Addis Ababa University
College of Natural Sciences

Attention-based Amharic-to-Wolaita Neural Machine Translation

Workineh Wogaso Gaga

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

October 2020
Addis Ababa, Ethiopia

Addis Ababa University

College of Natural Sciences

Workineh Wogaso Gaga

Advisor: Yaregal Assabie (PhD)

This is to certify that the thesis prepared by Workineh Wogaso, titled: *Attention-based Amharic-to-Wolaita Neural Machine Translation* and submitted in partial fulfilment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:	Yaregal Assabie (PhD)________	___________________	__________________
Examiner:	_Solomon Atinafu (PhD)_______	___________________	__________________
Examiner:	_Mulugeta Libsie (PhD)_______	___________________	__________________

Abstract

Natural language (NL) is one of the fundamental aspects of human behaviors and we can communicate all over the world through it. Natural Language Processing (NLP) is a branch of artificial intelligence that deals with the interaction between computers and humans using the NL. In the NLP world, every NL should be well understood by the machine. Machine Translation (MT) is a process by which computer software understand NL and an automatic translation of a text from one NL (source language) to another (target language). Neural Machine Translation (NMT) is a recently proposed approach to MT and has been able to gain State of The Art (SOTA) translation quality for the last few years. Unlike the traditional MT approaches, NMT aims at building a single neural network that can be jointly tuned to maximize translation performance. In this thesis we proposed Attention-based Amharic-to-Wolaita NMT. We built our system based on Encoder-Decoder architecture also called Sequence to Sequence (Seq2Seq) model by integrating a Recurrent Neural Network (RNN) and Gated Recurrent Unit (GRU). For comparison of our translation performance, we developed non-attention-based Amharic-to-Wolaita NMT. An encoder in basic (non-attention-based) Encoder-Decoder architecture encodes the complete information of the source (Amharic) sequence into a single real-valued vector called context vector which is passed to the Decoder to produce an output (Wolaita) sequence. Here, a context vector summarizes the entire input sequence into a single vector. As the length of the sentence increases, the inter-dependency of words is loosely related and it is a major drawback. The second problem of basic Encoder-Decoder model is handling of a large number of vocabulary sizes. As each word in the sentence is visited, it must be assigned a new identity number. But when the length of the corpus increases, the number used for word representation and dimension of word vector needed becomes higher. These two basic issues are solved using an attention mechanism. However, either attention-based or non-attention based NMT have not been developed for Amharic-to-Wolaita language pair. Thus, we developed attention-based NMT for Amharic-to-Wolaita translation and compare it against a non-attention-based NMT system. We used a BLEU score evaluation technique for system evaluation and we got 0.5960 BLEU score for non-attention-based system and 0.6258 BLEU score for attention-based system. Thus, attention-based system obtains up to +0.02978 BLEU improvements over non-attention-based NMT system.

Keywords: Natural Language Processing, Machine Translation, Neural Machine Translation, Recurrent Neural Network, Language Modelling, Attention-based Encoder-Decoder Model.

Dedication

It was dedicated to

1. My lovely family. Especially to My lovely Mother Workinesh Dana. Mom, I love you whole my remaining life and I promise to live a life that will do justice to all the sacrifices you have made to me.
2. An innocent people who lost their life in different Ethiopian regions.

Acknowledgements

First and foremost, I would like to thank the Almighty **God** and **St. Mary** for giving me strength, determination, endurance and wisdom to accomplish this study and for being always with me in whole my journeys I passed.

Next, I would like to extend my deepest appreciation to my advisor, **Dr. Yaregal Assabie**, for his excellent and enduring support. He has given me so much inspiration for my future academic career. I taught more from him how to build up my academic brand, when to say no to distractors, what to give to others, and many, many more great things. Without him, it will be uncertain if I can make it all the way through my MSc.

I also wish to express my sincere gratitude to **all** whom through their supports contributed to the successful completion of this work. First, I am highly indebted to **Tewodros Abebe**, currently PhD Student of Addis Ababa University, for his expertise, generous time, and patience in helping me to complete this thesis especially by informing some previous work of Wolaita language.

“Yoho ubbaa ba wodiyan puullayidi oottis; xoossay benippe doommidi alamiya wurssettay gakkanaw oottido oosuwa asa na’ay pilggidi gakkenna mala meri merinatettaa garssan immis”.

“ነገርን ሁሉ በጊዜው ውብ አድርጎ ሠራው፤ እግዚአብሔርም ከጥንት ጀምሮ እስከ ፍጻሜ ድረስ የሠራውን ሥራ ሰው መርምሮ እንዳያገኝ ዘላለምነትን በልቡ ሰጠው”።

መጽሐፈ መክብብ 3:11.

