Sebat Bet Gurage (Chaha)-Amharic Machine Translation using Deep Learning

A Thesis Presented

by

Dilu Yirga

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

Declaration

The signatory hereby certifies that the work included in this thesis is unique, has not been submitted for credit toward a degree at this or any other university, and that all references to materials utilized in the thesis have been properly cited.

Dilu Yirga
Full Name of Student

____
Signature

Addis Ababa

Ethiopia

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

Alemebante Mulu Kumlign (PhD)
Full Name of Advisor

____
Signature

Addis Ababa

Ethiopia

January 2024

Acknowledgment

First and foremost, I would like to thank my God and Ever-Virgin, St. Marry, Mother of our Lord, for your blessing and giving me the courage and wisdom to accomplish this thesis.

Next, I would like to thank my my advisor Dr. Alemebante Mulu, for the continuous support of my thesis, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of writing of this thesis. It could not be easier to finish this thesis work without his valuable comments and corrections.

Next I would like to thank Ato Kaleab Zewedu and teacher Ferede Shiketa and, teacher Tewodiros Kifile who are teachers of Agena preparatory school in Gurage Zone, for their support during data translation. I would like to say “Thank you all” for your support in my work.

Table of Contents

Table of Acronyms

AI—Artificial Intelligence

ANN— Artificial Neural Network

DL—Deep Learning

Bi-LSTM—Bidirectional Long Short Term Memory

CNN—Convolutional Neural Network GAMT—

Ge’ez-Amharic Machine Translation HCI—Human-

Computer Interaction

LSTM—Long Short Term Memory

MT—Machine Translation NLP—

Natural language processing NMT—

Neural Machine Translation RMT—

Rule Based Machine Translation SBG

—Sebat Bet Gurage

SMT—Statistical Based Machine Translation

List of Figures

List of table

Abstract

Natural Language Processing (NLP) is defined as a method for computers to intelligently analyze, understand, and derive meaning from human language. Machine translation is a branch of natural language processing that is used to translate text or speech from one language to another. Since before the thirteenth century, the sociolinguistic group of people living in the southwest of Ethiopia known as the administrative "Gurage Zone" has been referred to as "Gurage" (“□□□” for the people and “□□□□” for the language). In this days with the advancement of technology there is the need to translate different official documents, news and other written texts in different languages. The Sebat Bet Gurage-Amharic language translation is one of the concern that needs such translation technologies. However there is no research conducted on machine translation between Sebat Bet Gurage particularly Chaha to Amharic. In this study, we have developed a Chaha-Amharic machine translation model using an encoder decoder machine translation approach. In the study we have collected 5200 Chaha-Amharic parallel sentences from different sources. We then perform cleaning, normalization and tokenization stages to preprocess the dataset. We have experimented an encoder decoder model using LSTM, Bi-LSTM and GRU deep learning algorithms. Based on the result of our experiments done in this study, the encoder decoder model using the Bi-LSTM algorithm has a better BLEU score. The encoder decoder model using the Bi-LSTM algorithm scored 22, the encoder decoder model using the LSTM algorithm scored 17 and the encoder decoder model using the GRU algorithm scored 20. From the experiment the encoder decoder model using the Bi-LSTM algorithm took a long training time of 1:30 hours.

CHAPTER ONE

1 introduction

1.1. Background

Ethiopia is Africa's sole country with its own script (Fidel) and numbering system (Ahaze). According to many manuscripts, Ethiopia is a nation with a great deal of wisdom in terms of various types of literature and a home to a diverse range of literary works. For thousands of years, our forefathers and mothers have worked hard to preserve and build the country's traditions, handing them down to the next generation. This is one of the reasons Ethiopians are proud of their country, as well as the continent of Africa [1].

Since before the thirteenth century, the sociolinguistic group of people living in the southwest of Ethiopia known as the administrative "Gurage Zone" has been referred to as "Gurage" (“□□□” for the people and “□□□□” for the language). It has no negative connotations and is the Gurage people's favored phrase. Geographically speaking, linguistic groups within the current Gurage Zone include Soddo ((□□)) also known as □□□□ Kistane, Dobbi (□□)also known as Gogot (□□□),, Mesqan (□□□□) spoken in the north, Silt'e (□□□), Zay (□□),, Wolane (□□□) spoken in the east, and Sebat Bet (□□□ □□) spoken in the west. The term Sebat-Bet, which means to "Seven Houses," refers to the seven regions (districts) of Sebat Bet that developed a history of political alliances to defend themselves against outside aggressors. It also corresponds to the seven dialects spoken in the various districts. The dialects of Sebat Bet are Chaha (□□), Ezha (□□), Muher (□□□), Gura (□□), Gumer (□□□), Inor (□□□), Gyeta (□□), Endegegn (□□□□□) and Mesmes (□□□□), which is now extinct but which may once more be used interchangeably with the parent language—a frequent cause of linguistic ambiguity. Sebat Bet is frequently transliterated into other lexical forms and is frequently condensed into the single word "Sebatbeit," which is used in the Unicode Standard to name the letters that are unique to the language. The term "Sebat Bet" has also been used interchangeably with "Gurage" in numerous settings, which has led to ongoing misunderstandings about the peoples and languages it refers to [2].

Amharic is the world's second most widely spoken Semitic language next to Arabic. Although Ethiopians speak a variety of languages, Amharic is the country's lingua franca and most literary language, having long served as a medium of instruction in the country's educational system.

Amharic uses the Ethiopic alphabet and has 34 base characters, each with its own set of variations. Amharic has a complicated morphology, with many words derived from different word forms through inflectional and derivational processes [3]. Among the 89 languages listed in Ethiopia, Amharic is the official working language of the government. The bulk of Amharic speakers live in Ethiopia, but it is also spoken in Israel, Eritrea, Canada, the United States, and Sweden. Different sections of the country have five dialectical variations. Addis Ababa, Gojam, Gonder, Wollo, and Menz are among these dialects [4]. The Amharic language has tens of millions of native speakers, but it is one of the most under-resourced languages in dataset linguistics. Although there are few corpora and language tools for Amharic, progress is being made in the creation of both language data and tools [5].

A set of rules or a set of symbols can be used to define a language. Symbols are mixed and utilized to transmit or broadcast information. Natural Language Processing (NLP) is a branch of AI and linguistics concerned with making computers understand statements or words written in human languages. Natural Language Processing is divided into two parts: Natural Language Understanding and Natural Language Generation, which evolves the work of comprehending and producing text. Natural language processing was created to make users' lives easier and to fulfill their desire to connect with computers in natural language. NLP caters to those users who do not have enough time to learn new languages or perfect them, as not all users are well-versed in machine specific language [6]. Natural language processing (NLP) is the study of the interactions between human and the computer using natural language. Computer science, artificial intelligence, and computational linguistics all cross in this field. NLP is a textual content evaluation approach that permits machines to interpret human speech. Many NLP applications enables automatic text summarization, sentiment analysis, topic extraction, named entity recognition, parts-of-speech tagging, connection extraction, stemming, and other real-world applications which helps human-computer interaction like applications including text mining, machine translation, and automated question answering [7].

Machine translation (MT) is a well-known subfield of Natural Language Processing (NLP) that studies how to utilize computer software to translate text or speech from one language to another without using humans. Because the MT task has a goal that is comparable to the eventual goal of NLP and AI, namely, to fully understand human text (voice) at the semantic level, it has gotten a lot of interest in recent years. Apart from its scientific importance, MT offers a significant potential for reducing labor costs in a variety of practical applications, including scholarly communication and international business negotiations [8]. The benefits of machine translation are numerous. Machine translation has several advantages over human translation, the foremost of which is speed. Each minute, a machine can translate thousands of words. However, a human can translate around 2000 words each day on average. Cost is the second benefit. In comparison to manual translation, the price is also reasonable. The main expense of a typical human translation project is the salary of the translators, but the main expense of a machine translation project is in the post-editing stage, which saves the customer money. Term or phrase memorization is a capability of machine translation. When using numerous human translators, this makes it more difficult to achieve translations that are consistently applied throughout the entire file [9].

One of the most difficult academic problems in linguistics and computer science is developing systems for high-quality translation and multi-linguistic processing. Several scholars attempted to use English NLP technologies to understand and analyze other languages due to a lack of research and development in under-resourced languages. As a result, several academics have attempted to develop cutting-edge translation services and methods in order to improve translation results. A two-step strategy can be utilized to apply NLP technology to low-resource languages. To begin, well-designed translation procedures should be used to translate low-resource language materials into high-resource language. Second, various word embedding methods encode the translated material as vectors. As a result, enhanced translation approaches can help NLP systems perform better in different languages [10].

During the early years of research, machine translation systems were built using bilingual dictionaries and some handcrafted rules; however, it proved difficult to handle all language anomalies with these handcrafted rules. In the 1980s, increased processing capability prompted a shift from rule-based to statistical machine translation. As a result of the availability of massive parallel corpora and advances in deep learning, a paradigm shift from statistical to neural models occurred [11].

Deep learning is a form of machine learning that use artificial neural networks to imitate the functionality of the human brain. It can process enormous amounts of data and identify vital patterns that can aid in critical decision-making. It provides cutting-edge accuracy in many NLP tasks such as text classification and language translation [12]. Deep learning architectures like deep belief networks, deep neural networks, and convolutional neural networks are used in a wide range of applications, including natural language processing tasks such as speech recognition, audio recognition, bioinformatics, and machine translation, and in some cases, they outperform human experts [13]. Deep learning has revolutionized a variety of fields in recent years, ranging from computer vision to game artificial intelligence (AI). In response to these developments, the area of machine translation has switched to the use of deep-learning neural-based methods, which have largely supplanted previous approaches like as rule-based systems or statistical phrase-based methods. Deep learning techniques like neural machine translation (NMT) models can now access the whole information accessible anywhere in the source phrase and automatically learn which piece is useful at which stage of synthesizing the output text. The primary explanation for the huge improvement in translation quality is the removal of previous independence assumptions [14].

In many NLP tasks, deep learning algorithms have recently been used with astonishing precision. Word embedding is one of these techniques that uses the principles of deep learning, which detects both semantic and syntactic links between words and enables for the capture of more contextual cues available in human language. As a result, one of the word embedding approaches used to determine semantic and syntactic word relations is Word2Vec. Word2Vec is a two-layer neural network that acts on a sequence of texts to construct a vocabulary based on words that appear more than a user-defined threshold of times [15].

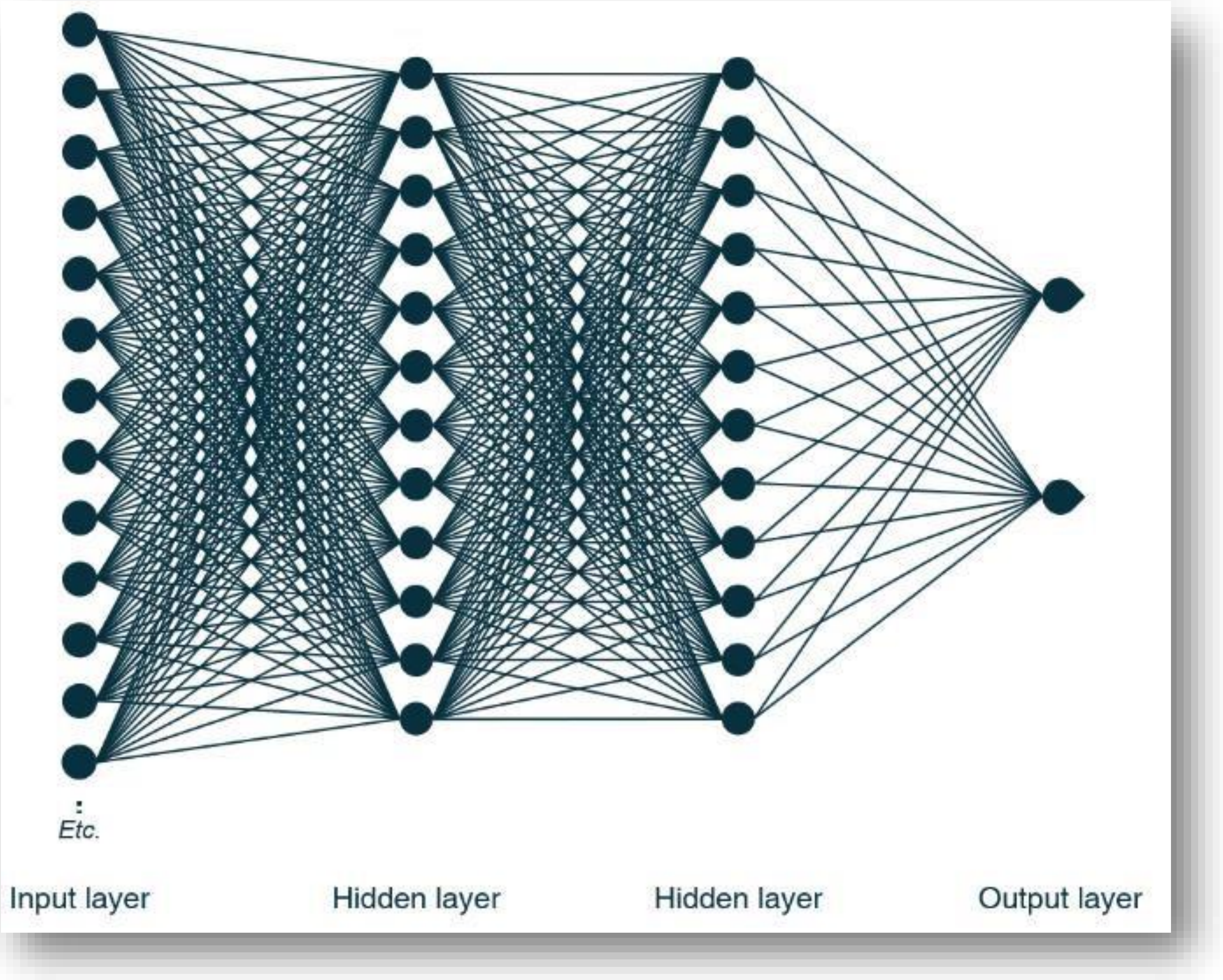


Figure 1 architecture of deep learning [16]

1.2. Problem statement

Sebat Bet Gurage, sometimes referred to as Central West Gurage, Gouraghie, Guragie, Gurague, or West Gurage, is a language used by the Gurage/Sebat bet people in the West Gurage Zone. It is a member of the Gurage family of languages. Text can be translated from one natural language to another using machine translation, which is an application of NLP. The goal of this research is to build a machine translation model for Sebat Bet Gurage (Chaha) -Amharic languages. There has never been any study done on machine translation between Sebat Bet Gurage and Amharic languages. It is crucial to develop machine translation between Sebat Bet Gurage and Amharic languages due to an increase in language users, addressing the concerns of the endangered of the Sebat Bet Gurage language, and increasing the content of the language in the web. It is necessary to translate many official documents, news articles, and other written texts into both languages in order to exchange information.

Neural Machine Translation (NMT) is a recently proposed approach to machine translation (MT) that has achieved the state-of-the-art translation quality in recent years. Unlike traditional MT approaches, NMT aims to create a single neural network that can be tuned collaboratively to maximize translation performance. Deep learning approaches, as opposed to SMT, are capable of capturing long dependencies in sentences, indicating that they have a lot of potential to become a new language translation trend. One of the early attempts to increase the efficiency of capturing long-term dependency was the hierarchical recurrent neural network [17]. So, the aim of this study is to develop Sebat Bet Gurage (Chaha)-Amharic machine translation model using Deep learning.

1.3. Research motivation

Machine translation plays an important role in strengthening communications between people residing in different areas. It enables peoples to use documents and data produced in different languages. Now a days there is a high demand for translation due to the requirement for information sharing among different languages. Among the languages that needs such information sharing are Ethiopian languages such as Sebat Bet Gurage (Chaha). Previously there are no any machine translation systems developed for Chaha language. There are Chaha documents that need to be shared for Amharic speakers, like culture, indigenous knowledge of the society, etc. So, this necessitates to study Chaha-Amharic machine translation. This has motivated us to work on Chaha-to-Amharic machine translation.

1.4. Research question

The following are research questions answered at the end of our study.

A. Which design and evaluate method is best Sebat Bet Gurage (Chaha) to Amharic machine translation model?

B. Which deep learning method is best for Sebat Bet Gurage (Chaha) to Amharic translation problem?

C. Which extent does the proposed bidirectional translation model work?

1.5. The objective of the study

1.5.1. General objective

The main objective of this study is to develop Sebat Bet Gurage (Chaha)-Amharic machine translation model using deep learning techniques.

1.5.2. Specific objectives

To realize the aforementioned general objective, our study will carry out the following specific objectives:

- To prepare a parallel dataset for Sebat Bet Gurage (Chaha)-Amharic machine translation model
- To design and evaluate Sebat Bet Gurage (Chaha)-Amharic machine translation model.
- To train and compare different deep learning algorithms for Sebat Bet Gurage (Chaha)-Amharic machine translation.
- To Prevent cheah language from vanish
- To learn deppely the cheha language

1.6. Scope and limitation of the study

1.6.1. Scope

The scope of our study is:

- Sebat Bet Gurage (Chaha)-Amharic translation model and not the reverse

1.6.2. Limitations

The following are out of the scope of our research work:

- Word sense disambiguation
- Amharic to Sebat Bet Gurage (Chaha) translation
- Considering other Sebat Bet Gurage languages.

1.7. Research methodology

In this study we followed experimental research technique. In experimental research technique one or more independent variables are manipulated and applied to 1 or greater structured variables to degree their impact at the latter. The following methods are applied to achieve our objective.

1.7.1. Method of data collection

We collected the dataset for our research from different sources like dictionaries and Gurage zone offices. The dataset have been collected from sources like the Holy Bible such as Saint Marikos, Saint Matiwos and Sebat Bet Gurage-Amharic-English dictionary available in Gurage zone.

After collecting our dataset, we will pre-processes it using the following techniques.

- Data labeling: involves preparing the data collected to make Sebat Bet Gurage-Amharic translation dataset.
- Data cleaning: involves the removal of unnecessary characters.
- Normalizing: involves making homophone characters follow the same script. For example, we changed the family of □, □ and □ to the family of either □ or □.

1.7.2. Tools for Implementation

In this study, we used different tools to develop the proposed model. We used tools like Pycharm, Jupyter notebook as editor. We used Python programming to develop the translation model.

Pycharm

PyCharm is a Python-specific Integrated Development Environment (IDE) that provides a wide range of essential tools for Python developers. These tools are tightly integrated to create a convenient environment for productive Python, web, and data science development.

Jupyter notebook

Researchers can mix software code, computational results, explanatory text, and multimedia materials in a single document by using Jupyter, a free, open-source interactive web platform. Although computational notebooks are not new, Jupyter in particular has been more and more well-known in recent years. An active user-developer community and a revamped architecture that enables the notebook to speak dozens of programming languages have contributed to this quick adoption.

Python

Python is a high-level, object-oriented, interpreted language with dynamic semantics. Because of its high-level built-in data structures, dynamic typing, and dynamic binding, it's a very attractive language to employ as a glue language or scripting language to join existing components. Python's straightforward, learnable syntax prioritizes readability and reduces software maintenance expenses. Python supports packages and modules, which encourages code reuse and program modularity. You can use and distribute the large standard library and Python interpreter for free on all major platforms, either in source or binary form.

1.8. Techniques

1.8.1. Deep learning Algorithms

In recent trends of NLP areas deep learning (DL) is gaining importance due to its best accuracy when it is trained with large amount of data. It is a technique for simulating the network of neurons in the human brain. It is a subset of machine learning that uses deep neural networks and is referred to as deep learning. Deep learning systems are constructed using connected layers. The first layer is the Input Layer, and the final is the Output Layer. All of the layers in between are known as Hidden Layers. Each of the Hidden layers is made up of neurons. All of the neurons are interconnected. The neuron will process and then propagate the incoming signal from the layer above [18].

1.9. Significance

The study has significance for both the area of machine translation and the preserving of the Sebat Bet Gurage language for future generations. Since under resourced languages like the Sebat Bet Gurage language is not studied in many NLP areas, the study will introduce new knowledge into the field of machine translation of Sebat Bet Gurage language. The study will create datasets, new insights for future researchers about machine translation of low resourced Ethiopian languages like the Sebat Bet Gurage language.

1.10. Thesis organization

The second chapter presents a literature review on the two languages, natural language processing, machine translation and approaches of machine translation, background of Amharic and Sebat Bet Gurage (Chaha) languages and related work on both local and international languages. The third chapter discusses the methodology, which include the dataset preparation and the proposed model architecture. The fourth chapter discuss about the result, discussion and findings of the study. The last chapter discusses the conclusion of the study and recommendations based on the study's results and findings.

.

CHAPTER TWO

2 LITERATURE REVIEW

2.1. Sebat Bet Gurage language

Gurage refers to the group of languages and dialects spoken in the Gurage Zone, Silt'e Zone, and the Oromo region as well as the small communities of South Ethiosemitic language speakers who reside in the Gurage Zone of the Southern Nations, Nationalities and Peoples' Regional State. The Gurage languages are bounded by the Omotic language Yemsa in the West, the Cushitic language Afan Oromo in the North, North East, and South East, Libido, also known as Mareqo, in the East, K'abeena in the North West, Hadiyya in the South West, and Alaba in the South. Gurage has a more convoluted classification history than other language groups. The linguistic traits used to distinguish each of the Gurage languages' subgroups were many and occasionally inaccurate [19]. Semitic-speaking ethnic groups that are encircled by Cushitic-speaking people have historically been referred to be "Gurage." The name Gurage's derivation is still unclear. It was discovered that the Gurage land, a Semitic enclave in a Cushitic region, is a relatively tiny area. Geographically, the Gurage people reside on the route to Jimma, which is bordered by the Rift Valley in the east and northeast and the Awash River, about 200 km to the southwest of Addis Ababa, the capital city of Ethiopia. The Southern Nations, Nationalities and Peoples' Regional state administration currently oversees it [20]. The term "Gurage" refers to the southernmost Semitic-speaking populations that are surrounded by Cushitic and Omotic languages in an area roughly bounded by the Rift Valley lakes in the east, the Awash River in the north, and the Gibe River in the west and southwest. It does not refer to a single linguistic entity [21].

The term Sebat-Bet, which means to "Seven Houses," refers to the seven regions (districts) of Sebat Bet that developed a history of political alliances to defend themselves against outside aggressors. It also corresponds to the seven dialects spoken in the various districts. The dialects of Sebat Bet are Chaha (□□), Ezha (□□), Muher (□□□), Gura (□□), Gumer (□□□), Inor (□□□), Gyeta (□□), Endegegn (□□□□□) and Mesmes (□□□□), which is now extinct but which may once more be used interchangeably with the parent language—a frequent cause of linguistic ambiguity. Sebat Bet is frequently transliterated into other lexical forms and is frequently condensed into the single word "Sebatbeit," which is used in the Unicode Standard to name the letters that are unique to the language. The term "Sebat Bet" has also been used interchangeably with "Gurage" in numerous settings, which has led to ongoing misunderstandings about the peoples and languages it refers to [2].

