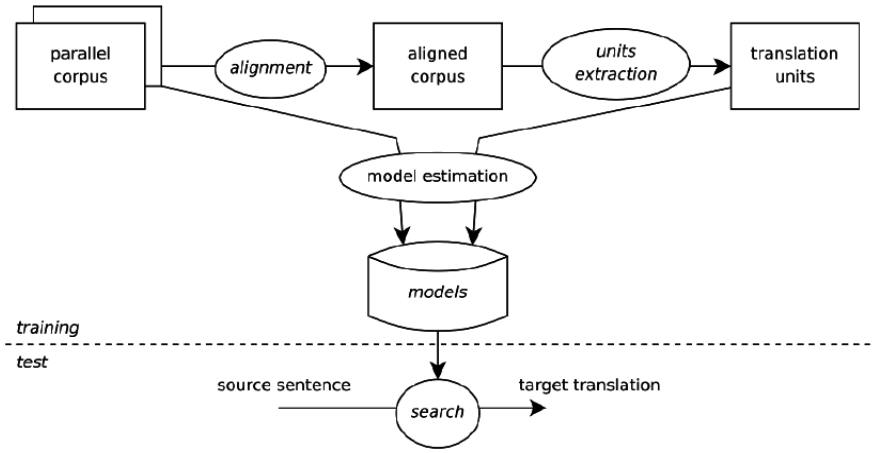

Figure 2.4: Architecture of SMT

According to Karan the above figure is for describing the pipe line of SMT. the process can be broken down into two primary stages: training and testing. "During the training stage, the system is constructed using available translation examples [63]. The first step of the training process involves the automatic induction of a word-to-word alignment from a parallel corpus that was previously aligned on a sentence-to-sentence basis. Additionally, prior to word alignment, the tokenization and categorization of the training corpus can also be viewed as part of the SMT architecture. The testing stage involves the translation of new sentences."

One of the earliest and most influential models of SMT is the IBM Model 1, which was developed in the 1990s [41]. Since then, "SMT has evolved to incorporate more sophisticated models, such as phrase-based models [42] and neural machine translation models [32]. While SMT has been largely surpassed by neural machine translation in recent years, it remains a valuable tool for low-resource language pairs and certain specialized domains." Below in Table 2.2. We will see the advantages and limitations of SMT.

Table 2.2: Advantages and Limitations of SMT

Advantages	Limitations
Good translation quality: especially for well-resourced language pairs [37].	Dependence on parallel corpora: the quality of the translations depends on the quality and size of the data [41].
Ability to learn from data: can automatically learn translation rules [41].	Domain-specific translation quality: may not perform well on domain-specific texts, such as technical or legal documents [42]
Flexibility: can be easily adapted to new domains by training on domain-specific parallel data [42].	Inability to handle morphology: where words can have multiple inflections and derivations [37].
Availability of open-source tools: such as Moses [62].	Lack of fluency: produce grammatically correct translations but not fluent [42].

2.4.3. Hybrid machine translation (HMT):

HMT is a translation approach that combines the strengths of both rule-based and statistical machine translation systems. HMT aims to improve the quality and accuracy of machine translation by utilizing the strengths of both systems while mitigating their weaknesses. According to Koehn, HMT is achieved by "combining the output of a statistical machine translation system with the output of a rule-based machine translation system, using various methods such as rule-based post-editing or statistical post-editing" [42].

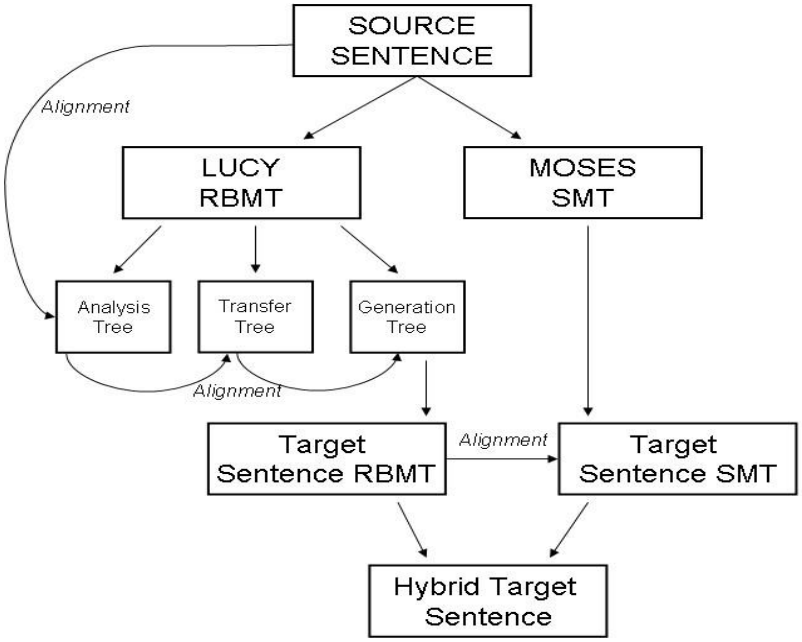

Figure 2.5: Architecture of HMT

The architecture of hybrid machine translation combines both rule-based and statistical machine translation as defined by Sabine's research the above figure is defined as: "The process involves initially detecting "interesting" phrases in the rule-based translation, followed by determining the most likely matching phrases in the output of other translation systems [64]." For the selected phrases, a factored substitution approach is utilized to decide whether to retain the original rule-based translation phrase or replace it with one of the alternative candidate phrases.

HMT has been shown to outperform both rule-based and statistical machine translation systems alone in terms of translation quality and accuracy [43]. However, "HMT systems can be complex and require significant resources and expertise to develop and maintain [37]. Despite these challenges, HMT remains a popular approach in machine translation research and development due to its potential for improving the overall quality and accuracy of machine translation." The advantages and limitations of hybrid machine translation are discussed in below.

Table 2.3: Advantages and limitations of HMT

Advantages	**Limitations**
Improved Translation Quality: by combining the strengths of different approaches [46].	**Complexity:** as they require the integration of multiple components [46].
Flexibility: can be designed to suit specific translation tasks or domains [46].	**Need for Expertise:** Developing and maintaining a hybrid MT system requires expertise [46].
Faster Deployment: as it can be built by combining pre-existing components [46].	**Limited Training Data:** such as a lack of sufficient training data [46].
Reduced Cost: as they can leverage existing resources and tools [46].	**Customization:** HMT may require customization for each new task [46].

2.4.4. Neural machine translation (NMT)

NMT is an approach that uses artificial neural networks to translate text from one language to another. NMT has gained significant attention in recent years due to its ability to learn from large amounts of data and produce high-quality translations. According to Sutskever, "NMT models consist of an encoder that encodes the input sequence into a continuous representation and a decoder that generates the output sequence from the encoded representation [35]. The neural network is a series of algorithms that endeavors to recognize underlying relationships in a se data set through a process that mimics how the human brain operates [12]." Neural networks are the simple clustering of primitive artificial neurons. This clustering occurs by creating layers that are then connected [13] as shown in figure below.

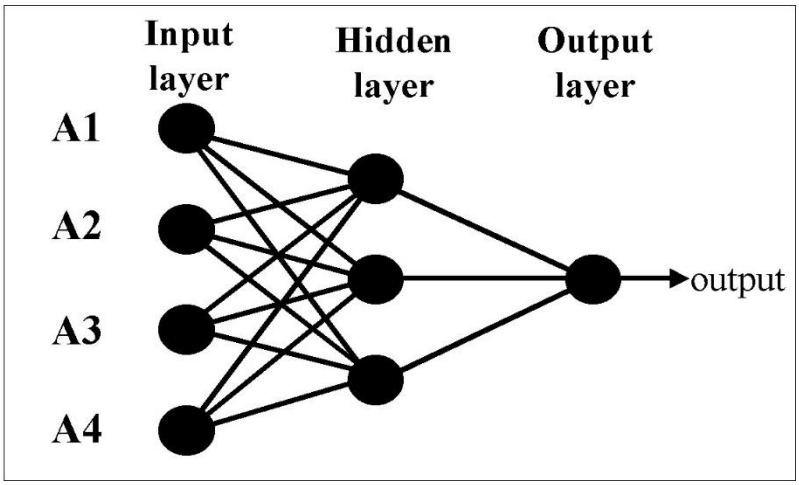

Figure 2.6: Architecture of NMT

The above figure is designed in the research conducted by Robert's which describe that neural network comprises multiple layers of interconnected neurons. "The neurons within each layer receive inputs from the preceding layer and transmit their outputs to the subsequent layer. The strength of these connections is represented by weights, which determine the behavior of the network. The architecture of a neural network is determined by the number of layers, the number of neurons in each layer, and the weights assigned to each connection." The neural network is trained by estimating these weights through an iterative process.

NMT has been shown to outperform traditional SMT approaches in terms of translation quality and fluency [44]. "Furthermore, NMT has also been shown to be effective in low-resource settings, where there is limited parallel data for training [45]. Despite these advantages, NMT models can be computationally expensive and require large amounts of training data [46]. However, the recent advancements in hardware and techniques for training large neural networks have made NMT a promising approach in machine translation research and development." There are both advantages and limitations associated with the use of neural networks for machine translations that are briefly described in Table below.

Table 2.4: Advantages and limitations of NMT

Advantages	Limitation
Improved Translation Quality: produce better translation quality compared to traditional statistical machine translation methods [46].	**Data Dependency:** NMT models require a large amount of parallel data to be trained effectively. [46].
End-to-End Approach: this means that the system learns to translate from the input language to the output language directly without requiring intermediate representations. [32].	**Training Time and Resource Intensive:** Training an NMT model can be time and resource-intensive [32].
Ability to Handle Long Sentences: NMT systems are better equipped to handle long sentences compared to SMT systems [32].	**Lack of Transparency:** difficult to understand how they arrive at a particular translation. [46].
Adaptability: learn to translate from a large amount of parallel data and can generalize to new language pairs and domains [46].	**Out-of-Vocabulary (OOV) Words:** which are words that are not present in the training data which lead to errors in translation when translating sentences with OOV words [32].

2.4.4.1. Recurrent neural network models

RNNs are designed to process sequential data by allowing information to flow in cycles. "RNNs are composed of a network of neurons that have feedback connections, allowing the output of one neuron to become the input of another neuron in the next time step. They are commonly used for applications such as speech recognition, language modeling, and natural language processing [10]." RNNs are a popular class of deep-learning models used for machine translation. Among the different types of RNNs, LSTM, BiLSTM, GRU, and BiGRU are some of the most commonly used approaches:

LSTM: is a type of RNN architecture that is designed to avoid the problem of vanishing gradients that occurs in traditional RNNs. It has a special memory cell that allows it to remember information for a long time. LSTM has been successfully used in machine

translation tasks, such as in the work by Sutskever who used LSTM-based encoder-decoder architecture for machine translation [35].

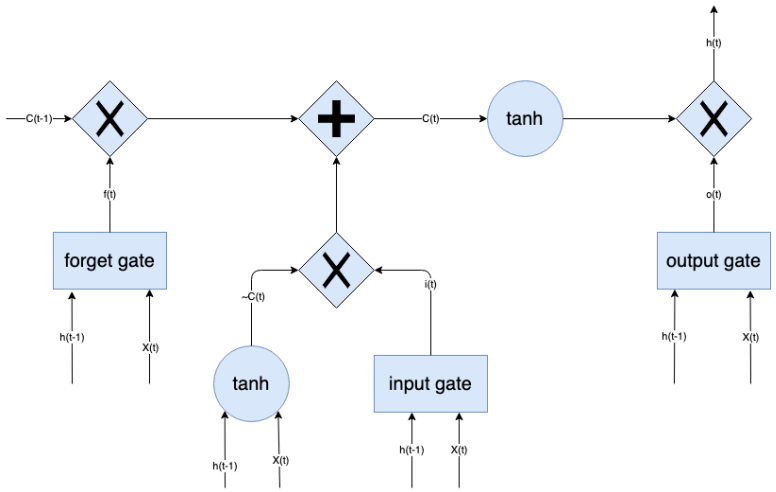

Figure 2.7: Architecture of LSTM

As shown in the figure above the “LSTM cell is comprised of three gates - an input gate, an output gate, and a forget gate - which regulate the flow of data within the network. At any given time step "t," the LSTM cell receives an input "Xt" and outputs a hidden state "Ht" and two different types of states - an input state "Cet" and an output state "Ct" - that are derived from the previous hidden state "Ht-1" and the input "Xt".” The values of the three gates - "it", "ft", and "ot" - are also computed at this time step. The values of the input state "Ct" and the hidden state "Ht" are then propagated through the network according to the LSTM architecture [65].

BLSTM: is a type of RNN that combines the advantages of both forward and backward RNNs. “It is able to capture context from both past and future inputs, which is particularly useful for sequence-to-sequence tasks such as machine translation. BLSTM has been used in many machine translation tasks, such as in the work by Bahdanau, where they used a BLSTM-based encoder-decoder architecture for machine translation [32].”

The BLSTM model utilizes both forward and backward LSTM connections to extract coarse-grained features. “The LSTM architecture includes an input gate i, forget gate f, and output gate o, which work together to determine how incoming information

should be incorporated into the inner memory cell C. Using predefined rules, the network decides whether to retain or discard the information [66].” This process ensures that only relevant information is preserved, while irrelevant information is discarded via the forget gate. Given an input sequence x = (x_0.....x_i) at time t and the hidden states of a BLSTM layer, the resulting hidden states h = (h_0.....h_t) can be obtained.