Table of Contents

List of Tables

List of Figures

Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Network
BLEU	Bi-Lingual Evaluation Understudy
BPTT	Backpropagate Through Time
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
EBMT	Example Based Machine Translation
FDRE	Federal Democratic Republic of Ethiopia
GLU	Gated Linear Unit
GPU	Graphical Processing Unit
GRU	Gated Recurrent Unit
LM	Language Modeling
LSTM	Long Short Term Memory
MT	Machine Translation
NL	Natural Language
NLM	Neural Language Modeling
NLP	Natural Language Processing
NMT	Neural Machine Translation
RBMT	Rule Based Machine Translation
RLM	Recurrent Language Model
RNN	Recurrent Neural Network
SMT	Statistical Machine Translation
SNNPRS	Southern Nations Nationalities and Peoples Regional State
SOTA	Start of The Art
SRLIM	SR Luthra Institute of Management
WMT	Conference on Machine Translation

Chapter 1: Introduction

1.1 Background

Natural language (NL) is one of the fundamental aspects of human behaviours and is a crucial component in our lives [1]. It is the method of communication in different ways which are by audio (spoken), text (written) and signs to exchange ideas, emotions, and information [2]. Thus, we can communicate all over the world through NL. The advancement of technology and the rise of the Internet as a means of communication led to an ever-increasing demand for Natural Language Processing [3]. Natural Language Processing (NLP), also called computational linguistics is widely regarded as a promising and critically important endeavour in the field of computer research [4]. It came into existence to ease the user's work [4]. The general goal for most computational linguists is to let the computer have the ability to understand and generate NL so that eventually people can address their computers through text and speech as though they were addressing another person [4]. NLP applications are useful in facilitating human-human, human-computer, computer-human, and computer-computer communication via computing systems. In the NLP world, every language should be well understood by the machine. The process that lets the machine understand the different languages used all around the world is called machine translation [2].

Machine Translation (MT) is a process by which computer software is used to translate a text from one NL to another [2]. It refers to an automatic translation of one language into one (bi-lingual) or more languages (multi-lingual) through electronic devices that contain a dictionary along with the programs needed to make logical choices as required for the new language [2]. MT is considered to be the most substantial way in which machines could communicate to humans and vice versa [4]. To process any translation, human or automated, the meaning of a text in the source language must be fully restored in the target language. While on the surface this seems straightforward, it is far more complex [33]. Translation is not a mere word-for-word substitution. A translator must interpret and analyze all of the elements in the text and know how each word may influence another [33]. This requires extensive expertise in grammar, syntax, semantics, etc., in the source and target languages, as well as familiarity with each local region in which syntax and semantic mean sentence structure and meanings respectively. The translation of NLs by machine was first dreamt of in the 17th century [5].

The advantages of MT over human translator is when time is a crucial factor; MT can make translations quickly [34]. We don't have to spend hours poring over dictionaries to translate the words and the machine translator can translate the content quickly. The next benefit of MT over human translator is, MT is comparatively cheap. This is because if we use the professional translator, s/he will charge us on a per-page basis which is going to be extremely costly. Confidentiality is another matter which makes MT favourable. Giving sensitive data to a professional translator might be risky. Universality is also another advantage of a machine translator. A machine translator usually translates a text which is in any language while a professional translator specializes in one particular field. Online translation and translation of web page content is also a favourable advantage of a machine translator. Online translation services are at hand and we can translate information quickly with this service. Furthermore, we can translate any web page content and query of a search engine by the use of MT systems [1, 8].

A number of approaches are available for MT, such as Rule-Based Machine Translation (RBMT), Corpus-based Machine Translation Approach (Corpus-based MT), Hybrid Machine Translation (Hybrid MT) and Neural Machine Translation (NMT) [6]. NMT is a recently proposed framework for MT based purely on neural networks. Neural networks are making in-roads into the MT industry, providing major advances in translation quality over the existing industry standard Statistical Machine Translation (SMT) technology. Because of how the technology functions, neural networks better capture the context of full sentences [7]. This technique has begun to show promising results when compared to other approaches [9, 10, 11, 12]. Unlike the traditional phrase-based translation system, NMT being an end-to-end trained model attempts to build and train a single large neural network [13]. The goal of neural network is to design a fully trainable model of which every component is tuned based on training corpora to maximize its translation performance [8]. People have been turning their heads towards NMT systems, which after being introduced seriously in 2014 have seen many refinements.

There are three big wins of NMT. The first one is end-to-end training, where all parameters are simultaneously optimized to minimize a loss function on a network's output. The second is a distributed representation of parameters. It is a way of mapping or encoding information to a neural network. The last one is better exploitation of the word and phrase similarities [13]. Attention-mechanism improves its performance when the length of words in a sentence becomes longer [8].

1.2 Motivation

Wolaita language belongs to the Omotic language family among Ethiopian language families (such as Semitic, Cushitic, Nilotic and Omotic). It is spoken by Wolaita people and some other parts of the Southern Nations, Nationalities, and People's Region of Ethiopia such as Gamo, Gofa, Mello, Kucha, and Dawro [14]. Wolaita language uses the Latin script for writing [14]. It is the working language in the Wolaita Zone. Currently, primary, secondary and higher education institutions, as well as different mass media, are using the Wolaita language in teaching and learning, and information transformation processes [14].

Amharic is one of the languages in the Semitic family which is widely spoken in Ethiopia [15]. It is the first largest spoken language in Ethiopia, the second most-spoken Semitic language in the world (after Arabic), and one of the five largest languages on the African continent [25]. Amharic is an official working language of the Federal Democratic Republic of Ethiopia (FDRE) [15]. Since Southern Nations Nationalities and Peoples Regional State (SNNPRS) is a collection of different Nation Nationalities having different languages and cultures, Amharic is also an official working language in the region [15]. Thus, different types of official documents, newspapers, and vacancies are written and produced in Amharic both at the federal and regional levels [15].