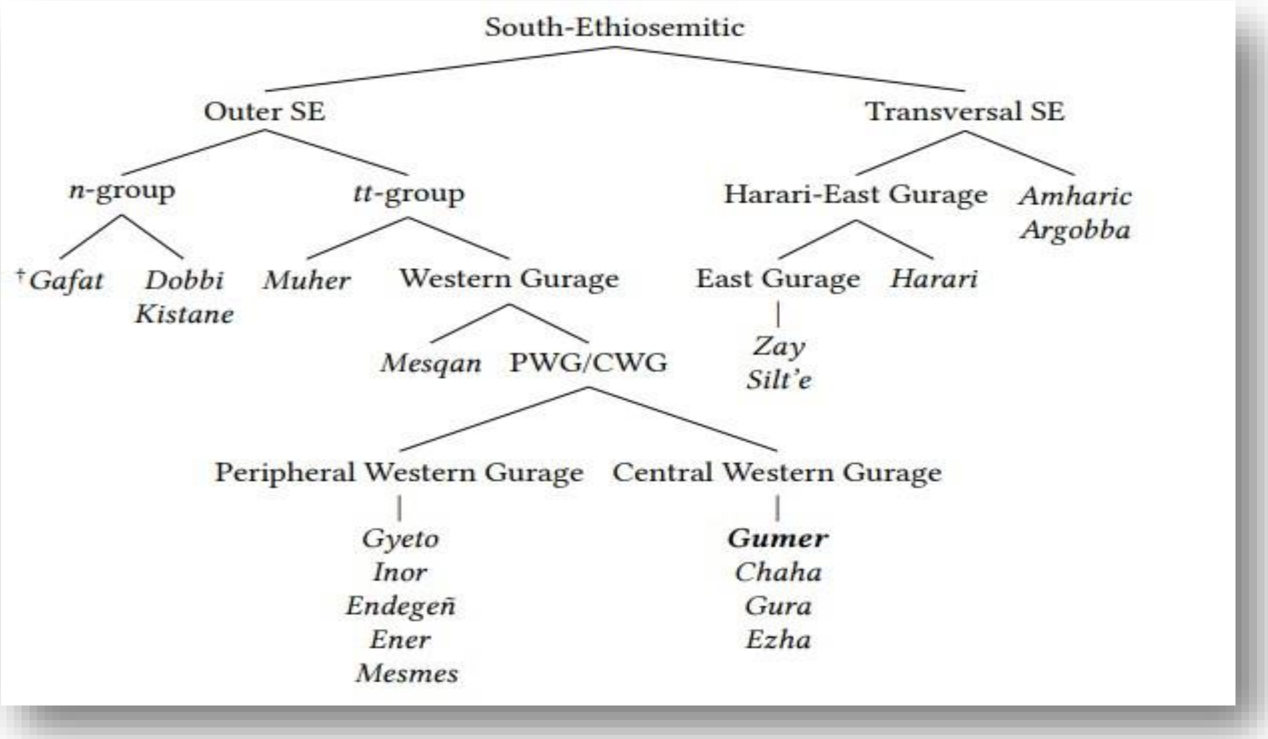


Figure 2 South Ethiosemitic [21]

Based on the literature from [22] SBG contains a set of phonemes that is very normal for an Ethiopian Semitic language. There are the typical set of ejective consonants in addition to straightforward voiced and voiceless consonants. However, compared to most other Ethiopian Semitic languages, the Chaha language also includes a greater variety of palatalized and labialized consonants. In addition to these languages' standard seven vowels, SBG also features open-mid front (ɛ) and back vowels (ɔ). Both short and long vowel phonemes can be found in some dialects, and some also use nasalized vowels. In addition to the intricate relationship between the set of consonants in a verb's root and how they are realized in a specific form of that verb or a noun derived from that verb, which is a feature of all Semitic languages, SBG exhibits another level of complexity. For instance, the word 'open' has a root consisting of the consonants $\{kft\}$, just like it does in the majority of other Ethiopian Semitic languages. All of these consonants are present in some forms. For example, käfätä-m is the third person singular masculine perfective Chaha form meaning ‘□□□□’ ('he opened'). However, two of the stem consonants are replaced when the impersonal form of this same verb, which basically means ‘□□□□□’ (‘he was opened’) is used: 'käf^{w}äč-i-m'. Like the Amharic language the writing method used to transcribe Chaha is Ge'ez (Ethiopic). However there are some variations in alphabets of the languages.

Based on the study by [23] linguists disagree as to whether Gurage is a single language or a fusion of several different languages, despite the fact that there are many different languages spoken here. As a result, it is unknown how many Gurage languages there are. Gurage language variations were categorized by various linguists according to various tenets. For instance, some have divided the language in to twelve dialects of Gurage clusters, calling them Chaha, Ezha, Soddo, Goggot, Ennemor, Selte, Endegagn, Masqan, Muher, Wolane, Gyeta, and Zay. Some of these dialects also have sub-dialects, such as Gumer of Chaha and Ener of Endagegn. Additionally, some have divided the Gurage languages into three dialect Clusters, which are as follows:

I. East : Selte, Wolane, and zay connected to Harari

II. West: Chaha, Ezha, Ennemor, Endegegn, Gyeta, Goggot, Muher, Masqan connected to Amharic

III. North (East) which includes Soddo, and considering Gurage as a single language

The tables below shows the vowels and consonants of the Chaha which is a sub group of the SBG language. Table 1 shows the vowels of Chaha and table 2 shows the consonants of Chaha language. The orthography of the Chaha language is stated in the Appendix A.

Table 1 Chaha Vowels [22]

	Front	**Central**	**Back**
High	i	ɨ ⟨ ə⟩	u
High-mid	e		o
Low-mid	ɛ	ɐ ⟨ ä⟩	
Low		a	

Table 2 Chaha Consonants [22]

		Labial		Dental	Post-alveolar	Palatal	Velar		Glottal
		plain	round				plain	round	
Nasal		m	m^{w}	n					
Plosive/ Affricate	voiced	b	b^{w}	d	d͡ʒ ⟨ ğ⟩	ɟ⟨ g^{y}⟩	g	g^{w}	
	voiceless	p	p^{w}	t	t͡ʃ ⟨ č⟩	c ⟨ k^{y}⟩	k	k^{w}	
	ejective			t’ ⟨ ṭ⟩	t͡ʃ’ ⟨ č□̣	c’ ⟨ $ḳ^{y}$⟩	k’ ⟨ ḳ⟩	$k’^{w}$ ⟨ $ḳ^{w}$⟩	
Fricative	voiced			z	ʒ ⟨ ž⟩				
	voiceless	f	f^{w}	s	ʃ ⟨ š⟩	ç ⟨ x^{y}⟩	x	x^{w}	h
Approximant		β̞		l		j ⟨ y⟩		w	
Rhotic				r					

2.2. Amharic language

After Arabic, Amharic is the world's second most widely spoken Semitic language. Although Ethiopians speak a variety of languages, Amharic is the country's lingua franca and most literary language, having long served as a medium of instruction in the country's educational system. Amharic uses the Ethiopic alphabet and has 34 base characters, each with its own set of variations. Amharic has a complicated morphology, with many words derived from different word forms through inflectional and derivational processes. Through intricate affixation, reduplication, and Semitic stem inter digitation, several surface forms of words can be created from a base form. A number of roles can be assigned to Amharic words. A verb, for instance, can be marked for person, case, gender, number, tense, aspect, and mood [3]. Among the 89 languages listed in Ethiopia, Amharic is the official working language of the government. The bulk of Amharic speakers live in Ethiopia, but it is also spoken in Israel, Eritrea, Canada, the United States, and Sweden. Different sections of the country have five dialectical variations. Addis Ababa, Gojam, Gonder, Wollo, and Menz are among these dialects [4]. The Amharic language has tens of millions of native speakers, but it is one of the most under-resourced languages in dataset linguistics. Although there are few corpora and language tools for Amharic, progress is being made in the creation of both language data and tools [5].

Amharic language is morphologically rich and like other Semitic languages, exhibits the root-pattern morphological phenomenon. A root is a group of consonants (also known as radicals) that have a basic lexical meaning. A pattern is made up of a series of vowels that are inserted between the consonants of a root to form a stem. Aside from this non-concatenative morphological feature, Amharic employs a variety of affixes to generate inflectional and derivational word forms [24]. Based on the study [25] Amharic words have 8 classes. These are noun (□□), adjective (□□□), verb (□□), preposition (□□□□□□) and adverb (□□□□ □□), pronoun (□□□□ □□), article (□□□□□□) and exclamation (□□ □□□).

A noun is a type of word that is used to refer to something (s). Nouns are used to indicate gender (Amharic has two gender types: feminine and masculine) and numbers.

Example

□□ □□ -- She is a thief. This indicates that the gender is
feminine. □□□□ □□ -- He is a lion. This indicates the gender is
masculine. □□ □□ -- it is a bull. This indicates singularity
□□□□ □□□ -- they are bulls. This indicates plurality

A verb is distinguished from other word classes by two characteristics. The first is that it comes at the end of an Amharic sentence, and the second is that it has a suffix attached to it that indicates the subject of the sentence.

Example

□□□ □□ □□□[abebe ate beso] □□ is the verb of the sentence.
□□ □□ □□□ [the boy came today] □□ is the verb of the sentence.

An adjective is used to modify the noun. Example □□ □□□ □□ □ [the boy is black] □□□ is adjective that modifies the name boy.

Adverbs can be derived from adjectives in some cases. By affixation and intercalation, nouns are formed from other basic nouns, adjectives, stems, roots, and the infinitive form of a verb. Nouns are inflected by case, number, definiteness, and gender marker affixes. Adjectives are formed by prefixing or suffixing nouns, stems, or verbal roots. Furthermore, adjectives can be formed by compounding. Adjectives, like nouns, are inflected for gender, number, and case. Roots are used to create Amharic verbs. Intercalation and affixation are both required for the conversion of a root to a basic verb stem [24].

2.3. Natural language processing

Natural language processing (NLP) emerged in the 1950s as a synthesis of artificial intelligence and linguistics. Text information retrieval (IR), which uses highly scalable statistics-based techniques to index and search vast quantities of text quickly, was first distinguished from NLP. However, NLP and IR have converged somewhat over time. NLP currently borrows from a variety of domains, requiring today's NLP researchers and developers to dramatically expand their conceptual knowledge base [26]. In computational linguistics, grammar refers to the study of certain structures and rules found in language, such as determining the principles of sentence order and classifying words. Language Model and Part-of-Speech Tagging are two ways for expressing linear laws in these languages. Syntactic Structure or Dependency Relationship between words in the sentence can be used to represent nonlinear information in the sentence. Although the analysis and expression of sentence structure may not be the final goal of natural language processing problems, it is frequently a key step in solving the problem [27].

Natural Language Processing, also known as Human Language Technology, Language Technology, or Speech and Language Processing, is an interdisciplinary field that aims to get computers to perform useful natural language tasks such as enabling human-machine communication, improving human-human communication, or simply performing useful text and speech processing. This field's research and development activities include human language coding, recognition, interpretation, translation, and generation. Speech and language technologies such as speech recognition and synthesis, machine translation, text categorization, and text mining are the end results of such activities [24].

Natural language, as opposed to computer languages, is human language. The distinction between them is that uncertainty is present. There is no ambiguity in any well-designed computer language. All known natural languages, on the other hand, have the trait of ambiguity. Ambiguity happens when an input can be interpreted in multiple ways. Ambiguity exists at every level of human communication. The study of computer programs that take natural, or human, language as input is known as natural language processing. Natural language processing software can tackle a variety of tasks, from low-level tasks like attributing components of speech to words to high-level ones like answering questions. Natural language processing (NLP) is required to convert meaningful information contained in text into structured data that may be utilized by computer operations [28].

2.4. Machine translation

One of the earliest and most fascinating areas of natural language processing is machine translation. The primary goal is to break down language barriers by creating a machine translation system that can translate from one human language to another. Machine translation is a branch of artificial intelligence that uses computers to translate one natural language into another. It is an interdisciplinary field of study that incorporates concepts from various disciplines such as languages, artificial intelligence, statistics, and mathematics. The concept of machine translation can be traced back to the early days of computing. The machine translation field first appeared in the memorandum of Warren Weaver, one of the field's pioneers, in 1949. In this digital age, various communities all over the world are connected and share vast resources. In this type of digital environment, different languages create a barrier to communication [11].

Approaches to MT can be divided into two categories based on methodology: rule-based methods and dataset-based methods. Rule-based methods dominated from the time the concept of MT was first proposed until the 1990s. Rule-based machine translation (RBMT) methods translate source language texts into target language texts using bilingual dictionaries and manually written rules. However, manually writing rules is time consuming. Furthermore, rules are difficult to keep and transfer from one domain to another, as well as from one language to another. As a result, rule-based systems are difficult to scale for open-domain translation and multilingual translation. MT systems were initially designed primarily for military applications. Georgetown University, in collaboration with the now-famous computer manufacturer International Business Machines Corporation (IBM), completed the first Russian-English MT experiment using the IBM-701 computer in 1954, demonstrating that the dream of MT had come true. After the 1954 demonstration, MT was a hot topic for more than a decade. It became extremely difficult to work on MT after the report, which was extremely skeptical of MT and resulted in a drastic cut in funding for MT research. The Association for Computational Linguistics (ACL), the dominant scientific society today, was originally named the Association for Machine Translation and Computational Linguistics in 1962, during the boom; however, it dropped the "MT" from its name in 1968, during the bust, following the ALPAC report. Meanwhile, MT researchers kept trying to improve translation quality. The first International Conference on Computational Linguistics, which focused on rule-based parsing and translation, was held in 1965 by NLP researchers. Beginning in the 1970s, RBMT methods became more refined. One of the first MT companies, SYSTRAN, launched a commercial translation system in 1978, which was one of the most well-known examples of a commercially successful rule-based system at the time. SYSTRAN's services were used by Google until 2007 [29].

Machine translators can be purchased as commercial computer solutions (for example, SYSTRAN Enterprise Server and IBM WebSphere) or as free Web-based applications (eg, Google Translate and Microsoft Bing Translator). Most machine translators are text-based and provide instant translations between various languages; however, audio output is sometimes available. A variety of language keyboards are occasionally available. Google Translate, for example, has a keyboard icon that allows users to switch between different language scripts by toggling an on-screen keyboard. To use the virtual keyboard, select the language from which you want to translate (i.e., uncheck "Detect language" and select a language other than English). The virtual keyboard icon will appear in the text box's lower left-hand corner. Smartphone apps that connect to online machine translation programs are also on the rise [30].

2.5. Approaches of machine translation

The core methodology of MT systems can be used to classify them. There are two main paradigms in this classification: the rule-based approach and the dataset-based approach. In the rule-based approach, human experts specify a set of rules to describe the translation process, necessitating a massive amount of human expert input. The dataset-based approach, on the other hand, extracts knowledge automatically by analyzing translation examples from a parallel dataset built by human experts. The Hybrid Machine Translation Approach was created by combining the characteristics of the two major classifications of MT systems [31].

2.5.1. Rule-based machine translation (RBMT)

Rule-Based Machine Translation (RBMT), also known as Knowledge-Based Machine Translation and the Classical Approach to MT, is a general term for machine translation systems that are based on linguistic information about the source and target languages that is retrieved from (bilingual) dictionaries and grammars that cover the main semantic, morphological, and syntactic regularities of each language. An RBMT system generates output sentences (in some target language) from input sentences (in some source language) based on morphological, syntactic, and semantic analysis of both the source and target languages involved in a concrete translation task [31]. The rule-based machine translation Approach includes three distinct approaches. They are the Direct, Transfer, and Interlingua Machine Translation Approaches, in that order. Despite being members of the RBMT, their approaches to achieving a representation of meaning or intent that is independent of language in both the source and target languages vary [31].

Direct Machine Translation (DMT) Approach

The Direct Machine Translation Approach is the shallowest level at the bottom of the pyramid. The DMT approach is the oldest and least popular. At the word level, direct translation is performed. This approach enables machine translation systems to translate from one language, known as the source language (SL), to another, known as the target language (TL) (TL). The SL words are translated without the use of an additional/intermediary representation. The analysis of SL texts is limited to a single TL. Direct translation systems are primarily bilingual and unidirectional in nature. A minimal amount of syntactic and semantic analysis is required for the direct translation approach. SL analysis is focused on producing representations that are appropriate for a single TL. DMT is a word-for-word translation method with some minor grammatical changes [31]. Figure 1 shows the steps in direct machine translation approach.

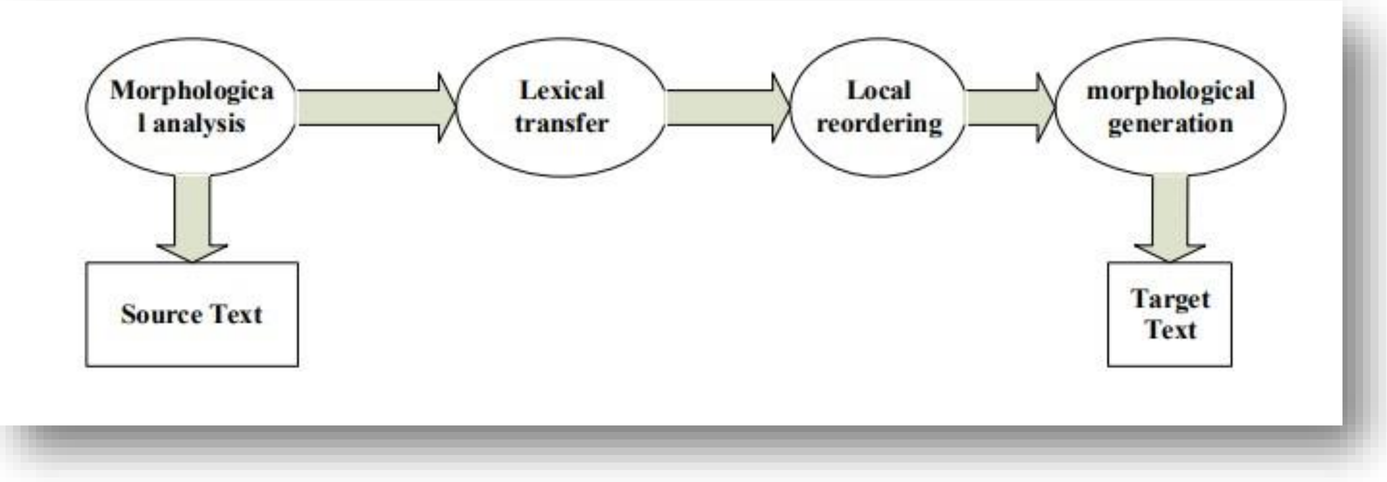


Figure 3 Steps of direct machine translation approaches

Transfer-based Machine Translation Approach

Because of the shortcomings of the Interlingua approach, a better rule-based translation approach known as the Transfer-based Approach was discovered. Transfer-based machine translation, like interlingual machine translation, generates a translation from an intermediate representation that simulates the original sentence's meaning. It is partially dependent on the language pair involved in the translation, as opposed to interlingual MT. A transfer system can be divided into three stages based on structural differences between the source and target languages: I analysis, ii) transfer, and iii) generation. The SL parser is used in the first stage to generate the syntactic representation of an SL sentence. The first stage's output is converted into equivalent TL-oriented representations in the following stage. A TL morphological analyzer is used to generate the final TL texts in the final step of this translation approach. This translation method can produce reasonably high-quality translations [31]. Figure 2 shows steps in Transfer-based approach of machine translation.

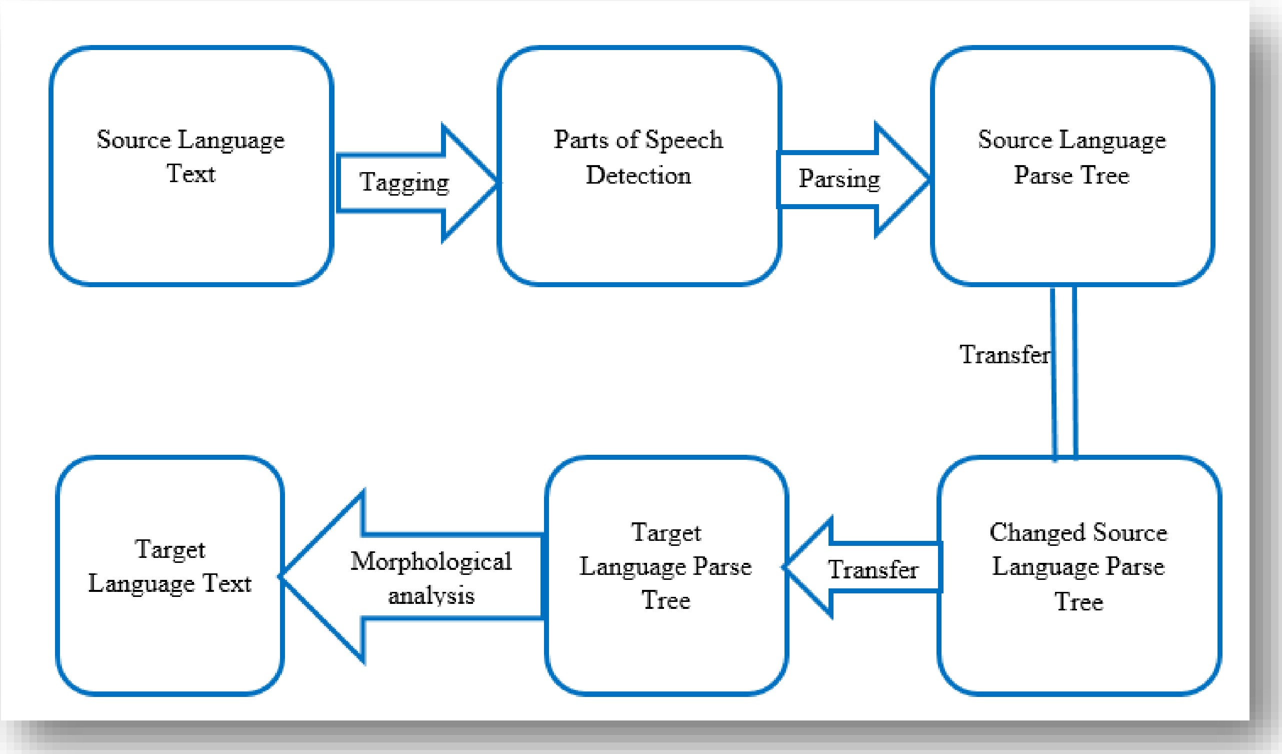

Source Language Text
Tagging
Parts of Speech Detection
Parsing
Source Language Parse Tree
Transfer
Changed Source Language Parse Tree
Transfer
Target Language Parse Tree
Morphological analysis
Target Language Text

Figure 4 Transfer-based Approach of MT

2.5.2. Dataset-based Machine Translation Approach

Dataset-based machine translation (also known as data-driven machine translation) is an alternative approach to machine translation that addresses the rule-based machine translation problem of knowledge acquisition. Dataset Based Machine Translation (CBMT) uses a bilingual parallel dataset to obtain knowledge for new incoming translations, as the name implies. This method makes extensive use of raw data in the form of parallel corpora. Text and translations are included in this raw data. These corpora are used to learn how to translate. The dataset-based approach is further subdivided into the two approaches listed below: Approaches to Statistical Machine Translation and Example-based Machine Translation [31].

Example-based Machine Translation Approach

Example-based machine translation (EBMT) is distinguished by its use of a bilingual dataset with parallel texts as its primary knowledge, with the main idea being translation by analogy. An EBMT system is given a set of sentences in the source language (from which it is translating) and point-to-point translations of each sentence in the target language. These examples are used to translate sentences from the source language to the target language. EBMT consists of four tasks: example acquisition, example base and management, example application, and example synthesis. The concept of translation by analogy is at the heart of example-based machine translation. Through the example translations used to train such a system, the principle of translation by analogy is encoded to example-based machine translation [31].

The majority of example-based MT systems use phrases or sentences as the unit for examples, allowing them to translate while taking case relations or idiomatic expressions into consideration. Example-based MT treats a bilingual dataset as a database and retrieves examples that are similar to an input sentence. However, example-based MT chooses the optimal example based on how similar the input and source parts of the example are when some examples conflict during retrieval. This suggests that example-based MT does not verify the accuracy of the translation of the provided input sentence [32]. Figure 3 shows the architecture of example based machine translation. The majority of example-based MT systems use phrases or sentences as the unit for examples, allowing them to translate while taking case relations or idiomatic expressions into consideration. Example-based MT treats a bilingual dataset as a database and retrieves examples that are similar to an input sentence. However, example-based MT chooses the optimal example based on how similar the input and source parts of the example are when some examples conflict during retrieval. This suggests that example-based MT does not verify the accuracy of the translation of the provided input sentence [34].