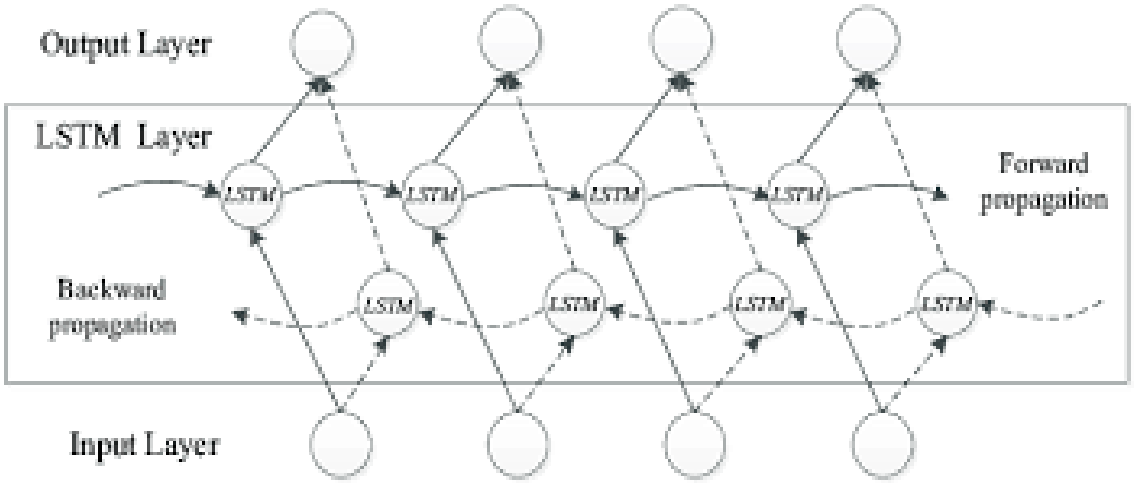

Figure 2.8: Architecture of BILSTM

“The forget gate receives two inputs [67], namely the output of the hidden layer at the previous moment (h_{t-1}) and the input at the current moment (x_t). It uses these inputs to selectively forget information in the cell state Ct.” Mathematically, this can be represented in Equation 2.1:

$$Ft= \text{Sigmoid} (W_{xf}X_t + W_{hf}h_{t-1}+b_f) \text{ …….. (Eq 2.1)}$$

“Where Ft is forget gate at time step t, sigmoid function, denoted as σ, is an activation function that squashes the input values to the range [0, 1], wfx as weight matrix for the input features (Xt) at time step t. It represents the influence of the current input on the forget gate, Xt as Input features at time step t, Whf as Weight matrix for the hidden state (ht-1) from the previous time step. It represents the influence of the previous hidden state on the forget gate, ht-1 as Hidden state from the previous time step, and bf as Bias term”

Together, the input gate and a tanh function collaborate to regulate the integration of fresh data. The tanh function produces a new candidate vector, while the input gate generates a value between 0 and 1 for each element in $\acute{C}_t$, determining the extent to which new information is incorporated [67]. This can be represented in 2.2:

$$Ct = \text{Sigmoid}\ (F_t.C_{t-1}+i_t.\acute{C}_t) \quad \ldots\ldots \text{(Eq 2.2)},$$

$$i_t = \text{Sigmoid}\ (W_{xi}Xt + W_{hi}h_{t-1} + b_t) \quad \ldots\ldots\ldots \text{(Eq 2.3)},$$

$$\acute{C}_t = \tanh\ (W_cX_t + W_ch_{t-1} + b_c) \quad \ldots\ldots\ldots\ldots \text{(Eq 2.4)}$$

In Equation 2.2, 2.3, and 2.4 the equations collectively describe how the LSTM cell state is updated at each time step based on the input features (Xt), the previous hidden state (ht-1), and the previous cell state (Ct-1). The forget gate (Ft) regulates what information to forget, the input gate (it) determines what new information to store, and the candidate value (Ćt) represents the new information to be considered for inclusion in the updated cell state. These mechanisms enable LSTMs to selectively retain and update information over long sequences, making them effective for capturing dependencies in sequential data.

The output unit state can be filtered by adjusting the output gate [67], which determines the amount of filtration applied, as shown in Equation 2.5:

$$Ot = \text{Sigmoid}\ (W_{xo}X_t + W_{ho}h_{t-1} + b_o) \quad \ldots\ldots\ldots \text{(Eq 2.5)}$$

Where Ot: Output gate at time step t, Sigmoid: The sigmoid activation function, Wxo: Weight matrix for the input features (Xt), Xt: Input features at time step t, Who: Weight matrix for the hidden state (ht-1) from the previous time step, ht-1: Hidden state from the previous time step, bo: Bias term.

The hidden states of "ht," which is a packet vector produced from each packet, can be described as the combination of "$\overleftarrow{h}_t$" and "$\vec{h}_t$ through concatenation [67]. This can be written as follows:

$h_t = \overleftarrow{h}_t + \overrightarrow{h}_t$ ..... (Eq 2.6),

$\overrightarrow{h}_t = \tanh (W_x\overrightarrow{h}X_t + W\overrightarrow{h}\overrightarrow{h}h_{t-1} + b_h)$ ..... (Eq 2.7)

$\overleftarrow{h}_t = \tanh (W_x\overleftarrow{h}X_t + W\overleftarrow{h}\overleftarrow{h}h_{t-1} + b_h)$ .....(Eq 2.8)

Where the symbol '.' refers to the point wise product. The variable x denotes the input of a time series dataset with heterogeneous components. Additionally, the variables $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ represent the hidden states of the forward LSTM layer and backward LSTM layer, respectively, at time t. The matrices W correspond to the connection weights between two units, and the vectors b denote the bias terms [67].

GRU: is another type of RNN that is similar to LSTM in that it also addresses the vanishing gradient problem. "It has a simpler architecture than LSTM and is faster to train. GRU has been used in machine translation tasks, such as in the work by Wu, who used a GRU-based encoder-decoder architecture for machine translation [44]."

The GRU is a variation of LSTM that features a less complex design, depicted in Figure below. "GRU incorporates two gates: the Update gate and the Reset gate. The Update gate governs the decision of whether to update the hidden state with a new one, while the Reset gate determines whether to disregard the previous hidden state [68]." Their outputs are z_t and r_t, respectively.

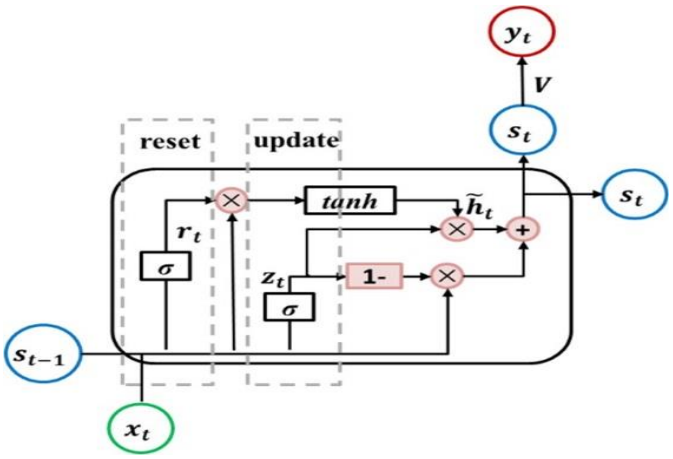

Figure 2.9: Architecture of GRU

$$z_t = \text{Sigmoid}\ (W_z.[S_{t-1}, X_t] + b_z) \ldots\ldots\ldots \text{(Eq 2.9)}$$

$$r_t = \text{Sigmoid}\ (W_r.[S_{t-1}], X_t] + b_r) \ldots\ldots\ldots \text{(Eq 2.10)}$$

$$\bar{S}_t = \tanh\ (W_s.[r_{t-1}, X_t] + b_s) \ldots\ldots\ldots\ldots\ldots \text{(Eq 2.11)}$$

$$S_t = (1-Z_t).S_{t-1} + Z_t + \bar{S}_t) \ldots\ldots\ldots\ldots\ldots\ldots \text{(Eq 2.12)}$$

In Equation 2.9, 2.10, 2.11, and 2.12: The update gate (Zt) determines how much of the previous state to keep, the reset gate (rt) determines how much to forget, and the memory content ($S\bar{t}$) represents the new candidate values. The final hidden state (St) is a combination of the old state and the new candidate values based on the update gate. These mechanisms help the model capture dependencies over sequential data.

BGRU: is a variant of GRU that, like BLSTM, can capture context from both past and future inputs. It has been used in machine translation tasks, such as in the work by Zhou, who used a BGRU-based encoder-decoder architecture for machine translation [33].

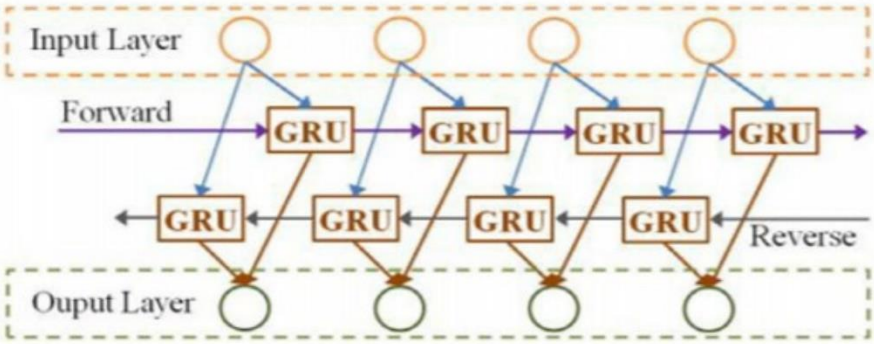

Figure 2.10: Architecture of BIGRU

The BIGRU model operates in parallel by utilizing two GRU layers that operate simultaneously. The model's output is a fusion of the outputs generated by the forward and backward GRU sequences [69]. This form of parallel operation is reminiscent of parallel computing, a computing method that executes multiple computational tasks concurrently. In this study, Bi-GRU was one of the models evaluated, and it was found to perform computations in parallel.

In conclusion, LSTM, BILSTM, GRU, and BGRU are all effective approaches for machine translation tasks in recurrent neural networks. The choice of which approach to use may depend on factors such as the size of the training data, the complexity of the

translation task, and the computational resources available. For the purpose of this research, we are planning to use BILSTM model.

2.5. Evaluation methods

Evaluation is crucial for measuring the quality of bilingual machine translation systems. A range of evaluation metrics is used to evaluate the performance of these systems. Among the most widely used metrics for evaluating neural network-based bilingual machine translation systems are the BLEU and METEOR scores. While BLEU is an n-gram-based metric that focuses on lexical similarity between the machine-generated translations and the reference translations, METEOR takes into account additional factors such as paraphrasing and synonyms. Recently, "there has been increasing interest in exploring alternative evaluation metrics that can better capture translation quality, such as human evaluation methods and multi-dimensional metrics that measure various aspects of translation quality [27]." However, since evaluation is a critical aspect of neural network machine translation. It allows us to measure the quality of translations produced by a model and compare it to other models or human translations. There are several evaluation methods for neural network machine translation, including:

NIST Score: Another evaluation metric is the National Institute of Standards and Technology (NIST) score. "It measures the similarity between the model output and a set of human translations using a weighted geometric mean of n-gram precision scores."

METEOR Score: The Metric for Evaluation of Translation with Explicit ORdering (METEOR) score is a metric that "considers both the fluency of the translation and the adequacy of the meaning conveyed".

Human Evaluation: In addition to automated metrics, human evaluation is also essential in evaluating machine translation models. It involves having human translators or bilingual speakers evaluate the quality of the translations produced by the model.

The Bilingual Evaluation Understudy (BLEU) score: is a widely used evaluation metric for machine translation. It measures the similarity between a machine-generated translation and a set of reference translations. "The BLEU score is calculated by counting the number of n-grams (continuous sequences of words) in the machine-generated

translation that appear in the reference translations. The precision score for each n-gram is then calculated as the number of times the n-gram appears in the machine-generated translation divided by the total number of n-grams in the machine-generated translation. The precision scores are then combined using a modified form of the geometric mean, known as the brevity penalty, to calculate the BLEU score. The brevity penalty is used to adjust the score based on the length of the machine-generated translation relative to the reference translations," to ensure that shorter translations are not unfairly rewarded. The formula for calculating the BLEU score is:

BLEU = brevity penalty * exp (sum (log (precision_n))) …… (Eq 2.13)

Where "n" represents the n-gram length and "precision_n" represents the precision score for n-grams of length "n". The brevity penalty is calculated as:

Brevity penalty = min (1, reference length/translation length) …….. (Eq 2.14)

Where "reference length" is the total length of the reference translations and "translation length" is the length of the machine-generated translation.

In Equation 2.14 the Brevity Penalty is a correction factor applied to the precision component of the BLEU score to address biases towards shorter translations. It helps provide a more balanced evaluation of the translation quality by considering the length of the generated translation in comparison to the reference translations.

Furthermore, BLEU has been shown to be effective for evaluating low-resource RNN-based machine translation systems, as it has been found to provide a good estimate of the performance of these systems even when only a small amount of training data is available [9]. In summary, the BLEU score is calculated by counting the number of overlapping n-grams between the machine-generated translation and the reference translations, calculating the precision scores for each n-gram, and combining them using a modified geometric mean with a brevity penalty to adjust for differences in length. The resulting score provides a measure of the similarity between the machine-generated translation and the reference translations. Comparing the above testing methods BLEU and human evaluation are selected.

2.6. Related Works

2.6.1. Machine translation for foreign language pairs

Machine translation in foreign languages has been evolving before computers were even created. In the mid-1930s, a French-Armenian Georges Artsrouni and a Russian Petr Troyanskii implemented patents for ‘translating machines’. Of the two, Troyanskii's become the more significant, providing now no longer the best technique for an automated bilingual dictionary. However, royanskii’s thoughts have been now no longer acknowledged till the end of the 1950s. Since then machine translation has been extremely progressing up to date. It is feasible to hint thoughts approximately mechanizing translation processes lower back to the 17th Century, however sensible opportunities got here best in the 20th century. Here are some researches reviews of machine translation that is conducted for foreign languages. Here are some examples;

"Neural Machine Translation by Jointly Learning to Align and Translate" is a seminal paper by Bahdanau et al. that proposed “an encoder-decoder architecture with an attention mechanism to address the problem of long-term dependencies in machine translation [32]. The attention mechanism allows the decoder to selectively focus on parts of the input sentence that are relevant to the current output word. They trained their NMT models on the WMT'14 English-to-French dataset and the WMT'14 English-to-German dataset with 36 million and 4.5 million sentence pairs respectively. They used a 4-layer LSTM with 1,000 hidden units for both the encoder and decoder. They also used a word embedding size of 1,000, a batch size of 80, and a beam search size of 12. The dropout rate was set to 0.5, and the learning rate was set to 0.001.” They used BLEU and METEOR evaluation methods with scores of 41.0 and 25.9 respectively.

"Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation" is a paper by Wu et al. that introduced Google's neural machine translation system [44]. The article describes an “NMT system that uses an encoder-decoder architecture with an attention mechanism. The system is trained on large-scale datasets, including WMT'14 English-to-German and English-to-French datasets, and achieves state-of-the-art results with a BLEU score of 25.9 and 38.0, respectively. The system uses a 4-layer LSTM with 1,024 hidden units for both the encoder and decoder, a

word embedding size of 1,024, a batch size of 128, and a beam search size of 10. The dropout rate is set to 0.2, and the learning rate is scheduled using an inverse square root decay strategy."

The paper "Attention Is All You Need" proposes a new neural network architecture called "Transformer for sequence-to-sequence learning tasks such as machine translation [71]. The Transformer model is a neural network architecture that does not use any recurrent or convolutional layers, and instead relies solely on self-attention mechanisms to compute representations of input and output sequences. The model consists of an encoder and decoder, each containing several layers of self-attention and feed-forward neural networks. The authors used several datasets for machine translation tasks, including WMT 2014 English-German (4.5 million sentence pairs) and WMT 2014 English-French (36 million sentence pairs). The experimental results showed that the Transformer model outperformed previous state-of-the-art methods on several machine translation tasks, achieving BLEU scores of 28.4 for English-German and 41.0 for English-French." The authors also demonstrated that the Transformer model can be used for other sequence-to-sequence tasks such as language modeling with lower perplexity scores compared to previous state-of-the-art methods.

2.6.2. Machine translation for Foreign and Local languages pairs

Machine translation is the process of using software to automatically translate text or speech from one language to another. For the Ethiopian language machine translation has been a challenge due to its complex grammatical structures and unique writing system. These systems use large amounts of data and sophisticated algorithms to learn the language's patterns and generate accurate translations. While there is still room for improvement, the development of machine translation for Ethiopian languages has the potential to facilitate communication and increase accessibility for Ethiopians both within their country and around the world.

Here are some of the researches that are conducted regarding machine translation of foreign languages to our local languages and vice-versa. This is helpful in order to see the complexities of languages and how they can be related. Here are some examples;

The paper titled "Preliminary experiments on English-Amharic statistical machine translation" presents an experimental study of English-Amharic statistical machine translation using various models and evaluation metrics [48]. “The paper compares three models for machine translation: PBSMT, NMT, and a hybrid approach using a dataset of 46,147 sentence pairs. PBSMT uses a phrase table with a maximum phrase length of 10 and a distortion limit of 6, while NMT uses a model with 3 layers, 512 hidden units, and an embedding size of 256. The hybrid model uses the same hyper-parameters as NMT but also includes a phrase-based translation model. They evaluate the models using three metrics: BLEU, NIST, and METEOR. The hybrid model outperformed PBSMT and NMT in all three metrics, achieving a BLEU score of 16.13, a NIST score of 6.24, and a METEOR score of 15.03.”

The paper “Bidirectional English – Afaan Oromo Machine Translation Using Hybrid Approach” proposes a bidirectional machine translation system that translate from Afaan Oromo to English Language and vice versa. “The general objective of this research work was to develop a bidirectional English – Afaan Oromo MT system using a hybrid approach [72]. Since the system is bidirectional, two language models are developed; one for English and the other for Afaan Oromo. Translation models that assign a probability that a given source language text generates a target language text are built and a decoder that searches for the shortest path is used. Two major experiments are conducted by using two different approaches and their results are recorded. The first experiment is carried out by using a statistical approach. The result obtained from the experiment has a BLEU score of 32.39% for English to Afaan Oromo translation and 41.50% for Afaan Oromo to English translation. The second experiment is carried out by using a hybrid approach and the result obtained has a BLEU score of 37.41% for English to Afaan Oromo translation and 52.02% for Afaan Oromo to English translation from the result, the researcher deduces that the hybrid approach is better than the statistical approach for the language pair and a better translation is acquired when Afaan Oromo is used as a source language and English is used as a target language.”

The paper titled "Tigrinya morphological segmentation with bidirectional long short-term memory neural networks and its effect on English-Tigrinya machine translation" presents an experimental study on Tigrinya morphological segmentation and its impact on English-Tigrinya machine translation [11]. The paper uses a “BILSTM neural network for Tigrinya morphological segmentation with two layers, 100 hidden units, and a learning rate of 0.001 and a PBSMT model for English-Tigrinya machine translation with a maximum phrase length of 10 and a distortion limit of 6. The authors used a dataset of 80,505 Tigrinya sentences and their corresponding English translations for training and testing the machine translation model. The authors used two evaluation metrics to assess the performance of the machine translation model: BLEU and ChrF. the model with morphological segmentation achieved a BLEU score of 19.39 and a ChrF score of 37.22, while the model without morphological segmentation achieved a BLEU score of 16.42 and a ChrF score of 35.30.”

The paper “Attention-based Neural Machine Translation from English-Wolaytta” by Mekdes melese [73] which is based on the “development of the English-Wolaytta machine translation system involved the meticulous training on a parallel corpus encompassing religious texts and frequently used sentences or phrases applicable to everyday communication. A comprehensive dataset comprising 27,351 parallel English-Wolaytta sentences was curated, with an 80/20 ratio employed for training and testing the system. Prior to training, the data underwent preprocessing to ensure compatibility with the neural machine translation framework. The proposed model adopted a Sequence-to-Sequence concept with an LSTM encoder and decoder architecture, integrating an attention mechanism for enhanced translation accuracy. The system's efficiency was evaluated using the BLUE score metric, revealing a notable BLEU score of 5.16 and an impressive 88.65% accuracy. To validate the significance of the attention mechanism, a non-attention model was developed and compared, affirming that the attention mechanism significantly improved translation quality. This research underscores the successful application of advanced neural machine translation techniques for English-Wolaytta language pairs, showcasing the potential for improved cross-language communication.”

2.6.3. Machine translation for local languages pairs

Machine translation can provide significant benefits for Ethiopian languages, including increased access to information, reduced translation costs, and preservation of linguistic diversity. Machine translation can make information available to speakers of Ethiopian languages who may not be proficient in other languages, which can improve access to education, healthcare, and government services [11]. Machine translation can also reduce the costs associated with translation, making it more accessible to individuals and organizations who may not have the resources to invest in professional translation services.

In addition, machine translation can help to preserve and promote Ethiopian languages, which are threatened by language shift and language endangerment. Machine translation can help to increase the visibility and accessibility of Ethiopian languages, which can promote language use and encourage language maintenance and revitalization efforts [49]. This can have important implications for preserving linguistic diversity, promoting cultural heritage, and improving access to information and resources for speakers of Ethiopian languages. Here are some machine translation research conducted for Ethiopian local languages.

The paper titled "Parallel Corpora for bi-Directional Statistical Machine Translation for Seven Ethiopian Language Pairs" extensively studies machine translation between seven Ethiopian languages [50]. The paper uses a “PBSMT model for machine translation between the seven Ethiopian language pairs. The authors also experimented with an NMT model but found it to be less effective due to the limited size of the training data. The authors collected parallel corpora for seven Ethiopian language pairs, including Amharic-Tigrinya, Amharic-Oromo, and Tigrinya-Wolaytta. The size of the parallel corpora varied from 1,142 to 58,373 sentence pairs per language pair. They also used monolingual data to train language models for each language. The paper provides a detailed description of the hyper-parameters used for each model. For PBSMT, the authors used a maximum phrase length of 10 and a distortion limit of 6. For language modeling, they used a modified Kneser-Ney smoothing with a discount of 0.75 and a back-off weight of 0.4. The best performance was achieved for the Amharic-Tigrinya language pair, with a BLEU

score of 21.22. The worst performance was achieved for the Tigrinya-Wolaytta language pair, with a BLEU score of 2.2."

"Lesan – Machine Translation for Low-Resource Languages" is a machine translation system designed specifically for low-resource languages, which often lack the training data necessary for effective machine translation. "The system uses a combination of rule-based and statistical machine translation methods and a neural machine translation model to produce both accurate and linguistically sound translations. One of the key features of Lesan is its ability to incorporate linguistic knowledge and resources, such as morphological analyzers and bilingual dictionaries, to improve the quality of translations. The system has been tested on a variety of low-resource languages, including Somali, Oromo, and Tigrinya, and has achieved significant improvements in translation quality compared to other machine translation systems." According to a study, "Lesan outperformed other state-of-the-art machine translation systems on multiple evaluation metrics when tested on the Tigrinya-English translation task [51]. The study found that Lesan achieved a human evaluation score with a top score of English to Amharic with a score of 3.25 for sentence and 3.17 for a story. And with a low score of Tigrinya to Amharic with a score of 1.94 for sentences and 1.77 for a story."

This research "Automatic Amharic-Tigrinya Translation using statistical machine translation approach" considers the use of unsupervised segmentation for SMT Amharic on Tigrigna [52]. "The experiment was conducted with 14,231 parallel sentences taken from parliamentary documents, school textbooks, medical records, and the Bible. In parallel, the collected data were randomly split into 90% of the training set and 10% of the testing and debugging set. The monolingual Tigrigna corpus of 25,875 sentences was assembled by DWOT, the Tigray Mass Media Agency (TMMA), and online texts by Tigrigna to develop a language model capable of ensuring fluent text output. The Moses open-source statistical machine translation system was used in the experiment for training, tuning, and decoding. Parallel data were aligned using the Giza++ toolkit. SRILM was used to create the language model and Morfessor was used for data segmentation. The first basic experiment of the translation system scored 37.98% with a BLEU score greater than 30, indicating an understandable translation. The second experiment was performed using

unsupervised data segmentation in Amharic and Tigrinya used in the underlying translation system, including the language model for Tigrinya. As a result, we get an increase of 2.74% over the benchmark."

2.7. Summary

Based on the related research review there are little research conducted for local language pair using RNN models. A research done by Mekdes Melese [73] have a related objective to this research by aiming to develop a English-Wollyta NMT using LSTM and LSTM with attention and compare which model is optimal considering their BLEU score. There is also a research conducted by Genet Worku [70] which uses BILSTM as an encoder and LSTM as a decoder. The research uses 50,000 sentence pairs of Ge'ez and Amharic sentences with a grid search for fine tuning. This research aims to address the existing gap in the literature regarding the identification of the most effective RNN model for Amharic-Tigrinya and vice versa translation. The primary objective of this research is to ascertain the optimal RNN model for Amharic-Tigrinya and vice versa translation. The RNN models under consideration include LSTM, LSTM with attention, BILSTM, BILSTM with attention, GRU, GRU with attention, BIGRU, and BIGRU with attention. The evaluation is based on their respective BLEU scores.

The research methodology first was aiming to fine-tune the hyper-parameter by a grid search just like Genet's research [70] but due to lack of memory resource this research uses a manual fine tuning techniques that is used by Mekdes [73]. The research methodology involves an initial fine-tuning of hyper-parameters for LSTM and GRU models. Subsequently, the BLEU scores of the other RNN models are observed, taking into account the fine-tuned hyper-parameters. Through this systematic approach, the research contributes valuable insights to the field of machine translation, aiding in the selection of optimal models for this specific language pair.

Chapter Three
Overview of the languages

3.1. Introduction

This chapter mainly focuses on the overview of both Amharic and Tigrinya languages. In this chapter, we will try to see both languages from the perspective of their structure based on alphabets, punctuation marks, syntax, accusative case, definitive article, and affixation.

3.2. The Amharic language

"Amharic is a Semitic language that is spoken by approximately 22 million people in Ethiopia, where it is the official language. It is the second-most spoken Semitic language in the world, after Arabic. Amharic has a long history and has been used for centuries in literature, religion, and government." The language is written using the Ge'ez script, which has been in use since the 4th century. "Amharic has a complex grammar system, with a total of 33 consonants and 7 vowels. The language has also had a significant influence on other Ethiopian languages, with many borrowing words and phrases from Amharic [1]."

Amharic plays a significant role in Ethiopian society, as it is the language of government, education, and the media. "It is also used in religious contexts, particularly in the Ethiopian Orthodox Church. Despite its widespread use, Amharic is classified as a vulnerable language by the UNESCO Atlas of the World's Languages in Danger, as it is increasingly being replaced by English and other foreign languages in some areas [54]." Efforts are being made to preserve and promote the language, including the establishment of Amharic language schools and the publication of Amharic-language literature [55].

3.3. The Tigrinya language

Tigrinya is a language spoken by over 7 million people primarily in Eritrea and Ethiopia [53]. "It is a member of the Semitic branch of the Afro-Asiatic language family and is closely related to other Semitic languages such as Amharic and Arabic. Tigrinya has its own unique writing system known as the Ge'ez script, which has been used since ancient

times to write various Semitic languages in Ethiopia and Eritrea. According to Ethnologue, Tigrinya is the third most widely spoken language in Eritrea after Tigre and Afar, and it is also spoken by significant populations in Ethiopia, Sudan, and Israel. Tigrinya has a rich literary tradition, with works dating back to the 13th century, and it has been used as a language of instruction in Eritrea since the country gained independence in 1993 [53]"

Tigrinya has a complex grammar system, with a total of 29 consonants and 7 vowels. The language has many loanwords from other languages, including Arabic, English, and Italian, due to Eritrea's colonial past [1]. Tigrinya is an important language in Eritrea and Ethiopia, and it plays a crucial role in the social and cultural identity of the Tigrinya-speaking people. It is used in literature, music, and other forms of media.

3.4. Structure of Amharic and Tigrinya language

The structure of a language is the set of rules, principles, and patterns that govern how words are formed, how sentences are constructed, and how meaning is conveyed. It encompasses the sounds (phonetics), grammar (syntax), vocabulary, and writing system of the language, and it can vary widely between different languages. Understanding the structure of a language is essential for learning and using the language effectively, as well as for linguistic analysis, as linguists seek to describe and understand the structure of different languages and how they relate to one another. Here are some structures of the languages.

Based on the alphabet although Tigrigna and Amharic share many similar alphabets, "certain letters have different sounds in Tigrigna than in Amharic. Specifically, the letters (ሐ, ሀ) and (ኣ, ዐ) have distinct sounds in Tigrigna, whereas they have the same sounds in Amharic. Furthermore, the alphabet (ሠ, ኀ) is not utilized in Tigrigna, and ቐ is not utilized in Amharic."

Based on punctuation marks in machine translation, "it's clear that identifying punctuation within a corpus is important for mitigating data sparseness in both the parallel corpus and language model." This is particularly relevant for Amharic and Tigrigna, which share similar punctuation marks as seen in Table 3.1.

Table 3.1. Punctuation marks which are used in Amharic and Tigrinya languages.