Speakers of Wolaita language who are unable to speak and understand Amharic cannot communicate and interact with Amharic speakers in an easy way without finding translators. Thus, non-Amharic speakers of Wolaita language speakers face problems of lack of information. The constitution of FDRE [15] recommends it is better if every regional official document is translated and documented in Amharic language parallel with the local language. Once Nelson Mandela talked at public speech as "*If you talk to a man in a language he understands, that goes to his head. If you talk to him in his own language that goes to his heart*" [16]. Thus, it is a good contribution if there is a way to translate federal written Amharic documents and literal news into Wolaita language which can prevent non-Amharic speakers of Wolaita peoples from lack of information and face to unwanted expenses, such as time and cost. Amharic-to-Wolaita Machine Translation can solve the aforementioned problems. This has motivated us to work on Attention-based Amharic-to-Wolaita Neural MT.

1.3 Statement of the Problem

Several studies and applications have been done for foreign languages MT using different approaches. Most of the works have been done on language pair of English and other languages, such as for Arabic-to-English Neural Machine Translation [17] and English-Japanese Machine Translation [18], French to English Statistical Machine Translation system [19], etc. This is because the English language is the dominant language spoken across the world [20]. However, only a little work has been done on the MT system among English and Ethiopian languages. Some of the studies are carried out on English-Amharic language pair [21, 22], English-Afaan Oromo language pair [6, 23]. Some of the MTs done between Ethiopian languages are Amharic-to-Tigrigna MT using a hybrid approach [24], bi-directional Ge'ez-Amharic MT [25] and so on.

Translations that have been made so far from Amharic-to-Wolaita and English-to-Wolaita are religious books using human translator [26]. People use human translation and they tend to be slower as compared to machines [27]. Sometimes it can be hard to get a precise translation that reveals what the text is about without everything being translated word-to-word. Translation software allows to translate entire text documents within seconds. Human translation takes much longer, especially if specific meanings have to be looked up in a dictionary. Thus, MT helps to save time [28]. This and the other issues are when MT comes in, which solves most of the problems caused by a human translator.

Since Wolaita language is used as a means of communication in different government and non-government institutions and serving as working language at zone, it benefits non-Amharic speakers of Wolaita people if documents and news articles are automatically translated into Wolaita language. In recent studies, NMT provides promising results than other MT approaches [9, 10, 11, 12]. Currently, the progress in technology is shifting from traditional MT approaches to deep learning-based NMT approach. Despite the recent success of NMT in standard benchmarks, it faces a problem when the length of words in a sentence becomes longer. This issue is solved when using an attention mechanism with NMT. As the researcher's knowledge is concerned, there is no prior study conducted on the development of Amharic-to-Wolaita Neural MT system. We found that NMT is also a very important NLP task that has to be done for the Wolaita language [30]. Thus, we propose attention-based Amharic-to-Wolaita NMT.

1.4 Objectives

General Objective

The general objective of this study is to design and develop an attention-based Amharic-to-Wolaita Neural Machine Translation.

Specific Objectives

The specific objectives are:

1. Review related systems and literature.
2. Develop parallel bi-lingual corpus for Amharic and Wolaita languages.
3. Identify the linguistic behaviors of Amharic and Wolaita languages.
4. Design a general architecture for attention-based Amharic-to-Wolaita NMT.
5. Develop a prototype.
6. Test and evaluate the performance of the system.

1.5 Methods

Literature Review

For the purpose of finding up-to-date methodologies in the MT domain, a thorough literature review will be conducted. For this study, secondary data sources, like books, articles, publications and other resources related to the topic will be reviewed. This helps to have a better understanding of the subject of the study. Studies related to this study will be compiled so as to know the pros and cons of various NMT techniques. MT systems in different languages will be studied with respect to the closeness and difference among the languages. The details of the approaches and algorithms followed to build the translation system will also be reviewed. The linguistic behavior of Amharic and Wolaita languages will also be investigated and identified.

Data Collection

To conduct NMT, a parallel corpus of source and target language is required. The translation system we are going to develop tries to generate translations using the Amharic-Wolaita corpus based on neural network methods. The sources for both languages are mostly from a religious book, and other relevant materials.

Prototype Development

In order to develop a prototype for NMT, some approaches and techniques are needed. Word alignment, reordering and language modeling can be performed with the help of a well-trained deep neural network. Word2vec generates the word-vectors that are used by recurrent auto-encoder in reconstruction task. RNN has the capability to implement reordering rules on sentences.

Evaluation Mechanism

NMT system can be evaluated either using by human (manual) or automatic evaluation methods. Manual evaluation is time-consuming and expensive to perform, BLEU score will be used to evaluate the performance of the system, which is an automatic evaluation technique.

1.6 Scope and Limitations

Attention-based Amharic-to-Wolaita NMT is designed to perform translation of texts written in Amharic to Wolaita.

Because of the unavailability of the standardized corpus (corpus ready for MT research purpose) for the dataset, the dataset we are going to use for training and testing is mostly from religious documents. Thus, words which are not in corpus is not translated in our system.

1.7 Application of Results

The results of this research work have many applications. The system can be used for translation from Amharic-to-Wolaita texts, and the translation system can be used as a tool for the teaching-learning process of the languages. This study can be used to simplify the barrier of language difficulty among language users. It enables to access information and interaction easily and fills the communication gap of speakers; Moreover, the study can be used as a component for other NLP applications such as speech translation. Since the future of NMT focuses on a multi-task learning, larger context, and mobile devices [1,30], this study may be used as input for the next researches.

1.8 Organization of the Rest of the Thesis

This thesis is organized into six chapters. Chapter Two presents a literature review which includes an overview of the languages and MT approaches especially NMT. Chapter Three presents different related works in the MT domain. Chapter Four presents the design of attention-based Amharic-to-Wolaita NMT system. The experiments and results are discussed in Chapter Five and Chapter Six presents conclusions, contribution and recommendations.