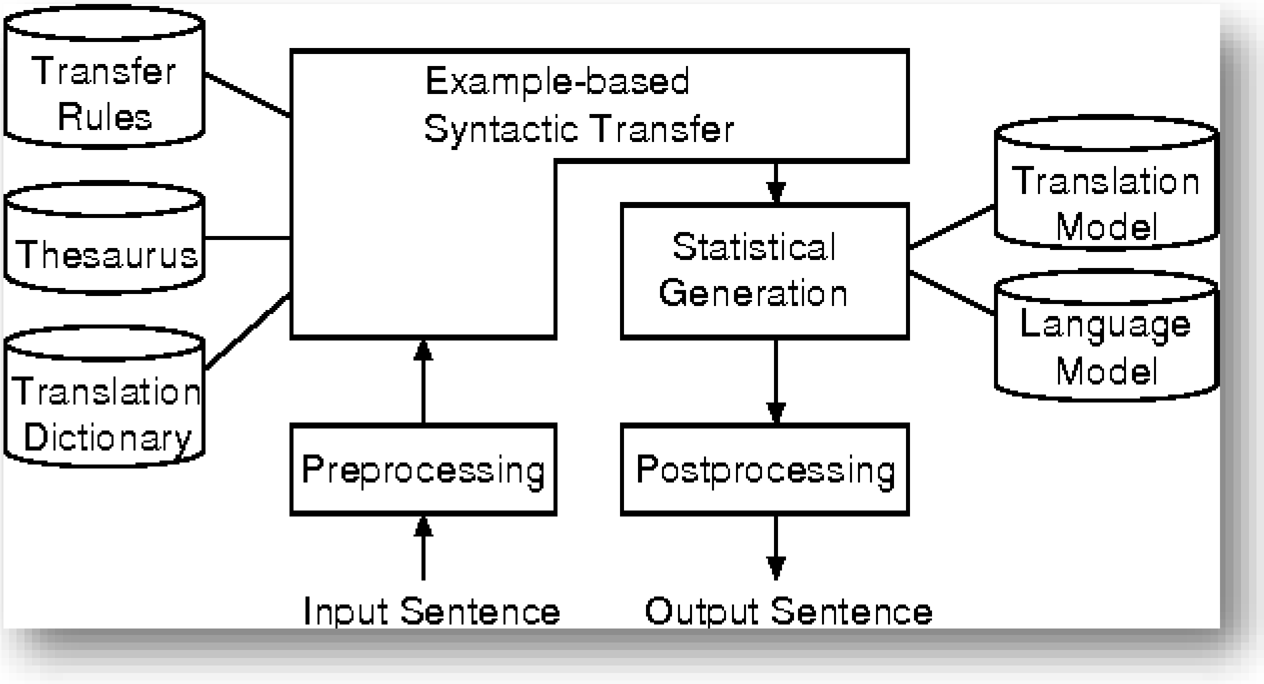


Figure 5 Example based MT architecture [32]

Statistical machine Translation

Warren Weaver proposed using computers to translate text from one natural language to another in 1949, using statistical and cryptanalytic techniques that were emerging from the nascent field of communication theory at the time. Efforts in this direction were quickly abandoned for a variety of philosophical and theoretical reasons, but at a time when the most advanced computers were on par with today's digital watch, any such approach was bound to fail. Anyone with a well-equipped workstation can now apply statistical methods to the study of machine translation and reap the benefits of their findings. Many different ways exist to translate a string of English words, e, into a string of French words. Knowing the larger context in which e occurs can often help to narrow the field of acceptable French translations, but many acceptable translations will still remain; the choice between them is largely a matter of taste. In statistical translation, we believe that every French string, f, can be translated as e. We assign a number Pr (f|e) to each pair of strings (e,f), which we interpret as the likelihood that a translator, when given e, will produce f as his translation [33].

A probability distribution over source-target sentence pairs is provided by a Source Language Model and a Translation Model (S,T). The joint probability Pt (S, T) of the pair (S, T) is the product of the language model's probability Pr (S) and the translation model's conditional probability Pr (T|S). The parameters of these models are automatically estimated from a large database of source-target sentence pairs using a statistical algorithm that optimizes the fit between the models and the data [34]. Figure 1 shows general architecture of Statistical machine translation.

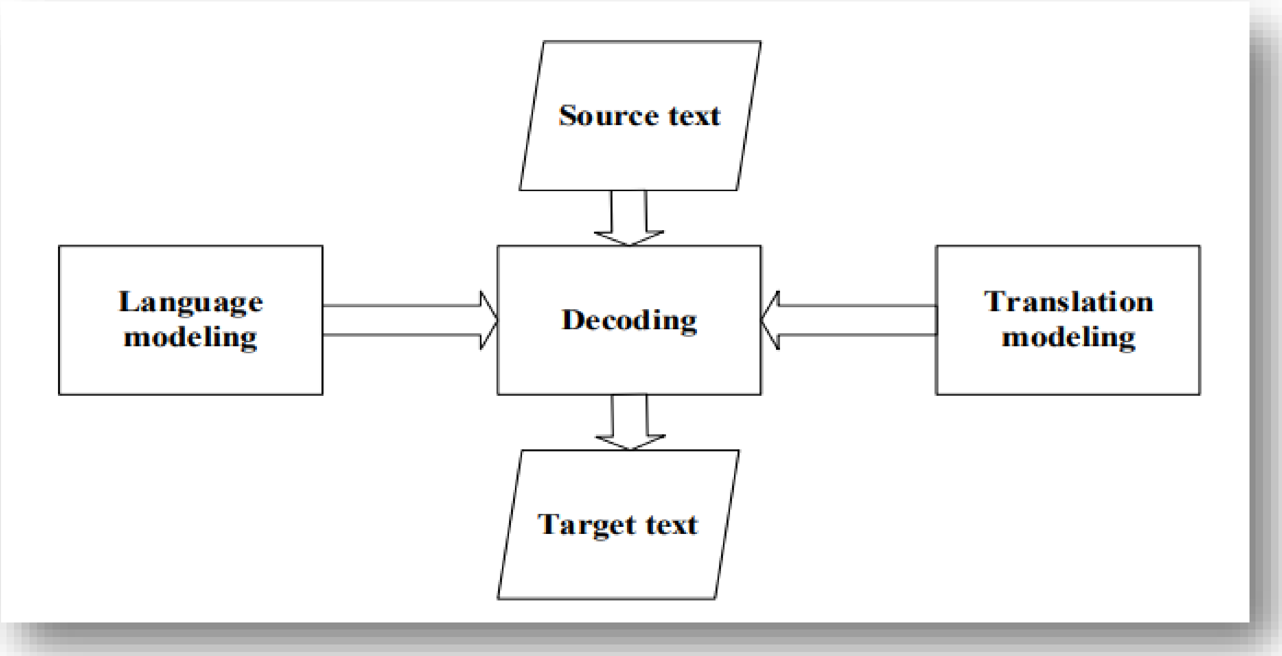


Figure 6 the general architecture of Statistical machine translation

2.5.3. Machine translation using Neural Network

Neural networks were introduced as an advancement tool for SMT to make the computation of the statistical probability assigned to each word in a sequence easier. Following additional research by various researchers, the idea of jointly training and translating data from one natural language to another natural language was modeled for pure neural network machine translation. It has been a long time since neural networks attempted for MT task. However, during the attempt at the early phase the performance was dismal. For many years, relevant machine translation research in neural network has gone overlooked [35].

Deep neural networks were introduced to the machine translation study, which improved existing machine translation systems in a number of ways, particularly in terms of translation quality. One of the main causes of this increase is deep neural networks' capacity to learn a logical representation of words. Deep neural architecture-based machine translation is producing cutting-edge outcomes when translating European languages [36].

2.5.3.1. Neural Machine Translation (NMT)

As a result of advancements in computer and communication technology, a new dataset-based method of machine translation has emerged that maps source and target languages end-to-end. It addresses the flaws of existing machine translation methods. NMT is made up of two neural networks: one encoder and one decoder.) The encoder converts the original sentence into a context vector c, which the decoder decodes to generate the target sentence. When the length of a sentence increases, encoding it into a fixed-length content vector v causes a problem. Incorporating the attention layer with the design can help to solve this problem and provide good performance. It is equal to finding a target sentence that optimizes the conditional probability, that is, arg max P(t|s), according to the probabilistic method. The encoder considers the source sentence S to be a series of vectors S = (x1, x2, x3,...) in vector v, also known as thought [11]. Neural machine translation is a method for automating translation that uses an end-to-end learning mode. Using a neural network of machines, the source text is encoded and decoded. It more closely resembles following a set of established rules. Neural machine translation (NMT), which can also handle traditional idioms and phrase-based content, has substantially improved the translation's quality. An exceptional tool that has a big impact on translation accuracy and precision is the neural machine, a unique discovery and development [37].

2.5.3.2. Different Neural Machine Translation Models

Deep Learning is a relatively new method for machine translation. Unlike traditional machine translation, neural machine translation is a better option for more accurate translation and performance. DNN can be used to improve existing systems and make them more efficient. For the development of a better machine translation system, various deep learning techniques and libraries are required. RNNs, LSTMs, and other neural networks are used to train the system that will convert the sentence from source to target language. Adapting appropriate networks and deep learning strategies is a good choice because it tunes the system toward maximizing the translation system's accuracy in comparison to others [38]. The following are deep learning algorithms used in machine learning applications.

Recurrent neural network (RNN)

Recurrent Neural Networks (RNN) eliminate the need to specify the context's size N and can represent more diverse patterns. Along with input, hidden, and output layers, recurrent matrices allow for time-delayed effects, or short-term memory, by connecting the hidden layer to itself.

Recurrent Neural Networks have recently gained popularity in language modeling tasks, particularly neural machine translation (NMT). Recent NMT models are based on Encoder-Decoder, in which a deep LSTM-based encoder projects the source sentence to a fixed dimensional vector, and then another deep LSTM decodes the vector [39]. Figure 1 shows architecture of RNN algorithm.

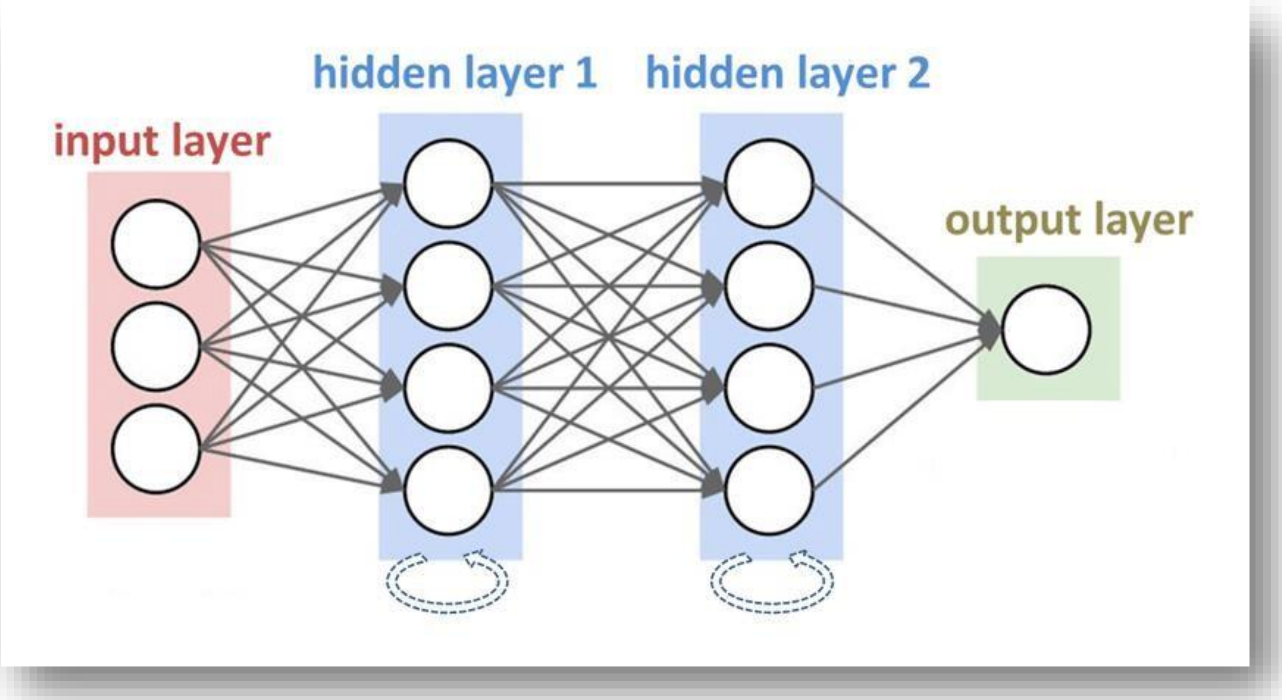


Figure 7 RNN architecture

Long short-term memory (LSTM)

A particular type of recurrent neural network (RNN) architecture called Long Short-Term Memory (LSTM) was created to better precisely simulate temporal sequences and their long-range relationships than standard RNNs [40]. Text is viewed as a sequence of words in RNN-based models, which are designed to capture word relationships and text structures. However, traditional RNN models are ineffective and frequently outperform feed-forward neural networks. The most prevalent RNN architecture is the Long Short-Term Memory (LSTM), which is designed to better capture long-term dependencies. By introducing a memory cell to remember values across arbitrary time periods and three gates (input gate, output gate, forget gate) to manage the flow of information into and out of the cell, LSTM addresses the gradient vanishing or exploding difficulties that vanilla RNNs suffer from. RNNs and LSTM models for TC have been improved by capturing additional information, such as natural language tree structures, long-span word relations in text, document subjects, and so on [41]. The structure of LSTM is shown in Figure 6.

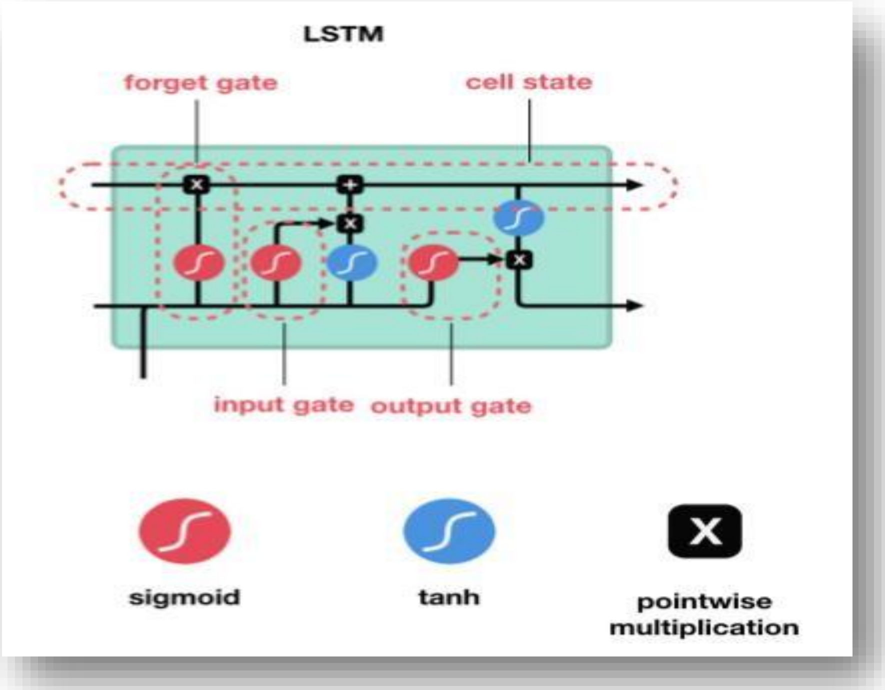


Figure 8 LSTM architecture

Bidirectional long short-term memory (Bi-LSTM)

A bidirectional LSTM is made up of two LSTMs running in parallel, one on the input sequence and the other on the output sequence. The Bidirectional LSTM's hidden state is the concatenation of the forward and backward hidden states at each time step. This configuration allows the hidden state to record both past and future data [42].

The Bi-LSTM neural network is made up of LSTM units that work in both directions to take into account past and future context. Long-term dependencies can be learned using Bi-LSTM without keeping redundant context information. As a result, it has shown to be quite good at solving sequential modeling problems and is commonly used for text categorization. The Bi-LSTM network includes two parallel layers that propagate in two directions using forward and reverse passes to capture interdependence in two contexts, unlike the LSTM network [43]. The structure of Bi-LSTM is shown in Figure 7.

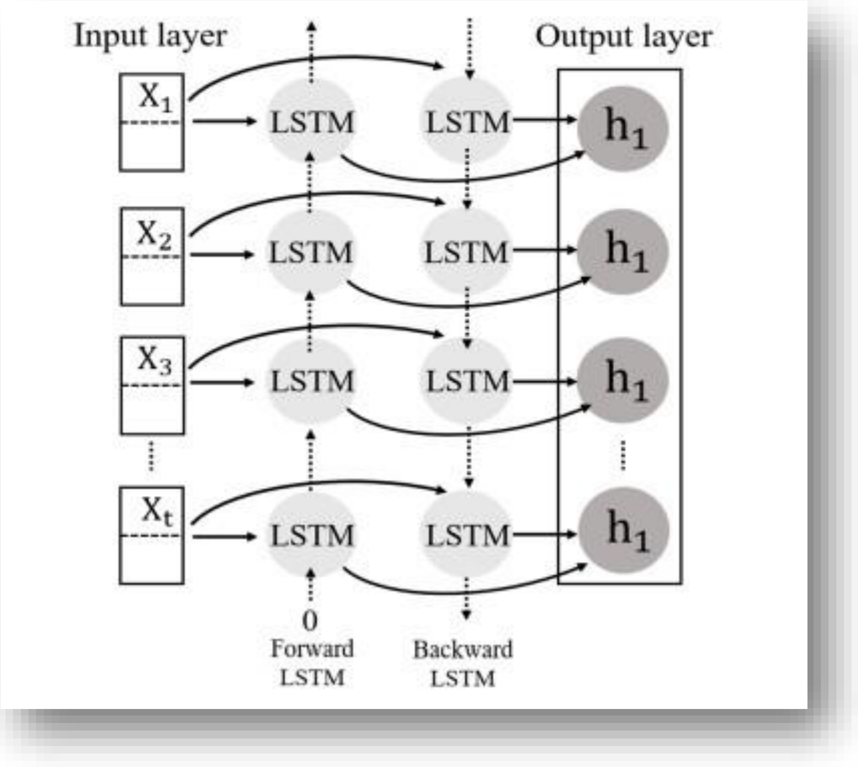


Figure 9 Structure of the Bi-LSTM network

Convolutional Neural Network (CNN)

Layers with convolving filters are applied to local features in convolutional neural networks (CNN). CNN models, which were first developed for computer vision, have now been proven to be useful for NLP, with outstanding results in semantic parsing, search query retrieval, sentence modeling, and other standard NLP tasks [44].

Convolutional Neural Networks (CNNs) are convolution-based networks that use pooling strategies to get their results. The advantages of CNN over others include parameter sharing, sparse interactions, and similar representations. For using the bi-dimensional structure of input data, local connections and shared weights in the network are used instead of completely connected networks. The CNN is the most well-known and widely used algorithm in the field of deep learning. The fundamental advantage of CNN over its predecessors is that it automatically detects significant features without the need for human intervention [45]. Figure 8 shows architecture of a CNN algorithm.

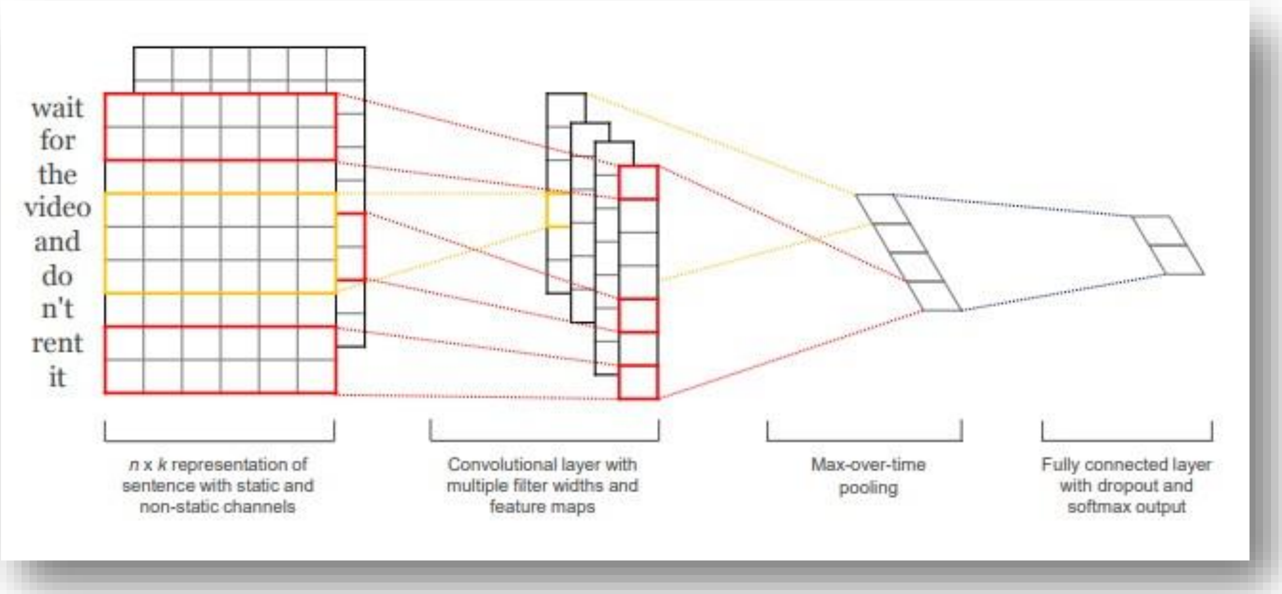

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 10 CNN architecture with two channels [46]

Encoder-Decoder Models

The architecture of encoder-decoder is commonly used in sequence-to-sequence modeling applications. Encoder-decoder neural networks remain the de facto neural network design for state-of-the-art models in machine translation, despite the transition from long short-term memory networks to Transformer networks as well as the introduction and development of attention mechanisms. Sequence-to-sequence modeling is often approached with Neural Networks (NNs), prominently encoder-decoder NNs, nowadays. For the task of Machine Translation (MT), which is by definition also a sequence-to-sequence task, the default choice of NN topology is also an encoder decoder architecture [47]. From a variable-length input sentence, the encoder extracts a fixed-length vector representation, from which the decoder produces an accurate, variable-length target translation. The task of translation can be understood from the perspective of machine learning as learning the conditional distribution p (f | e) of a target sentence (translation) f given a source sentence e. After a model has learned the conditional distribution, it may be used to sample a target phrase given a source sentence directly. This can be done through real sampling or by use an approximate search technique to identify the maximum of the distribution [48].

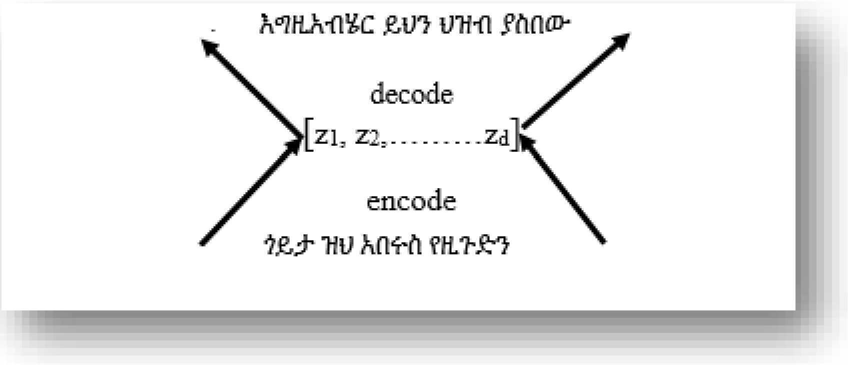


Figure 11 The encoder–decoder architecture [48]

Attention Mechanism

In order to enhance NMT performance, attention techniques were first developed to teach the alignment between source and target tokens. The classic word alignment in SMT, which learns the hard alignment between source and destination tokens, is different from the attention techniques. When creating a target token, attention mechanisms learn to take features from every source token. All of the concealed states of the source tokens are given weights. Larger weights are given to the concealed states that are more connected. Following that, attention mechanisms provide the decoder a context vector c_t that was retrieved from the encoder for target-side predictions [49]. The hidden state is represented with h and set is $\{h_1, h_2, \cdot \cdot \cdot , h_n\}$ in the encoder, where n is the number of source-side tokens. The context vector c_t is computed using

$$c_t = \alpha_t h$$

where α_t is the attention vector at time step t. α_t is a normalized distribution of a score computed by the hidden state set h and the decoder state $s_t{-}1$, as described by Equation 2:

$$\alpha_t = \text{softmax}\ (\text{score}(s_t{-}1, h))$$

The fact that early NMT models frequently provided inadequate translations for lengthy sentences is one issue that has yet to be fully resolved. The fixed-length source sentence encoding is the cause of this flaw. Sentences of various lengths transmit information in different ways. A fixed-length vector cannot therefore adequately capture a long sentence with a complex structure and meaning, even though it is fine for small sentences [50].