Name of the punctuation in Amharic	Name of the punctuation in Tigrinya	Symbol of the punctuation mark	Description
Arat Netib	Arbaete Netibi	።	End of the sentence
Tiyake Milikit	Mlikit Hto	?	End of the question
Timihirite Anikiro	Anikro Timihiriti	!	End of emphatic declaration or command.
Dirib netela serez	Dirib Koma	፤	Sentence Connector
Netela serez	Koma zbl tshuf	፣	Uses to list things

Based on syntax both Amharic and Tigrigna languages share a common linguistic feature, which is the subject-object-verb (SOV) arrangement. “This type of arrangement is prevalent in many languages around the world, and it doesn't usually create any significant order problems. However, while both languages have a similar arrangement, there are some differences in how certain phrases are arranged, such as the compound word. These differences can affect the overall structure and meaning of a sentence and require careful attention when translating or interpreting between the two languages.” Therefore, it's important to have a strong understanding of the unique features of each language to effectively communicate and avoid any misunderstandings.

Based on accusative cases the accusative case serves to indicate the direct object of the verb in a sentence. “To identify the direct object, one can ask "what" or "whom" in relation to the verb. For instance, in the sentence "The dog ate our lunch," the accusative case would be "our lunch" by answering the question "what" to the verb "ate." In Tigrigna,

the accusative case is indicated by a preposition, while in Amharic, it is marked with a suffix [56].” An illustration of this difference can be seen by examining the same English sentence in both languages.

Amharic: ውሻው ምሳችንን በላው።

Tigrinya: እቲ ከልቢ ነቲ ምሳሕና በሊዕዎ።

Based on a definitive article functions differently in Tigrinya and Amharic. In “Tigrinya, it is a standalone word, while in Amharic, it is attached to the end of the noun. When it comes to the accusative case, the definite article is added as a suffix to the noun in Amharic. For instance, in Amharic, the word for "dog" is ውሻው, with the suffix indicating the definite article. In Tigrinya, on the other hand, the definite article is not attached to the noun. For example, the word for "dog" in Tigrinya is ከልቢ (pronounced as "ek'libi") and the definite article is expressed independently as እቲ.”

Amharic: ውሻው

Tigrinya: እቲ ከልቢ

Based on affixations to obtain the smallest meaningful word, it is necessary to remove the affixes. “Affixes are either letters or words that are attached to stems to modify their meanings and grammatical functions. In Tigrinya, for instance, affixes like "ብ", "ታት", and "ን" are used in words such as "ብኣገባብ", "ህዝቢታት", and "ምፍጣሩን", while in Amharic, affixes such as "ከ" and "ዎች" are used in words like "ከሀግ" and "ብርቱዎች". These affixes can appear in three positions relative to the stem word: as a prefix before the stem word, as a suffix at the end of the stem word, or as an infix inserted within the stem word. Although most sentences in Amharic and Tigrinya have similar affix patterns, there are some differences between them.” For example, some affixes in Amharic become prepositions in Tigrinya, and vice versa. For example:-

Amharic: ሃይለ ሙዝ እና ብርትኳን ከገበያ ገዛ።

Tigrinya: ሃይለ ሙዝን ኣራንሺን ካብ ዕዳጋ ግዚኡ።

The sentence above shows that in the given parallel example, “the words "ሙዝ" and "ኣራንሺ" are listed using the suffix "ን" in Tigrigna, while in Amharic they are connected by the preposition "እና" to list "ሙዝ" and "ብርትኳን". Moreover, the preposition "ካብ" in

Tigrigna is replaced with the prefix "ከ" in Amharic, with the stem word "ገበያ". Finally, to list two items using the conjunction "and", Tigrigna uses a suffix, while Amharic uses a preposition".

Amharic and Tigrinya share some similarities in terms of grammar, vocabulary, and writing systems. Both languages use the Ge'ez script, which is also used for other Ethiopian languages. They also have a similar system of verb conjugation, with several forms for each verb. However, there are also notable differences between the two languages. For example, Tigrinya has more vowel sounds than Amharic, and the two languages have different systems of noun declension. Amharic also has a larger literary tradition, with a rich history of poetry, prose, and religious texts.

Chapter Four

Proposed Architecture and Research Methodology

4.1. Introduction

The objective of this study is to develop an Amharic–Tigrinya neural machine translation with LSTM, LSTM with attention, BILSTM, BILSTM with attention, GRU, GRU with attention, BIGRU, and BIGRU with attention. And compare which method is best suitable for Amharic-Tigrinya neural machine translation. The Parallel Amharic-Tigrinya corpus was systematically gathered and meticulously prepared to serve as the foundation for the NMT model. This chapter provides a concise overview of the proposed architecture, delineates the methodology employed in the study, and elucidates the tools and techniques utilized in the development of the Amharic-Tigrinya NMT model.

4.2. Proposed Architecture of Amharic-Tigrinya NMT

NMT stands as the cutting-edge technology in the field, surpassing all conventional translation methods. It employs word vector representation, harnessing the power of multiple neural networks to predict the probability of word sequences. Unlike other translation approaches, it has the capability to translate entire sentences in one go, showcasing a level of efficiency that was previously unattainable [44]. Sequence-to-sequence (seq2seq), is a type of neural network architecture designed for tasks involving sequential data, such as natural language processing, machine translation, and speech recognition. The fundamental idea behind seq2seq is to enable the model to process input sequences and generate output sequences of variable lengths. It take a sequence of Amharic sentence and convert it to sequences of numbers, and do the same for translated Tigrinya sentence using encoder-decoder architecture. Not only has that but also, added an attention mechanism to the architecture which gives it the capability to use previous output as seen in Figure 4.1.

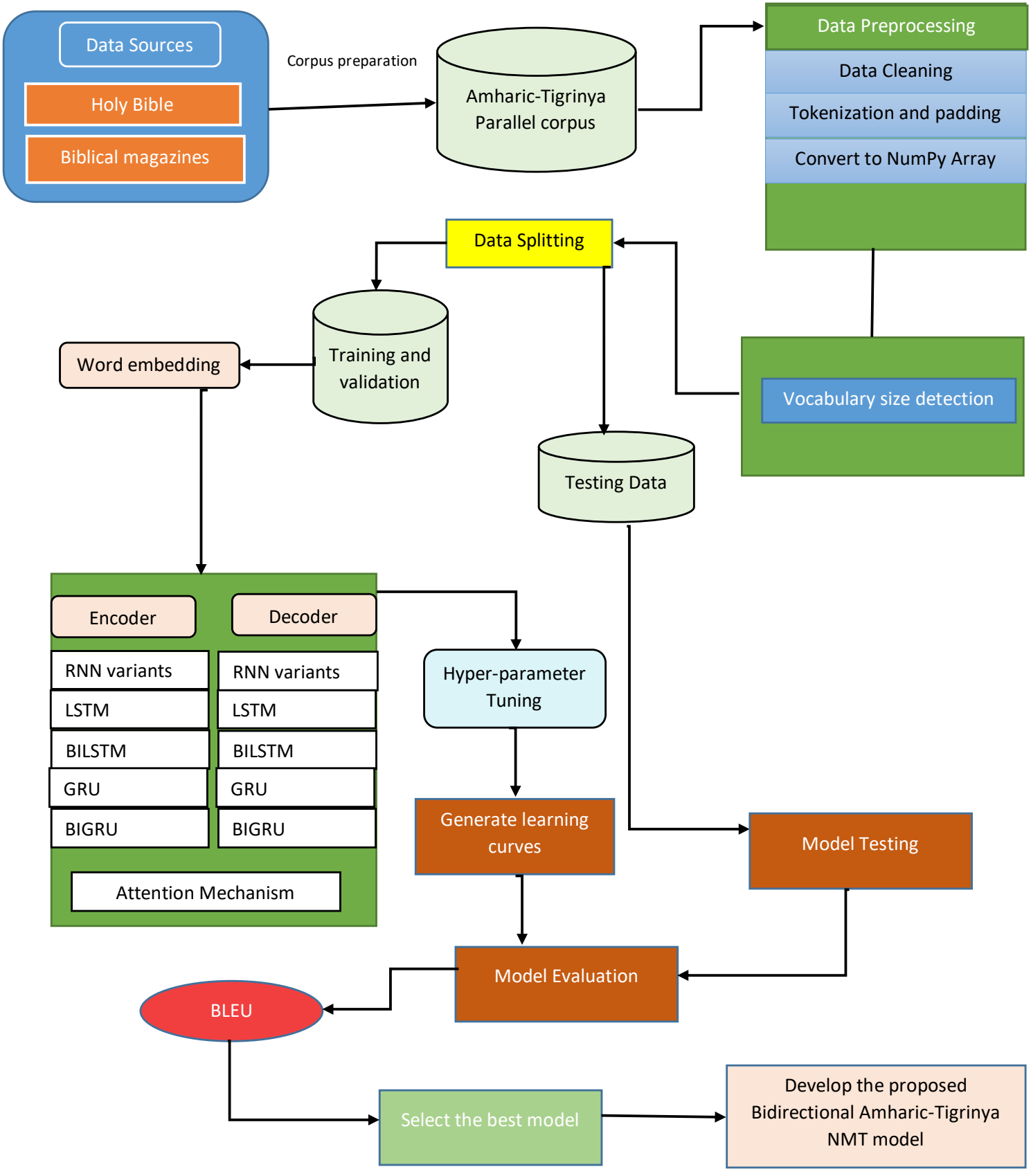

Figure 4.1: proposed architecture for Amharic-Tigrinya NMT

In Figure 4.1, the proposed architecture begins by gathering data from the Holy Bible and Biblical magazines, followed by the conversion of the collected data into an Amharic-Tigrinya parallel sentence corpus. The subsequent step involves pre-processing, commencing with the cleansing of the data to remove symbols and numerical representations from the parallel corpus. Additionally, the researcher has standardized Amharic and Tigrinya alphabets that share the same phonetic sound into a unified alphabet.

Subsequent to data cleaning, tokenization is performed on the parallel corpus, breaking down the text into tokens and converting variable-length sentences into sequences of discrete elements. This facilitates effective processing and learning by the model. Padding is then applied to ensure uniform sequence length, a critical aspect for efficient batch processing during training. The output is subsequently converted into a NumPy array to enhance integration with the TensorFlow framework, leveraging its optimized operations for numerical computations.

The research determines the vocabulary size by examining the length of the tokenized data for utilization in the embedding layer. Following this, the data is partitioned into an 80/20 ratio for training and testing purposes. Word embedding techniques are applied to capture semantic relationships and contextual information about words within a continuous vector space. The model undergoes training using various architectures to identify the optimal hyper-parameters.

In instances where the selected model necessitates integration with an attention mechanism, an 'Attention' layer is implemented. This mechanism, applied to the output of the last layer, enables the model to focus on different segments of the input sequence during the generation of each part of the sequence.

Upon completion of the training phase, the research proceeds to generate a learning curve depicting accuracy and loss to assess whether the trained model is under-fitting, over-fitting, or optimal. The model's performance is then evaluated using candidate and reference sentences to compute a BLEU score for each model. This process aids in the selection of the optimal model, which is subsequently developed for Bidirectional Amharic-Tigrinya NMT.

4.2.1. Corpus collection

To formulate a proficient translation model, the field of machine translation research is significantly dependent on parallel corpora comprising texts in both the source and target

languages. Consequently, the initial phase in undertaking any machine translation endeavor involves the acquisition of a parallel corpus for the source and target languages. In a related research that was conducted by Mekdes Melese[73] which is about developing LSTM and LSTM with attention models for English to Wolaytta and used 27,351 parallel corpus with 80/20 for training and testing. In the context of this research study, the corpus employed was meticulously gathered and curated from the religious domain. It consists of 34,350 parallel Amharic-Tigrinya sentences. Adding up to 68,700 sentences.

4.2.2. Corpus preprocessing

To preprocess the collected raw parallel corpus first the research first clean the data, then tokenizes and pad sequences the dataset, then converts to NumPy Arrays, and last but not least, vocabulary size calculation.

Data cleaning: to clean the data first the researchers have removed any symbols and numbers that would be bias. Then it moves on to cleaning the data by removing alphabet that have the same sound.

Tokenize and Pad Sequences: Tokenization involves the segmentation of a word or sequence into constituent elements such as keywords, words, symbols, and more. This process may utilize a single word or a group of words for tokenization. Furthermore, tokenization extends to the division of paragraphs or sentences into individual phrases or words. In essence, tokenization is the act of breaking down strings or words within a sentence, often achieved through the utilization of a space delimiter, and is applied across the entirety of the document [73].

The tokenization method used is based on the Keras Tokenizer class. The Tokenizer class is part of the Keras library and is used for converting text data into sequences of integers. It is initialized without any specific parameters. The “fit_on_texts” method is called with all the sentences (both Amharic and Tigrinya combined) in the dataset. This method updates the internal vocabulary of the tokenizer based on the unique words present in the input texts. It assigns a unique integer index to each unique word. The “word_index” attribute of the tokenizer is a dictionary that maps words to their integer indices. The total vocabulary size is calculated by adding 1 to the length of this dictionary. The additional 1 is reserved for a padding token. The “texts_to_sequences” method is used to convert each sentence in both the Amharic and Tigrinya

datasets into sequences of integers. Each word in the sentences is replaced by its corresponding integer index as per the vocabulary learned during the fitting step. The “pad_sequences” function from Keras is then used to ensure that all sequences have the same length. It pads sequences shorter than “max_sequence_length” with zeros at the end ('post' padding).

Overall, the tokenization process involves creating a vocabulary of unique words from the combined Amharic and Tigrinya sentences, assigning integer indices to each word, and converting the sentences into sequences of these integer indices. This is a common preprocessing step in natural language processing tasks before feeding text data into neural networks. The main purpose of this preprocessing is to convert raw text data into numerical sequences suitable for model input.

Convert to NumPy Array: this converts the padded sequences into NumPy arrays, which are compatible with TensorFlow for training for training neural network models. The NumPy ‘array’ function is used to convert these lists of sequences into NumPy arrays. Preparing the data in NumPy array format is essential for efficiently feeding it into neural network during the training process. NumPy arrays provide a convenient and efficient way to work with numerical data in the context of machine learning.

Vocabulary Size Calculation: this calculates the vocabulary size based on the unique word in the training data, to determine the size of the embedding layer in the neural network.