Chapter 2: Literature Review

2.1 Introduction

In this Chapter, a brief overview of Amharic and Wolaita languages, and MT in general and NMT in detail are discussed. Additionally, the advancement of NMT over other MT approaches: Statistical Machine Translation (SMT), Rule-Based Machine Translation (RBMT), Example-Based Machine Translation (EBMT), and Hybrid Machine Translation (HMT) are described.

2.2 Overview of Amharic Language

Amharic is one of the Semitic languages spoken in Ethiopia. Next to Arabic, it is the second most-spoken Semitic language in the world and it is the official working language of the Federal Democratic Republic of Ethiopia. It is also the first largest spoken language in Ethiopia and possibly one of the five largest languages on the African continent. The Amharic alphabet is called Fidel, which grew out of the Ge'ez abugida-called in Ethiopian Semitic language. The usual word order of Amharic is Subject-Object-Verb (SOV) [25]. Modern written Amharic uses a unique script called hohiyat (ሆሀያት) which is conveniently written in a tabular format of seven columns [35]. The first order is the basic form; the other orders are derived from it by more or less regular modifications indicating the different vowels [36]. The alphabet is written from left to right in contrast to some other Semitic languages such as Arabic and Hebrew. It consists of 34 consonants, giving 7*34=238 syllable patterns, or fidels [37]. In addition to the 238 characters, there are other non-standard alphabets which contain special features usually representing labialization. Each alphabet represents a consonant together with its vowel. The vowels are fused to the consonant form in the form of diacritic markings. The diacritic markings are strokes attached to the base characters to change their order [38].

2.2.1 Amharic Morphology

Amharic morpheme can be free or bound, where a free morpheme can stand as a word on its own whereas a bound morpheme cannot [39]. An Amharic root is a sequence of consonants and is the basis for the derivation of verbs. On the other hand, a stem is a consonant or consonant-vowel sequence which can be free or bound where a free stem can stand as a word on its own whereas a bound stem has a bound morpheme affixed to it. An example of morpheme could be, the word በልጅነት can be morphologically scrutinized into three separate morphemes: the prefix በ-, the root

ልጅ, and the suffix -ነት. A word, which can be as simple as a single morpheme or can contain several of them, is formed from a collection of phonemes or sounds [40, 41]. In Amharic, words can be formed from morphemes in two ways. These are by inflection and by derivation.

A. Inflection

Inflectional Morphology is a morphology that deals with the combination of a word with a morpheme, usually resulting in a word of the same class as the original stem, and serving the same syntactic function. Inflection can be achieved by marking a word category for gender, number, case, definiteness, aspect and politeness [24, 25]. Since the Amharic language is highly inflectional, a given root of a language word can be found in different forms [24]. The author in [42] states that Amharic word classes belong to five classes. These are noun (ስም), verb (ግስ), adjective (ቅጽል), preposition (መስተዋደድ) and adverb (ተውሳከ ግስ). From these classes, highly inflected parts in Amharic are discussed as follows.

Nouns (ስም): Noun is a name that represents a person, places, animal, thing, feeling and idea [25]. Amharic nouns are marked for gender, case, number, and definiteness, and results in an infected word with affixes to the noun. It can be achieved by marking a word category for number as singular and plural by affixation of morphemes (and vowel changes) or repetition of words as shown in Table 2.1 [43, 44].

Table 2.1: Amharic Noun Plural Formation

Noun in Singular Form	Morpheme	Plural Form
ልብ	-ኦች	ልቦች
ባርያ	-ዎች	ባርያዎች
ጻድቅ	-ን/-ኦች	ጻድቃን/ጻድቃኖች
ገዳም	-ት/ኦች	ገዳማት/ገዳሞች
ዘበነ	እነ-	እነዘበነ
ቅጠል	Plural formation by repetition	ቅጠል-ኣ-ቅጠል[ቅጠላቅጠል]

Amharic nouns are marked for a word category of definiteness and it can be achieved by the affixation of morphemes or vowels based on a number, gender, and/or ending of the noun as shown in Table 2.2 [43, 44].

Table 2.2: Amharic Nouns Marked Definiteness

Indefinite Noun	Number	Gender	Definite noun
ልጅ	Singular	Feminine	ልጅ-ዋ[ልጅዋ]
		Masculine	ልጅ-ኡ [ልጁ]
	Plural		ልጆች-ኡ [ልጆቹ]
በግ	Singular	Feminine	በግ-ዋ[በግዋ] በግ-ኢቱ [በጊቱ]
		Masculine	በግ-ኡ [በጉ]
	Plural		በጎች-ኡ[በጎቹ]

Amharic nouns are marked for a word category of gender as shown in Table 2.3 [43, 44] and it can be achieved by the affixation of the morphemes -ኢት. For example ልጅ-ኢት[ልጂት] በግ-ኢት[በጊት] አሮጌ-ኢት[አሮጊት]. Amharic nouns are marked for a word category of **case** and it can be in both objective and possessive case. Objective case can be achieved by affixation of morpheme -ን. For example, አህያ-ን [አህያን]. Possessive case can be achieved by the affixation of morphemes or vowels based on person, number, gender, and/or ending of the noun (personal pronouns by prefixing የ. የ-ለማ➔ [የለማ]).