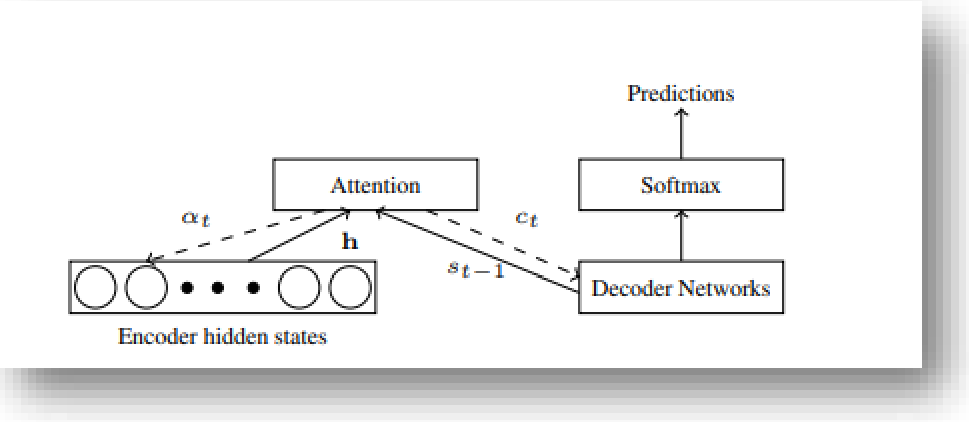


Figure 12 Vanilla attention mechanism [49]

Fully attention-based NMT has recently demonstrated promising performance. In particular, the attention mechanism has operated as a driving force rather than an assistant in text feature extraction. Transformer, which is a completely attention-based paradigm, is one of them. Transformer is a fully attention based NMT model, in contrast to earlier RNN- or CNN-based models. It can be a feature extractor that allows the complete sentence to be "read" and modelled once, meaning it is of self-attention with a feed-forward link. Multiple layers are frequently stacked, which improves the quality of the translation [51]. Figure 11 shows the architecture of the transformer.

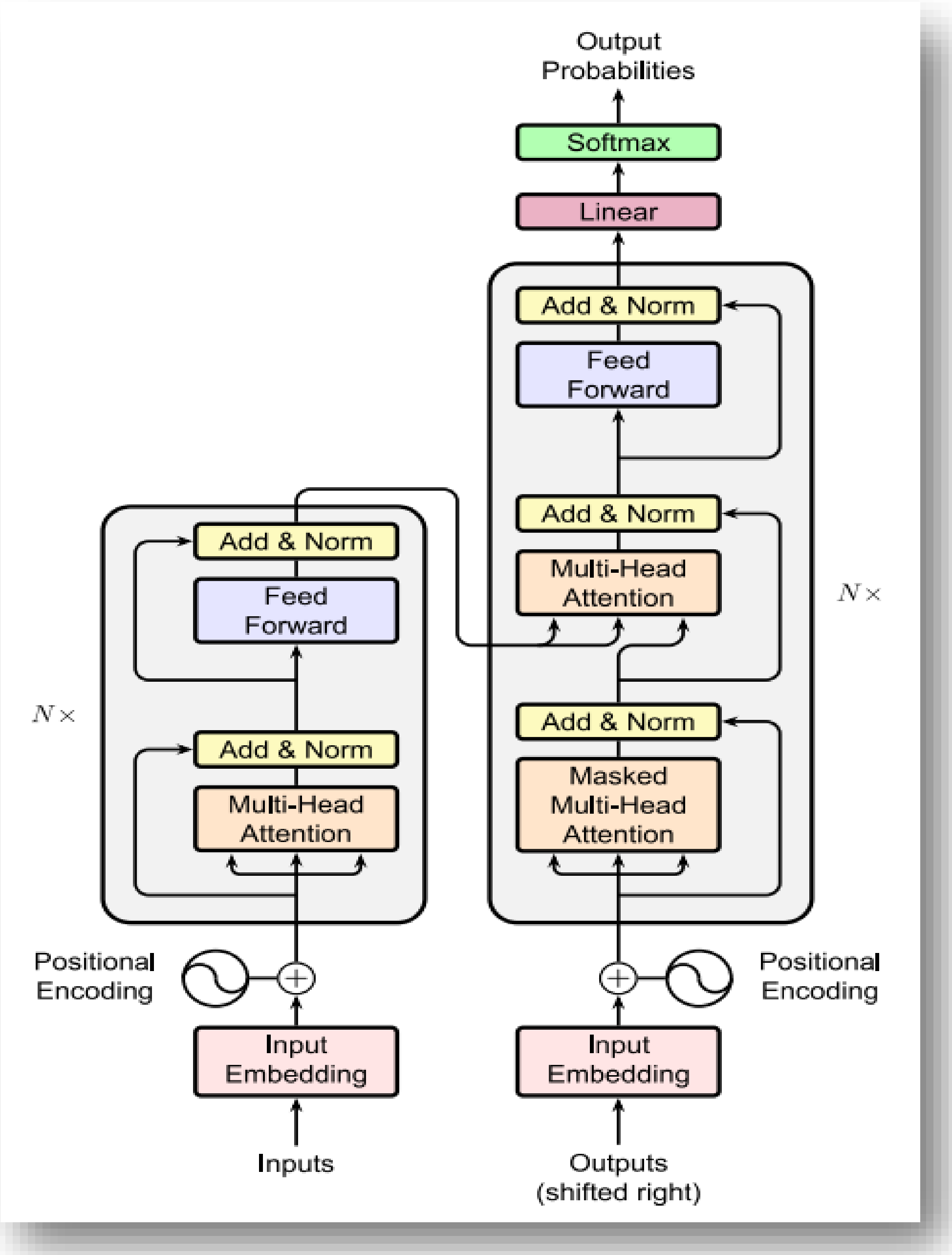


Figure 13 Transformer architecture [52]

2.6. Related work

Machine translation has been studied for both resource rich and under resourced languages using different techniques. Previous researchers has employed rule based and statistical based machine translation techniques for translation of Semitic languages. These techniques uses handcrafted features and are less performance for large size dataset.

The work by [53] proposed a bidirectional Amharic-Kistanigna machine translation using deep learning. The author employed an encoder decoder model for their experiment using a dataset collected from the bible. In the study the recommended for a study to be done on other Gurage languages like the Sebat Bet Gurage. The work of [54] proposed parallel corpora for bi-directional statistical machine translation for seven Ethiopian language pairs. The study employed statistical based machine translation (SMT). However SMT has a limitation of handling long term sentences dependencies. The authors reported that their system performs less for Ethio-Semitic language family. The author did not consider the Sebat Bet Gurage in their study. The authors also recommended ANN modelling as an attractive solution to the problems of machine translation that is the trend of the time.

The study [55] proposed Ge'ez Amharic machine translation using deep learning. The dataset for the study were collected from Bible and religious documents. The study reported that the performance of their system is much lower. This is because the study employed LSTM algorithm which has a drawback of handling long term dependencies that exist in a text. The dataset also makes the model limited to the domain where the dataset is collected. The study [56] proposed a morpheme based Ge'ez Amharic machine translation. The study used SMT which is based on morpheme and word level translation as a technique. However SMT approach may disregard the extended dependency that exists beyond the length of phrases and new words.

Studies also show that Neural Machine Translation (NMT) are better as compared to SMT systems. NMT is an end-to-end learning strategy for automated translation that has the potential to solve many of the flaws of traditional phrase-based translation systems. Unfortunately, both in training and translation inference, NMT systems are known to be computationally expensive – often excessively so in the case of very big data sets and complex models. NMT systems have also been accused of being unreliable, particularly when input sentences contain unusual terms. NMT's application in actual deployments and services, where both accuracy and speed are critical, has been hampered by these limitations [57].

SMT-based English-Amharic machine translation has been proposed in the study [58]. The study demonstrates the complexity of the task involved in creating machine translation (MT) for Amharic, which is regarded as one of the NLP scarce resource languages, using a rule-based approach. It is challenging for languages with limited resources because rule-based machine translation (MT) heavily utilizes integrated linguistic knowledge, rules, and resources of both the source and target languages. Linguistic knowledge of the source and target languages includes morphological, syntactic, semantic, lexical, and parsing information. The linguistic rules comprise rules for generating, analyzing, and transferring the source and/or target languages, including syntactic, semantic, and lexical rules.

In the work of [54], a machine translation from Amharic to English was proposed. Our literature review revealed that the linguistic properties of the target languages have a major influence on translation for SMT. The writing system, word ordering, and morphological complexity are just a few of the challenges. The Amharic, Tigrigna, and Ge'ez languages use the Ge'ez writing system, which uses different characters for words that have the same meaning. For instance, the word peace can be written as □□□ or □□□. The performance of SMT is directly impacted by these character differences in probability values.

From the work of [59] we observed that, one of the problems of developing effective MT systems is quantifying quality improvements. For example, an MT system that is trained and tested on religious texts may appear to have good outcomes but does not necessarily imply quality improvement. This could be due to (i) data leakage (similar/same words in both the train and the test) or (ii) domain mismatch (different domains in both the train and the test). To establish the quality of MT systems, a robust and dependable, diversified standard is required.

Based on the literature we reviewed, there is no previous study conducted on Sebat Bet Gurage (Chaha)-Amharic machine translation. In this study we developed Sebat Bet Gurage (Chaha)-Amharic machine translation model using deep learning techniques.

CHAPTER THREE

3 SYSTEM DESIGN AND IMPLEMENTATION

3.1. Introduction

The detailed architectural design and implementation of the Sebat Bet Gurage (Chaha) Amharic translation model is described in this chapter. This chapter has designed and discussed the suggested architecture. We collected the Sebat Bet Gurage (Chaha) dataset from churches. In addition to these the researcher has collected freely available Amharic dataset and translated to Sebat Bet Gurage (Chaha) language. Figure 14 shows the overall architecture of the proposed model.

3.2. System architecture

In this study we used the encoder decoder machine translation approaches. Figure 13 shows the model architecture we followed in our study.

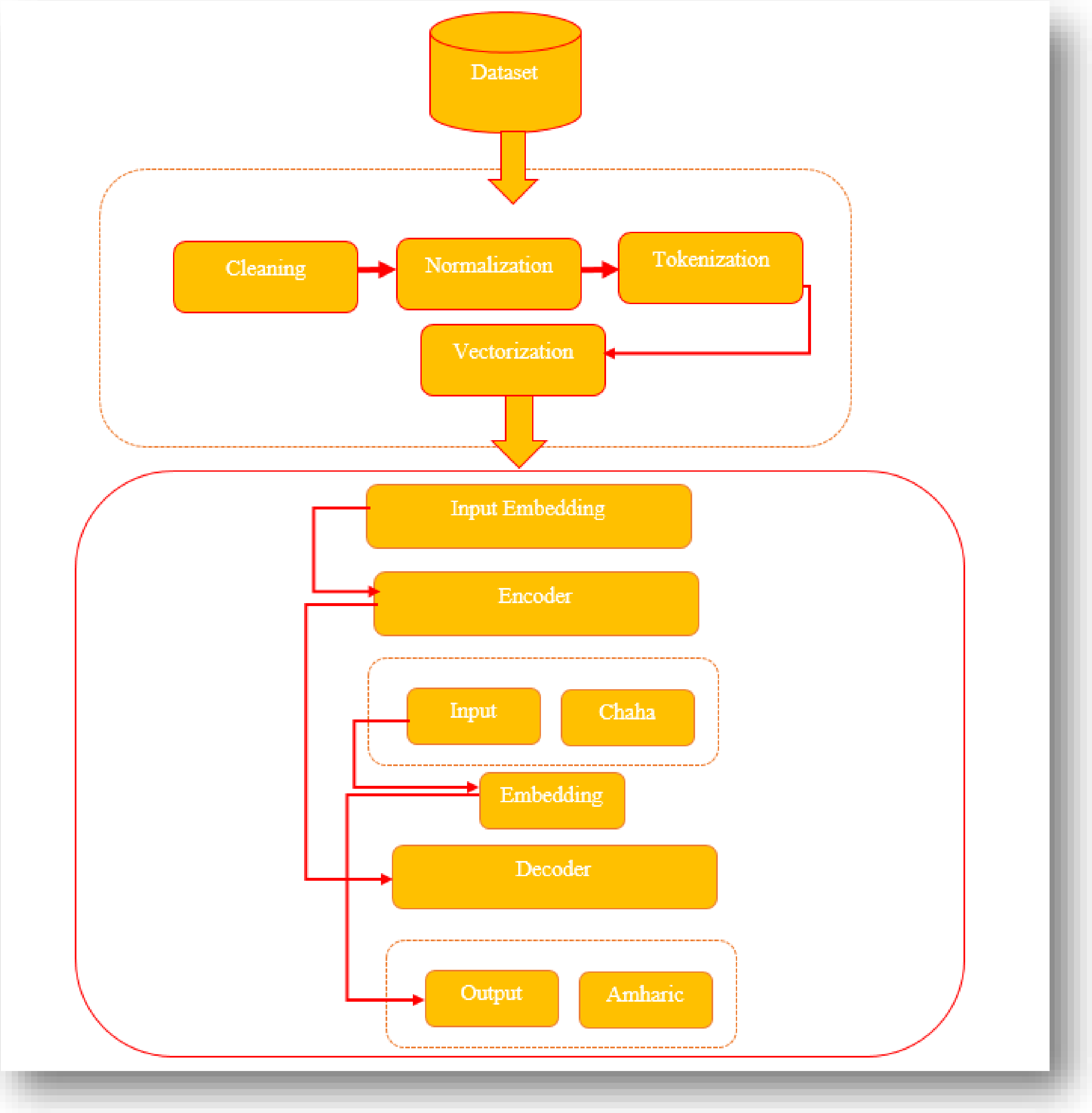


Figure 14 system architecture

3.2.1. Dataset collection

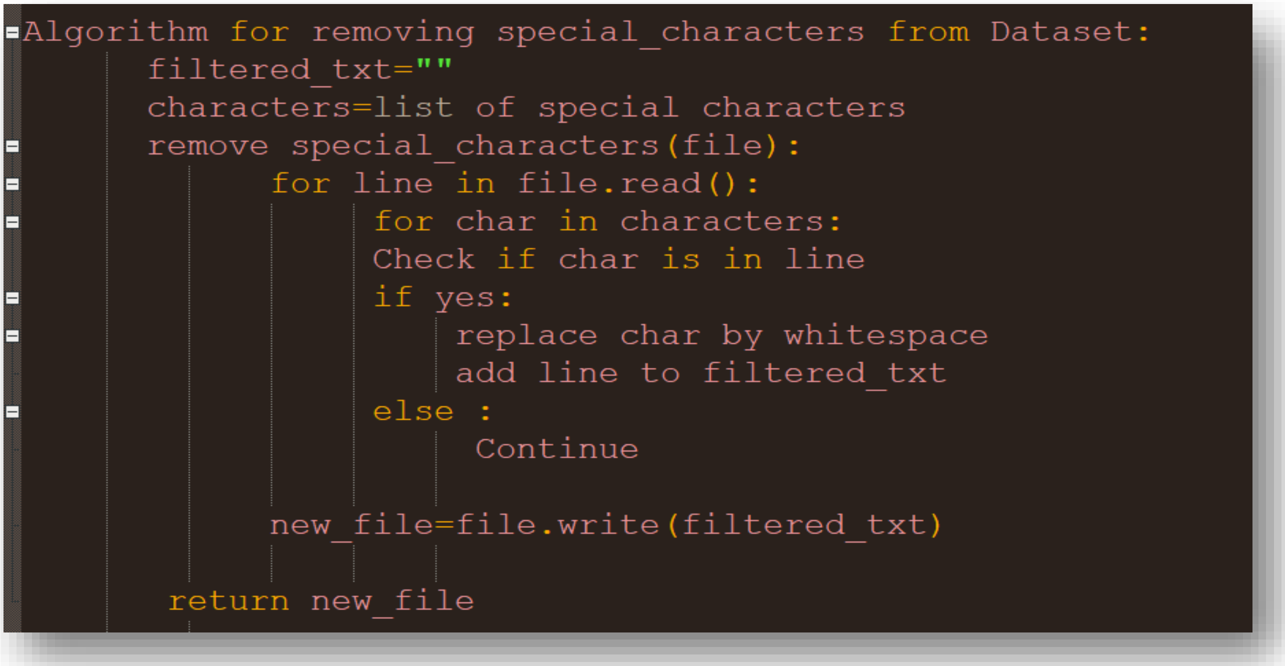
The Sebat Bet Gurage (Chaha) dataset is collected from different religious documents like bible and related documents. The dataset includes two text files: one containing Sebat Bet Gurage (Chaha) sentences and the other an Amharic dataset containing Amharic sentences of the corresponding Sebat Bet Gurage sentences in the Sebat Bet Gurage dataset. In addition to this the researcher has translated freely available Amharic dataset to the Sebat Bet Gurage language. This enables the model not to be domain specific.

3.2.2. Dataset preprocessing

The preprocessing phase allows us to prepare our dataset for our model's subsequent processing tasks. Text preprocessing is a phase in the natural language processing process that converts a text into a machine-readable format, making it easier for learning algorithms to work. There are various approaches to pre-process data, such as Cleaning, which deals with removing stop words, punctuations, and so on. Labeling, in which sentences are placed in parallel with their Amharic translation, standardization, in which text is prearranged for easy representation, and extraction of valuable features are all important for a specific task. In the study, we have performed the following preprocessing tasks.

Removal of special characters

In this stage, we removed special characters. We removed these characters, because it becomes noise later for the model during training. The algorithm we used to remove these characters is shown below in Figure 15.

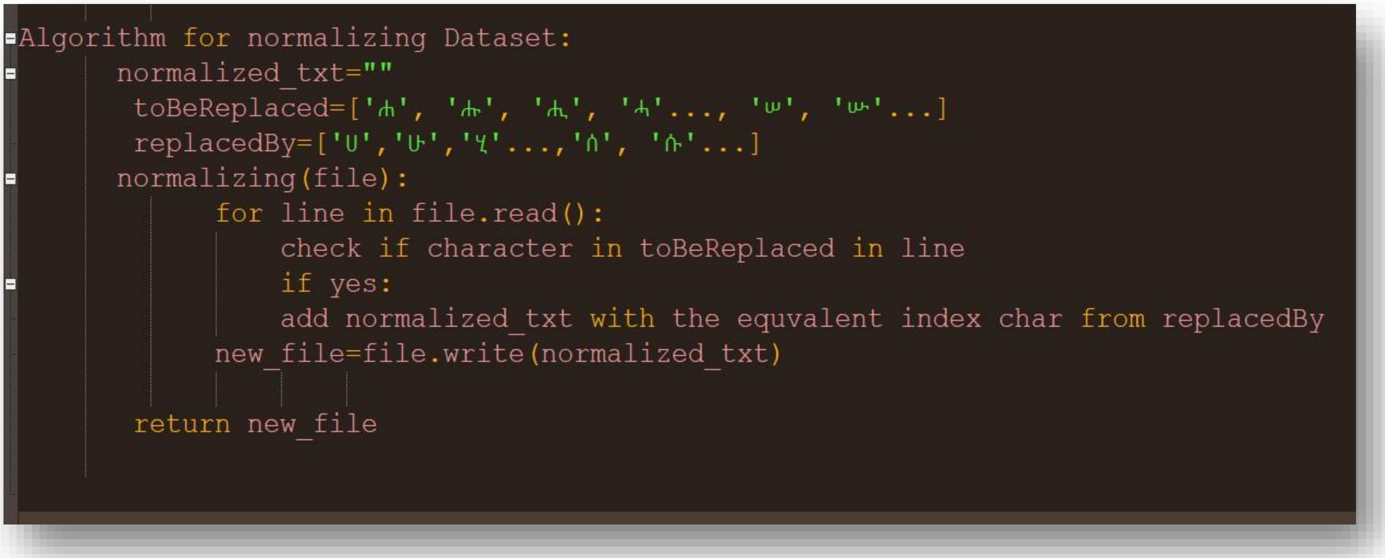


```
Algorithm for removing special_characters from Dataset:
      filtered_txt=""
      characters=list of special characters
      remove special_characters(file):
            for line in file.read():
                 for char in characters:
                 Check if char is in line
                 if yes:
                     replace char by whitespace
                     add line to filtered_txt
                 else :
                      Continue

            new_file=file.write(filtered_txt)

       return new_file
```

Figure 15 Algorithm to remove special characters

Normalization

Normalization helps preparing dataset by transforming the words to a standard format. For example, in both languages “□” and “□”, “□” and “□”, converting all words to a single representation will enable text representation techniques to give similar representation for similar words. The algorithm we used to normalize the dataset is shown below in Figure 16.



```
Algorithm for normalizing Dataset:
      normalized_txt=""
       toBeReplaced=['ሐ', 'ሑ', 'ሒ', 'ሓ'..., 'ሠ', 'ሡ'...]
       replacedBy=['ሀ','ሁ','ሂ'...,'ሰ', 'ሱ'...]
      normalizing(file):
            for line in file.read():
                check if character in toBeReplaced in line
                if yes:
                add normalized_txt with the equivalent index char from replacedBy
            new_file=file.write(normalized_txt)

       return new_file
```

Figure 16 Algorithm to normalize the dataset

Tokenization

Tokenization is a lexical analysis technique that is used to break sentences into their component tokens. We determined the vocabulary size, the maximum length of sequences, and the representation of words with unique numbers when we performed tokenization. In order to assign a distinct number to each word, we accessed and browsed through the entire parallel dataset of data. As a result, once a particular word is viewed, the process of giving it a unique number starts, and the word is then represented by the unique number. The last number is given to the term that has been accessed the most recently, depending on the size of each language's lexicon. By transforming the original dataset to integer number representation form, these actions make it easier to prepare data for training. Tokenized data is the data that has been translated to integer form. As a result, the vocabulary size is set for data pre-processing and the row data is tokenized. Based on the vocabulary's size, each word is represented by an integer. The index of the first vocabulary is zero, while the index of the last vocabulary is size of vocabulary-1 for each language.

“□□□ □□ □□□□ □□□□□” tokenized as ‘□□□’, ‘□□’, ‘□□□□’, ‘□□□□□’. The Figure 17 below shows sample of the tokenization stage of dataset preprocessing for Sebat Bet Gurage (Chaha) sentences.

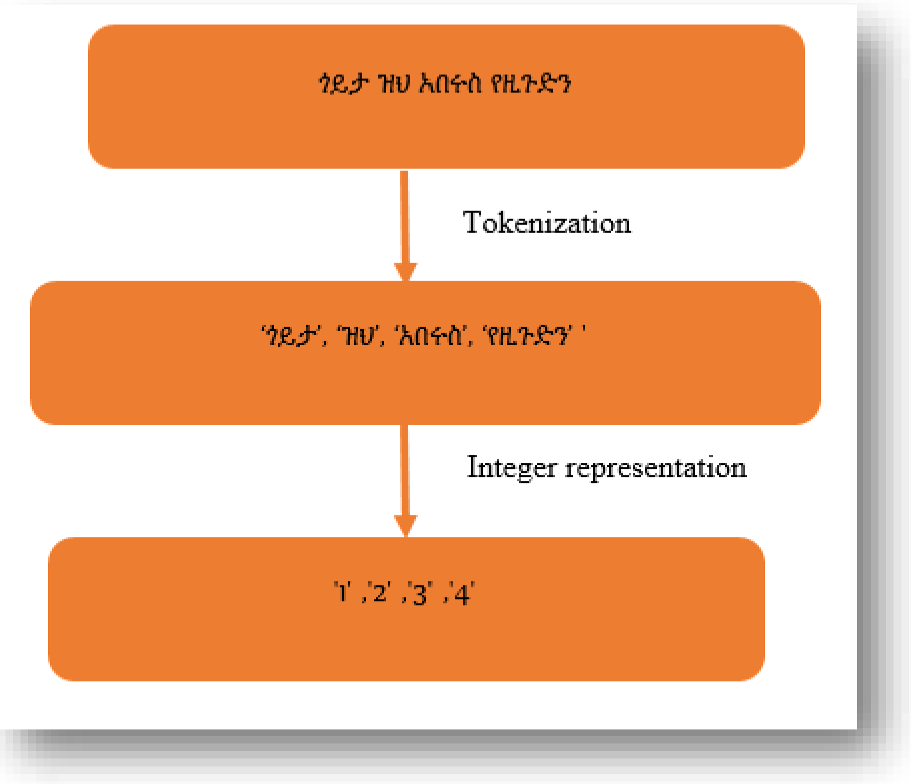


Figure 17 Tokenization of Sebat Bet Gurage sentence

Similarly the corresponding Amharic sentences is also tokenized the same way. “□□□□□□□□ □□□ □□□ □□□□” is tokenized as '□□□□□□□□ ', '□□□', '□□□', '□□□□'. However, in order to know the boundary of end of the sentences we used <sam> to indicate the beginning of a new sentence and <eam> to indicate the end of the sentences like ‘<sam>’, '□□□□□□□□' ,'□□□ ','□□□ ','□□□□'
<eam>’. The Figure 18 below shows sample of the tokenization stage of dataset preprocessing for Amharic sentences.