4.2.3. Data Splitting for Model Training and Testing

The ratio for splitting data into training and testing sets is a common consideration in machine learning, but there isn't a one-size-fits-all answer. The choice of the split ratio depends on various factors, including the size of the dataset, the complexity of the model, and the specific requirements of the task. In practice, a frequently used split ratio is 80-20 or 70-30, where 80% (or 70%) of the data is used for training, and the remaining 20% (or 30%) is used for testing. This ratio provides a good balance between having enough data to train a model effectively and having a sufficient amount of data to evaluate the model's performance. However, a related researches conducted by Mekdes Melese [73] which uses 80/20 ratio for training and testing, for comparing LSTM and LSTM with attention model for Wollyta-English, with a parallel corpus sentences of 27,351. This research cite Mekdes’s [73] splitting ratio for the Amharic and Tigrinya sentence pairs.

The Split of the data into training and testing sets part of the code is responsible for dividing the dataset into two subsets: one for training the machine translation model and another for validating its performance. This step is crucial for assessing how well the model generalizes to unseen data. Let's break down the relevant code:

Input Data: the corpus provided from the researcher

- amharic_sentence: The Amharic sentences after tokenization and padding.
- tigrinya_sentence: The corresponding Tigrinya sentences after tokenization and padding.

train_test_split Function: This function is from the scikit-learn library and is commonly used to split datasets into training and testing sets.

- **amharic_sentence and tigrinya_sentence:** The input datasets to be split.
- **test_size=0.2**: Specifies that 20% of the data should be reserved for validation, and the remaining 80% will be used for training.
- **random_state=42:** Provides a seed for the random number generator, ensuring reproducibility.

Output Variables: the generated output from our machine translation

- amharic_train and tigrinya_train: The training subsets for Amharic and Tigrinya sentences, respectively.
- amharic_test and tigrinya_test: The validation subsets for Amharic and Tigrinya sentences, respectively.

The purpose of splitting the data is to have separate sets for training and testing:

- Training Set: The model learns from this dataset during training. It adjusts its parameters to minimize the training loss.
- Test Set: After each training epoch, the model's performance is evaluated on this dataset. It helps monitor the model's ability to generalize to new, unseen data and avoid over-fitting.

This split is essential for assessing the model's performance on data it hasn't seen during training, providing insights into its generalization capabilities. The research has used 80/20 splitting technic for the trained and test data [73].

4.2.4. Model development

For this research eight models have used namely to compare which model is more optimal for Amharic-Tigrinya and vice versa machine translation. The models that have been chosen for this research are LSTM, LSTM with attention, BILSTM, BILSTM with attention, GRU, GRU with attention, BIGRU, and BIGRU with attention. These models are chosen by considering their pros and cons. Here are the details;

- **LSTM:** are effective at capturing long-range dependencies, making them suitable for translating languages with complex syntax and grammar. However, it's important to note that LSTMs may come with a higher computational cost compared to simpler architectures.
- **LSTM with attention:** Integrating attention mechanisms with LSTMs allows the model to focus on specific parts of the input sequence, improving its handling of long sentences. Although this adds complexity, the benefits in performance, especially for languages with non-trivial word order, can be substantial.
- **BILSTM:** capture information from both past and future context, potentially enhancing the understanding of context in both directions. However, the increased computational complexity compared to unidirectional LSTMs should be considered.
- **BILSTM with attention:** Combining bi-directionality with attention mechanisms provides a more comprehensive context understanding, particularly beneficial for languages with intricate structures.
- **GRU:** are similar to LSTMs but are computationally more efficient due to a simpler structure. They may not capture long-term dependencies as effectively as LSTMs.
- **GRU with attention:** Similar to LSTM with attention, integrating attention mechanisms with GRUs enhances the model's ability to focus on relevant parts of the input sequence.
- **BIGRU:** Similar to BILSTM, BIGRUs capture context information from both directions, offering advantages in understanding the broader context.

- **BIGRU with attention:** Combining bi-directionality with attention mechanisms in GRUs provides a comprehensive context understanding, making it a suitable choice for languages with complex linguistic structures.

For Amharic and Tigrinya machine translation, characterized by unique linguistic features, it is advisable to experiment with different architectures. The research aim to provide which model is more optimal considering their BLEU score.

4.2.5. Word representation

The word representation used is a dense word embedding. This embedding is learned as a part of the neural network during training. Here's a breakdown:

Embedding Layer: The code adds an Embedding layer to the neural network. This layer is responsible for learning a dense representation of words. Let's break down the parameters:

- **input_dim:** The size of the vocabulary, i.e., the total number of unique words in the dataset.
- **output_dim:** The dimensionality of the dense embedding. In this case, it's set to 100, meaning each word will be represented as a vector of 100 floating-point numbers.
- **input_length:** The length of input sequences. It should match the length of the padded sequences.

Training the Embedding: During training, the weights of the embedding layer are learned by the neural network. This means that the model learns to represent words in a way that is most useful for the specific machine translation task.

Word Indices to Word Vectors: After training, the embedding layer can convert integer indices (word indices obtained from tokenization) into dense vectors. For example, if the word "apple" has an index of 5, the embedding layer can map it to a 100-dimensional vector that represents the word "apple" in the learned embedding space.

Sequence Padding: Before being fed into the neural network, the sequences are padded to have a consistent length using the pad_sequences function. This ensures that all sequences have the same length, and the embedding layer can process them in batches.

In summary, the code uses a trainable word embedding layer to learn dense representations of words during the training process. This dense representation is crucial for capturing semantic relationships between words and improving the performance of the machine translation model.

4.2.6. Hyper-parameter tuning method

To fine tune the Hyper-parameter for optimal result the research first considered to use a grid search method but due to lack of memory and computational units the research was unable to perform a grid search. On the other hand, manual search for fine tuning the hyper-parameter is time consuming but it uses less memory and computational units. So manual search is selected for fine tuning the hyper-parameters. The manual tuning process involves a cycle of adjusting hyper-parameters, training the model, and evaluating its performance. It requires a good understanding of the data and the problem at hand. The goal is to find a set of hyper-parameters that results in a model with good generalization performance on unseen data.

To find the best hyper-parameter the research has conducted a manual search method. In order to do that first the research needs to have a base hyper-parameter for number of units, number of layer and number of epochs. To find the base hyper-parameter the research will use previous research hyper-parameter. The base hyper-parameter for this research is 128 for number of units, 4 for the number of layers, and 20 for number of epochs. After conducting the appropriate research the research will use the base hyper-parameter as a cornerstone to find the best hyper-parameter. To find the best hyper-parameter the research uses the base hyper-parameter and first replace the number of units while keeping the number of layers and number of epoch the same as the base hyper-parameter to determine at which number of unit is the bleu score is at its highest. After finding the best hyper-parameter for number of units the research will continue to determine the best hyper-parameter for the number of layers by making the number of unit at its optimal state, and keeping the number of epochs the same as the base hyper-parameter while we change the number of layer to find the best hyper-parameter for it. And finally we use the best hyper-parameter for number of units and number of layer and change the value of epochs to find the best hyper-parameter for epochs. And final, we have the best hyper-parameters for numbers of unit, numbers of layers, and number of epochs for the research model.

4.2.7. Training the models

Not only that research uses various numbers of epochs, number of layers, and number of units but also the research also uses LSTM, BILSTM, GRU, and BIGRU models with and without attention mechanism.

The attention mechanism [58] stands out as a key breakthrough in machine translation, particularly benefiting neural machine translation systems. In the conventional Encoder-decoder RNN setup detailed in the preceding section, the encoder transforms entire sequences of source language data into a singular real-valued vector, known as the context vector. This vector is then transmitted to the decoder for generating the output sequence. However, the limitation arises from the fact that the decoder relies solely on the encoder's output context vector. This approach proves inefficient, failing to adequately capture the nuances of intricate sentences and a vast vocabulary.

To address this inefficiency, the attention mechanism has emerged as a significant solution in contemporary neural network training. In an Encoder-decoder architecture incorporating attention mechanisms, the prediction of a target word is based on context vectors linked to the position of the source language and previously generated target words. This is a departure from the Encoder-Decoder architecture without attention, where the source representation is used only once to initialize the decoder's hidden state. This evolution allows for a more nuanced and effective representation of the input sequence, accommodating the complexity of diverse sentences and expansive vocabulary.

The attention layer is likely implemented to compute attention scores between the decoder's current hidden state and the entire sequence of encoder outputs. The "[model.layers[-1].output, model.layers[-1].output]" argument suggests that the attention mechanism is applied to the same set of inputs, possibly indicating a self-attention mechanism.

Then there will be two lines of codes to concatenate the attention and the output value. The first line code performs a dot product operation between the attention scores and the output of the last layer. The "axes = [2, 2]" argument specifies the axes along which the dot product is computed. This operation computes a weighted sum of the encoder outputs based on the attention scores, producing a context vector. The second line concatenates the context vector with the output of the last layer along the last axis (axis=-1). This concatenated vector is then likely used as the input to the decoder in a subsequent step.

In summary, these lines of code implement an attention mechanism, compute a context vector based on attention scores, and construct a decoder input by concatenating the context vector with the output of the last layer.

4.2.8. Evaluation and Testing Method

Utilizing the BLEU score for evaluating neural network machine translation models is advantageous for several reasons [51], [48], and [73]. BLEU provides a quantitative and objective measure of the quality of generated translations by comparing them against one or more reference translations. This metric is particularly well-suited for assessing the performance of neural network models in machine translation tasks, as it considers not only individual words but also evaluates the precision of entire phrases and sentences. The use of BLEU facilitates a holistic evaluation, capturing the model's ability to produce translations that align closely with human-generated references. Additionally, BLEU is widely accepted and adopted in the machine translation community, offering a standardized benchmark for model comparison and enabling researchers and practitioners to reliably assess and improve the effectiveness of neural network-based translation systems.

The model is used to predict Tigrinya sentences based on the Amharic test sentences. Then using “translate_sentence” function we have created we will convert the predicted sequence into words. “reference_sentences” The true Tigrinya sentences from the validation set. Then the BLEU score is calculated using the corpus_bleu function from the NLTK library. BLEU is a metric that measures the similarity between the predicted and reference sentences. It considers precision and brevity in its calculation. The same process is also taken for Tigrinya to Amharic translation too.

Chapter Five

Experimentation and Discussion

5.1. Introduction

Within this chapter the research discusses about what hardware and software required to perform this research. Then it continues on how the parallel sentences corpus was collected and prepared, then investigating to pinpoint the optimal hyper-parameters for the research is discussed. The overarching goal is to meticulously evaluate the efficacy of diverse RNN models within the domain of Amharic-Tigrinya and vice versa NMT. The spectrum of models scrutinized encompasses LSTM, LSTM with attention, BILSTM, BILSM with attention, GRU, GRU with attention, BIGRU and BIGRU with attention. After training and testing with above model for Amharic-Tigrinya and vice versa NMT each result is compared with their accuracy and BLEU score for both Amharic-Tigrinya and Tigrinya-Amharic NMT. The pivotal criteria employed to discern the model that excels in performance hinge on their individual BLEU scores [73], [38], providing a comprehensive and nuanced assessment of the translation quality achieved by each model in the Amharic-Tigrinya and vice versa context. And last but not least, the research tries to analyze the learning curves of accuracy and loss for top two models.

5.2. Development environment

In the course of the development of this research, a laptop computer of the 11th generation equipped with a 256 GB SSD and 8 GB RAM served as the hardware foundation. On the software front, the research utilized a Windows 10 Pro operating system, featuring the Chrome browser for web browsing. Additionally, the research employed the Google Colab Pro Plus platform for a duration of four months, requiring a subscription fee of $50 per month. The laptop has been used for collecting the corpus, reviewing other related researches, and last but not least for writing the code for this research. Google Colab Pro Plus platform is used to write the code, using Google Colab Pro Plus the research codes has been implemented and tested using a parallel sentence corpus file that is saved in Google Drive. The research chose Google Colab Pro Plus platform because it provide Graphical Processing Unit (GPU) which have more processing power than

Central Processing Unit (CPU). GPU is composed of hundreds of cores that can handle thousands of threads simultaneously.

5.3. Parallel corpus collection and preparation

The data is collected from Bible and Biblical magazines. Together the researcher have collected 34,350 parallel sentence for Amharic and Tigrinya in two different text file. These files are then assigned to a variables to create an Excel file with two column. The first column having Amharic sentence per cell and on the second column having its respective Tigrinya sentences. After combing the two text file into one Excel file the researcher then conducted data cleaning by removing symbols and letters that have the same sound. After that the data will be split into two for training and testing with 80/20 ratio. Giving 27,480 parallel sentences for training and 6,870 sentences for testing as seen in the Table 5.1.

Table 5.1: an overview of the corpus collected

Total sentences	Amharic sentences	Tigrinya sentences	Training pair sentences	Testing pair sentences
68,700	34,350	34,350	27,480	6,870

For the preprocess task, First the Excel parallel sentence data is loaded from Google Drive, then the loaded file is read into Pandas data frame to load the raw data for further preprocessing. Then the data is cleaned removing any symbols or numbers, after that the data is further cleaned by replacing Amharic and Tigrinya alphabet that have the same sound by a common alphabet. Then tokenizing and padding is done to ensure consistent input length which aims to convert raw text data into numerical sequences suitable for model input. After that the padded data is converted into NumPy array for compatibility with TensorFlow to prepare the data for training the neural network. And last but not least, the vocabulary size is calculated based on the unique words in the training data to determine the size of embedding layer in the neural network.

5.4. Hyper-parameter optimization

It's important to note that there is no universally "average" set of hyper-parameters for LSTM and GRU networks, as the choice of hyper-parameters depends on the specific task, dataset, and other factors. However, 4 layers, 128 units, 20 epochs seems like a common starting point or a rule

of thumb that might be used in certain scenarios. Here are some reasons why such hyper-parameters might be considered reasonable or average:

- **Number of layers:** a deep network can capture complex hierarchical patterns in data. However, increasing the layers also increase the risk of over-fitting and requires more computation. Four layers strike a balance between complexity and computational efficiency.
- **Number of units:** 128 unit per layer is a moderate size that provides a sufficient capacity for capturing patterns in data without being overly large.
- **Number of epochs:** represents how many times the learning algorithm will work through the entire training dataset. 20 epochs is a common starting point, and it might be sufficient for simpler tasks or smaller datasets.