Table 2.3: Amharic Nouns Marked for Gender

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

Verb (ግስ) ፦ Generally, Amharic verbs are derived from roots and use a combination of prefixes and suffixes to indicate the person, number, voice (active/passive), tense and gender [45]. A verb possesses two behaviors that make it different from other word classes in Amharic. The first one is, it is placed at the end of an Amharic sentence, and the second one is it has a suffix attached to it indicating the subject of the sentence [42]. **For example: -** ኢትዮጵያ የመጀመሪያ ሳታላይቷን **አ-መጠቀ-ች** /Ethiopia launched the first satellite/ the underlined word is the verb of the sentence with the prefix አ- and the suffix -ች. እሷ ምሳዋን **በላ-ች** /she ate her lunch/ the underlined word is a verb and the suffix "ች" indicates the subject of the sentence is she (እሷ) which is feminine gender. Amharic verbs, in general, show a high degree of inflection since person, case, gender, number, tense, aspect, mood and others are marked on the verb. For example, "አልሰበረንም" indicates: the subject እሱ (third person, masculine, singular), the object እኛን (first person, plural), negation አል.......ም, past tense -ሰበረ.

Adjective (ቅጽል): - An adjective is a word that describes, identifies, or further defines noun or pronoun. Nouns tell about things nature, but adjectives tell about things behavior or characteristics, like shape, size, colour, type, property. Adjectives of Amharic language are marked for gender and number and results in an inflected word with affixes [43, 44]. In Amharic, some of the morphemes that are used to inflect given adjectives are: -ት, -ኦ,-ኢት and ኦች as shown in Table 2.4 [43, 44].

Table 2.4: Amharic Adjectives Inflection

Singular form	Plural form	Prefix	Suffix
ድንበር	ድንበሮች		...ኦች
ሰማያዊ	ሰማያዊያን		..ያን
ማን	እነማን	እነ...	
እናት	እናቶች		..ኦች

B. Derivation

Derivational morphology is a morphology concerned with the way in which words are derived from morphemes through processes such as affixation or compounding.

Nouns: - Amharic nouns can be derived from verbal roots by infixing vowels between consonants like ጥ-ቅ- ም➔ ጥቅም. It can be derived from adjectives by suffixing bound morphemes ደግ (adjective)-ነት (morpheme)➔ ደግነት (derived noun). It can be derived from stems by prefixing or suffixing bound morphemes. For example, ጠቀም-ኤታ➔ ጠቀሜታ. It can be derived from Stem-like verbs by suffixing the bound morpheme -ታ. For example, ደስ-ታ➔ ደስታ. It can be derived from nouns by suffixing bound morphemes. For example, ኃይልእ-ኛ➔ ኃይለኛ. Compound Words (sometimes by affixing the vowels እ and ኦ). For example, Noun + [እ] + Noun. ቤት+[እ]+መንግስት➔ ቤተ-መንግስት.

Verbs: - Verbs are words which indicate action and they take place at the end of clause positions. Amharic verbs take subject markers as a suffix like -ሁ /-hu/ for subject 'I', -ህ /-h/ for subject 'You', ች /-c/ for subject 'She' and so on, to agree with the subject of the sentence [46]. Amharic verbs can be derived from Verbal Roots by affixing the vowel -እ- to produce CእC1C1እ C-, e.g. ስ-ብ-ር ➔ስእብብእር- [ሰበር] and repeating penultimate consonants and affixing the vowels -እ- and -ኣ- to produce CእC1ኣC1C1እC-, e.g, ፍ-ል-ግ➔ፍእልኣልልእግ [ፈላለግ]. Verbal Stems by affixing morphemes አ-, ተ-, አስ e.g, መጠቅ-(Verbal Stem) + አ-(morpheme)➔አመጠቅ.

Adjectives: Amharic adjectives modify nouns or pronouns by describing, identifying, or quantifying words [46]. Adjectives always come before nouns or pronouns which they modify. But all the words that come before nouns cannot always be adjectives. For example: - ይህ ቤት "this house" in this example ይህ "this" precedes the noun ቤት "house" but this doesn't mean ይህ "this" is an adjective, it is a pronoun. Amharic adjective can be derived from verbal roots by infixing vowels between consonants [43]. For example, ድ-ር-ቅ by applying the vowel, ድእርእቅ it gives the adjective ደረቅ. Nouns by suffixing bound morphemes (እኛ፣ ኣማ፣ኣም፣ አዊ). Stems by suffixing bound morphemes (ኣ፣ ኡ፣ ኢታ).

C. Affixation

Affix is a morpheme fastened to a stem or base form of a word, and modifies its meaning or creates a new word [40]. In Amharic affixes can be prefix, suffix, and infix. Prefix is a morpheme added at the beginning of a word whereas suffixes are added at the end to form derivatives. Infixes are inserted in the body of a word causing a change in meaning, which can be easily observed in an iterative and reciprocal aspect of a root word in the Amharic language [40, 42]. Amharic verbs can have up to four prefixes and up to four suffixes as shown in Figure 2.1 [43, 44].

Prefix					**Suffix**			
prep/conj	rel	neg	sbj	ROOT	sbj	obj/def	neg/aux/acc	conj

Figure 2.1: Placement of Affixes in Amharic Verbs

Here the first, second, third, and fourth options of prefix represent preposition or conjunction, relative, negation, and subject in terms of number, gender, person and definiteness respectively. Relative verbs are marked using "የ" /ye-, "የሚ"/yemi-, "እሚ"/Imi- and negation is marked with prefixes like "አይ"/ay-, "አል"/al-, etc. [22]. Similarly, the first and second options of suffix represent subject and object, in terms of gender, number, person, and definiteness, respectively. The third option represents negation or auxiliary or accusation, where negation can be marked with "-ም", auxiliary is usually marked with morpheme "አለ" , and accusative is marked with morpheme "ን"/-n. The fourth option represents conjunction like "ም", "-ስ" etc. [22, 47]. Amharic nouns have up to two prefixes and up to four suffixes. Similarly, the prefix and suffix slots have two and four sub-slots respectively. Figure 2.2 shows the placement of affixes in Amharic nouns [22, 47].