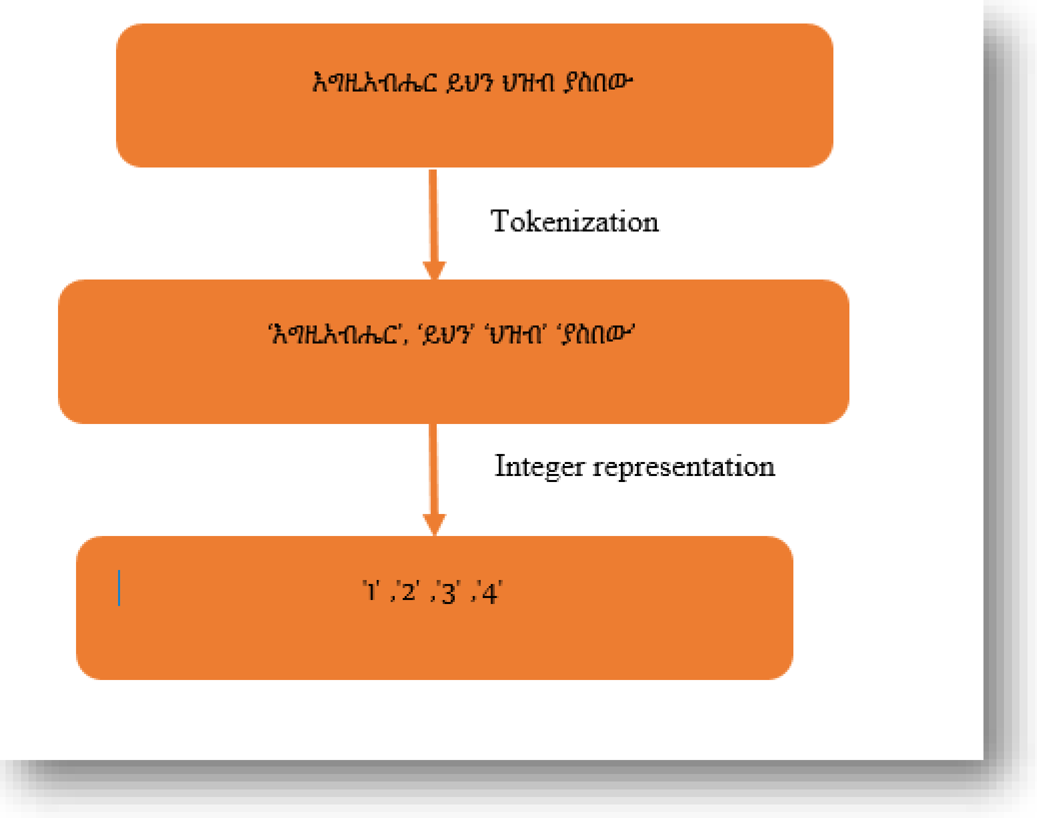
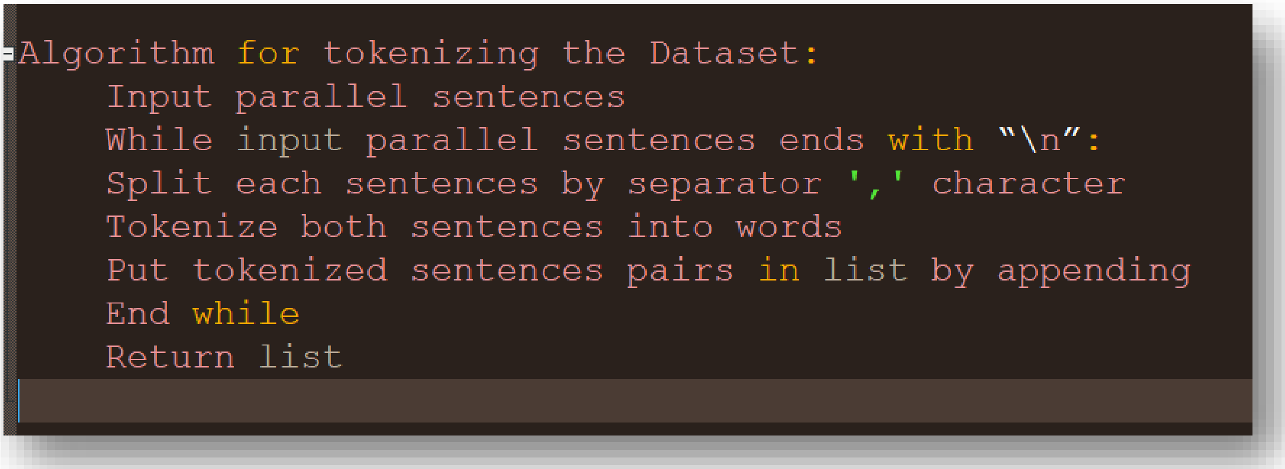


Figure 18 Tokenization of Amharic sentences

The algorithm we used to tokenize the dataset is shown below in Figure 19.



```
Algorithm for tokenizing the Dataset:
    Input parallel sentences
    While input parallel sentences ends with "\n”:
    Split each sentences by separator ',' character
    Tokenize both sentences into words
    Put tokenized sentences pairs in list by appending
    End while
    Return list
```

Figure 19 Algorithm we used to tokenize the dataset

One hot representation

In the tokenization stage the dataset is changed in to integer representation. However the integer representation cannot be directly input for the translation model. This is because the neural network does not directly operate on the vocabulary represented with integer. In order to operate the vocabulary by the neural network it should be changed in to vector representation, which is called one hot vector representation. In this stage the integer represented data is changed in to two-dimensional vector which is, one-hot vector representation. The one hot vector representation uses unique vector representation for each word of the sentences.

3.2.3. Model training

After data preprocessing, the researcher then padded the end of the tokenized parallel sequences with zeros, and batch the complete set of sequences into a single array. Using the parallel dataset preprocessed and padded in the above steps, the researcher used the encoder and decoder architecture to train the translation model.

3.2.3.1. Encoder decoder model

Modern machine learning encoder-decoder architectures can excel at sequence-to-sequence tasks like machine translation, language modeling, and speech-to-text translation [60]. Encoding is the process of converting data into the required format. In the context of this study, we convert a sequence of Sebat Bet Gurage (Chaha) words into a two-dimensional vector, which is also known as the hidden state. The researcher first defined the embedding layer which is the first layer to be defined. The embedding layer takes as input the shape of the input and shape of output. The input shape has a size equal to maximum length of the sentences and the output shape is the shape of the embedding vector. We then used an encoder that takes the Sebat Bet Gurage (Chaha) sentence as input and outputs a hidden vector.

In the study we have experimented an encoder using LSTM, Bi-LSTM and GRU models.

LSTM model

In this case the encoder is made up of an Embedding layer, which converts the words into vectors, and a long short-term memory (LSTM), which calculates the hidden state. The network is first provided the source embedding in a sequential manner. To learn how the steps in the input sequence are related to one another and create an internal representation of these relationships, we require an encoding level. To implement the encoder model, we used a single LSTM layer. This model generates a fixed-size vector as its output, which represents the internal representation of the input sequence. The length of this fixed-sized vector is determined by the number of memory cells in this layer. The learnt internal representation of the input sequence must be converted by the decoder into the right output sequence. We utilized a RepeatVector layer to link the encoder and the decoder. This is due to an error that arises from linking an encoder layer to a decoder layer directly. We cannot connect the encoder and decoder directly because the encoder will generate a 2-dimensional matrix of outputs and the decoder layer will require a 3-dimensional input in order to provide a decoded sequence with a varied length determined by the challenge. We used the RepeatVector layer that simply generates a 3D output by repeatedly iterating the specified 2D input. The encoder and decoder components of the network are fitted together using the RepeatVector layer as an adapter. We used one LSTM layer to implement the decoder model once more. The context vector and current predicted output are then accepted by the decoder model in order to predict the next output. In the first step there is no current predicted output. As a result, we used the start of the sentence, in our case <sam>. The decoder model predicts the next output until the end of the sequence, in our case <eam> to the end of each sentence. Next a dense layer is used as the output for the network. The same weights is used to output each time step in the output sequence by wrapping the dense layer in a TimeDistributed wrapper. Figures below shows the encoder and decoder architecture using the LSTM algorithm described here. The encoder architecture used in this case is shown in Figure 20.

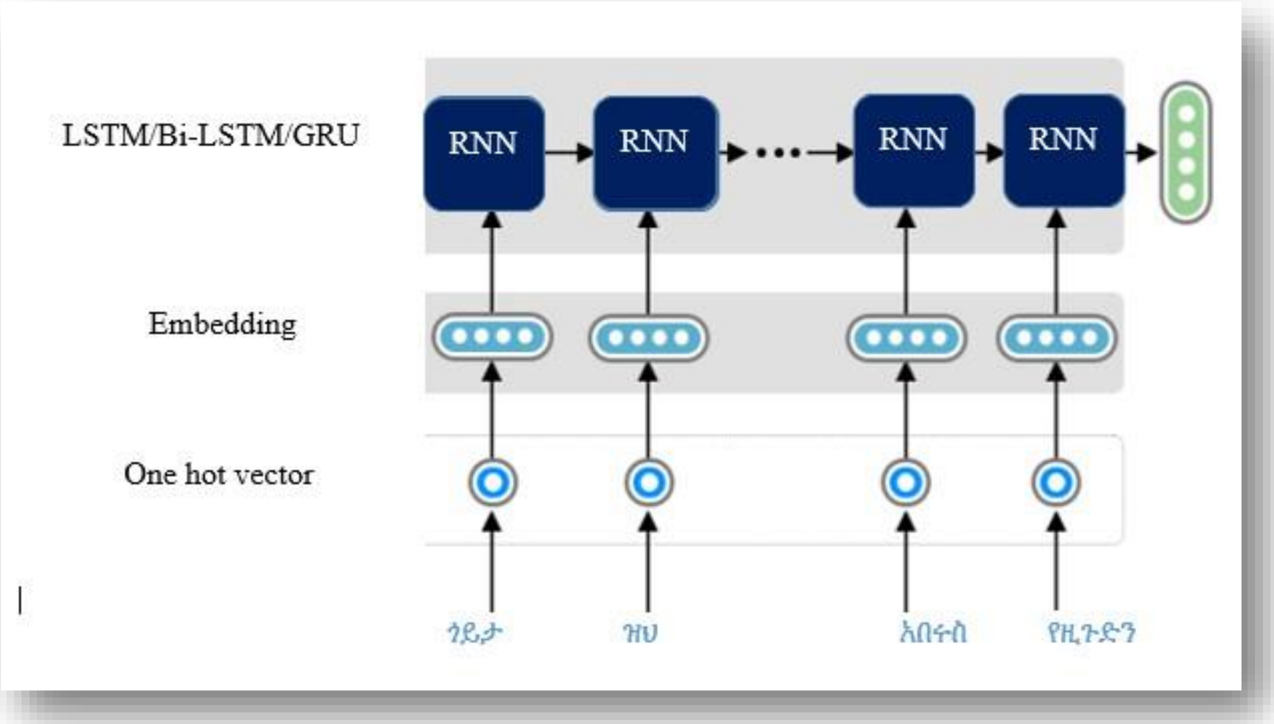


Figure 20 the encoder architecture

Figure 21 below shows the decoder architecture described above.

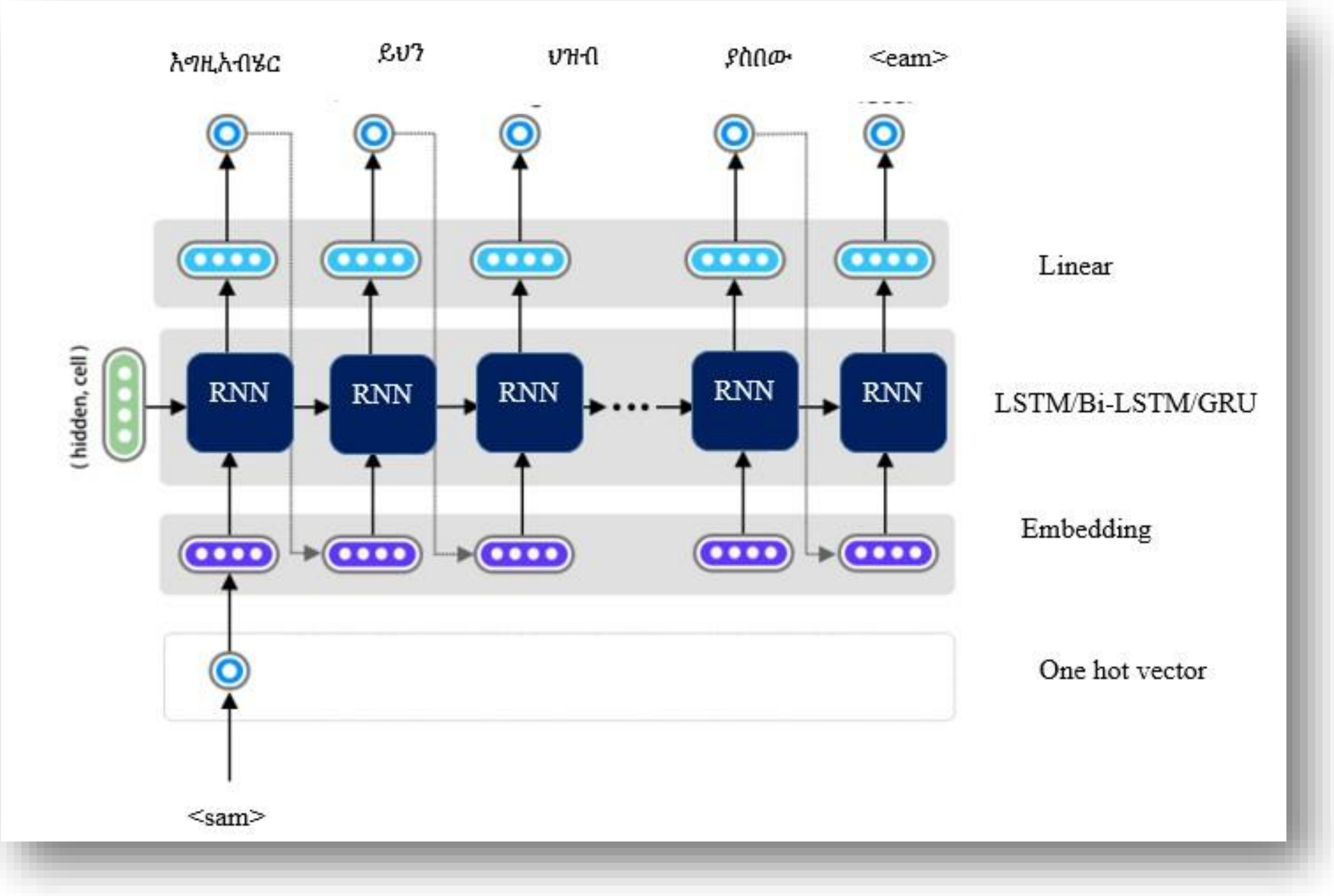
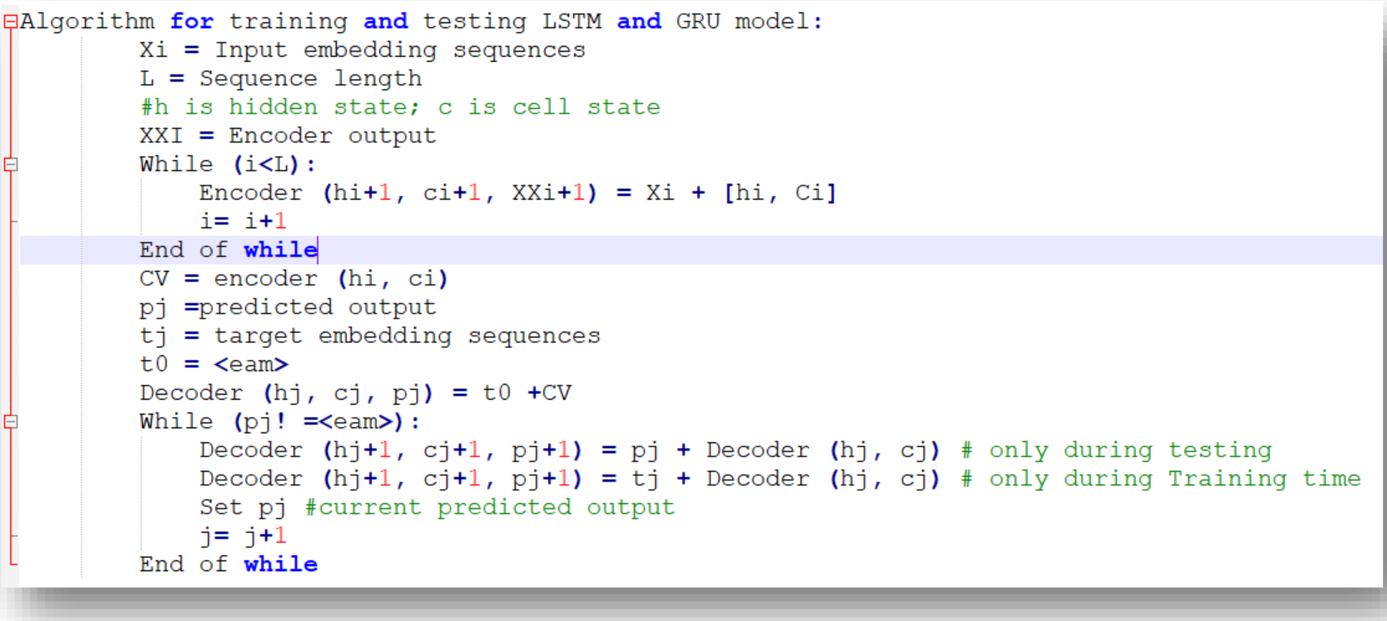


Figure 21 the decoder architecture

The algorithm we used for training and testing an encoder decoder model using the LSTM and GRU model is shown below in Figure 22.



```
Algorithm for training and testing LSTM and GRU model:
        Xi = Input embedding sequences
        L = Sequence length
        #h is hidden state; c is cell state
        XXI = Encoder output
        While (i<L):
            Encoder (hi+1, ci+1, XXi+1) = Xi + [hi, Ci]
            i= i+1
        End of while
        CV = encoder (hi, ci)
        pj =predicted output
        tj = target embedding sequences
        t0 = <eam>
        Decoder (hj, cj, pj) = t0 +CV
        While (pj! =<eam>):
            Decoder (hj+1, cj+1, pj+1) = pj + Decoder (hj, cj) # only during testing
            Decoder (hj+1, cj+1, pj+1) = tj + Decoder (hj, cj) # only during Training time
            Set pj #current predicted output
            j= j+1
        End of while
```

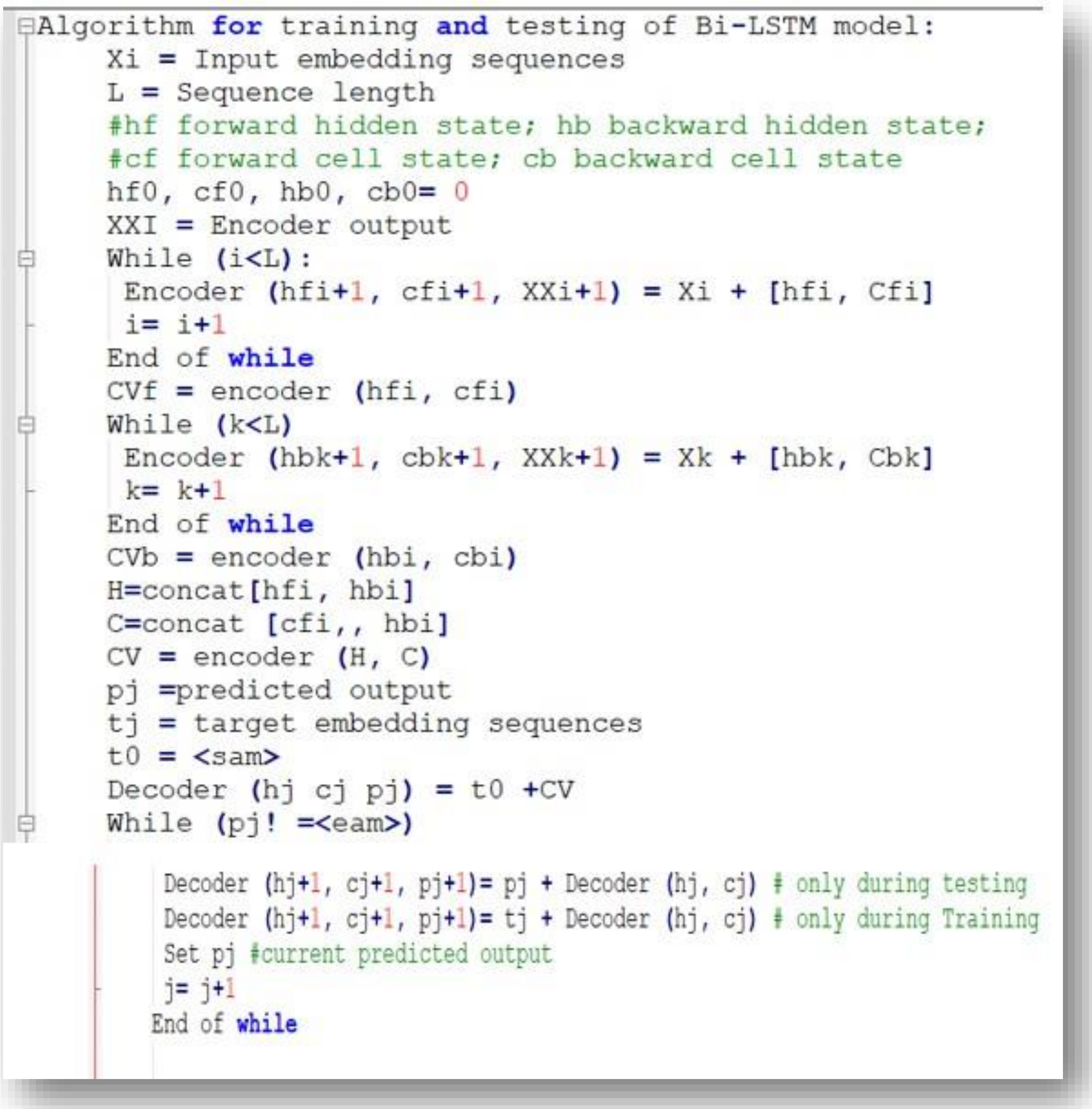
Figure 22 Algorithm for training and testing an encoder decoder model using the LSTM and GRU model

GRU model

The work flow of the GRU is similar with the LSTM model discussed above. As in LSTM model the encoder is made up of an Embedding layer, which converts the words into vectors, and a gated recurrent unit (GRU), which calculates the hidden state. The network is first provided the source embedding in a sequential manner. All the other operations are similar with the operation of LSTM model above.