5.4.1. Hyper-parameter tuning for LSTM model

To determine the most effective hyper-parameters, this research employs the average values from prior studies as the base configuration. The base hyper-parameters involve a model with 4 layers, 128 units, and 20 epochs. Subsequently, each parameter in the base configuration is systematically substituted, and the resulting BLEU score is analyzed.

A. Number of hidden units

Table 5.2: Hyper-parameter tuning for number of units for LSTM model

Unit	Number of layers	Epoch	Loss	Accuracy	BLEU score
64	4	20	1.7041	0.7466	1.2691
128	4	20	1.3534	0.7744	1.5589
256	4	20	0.9160	0.8283	2.2326
512	4	20	1.1236	0.7981	1.9373

As seen in Table 5.2 the initial focus is on finding the best unit hyper-parameter by varying the unit while maintaining other base hyper-parameters. The analysis reveals that the model achieves its lowest loss and highest accuracy and BLEU score when the unit is set to 256. Consequently, 256 units are identified as the optimal hyper-parameter for LSTM model.

B. Number of hidden layers

Table 5.3: Hyper-parameter tuning for number of Layers for LSTM model

unit	Number of layers	Epoch	Loss	Accuracy	BLEU score
256	3	20	1.0260	0.8138	2.1022
256	4	20	0.9160	0.8283	2.2326
256	5	20	1.5042	0.7571	2.2312

In the subsequent analysis as seen in Table 5.3, the number of layers is altered while keeping epochs and units at their best and base hyper-parameter values, respectively. The observations indicate that the model exhibits the lowest loss and highest accuracy and BLEU score when the number of layers is set to 4. Thus, 4 layers are determined to be the optimal hyper-parameter for the number of layers in the LSTM model.

C. Number of Epochs

Table 5.4: Hyper-parameter tuning for number of epochs for LSTM model

Unit	Number of layers	Epoch	Loss	Accuracy	BLEU score
256	4	5	2.2534	0.7310	1.9256
256	4	10	1.8625	0.7416	2.4315
256	4	15	1.4450	0.7621	1.9853
256	4	20	0.9160	0.8283	2.2326
256	4	30	0.9432	0.8231	1.9355

As seen in Table 5.4, the research further explores the impact of different epoch values on the LSTM model's performance while maintaining the best unit and number of layers. Although the lowest loss and highest accuracy are observed at 20 epochs, the BLEU score peaks at 10 epochs. Given the emphasis on the BLEU score in the comparison, 10 epochs are selected as the best hyper-parameter for the LSTM model.

In summary, the research concludes that the optimal hyper-parameters for the LSTM model are 256 units, 4 layers, and 10 epochs with a BLEU score of 2.4315, loss of 0.1.8625, and accuracy of 0.7416.

5.4.2. Hyper-parameter tuning for GRU model

In the pursuit of determining the best hyper-parameters for the GRU, this research leverages previously established average hyper-parameters as the base configuration. The foundational parameters involve a 4-layered architecture with 128 units and 20 epochs. The iterative process of identifying the optimal hyper-parameters involves substituting values for each parameter from the base configuration and assessing the resulting BLEU score. This methodology mirrors the approach employed in determining optimal hyper-parameters LSTM models.

A. Number of hidden units

Table 5.5: Hyper-parameter tuning for number of units for GRU model

Units	Number of layers	Epochs	Loss	Accuracy	BLEU score
64	4	20	1.3883	0.7751	2.1533
128	4	20	1.0023	0.8210	2.3124
256	4	20	0.8677	0.8334	2.6050
512	4	20	1.2369	0.7852	2.1742

As shown in the above Table 5.5, presents a comprehensive exploration of various unit configurations, where unit values, number of layers, and epochs are systematically altered. Notably, the analysis reveals that the GRU model attains its lowest loss, along with peak accuracy and BLEU score when employing 256 units. Consequently, 256 units are identified as the optimal hyper-parameter for the GRU model.

B. Number of hidden layers

Table 5.6: Hyper-parameter tuning for number of layers for GRU model

Units	Number of layers	Epochs	Loss	Accuracy	BLEU score
256	3	20	0.7485	0.8509	2.0886
256	4	20	0.8677	0.8334	2.6050
256	5	20	0.9965	0.8136	2.3102

Continuing the investigation, Table 5.6 explores different configurations for the number of layers while keeping units and epochs constant. Despite optimal states for loss and accuracy at three layers, the BLEU score is maximized with four layers. As a result, the research designates four layers as the optimal hyper-parameter for the number of layers in the GRU model.

C. Number of epochs

Table 5.7: Hyper-parameter tuning for number of epochs for GRU model

Units	Number of layers	Epochs	Loss	Accuracy	BLEU score
256	4	10	1.7105	0.7502	1.9586
256	4	15	1.5578	0.7519	1.9151
256	4	20	0.8677	0.8334	2.6050
256	4	30	0.6876	0.8608	2.3397

The subsequent analysis focuses on identifying the ideal number of epochs for the GRU model, as detailed in Table 5.7. While the training process achieves its lowest loss and highest accuracy at 30 epochs, the ultimate selection of the optimal epoch count is guided by the highest BLEU score, observed at 20 epochs. Consequently, the research concludes that the best hyper-parameters for the GRU model are 256 units, 4 layers, and 20 epochs with a BLEU score of 2.6050.

5.5. Models building for Amharic-Tigrinya

In the subsequent phase of the research, the focus shifts to evaluating the outcomes of employing the best hyper-parameters in LSTM models, including LSTM, LSTM with attention, BILSTM, and BILSTM with attention. The findings are succinctly summarized in Table 5.8.

Table 5.8: models building for Amharic-Tigrinya variety of LSTM models

Model	Units	Number of layers	Epochs	Loss	Accuracy	BLEU score
LSTM	256	4	10	1.8625	0.7416	2.4315
LSTM with attention	256	4	10	1.4917	0.7624	2.5941
BILSTM	256	4	10	1.5417	0.7534	3.2015
BILSTM with attention	256	4	10	0.7616	0.8403	3.3306

Table 5.8 provides a comparative analysis of diverse LSTM architectures with identical configurations of 256 units, 4 layers, and 10 epochs. Notably, BILSTM with attention stands out as the most effective model within this set, showcasing the lowest loss and the highest accuracy and BLEU score among all models. The inclusion of attention mechanisms in the architecture enhances the model's performance, making BILSTM with attention the optimal choice for the specified hyper-parameters.

The subsequent phase of this research involves scrutinizing the outcomes derived from implementing the optimal hyper-parameters in various GRU models, including GRU, GRU with attention, BIGRU, and BIGRU with attention. The tabulated data in Table 5.9 succinctly encapsulates the performance metrics of these models. Notably, BILSTM with attention outshines

Table 5.9: models building for Amharic-Tigrinya variety of GRU models

Model	Unit	Number of layers	Epochs	Loss	Accuracy	BLEU score
GRU	256	4	20	0.8677	0.8334	2.6050
GRU with attention	256	4	20	0.2949	0.9312	2.9539
BIGRU	256	4	20	0.0790	0.9782	3.2895
BIGRU with attention	256	4	20	0.0775	0.9786	3.3415

Table 5.9 provides a comprehensive overview of the key performance indicators for each GRU model under consideration. The comparison underscores the superior performance of BIGRU with attention, exhibiting the lowest loss and highest accuracy and BLEU score among the models. The tabular presentation facilitates a clear and concise examination of the diverse GRU architectures, aiding in the identification of the most effective model for the specific context of the study.

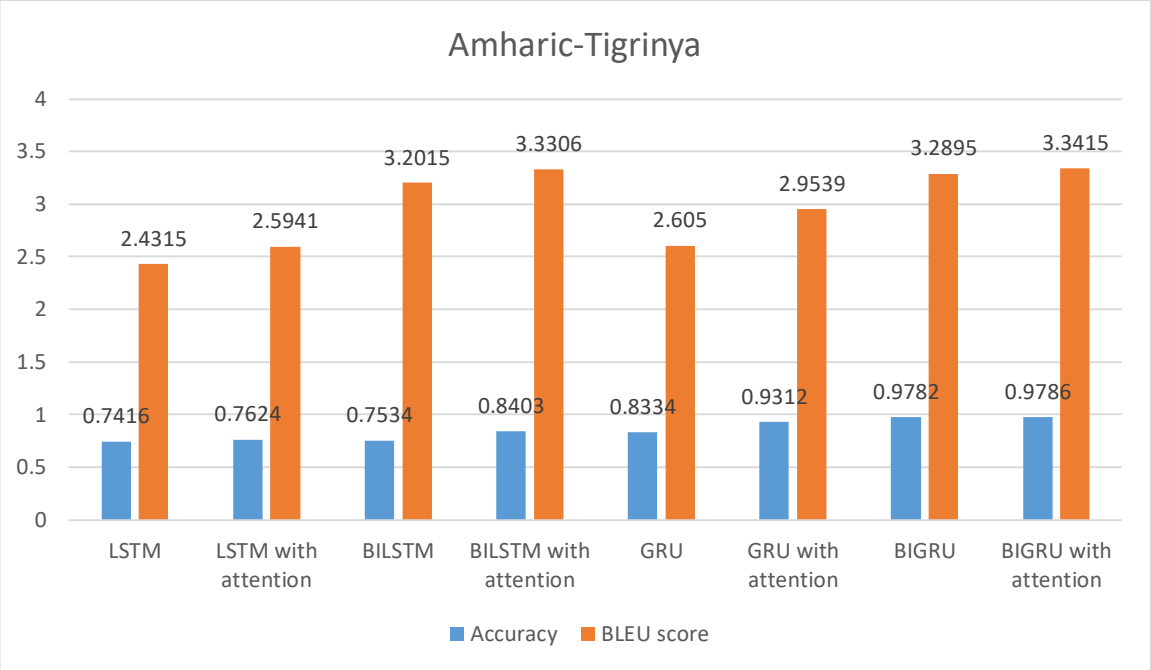

Figure 5.1: Amharic-Tigrinya accuracy and BLEU score results

In Figure 5.1, the accuracy and BLEU score peak at 0.9786 and 3.3415, respectively, when utilizing the BIGRU with attention model. Conversely, the lowest values are observed at 0.7416 for accuracy and 2.4315 for BLEU score. Arranged hierarchically based on their accuracy and BLEU score, from lowest to highest, are the following models: LSTM, LSTM with attention, BILSTM, BILSTM with attention, GRU, GRU with attention, BIGRU, and BIGRU with attention.

5.6. Models building for Tigrinya-Amharic

The research aims to assess the optimal model among various architectures, including LSTM, LSTM with attention, BILSTM, BILSTM with attention, GRU, GRU with attention, BIGRU, and BIGRU with attention, specifically tailored for the Tigrinya-Amharic language pair. This evaluation is conducted using established hyper-parameters derived from previous research, encompassing 256 units, 4 layers, and 10 epochs for the LSTM model, and 256 units, 4 layers, and 20 epochs for the GRU model. Through a comparative analysis, the study seeks to gauge the performance of each model under these predefined hyper-parameters, providing valuable insights into their effectiveness in the realm of Tigrinya-Amharic machine translation.

Table 5.10: models building for Tigrinya- Amharic variety of LSTM models

Model	Units	Number of layers	Epochs	Loss	Accuracy	BLEU score
LSTM	256	4	10	1.1280	0.8066	1.7812
LSTM with attention	256	4	10	1.0314	0.8186	1.8088
BILSTM	256	4	10	0.8118	0.8426	2.0141
BILSTM with attention	256	4	10	0.6271	0.8658	2.1506

As illustrated in Table 5.10, the initial phase of the study involves assessing the loss, accuracy, and BLEU score for various LSTM configurations, specifically those with 256 units, 4 layers, and 10 epochs. Subsequently, the research discerns that among the considered LSTM variants, including LSTM, LSTM with attention, and BILSTM, BILSTM with attention

Table 5.11: models building for Tigrinya- Amharic variety of GRU models

Model	Unit	Number of layers	Epochs	Loss	Accuracy	BLEU score
GRU	256	4	20	0.9862	0.8197	2.3599
GRU with attention	256	4	20	0.4516	0.9063	2.3812
BIGRU	256	4	20	0.4191	0.9105	2.5079
BIGRU with attention	256	4	20	0.0805	0.9785	2.5805

Table 5.11 presents an extensive overview of the outcomes derived from various GRU configurations, specifically utilizing a unit size of 256, 4 layers, and 20 epochs. Upon meticulous examination of each model's performance, a distinct pattern emerges, underscoring BIGRU with attention as the optimal model. This assertion holds not only when compared within the GRU variants, including GRU, GRU with attention, and BIGRU but also extends to outperforming counterparts in the LSTM, LSTM with attention, BILSTM, and BILSTM with attention categories, as evidenced by its superior BLEU score.

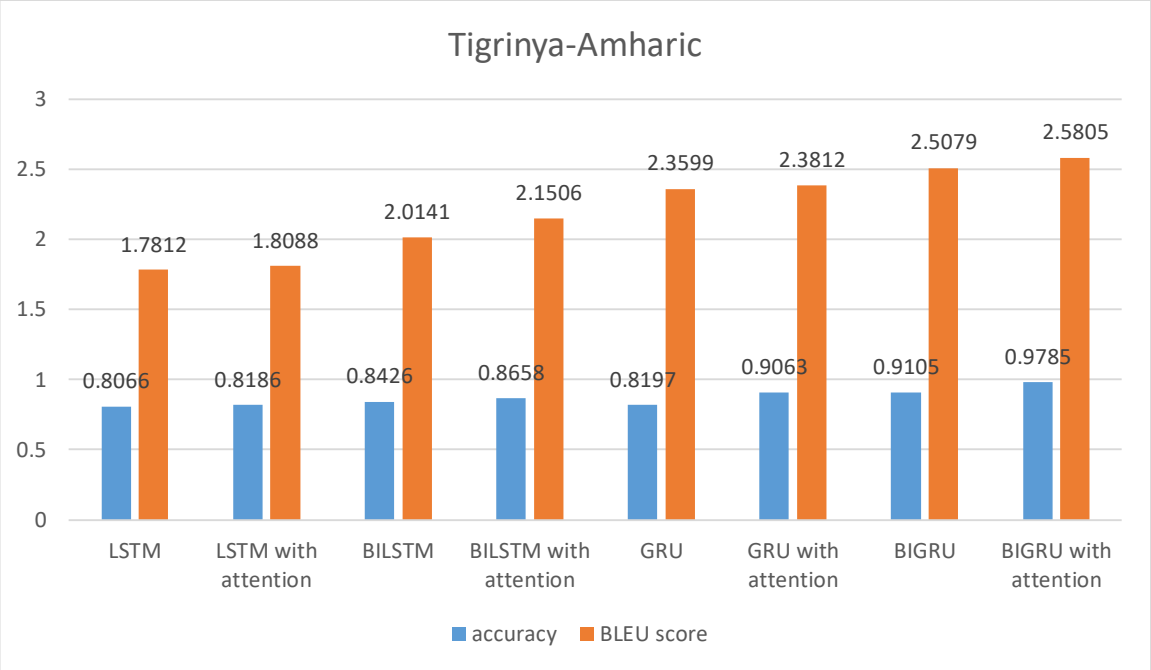

Figure 5.2: Tigrinya-Amharic accuracy and BLEU score results

In Figure 5.2, the BIGRU with attention model stands out with the highest accuracy and BLEU score, registering 0.9785 and 2.5808, respectively. Conversely, the LSTM model records the lowest accuracy and BLEU score at 0.8066 and 1.7812, respectively. In hierarchical order, based on their accuracy and BLEU score, from lowest to highest, are the following models: LSTM, LSTM with attention, BILSTM, BILSTM with attention, GRU, GRU with attention, BIGRU, and BIGRU with attention, with BIGRU with attention achieving the highest scores.