Prefix			**Suffix**			
prep/gen	distrib	STEM	plur	poss/def	acc	conj

Figure 2.2: The Placement of Affixes in Amharic Nouns

The prep/gen option of the prefix represents preposition or genitive. Genitive is marked using morphemes -ye-/"የ-". In the second option of prefix, distributive (*distrib*) is marked using -Iy_e-/"እየ-"morpheme. In the case of suffix, option one represents the number of information. Option two represents possessive or definiteness information. The third and fourth options represent accusative and conjunction, respectively [47].

2.2.2 Amharic Phrases

Phrases are syntactic structures that consist of one or more than one word but lack the subject predicate organization of a clause. These phrases are composed of either only head word or other words or phrases with the head combination. The other words or phrases that are combined with the head in phrase construction can be specifiers, modifiers and complements [46]. In Amharic, phrases are categorized into five, namely noun phrase (NP), verb phrase (VP), adjectival phrase (AdjP), adverbial phrase (AdvP) and prepositional phrase (PP) [43, 44].

A. Noun Phrases

A noun phrase (NP) is a phrase that has a noun as its head. In this phrase construction, the head of the phrase is always found at the end of the phrase. This type of phrase can be made from a single noun or combination of a noun with either other word classes including noun word class or phrases. That means one noun can be a noun phrase. See the following example: አንበሳው ሁለት ላሞችን ገደለ *"the lion killed two cows"* in this sentence, there are two parts: the subject አንበሳው" the loin "and the object with the verb ሁለት ላሞችን ገደለ" killed two cows". Thus, the first part (the subject) is a noun phrase and the second one is a verb phrase. Therefore, the noun phrase in the above example is only the noun አንበሳው" the lion" [46].

A Noun Phrase can be simple or complex [24]. The simple NP construction consists of a single noun or pronoun, for instance, በግ "sheep", መኪና "car", እሱ "he", እሷ "she" are simple NP and do not consist subordinate clauses. A complex NP can consist of a noun with other constituents (specifiers, modifiers and complements) but the phrase must contain at least one sentence [46].

Example: -In the ኢትዮጵያ ያመጠቀችው የመጀመርያው ሳታላይት *(the satellite that Ethiopia launched for the first time)* here ኢትዮጵያ ያመጠቀችው (*that Ethiopia launched*) is a sentence which is a modifier, whereas የመጀመርያው (the first) is the single word that is a complement.

B. Verb Phrases

A verb phrase (VP) is composed of a verb as a head, which is found at the end of the phrase, and other constituents such as complements modifiers and specifiers [43, 44]. ካሳ [ወደ ቤተ ክርስቲያን] ሄደ *"Kassa went to church"* [ወደ ቤተ ክርስቲያን] is a prepositional phrase modifying the verb ሄደ

‘went’ from place point of view. In general, the structural rule of verb phrases can be formulated as VP => PP V|V|AdjP V|NP VP|NP PPVP|AdvP PP VP.

C. Adjectival Phrases

Amharic Adverbial phrases (AdvP) are made up of one adverb as head word and one or more other lexical categories including adverbs itself as modifiers. The head of the AdvP is also found at the end. Unlike other phrases, AdvPs do not take complements. Most of the time, the modifiers of AdvPs are PPs that come always before adverbs [43, 44, 46].

Example: - ካሳ [እንደ አባቱ ክፉኛ] ታመመ *(Kassa is severely sick like his father)*. Here the phrase in the bracket is adverbial phrase and the head word is ክፉኛ (severely). The modifier that is found in the AdvP is እንደ አባቱ (like his father) which is comparative PP. Generally, the structural rule for Amharic languages can be formulated as follows: AdjP=>Adj|Spec Adv Adj|PP Adj|NP Adj.

D. Prepositional Phrase

Amharic prepositional phrase (PP) is made up of a preposition (Prep) head and other constituent such as nouns, noun phrases, prepositional phrase, etc [6, 43, 44]. Unlike other phrase constructions, prepositions cannot be taken as a phrase. Instead they should be combined with other constituents and the constituents may come either previous to or subsequent to the preposition which is the head of the phrase [46]. In a prepositional phrase, if the complements are nouns or NPs, the position of prepositions is in front of the complements. For example, እንደ ትልቅ ልጅ*” like a big child”* እንደ “like” is a preposition which is combined with the noun “ልጅ*”/ child and it came in front of complement”* ትልቅ*”/big. whereas if* the complements are PPs, the position will shift to the end of the phrase. For example, ከወንዙ አጠገብ” Next *to the river”,* አጠገብ is PP which is combined with the noun “ወንዙ*”/ river and it came at the end.* In general, the structural rule for Amharic can be written as: PP => PP PP |PP NP|PP NN |PPV|N PP

E. Adverbial Phrases

Amharic Adverbial phrases (AdvP) are made up of one adverb as head word and one or more other lexical categories including adverbs itself as modifiers [4, 43, 44]. The head of the AdvP is also found at the end. Unlike other phrases, AdvPs do not take complements. Most of the time, the modifiers of AdvPs are PPs that come always before adverbs [46].