Bi-LSTM model

The Bi-LSTM model follows the same workflow as the LSTM model discussed above. When the length of the input sequence is long, we merely add backward internal state (backward hidden state and backward cell state) to check for and give emphasize for missing information. The encoder using Bi-LSTM algorithm is provided the source embedding. All the other operation of encoder and decoder using the Bi-LSTM algorithm is similar with the encoder decoder model using the LSTM algorithm. The only difference is that this approach consists of a Bi-LSTM layer for both encoding and decoding layers, RepeatVector layer, and time distributed layer. Figure 22 below shows the algorithm used to show how the training and testing of an encoder decoder model using a Bi-LSTM algorithm works.



```
Algorithm for training and testing of Bi-LSTM model:
    Xi = Input embedding sequences
    L = Sequence length
    #hf forward hidden state; hb backward hidden state;
    #cf forward cell state; cb backward cell state
    hf0, cf0, hb0, cb0= 0
    XXI = Encoder output
    While (i<L):
     Encoder (hfi+1, cfi+1, XXi+1) = Xi + [hfi, Cfi]
     i= i+1
    End of while
    CVf = encoder (hfi, cfi)
    While (k<L)
     Encoder (hbk+1, cbk+1, XXk+1) = Xk + [hbk, Cbk]
     k= k+1
    End of while
    CVb = encoder (hbi, cbi)
    H=concat[hfi, hbi]
    C=concat [cfi,, hbi]
    CV = encoder (H, C)
    pj =predicted output
    tj = target embedding sequences
    t0 = <sam>
    Decoder (hj cj pj) = t0 +CV
    While (pj! =<eam>)
        Decoder (hj+1, cj+1, pj+1)= pj + Decoder (hj, cj) # only during testing
        Decoder (hj+1, cj+1, pj+1)= tj + Decoder (hj, cj) # only during Training
        Set pj #current predicted output
        j= j+1
       End of while
```

Figure 23 Algorithm for training and testing an encoder decoder model using the Bi-LSTM model

Figure 23 shows a plot of the encoder decoder based model using Bi-LSTM algorithm.

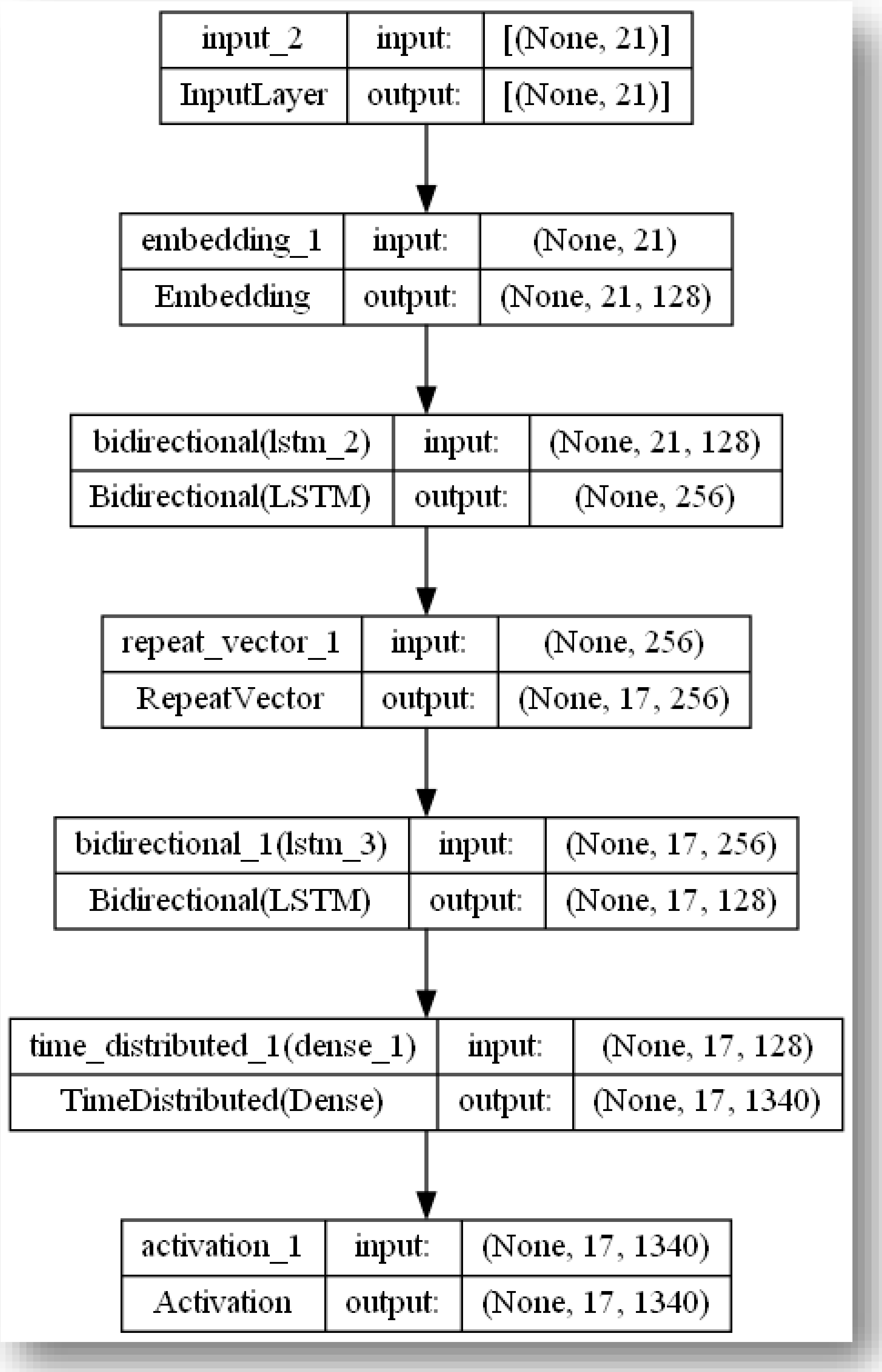

input_2 | input: | [(None, 21)]
InputLayer | output: | [(None, 21)]
embedding_1 | input: | (None, 21)
Embedding | output: | (None, 21, 128)
bidirectional(lstm_2) | input: | (None, 21, 128)
Bidirectional(LSTM) | output: | (None, 256)
repeat_vector_1 | input: | (None, 256)
RepeatVector | output: | (None, 17, 256)
bidirectional_1(lstm_3) | input: | (None, 17, 256)
Bidirectional(LSTM) | output: | (None, 17, 128)
time_distributed_1(dense_1) | input: | (None, 17, 128)
TimeDistributed(Dense) | output: | (None, 17, 1340)
activation_1 | input: | (None, 17, 1340)
Activation | output: | (None, 17, 1340)

Figure 24 plot of the encoder decoder model using Bi-LSTM algorithm

3.2.4. Model prediction

The model prediction phase is used to evaluate the corresponding Amharic text of new Sebat Bet Gurage (Chaha) text that has not been used in training and testing phases of the Sebat Bet Gurage (Chaha)-Amharic translation model. As discussed in above stages, the objective of this research is developing Sebat Bet Gurage (Chaha)-Amharic translation model that has a capability giving Amharic translation of Sebat Bet Gurage (Chaha) text. The researcher test this capability of the model in the prediction phase. To predict the Amharic translation of Sebat Bet Gurage (Chaha) text, the text is first preprocessed and feed to the model. The model then displays the Amharic translation of the give Sebat Bet Gurage (Chaha) text. The model prediction process is plotted in Figure 27.

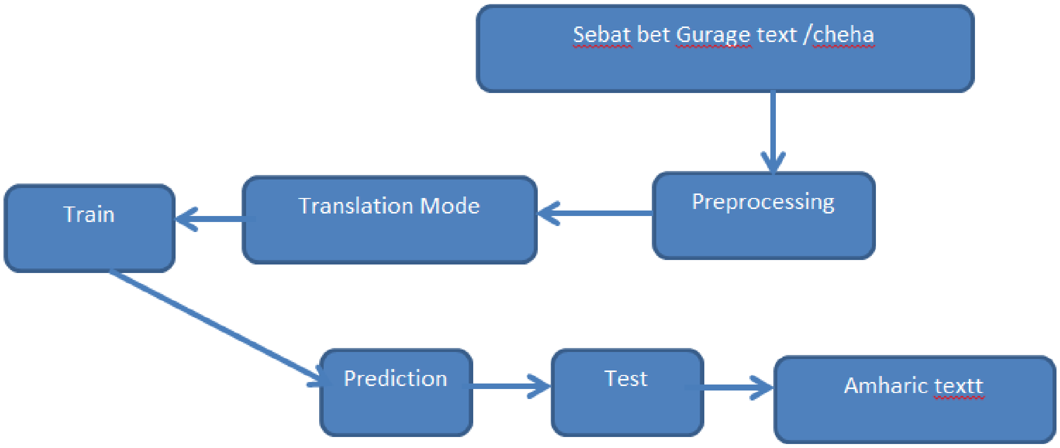


Figure 25 model prediction phase

3.2.5. Model evaluation

To assess the performance of our study, we used the Bilingual Evaluation Understudy (BLEU) score, which can automatically evaluate the machine translation system. The BLEU score is an algorithm that evaluates the quality of text that has been translated from source to target language. The text's quality is determined by comparing the results of machine translation to those of human translation. If the machine translated text is closely related to the human translated text, the result is considered to be of higher quality, and the prototype is effective in machine translation.

3.2.6. Development environments

Python

Python is a high-level, interpreted, dynamically semantic programming language. Because of its high-level built-in data structures, dynamic typing, and dynamic binding, it's perfect for scripting or as a language to connect existing components while doing Rapid Application Development. Python's short, simple-to-learn syntax encourages readability, which reduces the cost of software maintenance. Python supports modules and packages, which makes code reuse and program modularity easier. We wrote our software in the Python programming language.

Jupyter notebook

The Jupyter Notebook App is a web-based server-client application for editing and running notebook documents. The Jupyter Notebook App can be run locally on a computer without internet access (as explained in this paper) or remotely on a server and accessible via the internet. We used jupyter notebook to write our python code.

Keras

Keras is a high-level neural network library written in Python that works with TensorFlow or Theano. The intention behind its creation was to facilitate rapid experimentation. When conducting research, it is imperative to be able to move swiftly from idea to outcome. To train our deep learning model, we used Keras.

Anaconda

Aimed for streamlining package management and deployment in scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, and so on), Anaconda is a Python and R programming language distribution. The bundle includes data-science packages for Windows, Linux, and macOS. This platform was utilized by the researcher to create the suggested model.

Matplotlib

Matplotlib is a plotting library available as a component of NumPy, a big data numerical handling resource, for the Python programming language. Matplotlib embeds plots in Python applications using an object-oriented API. We used this library to visualize the performance of the model during different experiments. We visualize the accuracy and loss of the model using different algorithms.

TensorFlow

TensorFlow is a complete open source machine learning platform. It has a comprehensive, adaptable ecosystem of tools, libraries, and community resources that enable researchers to push the boundaries of ML and developers to easily build and deploy ML-powered applications. TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embeddings, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE (partial differential equation)-based simulations, competing with frameworks such as PyTorch and Apache MXNet. Best of all, TensorFlow can predict production at scale using the same models that were used for training.

CHAPTER FOUR

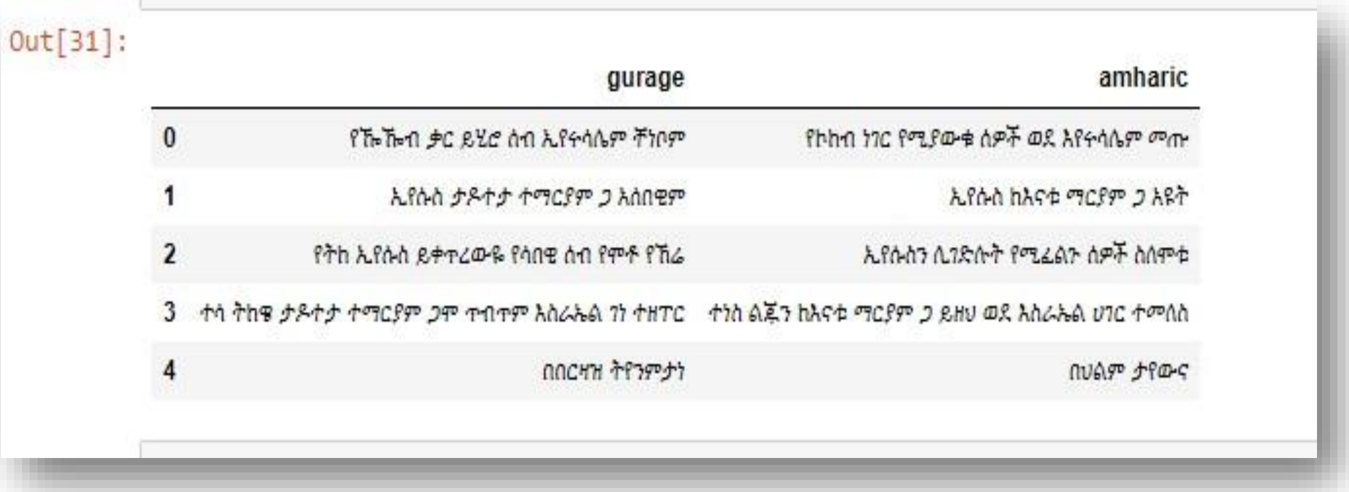
4. EXPERIMENTATION AND RESULTS

4.1. Introduction

In this we have discussed the experimental results by showing experimental setups and performance of testing results of the proposed model using BLEU score metrics. And we have compared the results of the experimental systems between the language pair. In this chapter we have followed the following steps.

4.2. Dataset Collection and Preparation

In the study we have collected 5200 parallel sentences of Amharic–Chaha corpus. The corpus, we prepared is from religious book (Saint Matiwos, and Saint Lukas wongel). In addition to this we have translated freely available Amharic corpus to the Chaha language.



Out[31]:

	gurage	amharic
0	የኾኾብ ቃር ይሂሮ ሰብ ኢየሩሳሌም ቾነቦም	የኮከብ ነገር የሚያውቁ ሰዎች ወደ እየሩሳሌም መጡ
1	ኢየሱስ ታዶተታ ተማርያም ጋ አሰበዊም	ኢየሱስ ከእናቱ ማርያም ጋ አዩት
2	የትከ ኢየሱስ ይቀጥረውዬ የሳበዊ ሰብ የሞቶ የኸሬ	ኢየሱስን ሊገድሉት የሚፈልጉ ሰዎች ስለሞቱ
3	ተሳ ትከዊ ታዶተታ ተማርያም ጋሞ ጥብጥም እስራኤል ገነ ተዘፐር	ተነስ ልጁን ከእናቱ ማርያም ጋ ይዘህ ወደ እስራኤል ሀገር ተመለስ
4	በበርዛዝ ትየንምታነ	በህልም ታየውና

Figure 26 sample of the dataset

4.3. Experimental setups

The experimentation for this research is done on windows 10 operating system. The machine we used has 4GB of RAM and Intel core i5 8th generation computer. To build our system, we used the Python programming language along with the Keras, TensorFlow, NumPy, and libraries. The experiment is done using 522 Amharic-Chaha parallel sentences. In the experimentation, we used a training and testing dataset with ratio 80% for training set and 20% of the dataset for testing purpose. To validate our model during training, we used 10% of the training dataset.

4.4. Parameter Selection

In the study we used the training data to carry out a number of experiments on different parameter combinations in order to get a model with best performance. The first thing we did was choosing the embedding dimension. We chose 64, 128 and 256 as embedding dimensions. We have conducted experiments using various combinations of embedding dimension, optimizers, dropout rates, and activation functions in order to determine the optimal settings. Our experiments led us to the following parameter combination shown in Table 3, which produced the best results. These parameter combinations produce performances that are nearly identical, with only minor variations.

Table 3 Parameters

Parameter	Setup 1	Setup 2
Number of neuron	64	128
Batch size	30	60
Number of Epoch	32	64
Dropout	0.2	0.5
Optimizer	Adam	Adam
Embedding dimension	128	256
Activation function	Softmax	Relu

From the above experiments, the best result were scored with parameter combinations of 128 embedding dimension, 64 neuron, 60 epochs, 30 a batch size and 0.2 dropout rate. The time taken for the experiment with this parameter combinations were around 1:30 hours.

4.5. Performance evaluation

4.5.1. Training and validation accuracy of Chaha to Amharic translation model

In this section, we have discussed the training and validation accuracy of the proposed Chaha-Amharic machine translation model. In Figure 27, the training and validation accuracy of the encoder decoder model using the LSTM algorithm is shown. As shown in the Figure 27, the model performs 77.21% and 80.9 % training accuracy and validation accuracy respectively. This shows the model using the LSTM algorithm over fits.

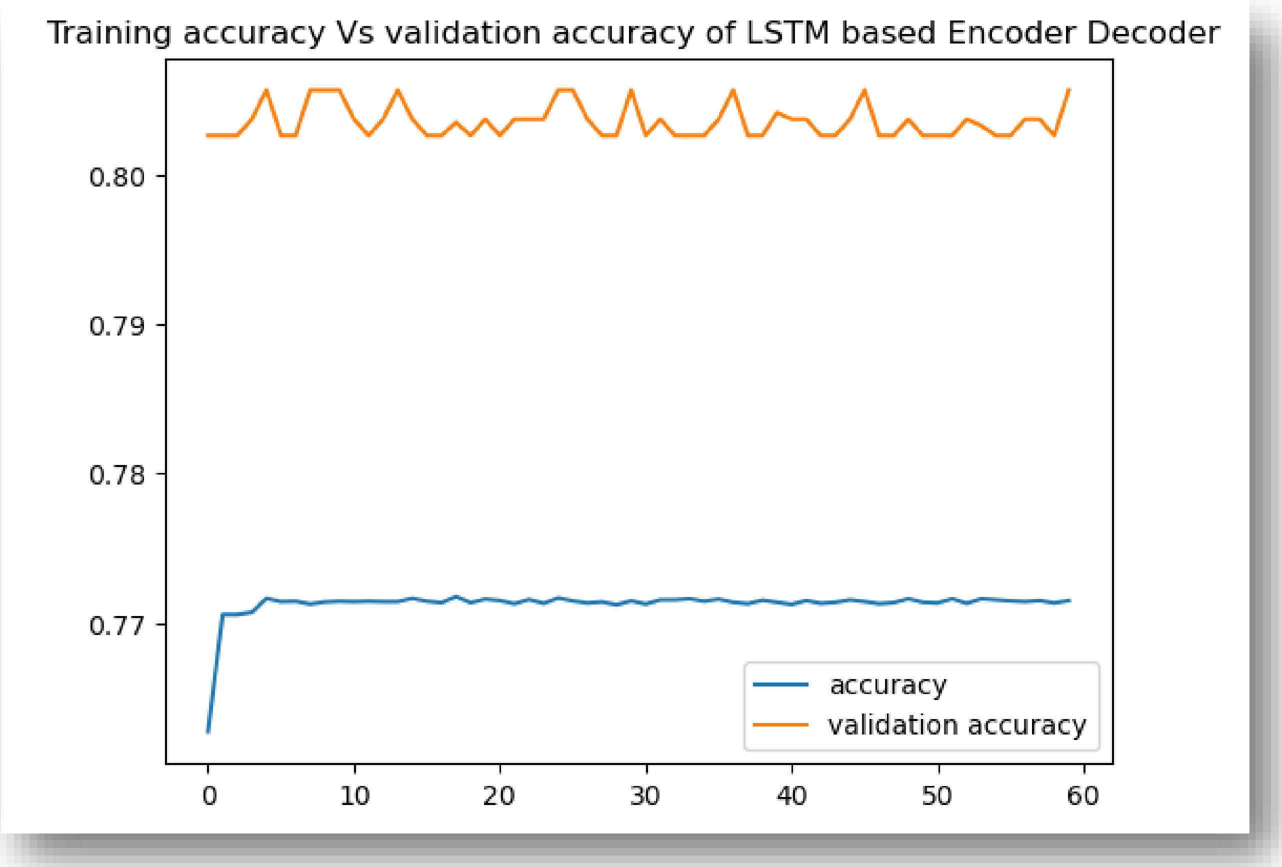


Figure 27 Training and validation accuracy of encoder decoder model with LSTM algorithm

The training and validation accuracy of GRU based encoder decoder model is shown in Figure 28. As shown in the Figure 28, the model performs 92.28% and 81.6 % training and validation accuracy respectively.

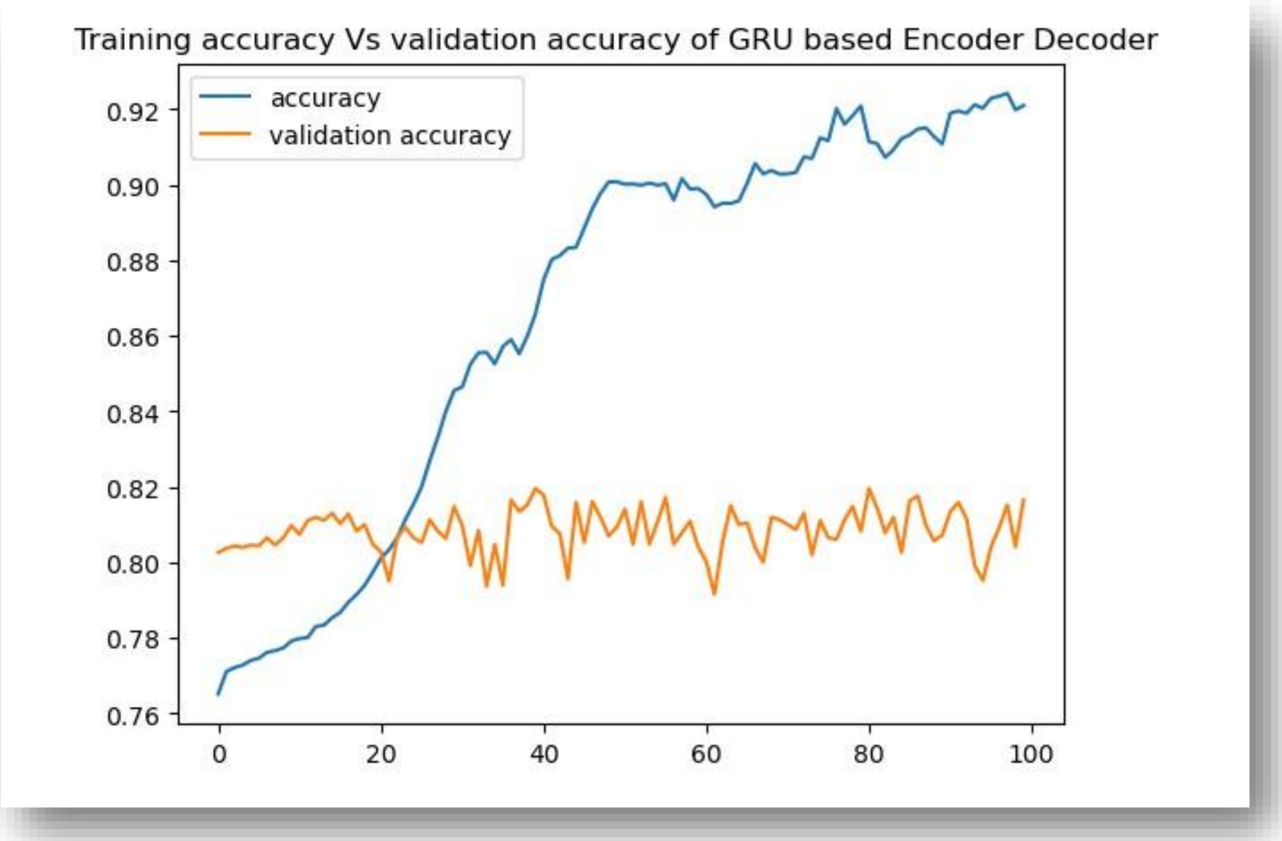


Figure 28 Training and validation accuracy of GRU algorithm

In Figure 29, the training and validation accuracy of the encoder decoder model using the Bi-LSTM algorithm is shown. As shown in the Figure 29, the model performs 89.5% and 83.5 % training and validation accuracy respectively.