5.7. Learning curves

In the course of investigation of the top two models learning curves of accuracy and loss as seen in Figure 5.3 and 5.4 for BIGRU with attention and BIGRU respectively. As seen in Figure 5.3 and 5.4 the learning curve of accuracies are over-fitting because training accuracy increases while test accuracy remains relatively stable often indicates over-fitting. And the learning curves of loss are over-fitting too because a decreasing training loss accompanied by an increasing test loss in a learning curve suggests over-fitting. Over-fitting occurs when a model learns the training data too well, capturing noise or specific patterns that do not generalize to new, unseen data. Factors contributing to this include the model's complexity, insufficient regularization, data mismatch between training and test sets, and potential data leakage. Strategies to address this issue

involve implementing regularization techniques, monitoring for signs of over-fitting, and ensuring that the training and test datasets are representative of each other.

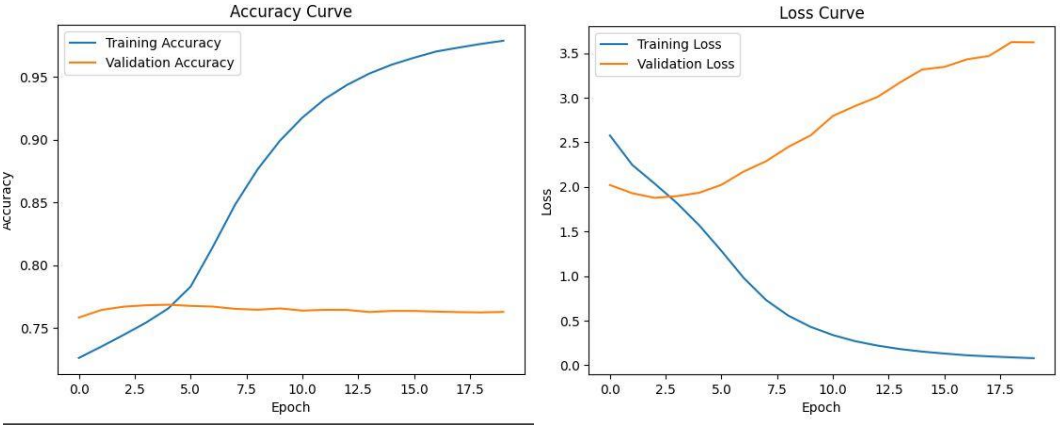

Figure 5.3: accuracy and loss learning curve for BIGRU with attention for Amharic-Tigrinya

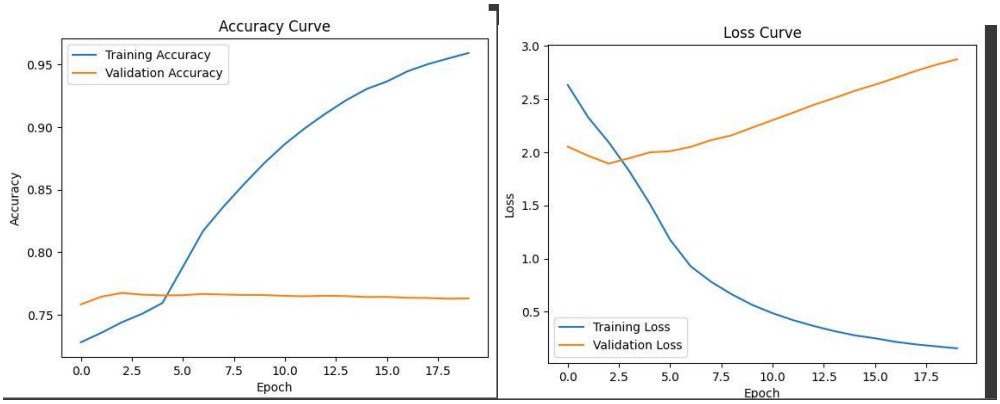

Figure 5.4: accuracy and loss learning curve for BIGRU for Amharic-Tigrinya

Chapter Six

Conclusion and Future Works

6.1. Introduction

In this concluding chapter, the researcher aims to summarize the findings of the research and propose potential avenues for future work The initial step involves determining the optimal model among various architectures, including LSTM, LSTM with attention, BILSTM, BILSTM with attention, GRU, GRU with attention, BIGRU, and BIGRU with attention.

6.2. Conclusion

Machine translation has undergone significant evolution from rule-based approaches to neural network models. In the earlier stages, rule-based systems were prevalent, relying on predefined linguistic rules and handcrafted dictionaries to translate text between languages. These systems, while structured, faced limitations in handling the nuances and complexities of natural language. With the advent of SMT, the focus shifted to probabilistic models that learned translation patterns from large bilingual corpora. However, SMT had challenges capturing long-range dependencies and context. In recent years, the emergence of neural network-based models, RNNs and their specialized variants LSTM and GRU, has revolutionized machine translation. Neural networks can effectively capture intricate linguistic structures and context, leading to significant improvements in translation quality. The transition from rule-based systems to neural networks reflects a paradigm shift towards data-driven learning, enabling machines to better comprehend and generate human-like translations across diverse language pairs.

As machine translation has been moving forward there is only little research conducted regarding Ethiopian language pairs. This research objective is to develop bidirectional Amharic-Tigrinya RNN machine translation and determine which RNN model is optimal from LSTM, LSTM with attention, BILSTM, BILSTM with attention, GRU, GRU with attention, BIGRU, and BIGRU with attention considering the BLEU score they have scored. The methods used are discussed below.

The data collection phase focused on extracting 34,350 parallel sentences for Amharic and Tigrinya from the Bible and Biblical magazines. Subsequently, these sentences were organized into an Excel file, with data-cleaning processes applied to remove symbols and letters with similar sounds. The resulting dataset was split into training and testing sets at an 80/20 ratio, yielding 27,480 sentences for training and 6,870 sentences for testing.

Hyper-parameter optimization, specifically for the LSTM model, involved a meticulous exploration of units, number of layers, and epochs. The research determined that the optimal hyper-parameters for LSTM are 256 units, 4 layers, and 10 epochs. Subsequently, the research extended this optimization process to the GRU model, employing established average hyper-parameters as a base configuration. The resulting exploration, detail indicated that the optimal hyper-parameters for GRU are 256 units, 4 layers, and 20 epochs.

The research then delved into model building for Amharic-Tigrinya and Tigrinya-Amharic language pairs, comparing various LSTM and GRU architectures. Notably, while BIGRU with attention comes up on top of the other models with the highest accuracy and BLEU score, LSTM is determined to be the least optimal model according to the recorded least accuracy and BLEU score result.

Last but not least, the research try to analyze the learning curves for the top model in this research case BIGRU with attention. The learning curve analysis further enriches the research narrative, revealing instances of over-fitting and emphasizing the need for vigilant model evaluation. The graphical representation of accuracy and loss curves enhances the comprehension of performance trends over successive epochs, providing a visual narrative of the models' training and testing dynamics. Resulting BIGRU with attention on top of the other models scoring a BLEU score of 3.3415, on the second place BILSTM with attention with a BLEU score of 3.3306, and on the third place there is BIGRU with a BLEU score of 3.2895.

In summary, this research has systematically investigated the experimental setup, hyper-parameter tuning, and model construction processes, providing a comprehensive overview of Amharic-Tigrinya NMT. Each chapter contributes to a nuanced understanding of the specific challenges posed by this linguistic context. The evaluation of various RNN models underscores the significance of attention mechanisms in improving BLEU scores, offering crucial contributions to the domain of machine translation. Notably, the BIGRU model with attention emerges as the

top performer, achieving the highest BLEU score of 3.3415, thereby substantiating its efficacy in enhancing translation accuracy for Amharic-Tigrinya language pairs.

6.3. Contribution

This research significantly contributes to the field of Amharic-Tigrinya and vice versa neural network machine translation by systematically evaluating and determining the optimal model among a diverse set of architectures. The exploration encompasses well-established models, including LSTM, LSTM with attention, BILSTM, BILSTM with attention, GRU, GRU with attention, BIGRU, and BIGRU with attention. The research methodology, spanning hardware and software configurations, parallel sentence corpus collection, hyper-parameter optimization, model building, and learning curve analysis, offers valuable insights and advancements in the following key areas:

- **Model Comparison and Selection:** By subjecting each model to a rigorous evaluation based on BLEU scores, the research provides a nuanced understanding of their relative strengths and weaknesses. This comparative analysis aids researchers, practitioners, and developers in selecting the most effective model for Amharic-Tigrinya and vice versa machine translation tasks.
- **Hyper-parameter Tuning Guidelines:** The study systematically explores hyper-parameter tuning for both LSTM and GRU models, unraveling optimal configurations. The research contributes to establishing practical guidelines for selecting the number of layers, hidden units, and epochs, offering valuable insights for researchers and practitioners engaged in similar tasks.
- **Learning Curve Analysis and Over-fitting Insights:** The investigation into learning curves provides a deeper understanding of model training dynamics. The identification of over-fitting instances and the graphical representation of accuracy and loss curves guide future researchers in refining their models and implementing effective regularization techniques to enhance generalization.
- **Performance Evaluation in Amharic-Tigrinya and Tigrinya-Amharic Contexts:** The research extends its impact by evaluating model performance in both Amharic-Tigrinya and Tigrinya-Amharic language pairs. This dual

assessment provides a comprehensive understanding of model adaptability and effectiveness across diverse translation tasks.

- **Practical Implementation Guidelines:** The detailed account of the development environment, software tools, and methodologies used in the research provides practical implementation guidelines. This information is valuable for researchers and practitioners seeking to replicate or build upon the study's findings.

Overall, the research significantly advances the understanding of neural network machine translation in the Amharic-Tigrinya linguistic context. The insights gained from model comparisons, hyper-parameter tuning, and performance evaluations contribute to the ongoing development and improvement of machine translation systems for these languages, addressing the unique challenges posed by the Amharic and Tigrinya scripts and linguistic structures.

6.4. Future works

Below are some additional avenues for future research or suggested actions derived from the discoveries made in this study.

- Recommended to add and expand corpus from various domains including law, business, news, entertainment, health and so on to make it more optimal
- Recommended to train the NMT models with a powerful processing machine by adding more RAM and CPU units so that they perform more efficiently
- Recommended to develop and implement the Amharic-Tigrinya and vice versa NMT system in web based so that it becomes more operational and useful for users
- Recommended to enhance the training of the NMT model by utilizing robust processing machines and augmenting the dataset. This is crucial for improving translation quality, as neural networks thrive on extensive data and efficient processing capabilities to operate effectively and yield precise translations.
- And last but not least, it is recommended that future studies should be conducted on speech-to-speech, speech-to-text, and text-to-speech translations.

References

[1] Britannica, the world standard in knowledge https://www.britannica.com/

[2] Hayward, Katrina and Richard J. Hayward. ”Amharic. In Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet.” Cambridge: the University Press, 1999

[3] "Tigrinya language". Encyclopedia Britannica.

[4] The Bible in Tigrinya, United Bible Society, 1997

[5] M. Gebre Michael, “Federalism and conflict management in Ethiopia: a case study of Benishangul-Gumuz Regional State”, University of Bradford, United Kingdom 2011

[6] Wikipedia, the free encyclopedia https://en.wikipedia.org

[7] Liu Qun, Zhang Xiaojun, “Routledge Encyclopedia of Translation Technology: Machine translation”, 2014

[8] Liu Yang, Zhang Min, “Routledge Encyclopedia of Translation Technology: Statistical machine translation”, 2014

[9] Sennrich, R., Haddow, B., & Birch, A., “Improving neural machine translation models with monolingual data”, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 86-96, 2016.

[10] Graves, A., “Supervised sequence labelling with recurrent neural networks”. Springer, 2012.

[11] Yohannes Gebreegzabher and Solomon Teferra Abate, “Tigrinya morphological segmentation with bidirectional long short-term memory neural networks and its effect on English-Tigrinya machine translation”, Nagaoka University of Technology, 2019

[12] James Chen, “Neural Network”, Investopedia Express Podcast, December 22, 2020

[13] Ms. Sonali. B. Maind, and Ms. Priyanka Wankar “research paper on basic of artificial neural network” Departments of Information technology and computer science, university of Wardha, India, 2014

[14] Will Koehrsen," how deep learning can represent war and peace as vectors", www.towardsdatascience.com, October 02, 2018

[15] Jurafsky, Daniel, H. James and Martin "speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition" Upper Saddle River, New Jersey U.S.A, 2000

[16] Farhad Malik, "Neural networks layers: understanding how neural network layers work", May, 2019

[17] Mark A. Kramer, "Nonlinear Principal Component Analysis Using Auto-associative Neural Networks", department of chemical engineering, Massachusetts institute of technology, Cambridge, 233-243, 1999

[18] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and Helmholtz free energy", Department of computer science and department of neuroscience laboratory, university of Toronto, Canada, November 1993

[19] Jason Brownlee, "Encoder-Decoder Recurrent Neural Network Models for Neural Machine Translation", January 2018

[20] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. & Bengio, Y., "Generative adversarial nets. Advances in Neural Information Processing Systems", 2672-2680, 2014.