Example: - ካሳ [እንደ አባቱ ክፉኛ] ታመመ *(Kassa is severely sick like his father).* Here the phrase in the bracket is adverbial phrase and the head word is ክፉኛ (severely). The modifier that is found in the AdvP is እንደ አባቱ (like his father) which is comparative PP. The general structural rule for an adverbial phrase is: AdvP => Adv|Adv Adv

2.2.3 Amharic Sentences

An Amharic sentence is formed from a noun phrase (NP) and a verb phrase (VP). Regarding their order in a given sentence, NP comes first then VP follows [35]. The sentence structure for the Amharic language is a Subject-Object-Verb (SOV), unlike English with a subject-verb-object combination [4, 43, 44]. We can take the following as an Example, አበበ ምሳውን በላ/Abebe ate his lunch, in this example, the sentences are composed of አበበ: subject, ምሳውን፡Object and በላ: Verb which is different from English. Amharic sentences are constructed from simple or complex NP and simple or complex VP but NP always comes first as a subject [46].

Example: - *አብዛኛው ፖለቲካ ፓርቲዎች "many political parties". ወደ ኢትዮጵያ ገብተዋል "went to Ethiopia". አብዛኛው ፖለቲካ ፓርቲዎች ወደ ኢትዮጵያ ገብተዋል "many political parties went to Ethiopia"*. The first two constructions do not express a full idea but the last one does. Because the last one expresses full information such as who did go to Ethiopia? Where the political parties went? etc. All these questions have been answered by the last word construction. In the last construction there are NP and VP which build the sentence and these are NP *አብዛኛው ፖለቲካ ፓርቲዎች* "*many political parties"* and VP *ወደ ኢትዮጵያ ገብተዋል "went to Ethiopia"*. The remaining phrases (other than NP and VP) are being constructed in NPs or VPs that are found in a sentence. Based on this construction, sentences can be simple or complex.

A. Simple Sentences

Simple sentences are sentences, which contain only one verb. A simple sentence can be constructed from NP followed by VP which only contain single verb [4, 43, 44].

Example 1: -አስቴር ብርጭቆውን ሰበረችው *"Aster broke the glass"*. Here the sentence contains only one verb ሰበረችው "she broke". This sentence contains transitive verb ሰበረችው "broke" that takes only one object ብርጭቆውን "the glass".

Example 2: - አስቴር ለካሳ መጽሀፍ ሰጠችው *"Aster gave Kassa a book"*. Here also the sentence contains only one verb ሰጠችው "gave" so it is a simple sentence. The difference in this example from the previous one is the sentence here contains transitive verb ሰጠችው "gave" with two objects ለካሳ "Kassa" and መጽሀፍ "book". Generally, all the above stated examples are simple sentences that contain different types of verbs. Simple sentences may contain intransitive verbs, transitive verbs with one object and transitive verbs with two objects. As explained in [4, 43, 44] simple sentences are classified into four namely: declarative sentences, interrogative sentences, negative sentences and imperative sentences.

Declarative sentences are used to convey ideas and feelings that the speaker has about things, happenings, etc., that could be physical, mental, real or imaginary. In Amharic, declarative sentences always end with the Amharic punctuation mark "።" which is equivalent of period (.) in English. Example: - አስቴር ትምህርት ቤት ውስጥ ነች።/ *"Aster is at school."* Here the sentence is declarative because it describes where Aster is.

Interrogative sentence is a sentence that questions about the subject, the complement, or the action that the verb specifies. Example: - አስቴር ምን ውስጥ ነች? In the above example, the question is for the unknown thing just to get full information about it. This type of question or interrogative sentences consist interrogative pronouns which are ማን "who", መቼ "when", ምን "what", ስንት "how many", የት "where", etc.

Negative sentences simply negate a declarative statement made about something. Example: - ካሳ በግ አልገዛም *"Kassa didn't buy a sheep"* In this example, the sentence is a negative declarative sentence. The verb አልገዛም "did not buy" is negated by the prefix አል-"not".

Imperative sentences convey instructions and mostly their subject is a second person pronoun that is usually implied by the suffix on the verb. But when the command is passed for the third person, the subject of the sentence can be third person pronoun or noun. Example: - ውጥ አምጪ *(bring wat)* Here the subject is (you) second person feminine singular. Example: - ካሳ ልብስ ይጠብ *(Kassa, wash clothes)* Here the command is for the third person that does not exist at the time the order is transferred. So, the subject is (he) third person singular masculine.

B. Complex Sentences

Complex sentences are sentences that contain at least one complex NP or complex VP or both complex NP and complex VP [4, 43, 44]. Complex NPs are phrases that contain at least one embedded sentence in the phrase construction. The embedded sentence can be complemented. Example: - [ኢትዮጵያ ያመጠቀችው የመጀመሪያው ሳተላይት] በጣም ዉጤታማ ነች *"the first satellite that Ethiopia launched is very effective".* Here the head of the noun phrase [ኢትዮጵያ ያመጠቀችው የመጀመሪያው ሳተላይት "that" is ሳተላይት] "*the first satellite that Ethiopia launched*". The head with the complement የመጀመሪያው "the first" form simple noun phrase የመጀመሪያው ሳተላይት "the first satellite" and this noun phrase has been combined with the embedded sentence or clause ኢትዮጵያ ያመጠቀችው "that Ethiopia launched" to form a complex noun phrase.