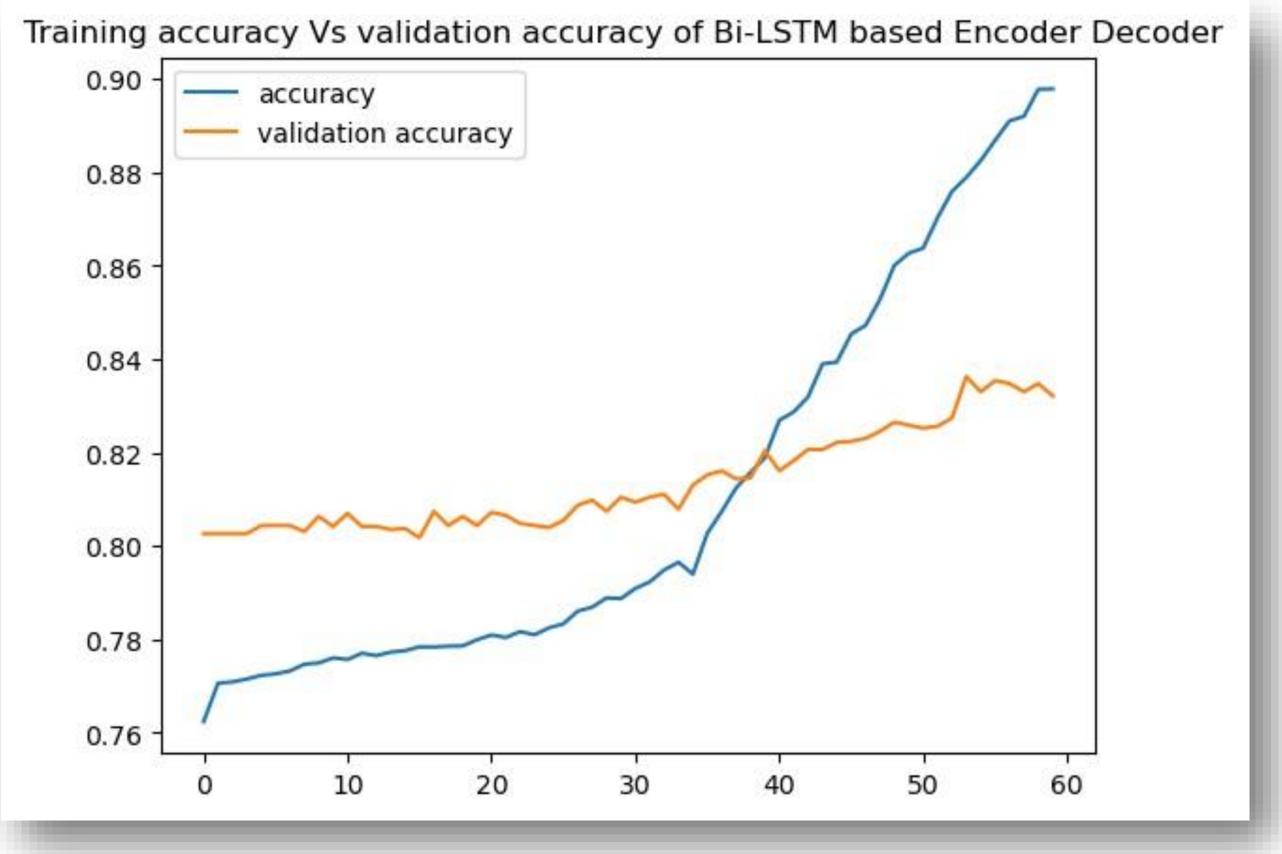


Figure 29 Training and validation accuracy of encoder decoder model with Bi-LSTM algorithm

4.1.1. Training and validation loss of Chaha to Amharic translation model

In Figure 30, the training and validation loss of the encoder decoder model using the LSTM algorithm is shown. As shown in the Figure 30, the model scored 1.64 and 2.28 training loss and validation loss respectively.

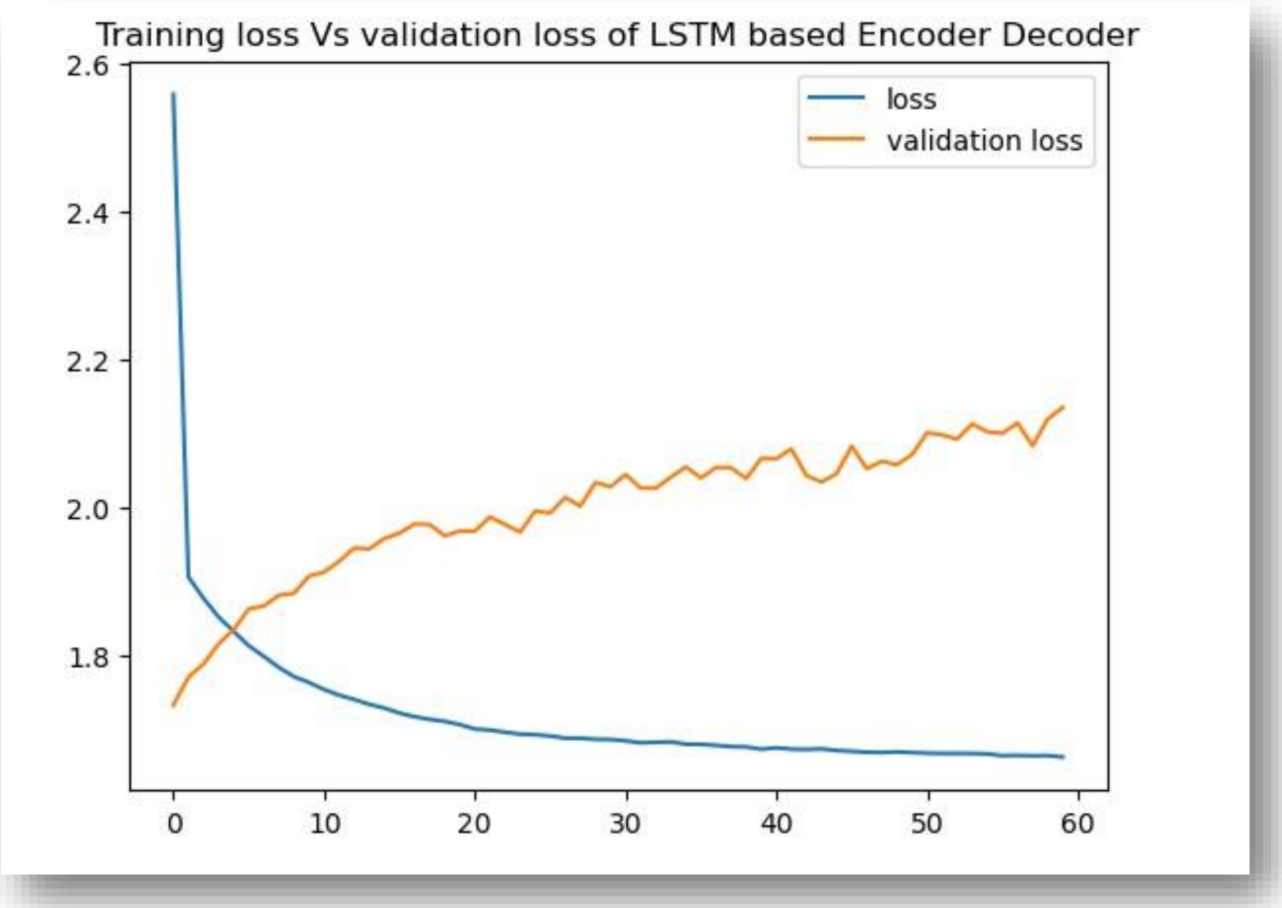


Figure 30 Training and validation loss of encoder decoder model with LSTM algorithm

In Figure 32, the training and validation loss of the GRU based encoder decoder model is shown. As shown in the Figure 32, the model scored 0.28 and 2.7 training loss and validation loss respectively.

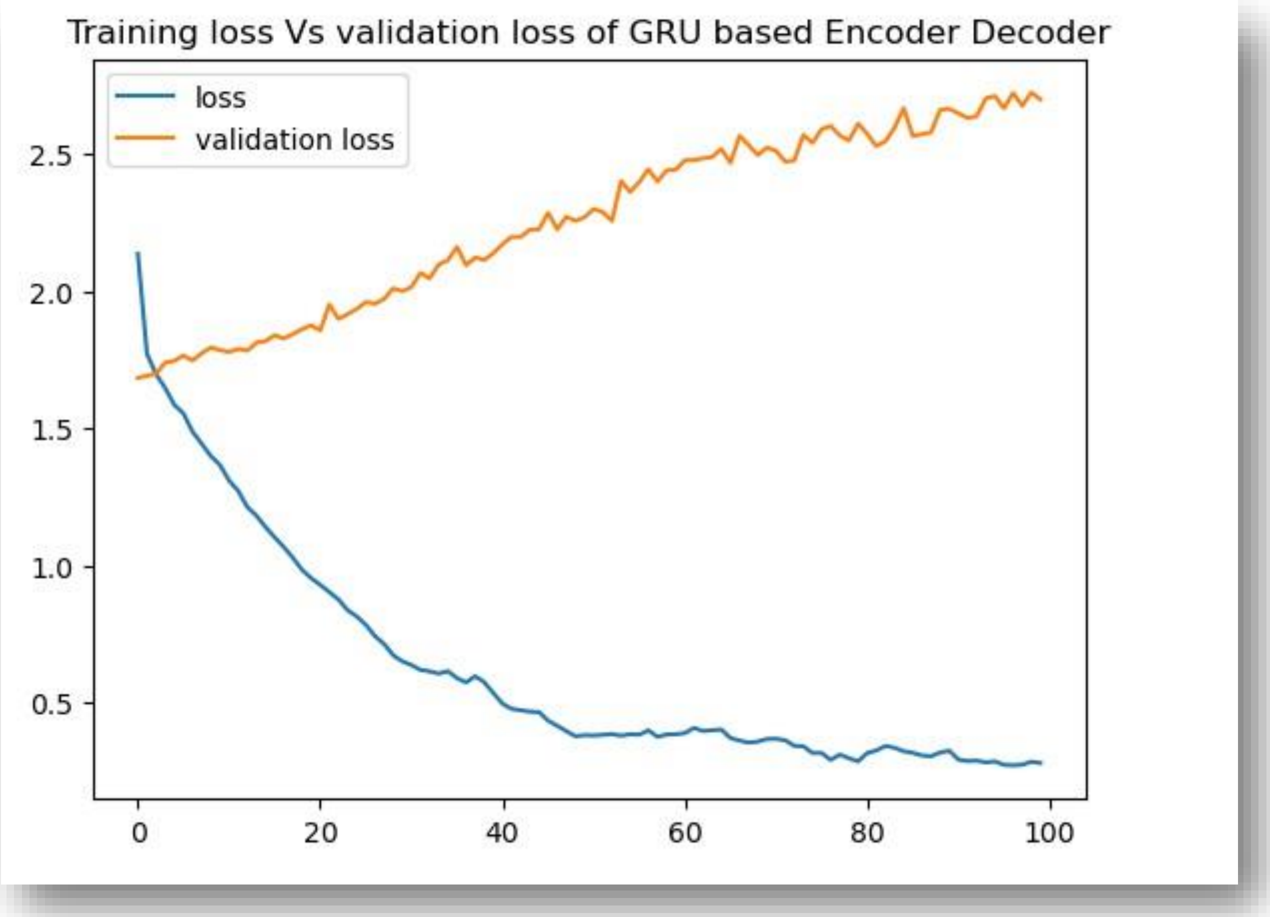


Figure 31 Training and validation loss of encoder decoder model with GRU algorithm

In Figure 33, the training and validation loss of the encoder decoder model using the Bi-LSTM algorithm is shown. As shown in the Figure 33, the model scored 0.4 and 0.71 training loss and validation loss respectively.

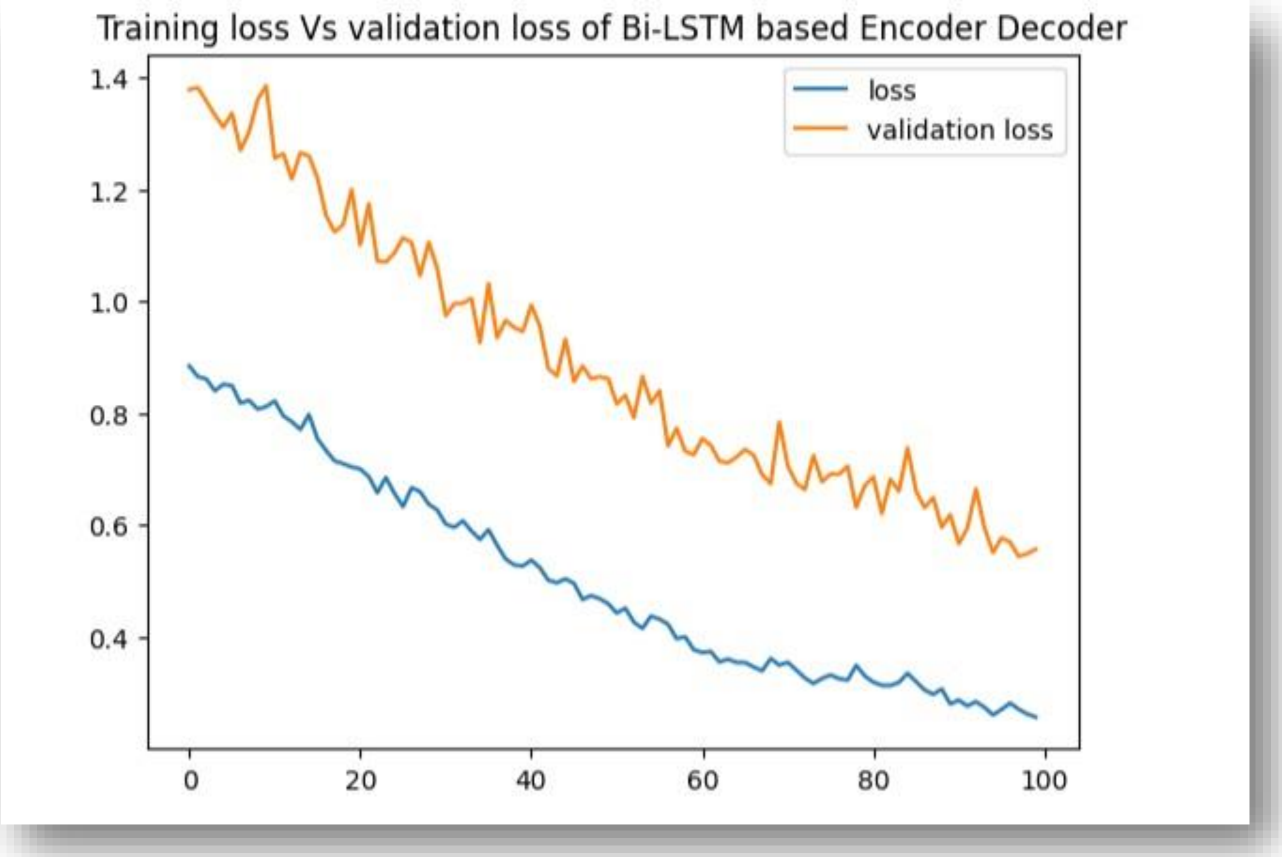


Figure 32 Training and validation loss of encoder decoder model with Bi-LSTM algorithm

4.2. Experimental results and discussion

We have followed experimental research methodology. This methodology enables us conducting many experiments by taking different combination of model hyper parameters and dataset distributions. We have conducted different experiments with different values of model parameter combinations like embedding size, learning rate, activation function and so on. We selected the parameter values discussed in section 4.4 due to their best performance in the experiment. Using the experiments conducted above, the researcher developed a Chaha-Amharic machine translation model. We have conducted different experiments using the 3 deep learning of LSTM, GRU and Bi-LSTM. To evaluate the translation model the BLEU score metrics is used in the study. We used the encoder decoder model in our experiment. We have experimented encoder decoder model using the LSTM, GRU and Bi-LSTM algorithms. Based on our experiment we got best BLEU score result using encoder decoder model with Bi-LSTM algorithm.

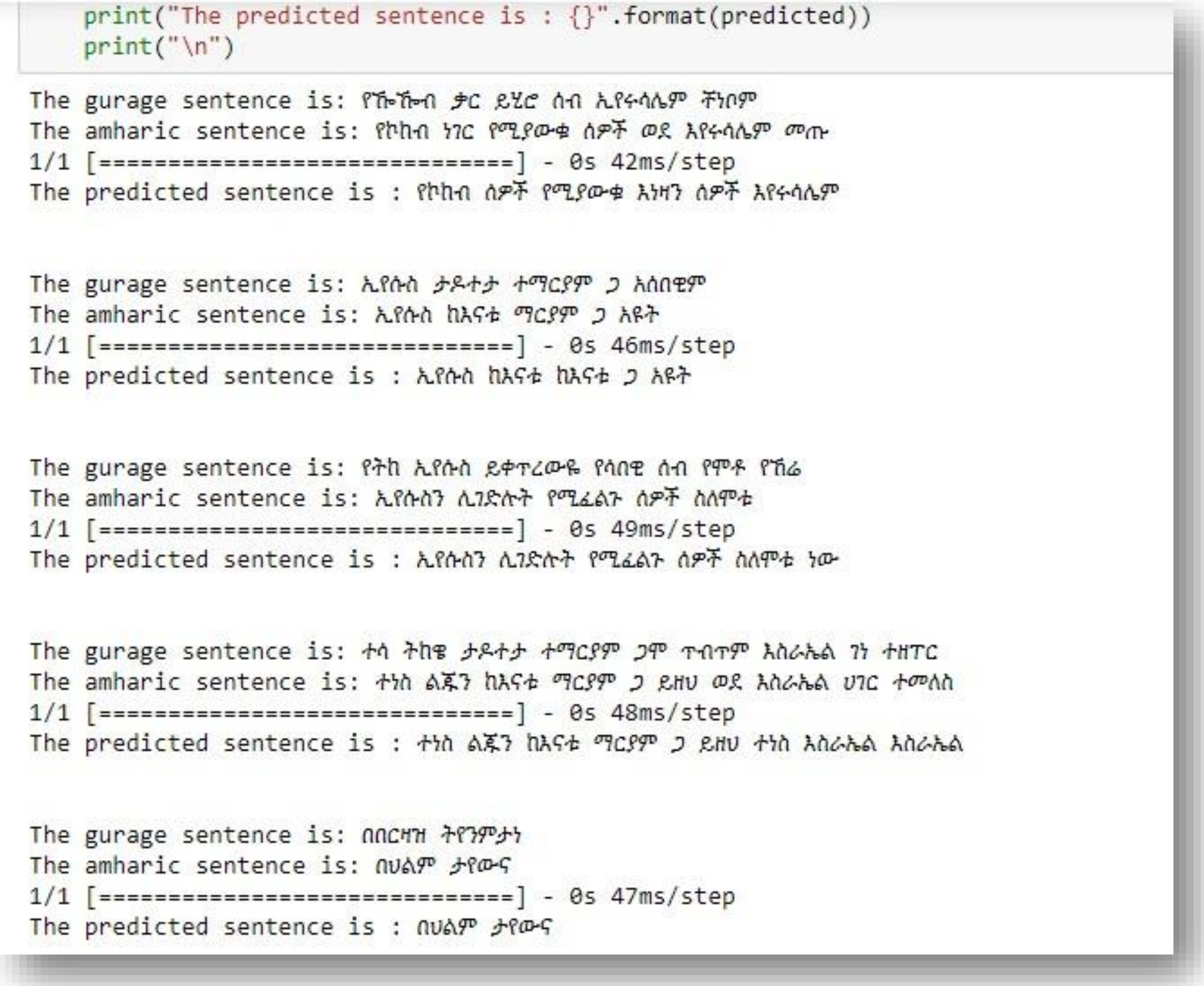
According to the experiments done, encoder decoder using the Bi-LSTM algorithm performs better than the encoder decoder model using the GRU and LSTM. Performance of the model using different deep learning algorithms is shown in Table 2.

Table 4 Performance of the model

Encoder decoder model using	Bleu score
LSTM Algorithm	17
Bi-LSTM Algorithm	**22**
GRU Algorithm	20

4.3. Prediction phase

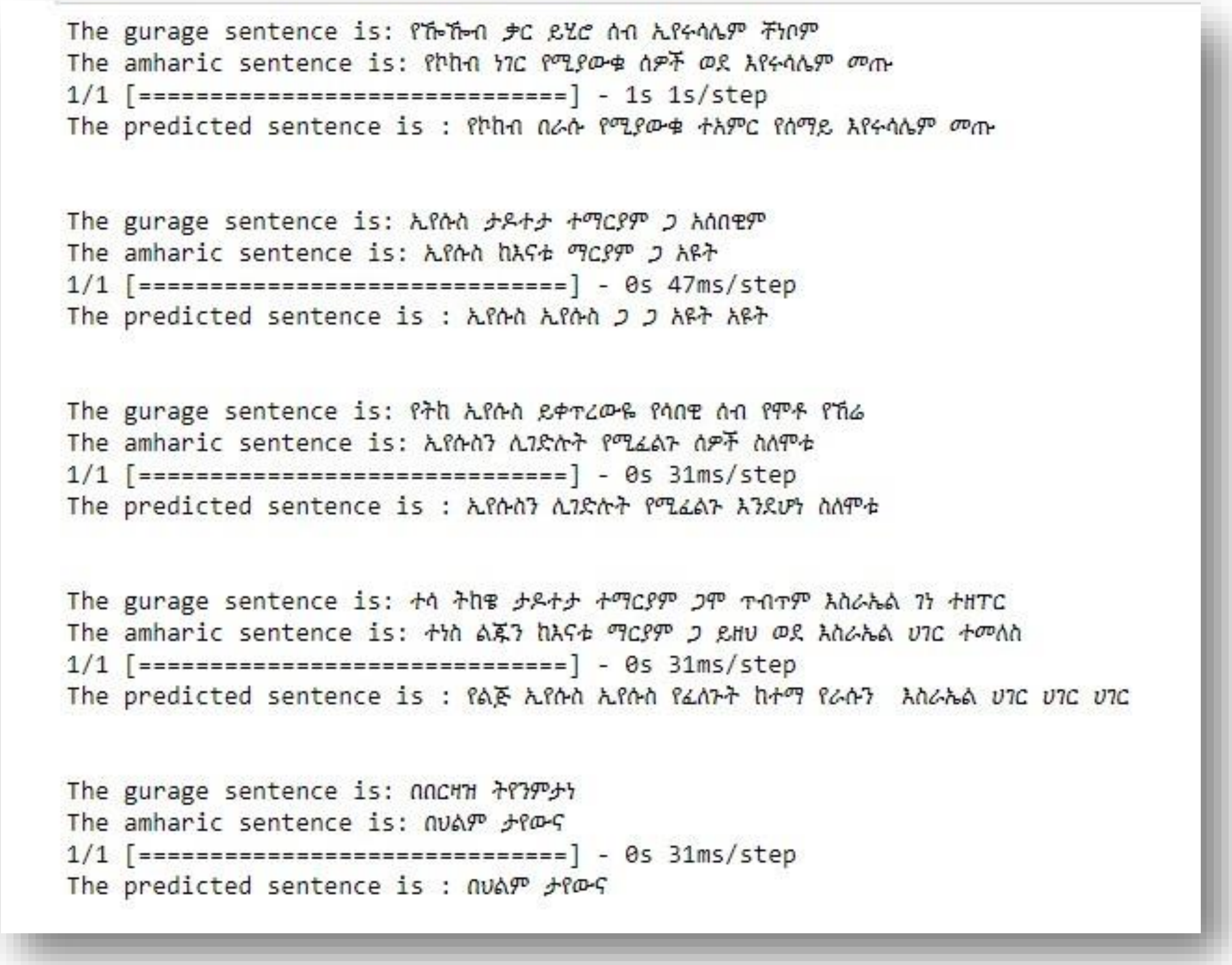
In the prediction phase, features from new (unseen) data samples are extracted and processed through the model during the prediction phase. Depending on the type of machine learning task to be completed—clustering, pattern identification, association rules, or dimensionality reduction, the model returns relevant results [61]. In the study the trained translation model predicts the Amharic translation of the Chaha text. For prediction the text is preprocessed, and fed to the model after reading the model file. Samples of the model's prediction results are shown in below snapshots. The Figures are samples snapshots of the prediction phase of different translation approaches followed in our experiment. Figure 37 shows the prediction output of Bi-LSTM based encoder decoder model.



```
    print("The predicted sentence is : {}".format(predicted))
    print("\n")
```