[21] Michael Woldeyohannis and Million Meshesha, "Experimenting Statistical Machine Translation for Ethiopic Semitic Languages: The Case of Amharic-Tigrigna", Addis Ababa University, Ethiopia 2017

[22] Kjell Erik Rudestam, Rae R. Newton, "surviving your dissertation: a comprehensive guide to content and process", fielding graduate university, USA 1992

[23] Wikipedia, the free encyclopedia https://en.wikipedia.org

[24] Ben Luckovich, "language modeling – machine learning", March 2020

[25] Elizabeth D. Liddy, "Natural Language Processing", Syracuse University, 2001

[26] Muhammad Imran, “Why Natural Language Processing is Difficult – Considering how challenging human language is and how differently each individual uses it.”, December 10, 2021

[27] Fomicheva, M., Bojar, O., Federmann, C., & Fishel, M., “Machine Translation Evaluation beyond BLEU and METEOR: An Overview. Computational Linguistics”, 47(1), 1-49, 2021

[28] LeCun, Y., Bengio, Y., & Hinton, G., “Deep learning. Nature”, 521(7553), 436-444., 1998.

[29] Hochreiter, S., & Schmidhuber, J., “Long short-term memory. Neural computation”, 9(8), 1735-1780, 1997

[30] Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y., “On the properties of neural machine translation: Encoder-decoder approaches”. arXiv preprint arXiv:1409.1259, 2014.

[31] Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.0, 1991

[32] Bahdanau, D., Cho, K., & Bengio, Y., “Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[33] Zhou, J., Xu, W., & He, Y., Bidirectional LSTM-CRF Models for Chinese Word Segmentation. arXiv preprint arXiv:1603.09457, 2016.

[34] Ma, X., Hovy, E., & Liu, Y., End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. arXiv preprint arXiv:1603.01354, 2016.

[35] Sutskever, I., Vinyals, O., & Le, Q. V., “Sequence to sequence learning with neural networks”, pp. 3104-3112, 2014.

[36] Fink, A. “Conducting research literature reviews: From the internet to paper”, Sage publication, 2013.

[37] Koehn, Philipp. "Statistical machine translation." Cambridge University Press, 2009

[38] Hutchins, J., & Somers, H., “An introduction to machine translation.”, Academic Press, 1992.

[39] Zeng, X., & Wang, H., “Research on rule-based machine translation. Journal of Software Engineering and Applications”, 6(1), 1-7, 2001

[40] Daelemans, W., Hoste, V., & De Pauw, G., “Language technology for cultural”, 2018.

[41] Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., & Mercer, R. L., "Statistical speech translation" Proceedings of the 28th annual meeting on Association for Computational Linguistics, 3-12, 1990.

[42] Koehn, P., Och, F. J., & Marcu, D, "Statistical phrase-based translation. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology", 48-54, 2003.

[43] Bangalore, S., Riccardi, G., & Stent, A., "Hybrid machine translation: combining statistical and rule-based machine translation in a single system", Natural Language Engineering, 15(1), 31-53, 2009.

[44] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Dean, J., "Google's neural machine translation system: Bridging the gap between human and machine translation", arXiv preprint arXiv:1609.08144, 2016.

[45] Zoph, B., Knight, K., & Merity, S., "Transfer learning for low-resource neural machine translation", arXiv preprint arXiv: 1604.02201, 2016.

[46] Koehn, P. and Knowles, R., "Six Challenges for Neural Machine Translation. Proceedings of the First Workshop on Neural Machine Translation", 2017.

[47] Haykin, S., "Neural networks and learning machines", Pearson Education, 2009

[48] Mulu Gebreegziabher Teshome and Laurent Besacier (Prof.), "Preliminary experiments on English-Amharic statistical machine translation", IT Doctoral Program, Addis Ababa University, 2012

[49] Alemneh, D. G., & Alemu, G. S, "Ethiopian languages and machine translation: Issues and challenges", In Handbook of Research on Machine Learning Innovations and Trends (pp. 415-437). IGI Global, 2017.

[50] Solomon Teferra Abate, Michael Melese, Martha Yifiru Tachbelie, Million Meshesha, Solomon Atinafu, Wondwossen Mulugeta, Yaregal Assabie, Hafte Abera, Binyam Ephrem, Tewodros Abebe, Wondimagegnhue Tsegaye, Amanuel Lemma, Tsegaye Andargie, Seifedin Shifaw, "Parallel Corpora for bi-Directional Statistical Machine Translation for Seven Ethiopian Language Pairs", Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018.

[51] Zemene, E., Hussein, M., & van der Goot, E., “Lesan – Machine Translation for Low Resource Languages”, In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 2441-2444), 2020.

[52] Kidane, G., Gebre, Y., & Tewelde, G., “Automatic Amharic-Tigrigna Translation Using Statistical Machine Translation Approach”, International Journal of Advanced Computer Science and Applications, 8(1), 107-111, (2017).

[53] Ethnologue, Tigrinya. https://www.ethnologue.com/language/tir, 2021.

[54] Ethnologue, Tigrinya. https://www.ethnologue.com/language/tir, 2022.

[55] The Reporter, “Challenges of promoting Amharic language”, https://www.thereporterethiopia.com/article/challenges-promoting-amharic-language, 2020, August 22.

[56] Imed Zitouni, “Natural Language Processing of Semitic Languages”, March 2014.

[57] Akmajian, A.Demers, K.Farmer, and M.Harnish, "an introduction to language and communication" MIT press, vol. fifth, p. 616, 2017

[58] Weaver, W., “Translation. Machine translation of languages”, Technology Press of Massachusetts Institute of Technology, 1955.

[59] Hutchins, W. J., “Machine translation: Past, present, future”, Ellis Horwood, 1986.

[60] Somers, H., “Computers and translation: A translator's guide”, John Benjamins Publishing, 2003.

[61] Wong, D. F., & Wu, D., “A survey of rule-based machine translation” Journal of Computer Science and Technology, 28(4), 608-625, 2013.

[62] Vauquois, J., “La structure de la phrase simple en français”, Lingua, 11, 97-121.

[63] Karan Singla, “Methods for Leveraging Lexical Information in SMT”, June 2015

[64] Sabine Hunsicker, “Machine Learning for Hybrid Machine Translation”, 2012.

[65] Jorge Vicente-Gabriel, Ana-Belén Gil-González , Ana Luis-Reboredo, Pablo Chamoso and Juan M. Corchado, “LSTM Networks for Overcoming the Challenges Associated with Photovoltaic Module Maintenance in Smart Cities”, 2021

[66] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM,” Neural Computation, vol. 12, no. 10, pp. 2451–2471, 2000

[67] Tongtong Su, Huazhi Sun, Jinqi Zhu, Sheng Wang and Yabo Li, “BAT: Deep Learning Methods on Network Intrusion Detection using NSL-KDD dataset”, 2017.

[68] Chung J., Gulcehre C., Cho K. and Bengio Y., "Empirical evaluation of gated recurrent neural networks on sequence modeling.", arXiv:1412.3555, 2014.

[69] Aloraifan D., Ahmad I., Alrashed E., “Deep learning based network traffic matrix prediction”. Int. J. Intell. Netw, 2, 46–56, 2021.

[70] Genet worku, “GE’EZ-AMHARIC MACHINE TRANSLATION USING DEEP LEARNING”, Bahirdar University, 2021.

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhinm., “Attention is all you need”, 2018

[72] Jabesa Daba and Yaregal Assabie, “Bidirectional English – Afaan Oromo Machine Translation Using Hybrid Approach”, Vol. 8686, pp. 228-235, 2014.

[73] Mekdes Melese, “Attention-based Neural Machine Translation from English-Wolaytta”, St. Mary’s university, January 2023.

Appendixes

Appendix A: data cleaning

```
import re
import pandas as pd
# Read the Excel file into a DataFrame
df = pd.read_excel('/gdrive/MyDrive/corpus/combo.xlsx',
engine='openpyxl')
# Replace multiple characters in a specific column
column_name_to_replace = 'Column2'
column_name_to_replace2 = 'Column1'
replacements = {'።': ' ', '፣': ' ', "“": ",”":", '፤': ' ', '?': ' ', '፡': ' ', '፡-': ' ', '!': ' ', ',': ',
        'ሃ':'ሀ','ሐ': 'ሀ', 'ሑ': 'ሁ', 'ሒ': 'ሂ','ሓ':'ሀ', 'ሔ': 'ሄ', 'ሕ': 'ህ', 'ሖ': 'ሆ',
        'ኅ': 'ሀ', 'ኍ': 'ሁ', 'ኚ': 'ሂ','ኃ':'ሀ', 'ኄ': 'ሄ', 'ኅ': 'ህ', 'ኆ': 'ሆ',
        'ኹ': 'ሁ', 'ኺ': 'ሂ','ኻ':'ሀ', 'ኼ': 'ሄ', 'ኽ': 'ህ', 'ኾ': 'ሆ',
        'ሰ': 'ሠ', 'ሱ': 'ሡ', 'ሲ': 'ሢ','ሳ':'ሣ', 'ሴ': 'ሤ', 'ስ': 'ሥ', 'ሶ': 'ሦ',
        'ኣ':'አ','ዐ': 'አ', 'ዑ': 'ኡ', 'ዒ': 'ኢ','ዓ':'አ', 'ዔ': 'ኤ', 'ዕ': 'እ', 'ዖ': 'ኦ',
        'ጸ': 'ፀ', 'ጹ': 'ፁ', 'ጺ': 'ፂ','ጻ':'ፃ', 'ጼ': 'ፄ', 'ጽ': 'ፅ', 'ጾ': 'ፆ',}
# Escape special characters in replacement strings
replacements = {re.escape(key): re.escape(value) for key, value in
replacements.items()}
# Use the `replace` method with the replacements dictionary
df[column_name_to_replace] =
df[column_name_to_replace].replace(replacements, regex=True)
df[column_name_to_replace2] =
df[column_name_to_replace2].replace(replacements, regex=True)
# Save the modified DataFrame back to the Excel file
df.to_excel('/gdrive/MyDrive/corpus/combo.xlsx', index=False,
engine='openpyxl')
```

Appendix B: splitting the data for train and test

```
from sklearn.model_selection import train_test_split
# Split your data into training and test sets
amharic_train, amharic_test, tigrinya_train, tigrinya_test = train_test_split(
    amharic_sentences, tigrinya_sentences, test_size=0.2, random_state=42)
```

Appendix C: Tokenize and pad sequences for train and test data

```
# Tokenize and pad sequences for train and test data
tokenizer = Tokenizer()
tokenizer.fit_on_texts(amharic_train + tigrinya_train)

amharic_train_sequences = tokenizer.texts_to_sequences(amharic_train)
tigrinya_train_sequences = tokenizer.texts_to_sequences(tigrinya_train)

amharic_test_sequences = tokenizer.texts_to_sequences(amharic_test)
tigrinya_test_sequences = tokenizer.texts_to_sequences(tigrinya_test)

max_sequence_length = max(len(seq) for seq in amharic_train_sequences +
tigrinya_train_sequences)

amharic_train_padded = pad_sequences(amharic_train_sequences,
maxlen=max_sequence_length, padding='post')
tigrinya_train_padded = pad_sequences(tigrinya_train_sequences,
maxlen=max_sequence_length, padding='post')

amharic_test_padded = pad_sequences(amharic_test_sequences,
maxlen=max_sequence_length, padding='post')
tigrinya_test_padded = pad_sequences(tigrinya_test_sequences,
maxlen=max_sequence_length, padding='post')
```

```
# Convert to NumPy arrays
amharic_train_np = np.array(amharic_train_padded)
tigrinya_train_np = np.array(tigrinya_train_padded)

amharic_test_np = np.array(amharic_test_padded)
tigrinya_test_np = np.array(tigrinya_test_padded)

vocab_size = len(tokenizer.word_index)
```

Appendix D: building the model

```
# Build the model
model = Sequential()
model.add(Embedding(input_dim=vocab_size, output_dim=128,
input_length=max_sequence_length))
#GRU layer
model.add(GRU(128, return_sequences=True))
model.add(GRU(128, return_sequences=True))
model.add(GRU(128, return_sequences=True))
model.add(GRU(128, return_sequences=True))
# Attention mechanism
attention = Attention()([model.layers[-1].output, model.layers[-1].output])
# Combine GRU output with attention output
context = Dot(axes=[2, 2])([attention, model.layers[-1].output])
decoder_input = Concatenate(axis=-1)([context, model.layers[-1].output])
# Dense layer for prediction
output_layer = Dense(vocab_size, activation='softmax')(decoder_input)
model = tf.keras.Model(inputs=model.inputs, outputs=output_layer)
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
```

Appendix E: Training the model

```
model.fit(
   x=amharic_train,
   y=tigrinya_train,
   epochs=20,
   batch_size=32,
   validation_data=(amharic_test, tigrinya_test)
)
```

Appendix F: generating a learning curve for the model

```
import matplotlib.pyplot as plt

#for Loss curve
plt.plot(model.history.history['loss'])
plt.plot(model.history.history['val_loss'])
plt.title('Loss Curve')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(['Training Loss', 'Validation Loss'])
plt.show()

#for Accuracy curve
plt.plot(model.history.history['accuracy'])
plt.plot(model.history.history['val_accuracy'])
plt.title('Accuracy Curve')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(['Training Accuracy', 'Validation Accuracy'])
plt.show()
```

Appendix G: defining the sentence translator

```
def translate_sentence(sentence):
    input_seq = tokenizer.texts_to_sequences([sentence])
    input_seq_padded = pad_sequences(input_seq,
maxlen=max_sequence_length, padding='post')
    prediction = model.predict(input_seq_padded)
    predicted_ids = np.argmax(prediction, axis=-1)
    translated_sentence = ""
    for idx in predicted_ids[0]:
      if idx == 0:  # Padding token
        break
      word = tokenizer.index_word[idx]
      translated_sentence += word + " "
    return translated_sentence
```

Appendix H: BLEU score evaluation

```
# Prepare reference and candidate sentences
reference_sentences = [line.split() for line in tigrinya_test]
candidate_sentences = [translate_sentence(sentence).split() for sentence in
amharic_test]
# Calculate BLEU score
bleu_score = corpus_bleu(reference_sentences, candidate_sentences)
print("BLEU Score:", bleu_score)
```