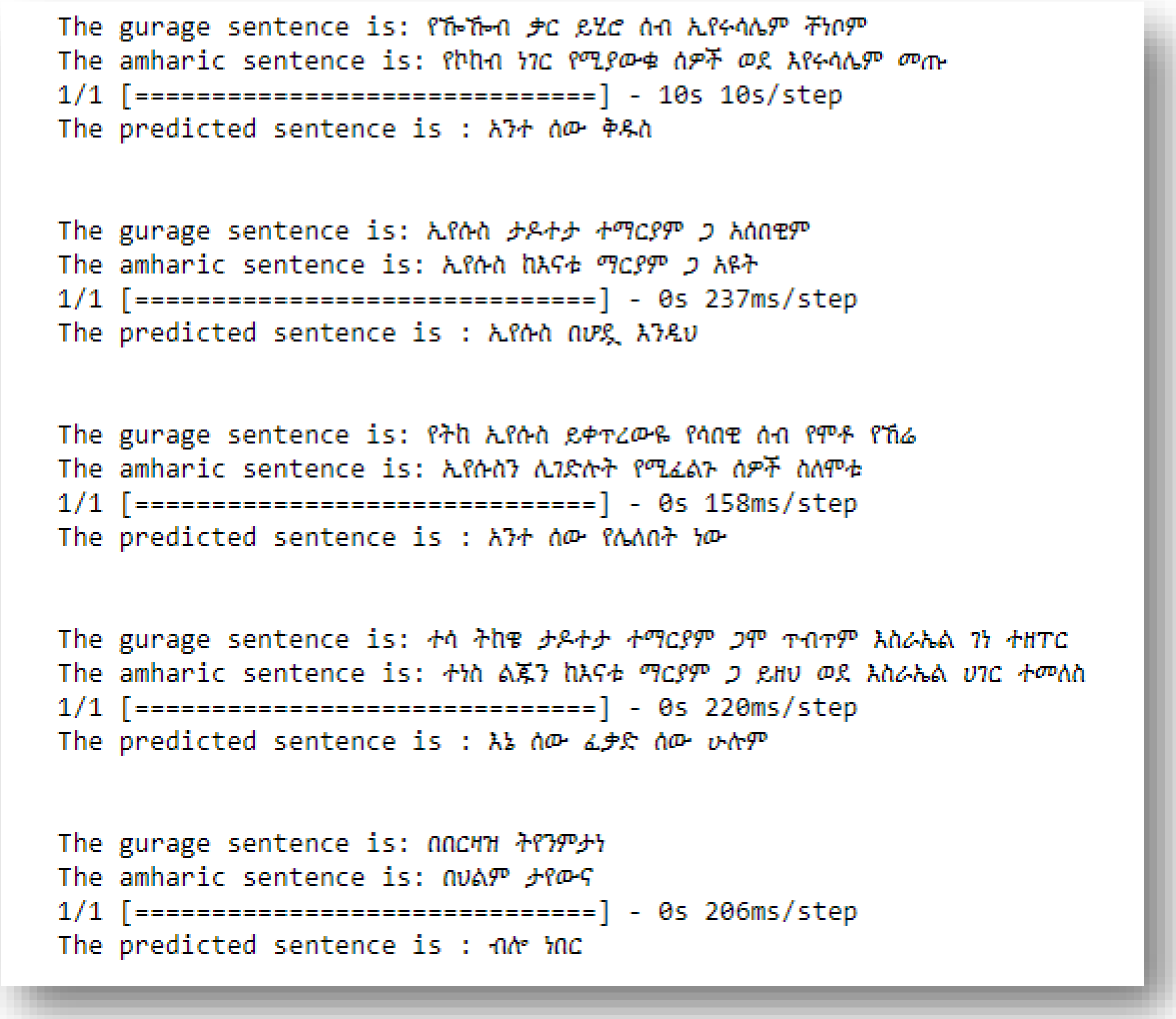
```
The gurage sentence is: የኾኾብ ቃር ይሂሮ ሰብ ኢየሩሳሌም ቾነቦም
The amharic sentence is: የኮከብ ነገር የሚያውቁ ሰዎች ወደ እየሩሳሌም መጡ
1/1 [==============================] - 0s 42ms/step
The predicted sentence is : የኮከብ ሰዎች የሚያውቁ እነዛን ሰዎች እየሩሳሌም


The gurage sentence is: ኢየሱስ ታዶተታ ተማርያም ጋ አሰበቺም
The amharic sentence is: ኢየሱስ ከእናቱ ማርያም ጋ አዩት
1/1 [==============================] - 0s 46ms/step
The predicted sentence is : ኢየሱስ ከእናቱ ከእናቱ ጋ አዩት


The gurage sentence is: የትከ ኢየሱስ ይቀጥረውዬ የሳበቺ ሰብ የሞቶ የኸሬ
The amharic sentence is: ኢየሱስን ሊገድሉት የሚፈልጉ ሰዎች ስለሞቱ
1/1 [==============================] - 0s 49ms/step
The predicted sentence is : ኢየሱስን ሊገድሉት የሚፈልጉ ሰዎች ስለሞቱ ነው


The gurage sentence is: ተሳ ትከቺ ታዶተታ ተማርያም ጋሞ ጉብጉም እስራኤል ገነ ተዘፐር
The amharic sentence is: ተነስ ልጁን ከእናቱ ማርያም ጋ ይዘህ ወደ እስራኤል ሀገር ተመለስ
1/1 [==============================] - 0s 48ms/step
The predicted sentence is : ተነስ ልጁን ከእናቱ ማርያም ጋ ይዘህ ተነስ እስራኤል እስራኤል


The gurage sentence is: በበርዛዝ ትየንምታነ
The amharic sentence is: በህልም ታየውና
1/1 [==============================] - 0s 47ms/step
The predicted sentence is : በህልም ታየውና
```

Figure 33 prediction result of Bi-LSTM based encoder decoder model



```
The gurage sentence is: የኾኾብ ቃር ይሂሮ ሰብ ኢየሩሳሌም ቾነቦም
The amharic sentence is: የኮከብ ነገር የሚያውቁ ሰዎች ወደ እየሩሳሌም መጡ
1/1 [==============================] - 1s 1s/step
The predicted sentence is : የኮከብ በራሱ የሚያውቁ ተአምር የሰማይ እየሩሳሌም መጡ


The gurage sentence is: ኢየሱስ ታዶተታ ተማርያም ጋ አሰበዊም
The amharic sentence is: ኢየሱስ ከእናቱ ማርያም ጋ አዩት
1/1 [==============================] - 0s 47ms/step
The predicted sentence is : ኢየሱስ ኢየሱስ ጋ ጋ አዩት አዩት


The gurage sentence is: የትከ ኢየሱስ ይቀጥረውዬ የሳበዊ ሰብ የሞቶ የኸፌ
The amharic sentence is: ኢየሱስን ሊገድሉት የሚፈልጉ ሰዎች ስለሞቱ
1/1 [==============================] - 0s 31ms/step
The predicted sentence is : ኢየሱስን ሊገድሉት የሚፈልጉ እንደሆነ ስለሞቱ


The gurage sentence is: ተሳ ትከቼ ታዶተታ ተማርያም ጋሞ ጥብጥም እስራኤል ገነ ተዘፐር
The amharic sentence is: ተነስ ልጁን ከእናቱ ማርያም ጋ ይዘህ ወደ እስራኤል ሀገር ተመለስ
1/1 [==============================] - 0s 31ms/step
The predicted sentence is : የልጅ ኢየሱስ ኢየሱስ የፈለጉት ከተማ የራሱን  እስራኤል ሀገር ሀገር ሀገር


The gurage sentence is: በበርዛዝ ትየንምታነ
The amharic sentence is: በህልም ታየውና
1/1 [==============================] - 0s 31ms/step
The predicted sentence is : በህልም ታየውና
```

Figure 34 prediction result of GRU based encoder decoder model



```
The gurage sentence is: የኰኰብ ቓር ይሄሮ ሰብ ኢየሩሳሌም ቾነቦም
The amharic sentence is: የኮከብ ነገር የሚያውቁ ሰዎች ወደ እየሩሳሌም መጡ
1/1 [==============================] - 10s 10s/step
The predicted sentence is : አንተ ሰው ቅዱስ


The gurage sentence is: ኢየሱስ ታዶተታ ተማርያም ጋ አሰበዊም
The amharic sentence is: ኢየሱስ ከእናቱ ማርያም ጋ አዩት
1/1 [==============================] - 0s 237ms/step
The predicted sentence is : ኢየሱስ በሆዷ እንዴህ


The gurage sentence is: የትከ ኢየሱስ ይቀጥረውዬ የሳበዊ ሰብ የሞቶ የኸፊ
The amharic sentence is: ኢየሱስን ሊገድሉት የሚፈልጉ ሰዎች ስለሞቱ
1/1 [==============================] - 0s 158ms/step
The predicted sentence is : አንተ ሰው የሌለበት ነው


The gurage sentence is: ተሳ ትከዌ ታዶተታ ተማርያም ጋሞ ጥብጥም እስራኤል ገነ ተዘፐር
The amharic sentence is: ተነስ ልጁን ከእናቱ ማርያም ጋ ይዘህ ወደ እስራኤል ሀገር ተመለስ
1/1 [==============================] - 0s 220ms/step
The predicted sentence is : እኔ ሰው ፈቃድ ሰው ሁሉም


The gurage sentence is: በበርዛዝ ትየንምታነ
The amharic sentence is: በህልም ታየውና
1/1 [==============================] - 0s 206ms/step
The predicted sentence is : ብሎ ነበር
```

Figure 35 prediction result of LSTM based encoder decoder model

CHAPTER FIVE

5. CONCLUSION AND RECOMMENDATION

5.1. Introduction

This chapter discusses the research findings and recommendations for future researchers interested in working on Sebat bet Gurage to Amharic and other language machine translation works. In this study we have developed a Sebat bet Gurage (Chaha) to Amharic language translation model using different deep learning algorithms. The model is able to predict the Amharic translation of a Chaha text. In this study encoder decoder based models are tested for their performance on Chaha-Amharic machine translation.

5.2. Conclusion

Machine translation (MT) is a well-known subfield of Natural Language Processing (NLP) that studies how to utilize computer software to translate text or speech from one language to another without using humans. During the early years of research, machine translation systems were built using bilingual dictionaries and some handcrafted rules; however, it proved difficult to handle all language anomalies with these handcrafted rules.

Chaha is one of the sebat bet gurage language families with no resource on the web. It is crucial to develop machine translation between Chaha and Amharic languages due to an increase in language users, and increasing the content of the language in the web. In this study, we have developed a Chaha-Amharic machine translation model using different machine translation approaches. In the study we have collected Chaha-Amharic parallel corpus from different sources. The model is trained using 5200 Chaha-Amharic parallel sentences. We have experimented an encoder decoder model using LSTM, Bi-LSTM and GRU deep learning algorithms.

Based on the result of our experiments done in this study, the encoder decoder model using the Bi-LSTM algorithm has a better BLEU score. The encoder decoder model using the Bi-LSTM algorithm scored 22, the encoder decoder model using the LSTM algorithm scored 17 and the encoder decoder model using the GRU algorithm scored 20. From the experiment the encoder decoder model using the Bi-LSTM algorithm took a long training time of 1:30 hours.

5.3. Contribution

After the end of the study, the researcher has contributed the things.

- The researcher has collected around 5.2K new Chaha-Amharic sentences pairs that can be used by other researchers on the area.
- The researcher has showed that encoder decoder model using Bi-LSTM algorithm outperforms other approaches in our experiments.
- The researcher has built a Chaha-Amharic machine translation model.

5.4. Recommendation

After all the researcher recommend the following issues to be addressed for future.

- This translation model is trained with limited dataset, extending the study with a large dataset is recommended.
- Morphological analysis of the languages might increase the performance of the translation model and is recommended.
- Exploring other approaches to improve the performance of the model
- Exploring bidirectional machine translation model for the languages

References

[1] S. Weninger, M. P. Streck, and J. C. E. Watson, “An International Handbook Edited by Geoffrey Khan”.

[2] D. Yacob and F. Menuta, “A Review of Shifts in Gurage Orthography”.

[3] T. Yeshambel and J. Mothe, “applied sciences Amharic Adhoc Information Retrieval System Based on Morphological Features,” 2022.

[4] M. M. Woldeyohannis, L. Besacier, M. Meshesha, and A. Ababa, “Amharic Speech Recognition for Speech Translation,” vol. 11, no. 1, pp. 11–21, 2016.

[5] P. Rychlý and G. T. Lemma, “An Update of the Manually Annotated Amharic Corpus,” pp. 124–128.

[6] D. Khurana, A. Koli, K. Khatter, and S. Singh, “Natural language processing: state of the art, current trends and challenges,” *Multimed. Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, 2023, doi: 10.1007/s11042-022-13428-4.

[7] S. Processing, S. Interpretation, and F. Readings, “Allen 1995 : Natural Language Understanding 1 . Introduction to Natural Language Understanding,” pp. 1–24, 1995.

[8] S. Yang, Y. Wang, and X. Chu, “A Survey of Deep Learning Techniques for Neural Machine Translation”.

[9] H. Peng, “The Impact of Machine Translation and Computer-aided Translation on Translators,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 322, no. 5, 2018, doi: 10.1088/1757-899X/322/5/052024.

[10] H. Ji, S. Oh, J. Kim, S. Choi, and E. Park, “Integrating Deep Learning and Machine Translation for Understanding Unrefined Languages,” 2022, doi: 10.32604/cmc.2022.019521.

[11] A. Wahid, “Machine Translation System Using Deep Learning for English to Urdu,” vol. 2022, 2022.

[12] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing [Review Article],” *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, 2018, doi: 10.1109/MCI.2018.2840738.

[13] D. Goularas and S. Kamis, “Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data,” *2019 Int. Conf. Deep Learn. Mach. Learn. Emerg. Appl.*, pp. 12–17, 2019, doi: 10.1109/Deep-ML.2019.00011.

[14] M. Popel, M. Tomkova, Ł. Kaiser, and J. Uszkoreit, “Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals,” pp. 1–15, 2020, doi: 10.1038/s41467-020-18073-9.

[15] S. Yuan, “A Two Phase Deep Learning Model for Identifying Discrimination from Tweets,” pp. 696–697, 2016.

[16] V. FORTUNATI, “How does deep learning in radiology work?” [Online]. Available: https://www.quantib.com/blog/how-does-deep-learning-work-in-radiology

[17] Y. Shi, Y. Tian, Y. Wang, W. Zeng, and T. Huang, “Learning long-term dependencies for action recognition with a biologically-inspired deep network,” pp. 716–725.

[18] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, “Deep learning applications and challenges in big data analytics,” *J. Big Data*, vol. 2, no. 1, pp. 1–21, 2015, doi: 10.1186/s40537-014-0007-7.

[19] F. Menuta, “Linguistic Distance among Gurage Language Varieties,” pp. 1–17, 2023.

[20] E. H. Tessema, “Description of the social varieties used by the Gurage : Fedwet and Fuga,” no. June, 2019.

[21] W. G. Variety, “Towards a Grammar of Gumer,” 2017.

[22] “Chaha language.” [Online]. Available: https://en.wikipedia.org/wiki/Chaha_language

[23] L. Studies and B. A. Keleta, “Gura Documentation and Description of Morphology and Syntax Gura Documentation and Description of Morphology and Syntax,” 2020.

[24] M. Y. Tachbelie, “No Title,” no. August, 2010.

[25] B. Abel, “Geez to Amharic Machine Translation,” 2018.

[26] P. M. Nadkarni, L. Ohno-machado, and W. W. Chapman, “Natural language processing : an introduction,” 2011, doi: 10.1136/amiajnl-2011-000464.

[27] Y. Zhou, “Natural Language Processing with Improved Deep Learning,” vol. 2022, 2022.

[28] K. B. Cohen, *Biomedical Natural Language Processing and Text Mining*, Error. Elsevier Inc., 2014. doi: 10.1016/B978-0-12-401678-1.00006-3.

[29] H. Wang, H. Wu, Z. He, L. Huang, and K. Ward, “Progress in Machine Translation,” *Engineering*, 2021, doi: 10.1016/j.eng.2021.03.023.

[30] G. Randhawa, M. Ferreyra, R. Ahmed, O. Ezzat, and K. Pottie, “Using machine translation in clinical practice.,” *Can. Fam. Physician*, vol. 59, no. 4, pp. 382–383, Apr. 2013.

[31] M. D. Okpor, “Machine Translation Approaches,” vol. 11, no. 5, pp. 159–165, 2014.

[32] K. Imamura, H. Okuma, T. Watanabe, and E. Sumita, “Example-based machine translation based on syntactic transfer with statistical models,” *COLING 2004 - Proc. 20th Int. Conf. Comput. Linguist.*, no. Figure 2, 2004, doi: 10.3115/1220355.1220370.

[33] P. E. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, “The Mathematics of Statistical Machine Translation : Parameter Estimation,” vol. 10598, 1993.

[34] P. F. Brown *et al.*, “Statistical approach to machine translation,” vol. 16, no. 2, pp. 79–85, 1990.

[35] Z. Yang, Y. Gao, W. Wang, and H. Ney, “Predicting and Using Target Length in Neural Machine Translation,” *Proc. 1st Conf. Asia-Pacific Chapter Assoc. Comput. Linguist. 10th Int. Jt. Conf. Nat. Lang. Process.*, pp. 389–395, 2020, [Online]. Available: https://www.aclweb.org/anthology/2020.aacl-main.41

[36] B. Premjith, M. A. Kumar, and K. P. Soman, “Neural machine translation system for English to Indian language translation using MTIL parallel corpus,” *J. Intell. Syst.*, vol. 28, no. 3, pp. 387–398, 2019, doi: 10.1515/jisys-2019-2510.

[37] J. Yang, S. Ma, D. Zhang, Z. Li, and M. Zhou, “Improving neural machine translation

with soft template prediction," *Proc. Annu. Meet. Assoc. Comput. Linguist.*, pp. 5979–5989, 2020, doi: 10.18653/v1/2020.acl-main.531.

[38] S. P. Singh *et al.*, "Machine Translation using Deep Learning: An Overview," pp. 162–167, 2017.

[39] M. Parmar and V. S. Devi, "Neural Machine Translation with Recurrent Highway Networks," pp. 1–10.

[40] D. Lee *et al.*, "Long short-term memory recurrent neural network-based acoustic model using connectionist temporal classification on a large-scale training corpus," *China Commun.*, vol. 14, no. 9, pp. 23–31, 2017, doi: 10.1109/CC.2017.8068761.

[41] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning Based Text Classification: A Comprehensive Review," *arXiv*, vol. 1, no. 1, pp. 1–43, 2020.

[42] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-Term memory networks," *ACL-IJCNLP 2015 - 53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Jt. Conf. Nat. Lang. Process. Asian Fed. Nat. Lang. Process. Proc. Conf.*, vol. 1, pp. 1556–1566, 2015, doi: 10.3115/v1/p15-1150.

[43] B. Jang, M. Kim, G. Harerimana, S. Kang, and J. W. Kim, "applied sciences Bi-LSTM Model to Increase Accuracy in Text Classification : Combining Word2vec CNN and Attention Mechanism," 2020.

[44] M. Z. Amin and N. Nadeem, "Convolutional Neural Network : Text Classification Model for Open Domain Question Answering System".

[45] L. Alzubaidi *et al.*, *Review of deep learning : concepts , CNN architectures , challenges , applications , future directions*. Springer International Publishing, 2021. doi: 10.1186/s40537-021-00444-8.

[46] Y. Kim, "Convolutional Neural Networks for Sentence Classification," 2014.

[47] Y. Gao, C. Herold, and Z. Yang, "Is Encoder-Decoder Redundant for Neural Machine Translation?," 2019.

[48] B. Van Merri, “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches,” 2014.

[49] G. Tang, “An Analysis of Attention Mechanisms : The Case of Word Sense Disambiguation in Neural Machine Translation,” vol. 1, pp. 26–35, 2018.

[50] F. Stahlberg, “Neural machine translation: A review,” *J. Artif. Intell. Res.*, vol. 69, pp. 343–418, 2020, doi: 10.1613/JAIR.1.12007.

[51] L. Zhao, W. Gao, and J. Fang, “applied sciences High-Performance English – Chinese Machine Translation Based on GPU-Enabled Deep Neural Networks with Domain Corpus,” 2021.

[52] Z. Tan, S. Wang, Z. Yang, G. Chen, and X. Huang, “Neural machine translation : A review of methods , resources , and tools,” *AI Open*, vol. 1, no. November 2020, pp. 5–21, 2021, doi: 10.1016/j.aiopen.2020.11.001.

[53] M. Kinfe, “Amharic-Kistanigna Bi-directional Machine Translation using Deep Learning,” *ריקדא*, no. 8.5.2017, pp. 2003–2005, 2022.

[54] S. T. Abate and S. Atinafu, “Parallel Corpora for bi-Directional Statistical Machine Translation for Seven Ethiopian Language Pairs,” pp. 83–90, 2018.

[55] G. W. GEBEYEHU, “GE EZ-AMHARIC MACHINE TRANSLATION USING DEEP LEARNING,” 2021.

[56] T. Kassa, “Morpheme-Based Bi-directional Ge’ez -Amharic Machine Translation,” 2018.

[57] Y. Wu *et al.*, “Google ’ s Neural Machine Translation System : Bridging the Gap between Human and Machine Translation,” pp. 1–23.

[58] C. Town and S. Africa, “ENGLISH-AMHARIC STATISTICAL MACHINE TRANSLATION (1) Addis Ababa , Ethiopia : IT Doctoral Program , Addis Ababa University (2) Grenoble , France : University Joseph Fourier,” 2012.

[59] I. Gashaw and H. L. Shashirekha, “AMHARIC-ARABIC NEURAL MACHINE TRANSLATION,” pp. 55–68, 2019, doi: 10.5121/csit.2019.91606.

[60] K. Aitken and N. Maheswaranathan, “Understanding How Encoder-Decoder

Architectures Attend,” no. NeurIPS, 2021.

[61] A. Subasi, “Chapter 1 - Introduction,” in *Practical Machine Learning for Data Analysis Using Python*, A. Subasi, Ed., Academic Press, 2020, pp. 1–26. doi: https://doi.org/10.1016/B978-0-12-821379-7.00001-1.

Appendix A. Chaha language orthographic Table

ä **[ə]**	**U**	**i**	**a**	**e**	**ə** **[ɨ]**	**o**	**ʷä** **[ʷə]**	**ʷi**	**ʷa**	**ʷe**	**ʷə [ʷɨ]**	
x	□	□	□	□	□	□	□	□	□	□	□	□
xʸ	□	□	□	□	□	□	□					
l	□	□	□	□	□	□	□					
m	□	□	□	□	□	□	□	□	□	□	□	□
r	□	□	□	□	□	□	□					
s	□	□	□	□	□	□	□					
š	□	□	□	□	□	□	□					
ḳ	□	□	□	□	□	□	□	□	□	□	□	□
ḳʸ	□	□	□	□	□	□	□					
b	□	□	□	□	□	□	□	□	□	□	□	□
β	□	□	□	□	□	□	□					
t	□	□	□	□	□	□	□					

č	□	□	□	□	□	□	□					
n	□	□	□	□	□	□	□					
ñ	□	□	□	□	□	□	□					
ʾ	□	□	□	□	□	□	□					
k	□	□	□	□	□	□	□	□	□	□	□	□
k^{y}	□	□	□	□	□	□	□					
w	□	□	□	□	□	□	□					
z	□	□	□	□	□	□	□					
ž	□	□	□	□	□	□	□					
y	□	□	□	□	□	□	□					
d	□	□	□	□	□	□	□					
ǧ	□	□	□	□	□	□	□					
g	□	□	□	□	□	□	□	□	□	□	□	□
g^{y}	□	□	□	□	□	□	□					
ṭ	□	□	□	□	□	□	□					
č̣	□	□	□	□	□	□	□					
p̣	□	□	□	□	□	□	□					
ṣ	□	□	□	□	□	□	□					
f	□	□	□	□	□	□	□	□	□	□	□	□

p	□	□	□	□	□	□	□	□	□	□	□	□
	ä **[ə]**	**u**	**i**	**a**	**e**	**ə [ɨ]**	**o**	**^{w}ä** **[^{w}ə]**	**^{w}i**	**^{w}a**	**^{w}e**	**^{w}ə [^{w}ɨ]**