2.2 Overview of Wolaita language

Wolaita language belongs to the Omotic language family among Ethiopian language families (such as Semitic, Cushitic, Nilotic and Omotic). It is one of the main languages of the Ometo group of the Omotic (named as 'west Cushitic' in [26]) family, which belongs to the Afro-Asiatic language phylum. It is spoken by Wolaita people and some other parts of the Southern Nations, Nationalities, and People's Region of Ethiopia such as Gamo, Gofa, Mello, Kucha, and Dawro [14, 50]. It is because the people of Wolaytta are surrounded in the west and in the south by populations (such as the Dawro, the K'uc'a, the Borodda and the Gamu) who speak another Ometo dialect and thus an idiom very similar to Wolaytta [26]. The native people call the language "Wolaytta" (wolaittáttuwa in local language). The language is also referred to as woláítta dóónaa (ወላይታ ኣፍ) or woláítta Káálaa (ወላይታ ቃል). Wolaita is the working language of the Wolaita Zone. Currently, primary, secondary and higher education institutions, as well as different mass media, are using the Wolaita language in teaching and learning, and information transformation processes [14].

Wolaita language uses the Latin script for writing called WPXW which literally mean “The custom in writing the Wolaytta letters”. [26, 34, 50]. WPXW was published in 1985 E.C by Wolaytta Qaala Hiwote Maatteme Keettaa (WQHMK) [26, 34, 50]. WPXW’s writing system is as convenient as Qubee, which is a system based on the Latin alphabet for writing the Afan-Oromo language [34, 50]. According to [26], Wolaita was first written in the late 1940s by a team of missionaries led by Dr. Bruce Adams. They translated the Bible into Wolaita in 2002. The language is used in social, political and economic activity. At the primary school level, the language is used as a medium of instruction and taught as a subject in secondary and high school. Currently, the language is offered as a subject at Wolaita Soddo University and Arbaminch TVET College. But it was planned for the future to be thought in all universities of Ethiopia. The language was written as Wolaytta, Wolayta, Wolaita, Woleyta, Ometo by different writers in different time [50]. But in our case, we use Wolaita for our study.

2.2.4 Wolaita Morphology

Morphology is a branch of linguistics that studies and describes how words are formed in language [4, 43, 44]. Like Amharic, there are two categories of morphemes in Wolaita: free and bound morphemes. A free morpheme can stand as a word on its own whereas bound morpheme does not occur as a word on its own. Affixation and compounding are two basic word-formation processes in Wolaita [14, 49]. Affixation is a process by which affixes are added/attached in some manner to the root, which serves as a base. Affixes are morphemes that cannot occur independently. Prefix, suffix, and infix are the three types of affixes. Wolaita language does not have prefix and infix. Instead, Suffixation is the basic way of word formation in Wolaita. In forming a word, adding one suffix to another is common in Wolaita. This process of adding one suffix to another suffix can result in relatively long word, which often contains an amount of semantic information equivalent to a whole Amharic phrase, clause or sentence [14, 50]. The second word-formation process in Wolaita is compounding. According to [14], compounding is the joining together of two linguistic forms, which function independently. Although Wolaita is very rich in compounds, compound morphemes are rare in Wolaita and their formation process is irregular [49]. As a result, itis difficult to determine the stem of compounds from which the words are made. As discussed in [14, 49], there are two kinds of morphology in Wolaita language: inflectional and derivational.

A. Inflectional Morphology in Wolaita

Inflectional morphology is concerned with the inflectional changes in words where word stems are combined with grammatical markers for things like a person, gender, number, tense, case and mode [14, 50]. It does not result in changes of parts of speech. Like Amharic language, Wolaita is highly inflectional, a given root of a language word can be found in different forms [14, 49]. Highly inflected word classes in Wolaita are discussed as follows.

Nouns:

Based on the syllable formulation of Lamberti and Sottile in [49] Wolaita nouns have the form C1V1C2V2, where C and V stand for consonant and vowel respectively. However, C1 represents a glottal stop while V1 seldom represents diphthong (diphthongs are sounds formed by the combination of two vowels in a single syllable). In the same way, C2 may represent simple or germinated constant and V2 either represents ending in the absolute case or ending required by the syntactic function associated with noun stems. Most Wolaita words are bi-radical (the number of consonants in the word). However, there are some pluri-radical words. See Table 2.5 [48, 49] for Wolaita noun formation.

Table 2.5: Noun Formation in Wolaita Language

Character sequence	Example	Character sequence	Example
V	Aguntta/እሾክ/ thorn	CVCV	Kayisoi/ሌባ / Theaf
VV	Aawaa/አባት/father	CCVCCV	Shappa/ወንዝ /river
VC	Intarssa/ምላስ/tongue	CVVCCVV	Keettaa/ቤት/house
VCC	oydda/አራት/four	CVCCVV	Mattaa/ንብ/bee
CVV	Maataa/ሳር/grass	CVC	Zokkuwa/ጀርባ/back

As Demewoz Beldados discussed in [50], Wolaita nouns are formed from stems and suffixes in which the stems never change while suffixes exhibit changes. Table 2.6 [48, 49] is an illustration of his discussion.

Table 2.6: Wolaita Nouns Formation from Stems and Suffixes

Nouns	Stems	Suffixes
keett-a/ቤት/house	Keett-	-a
keett-I/ቤት/house	Keett-	-i
keett-eta/ቤት/house	Keett-	-eta
na?-eti/ልጅ/ a boy	na?-	-eti

According to the ending they take in inflection Wolaita noun is classified into four major classes [49]. First class nouns are nouns ending in –a in absolute case with the stress on the last syllable. The second class nouns are nouns that have an absolute case ending in –iya and the stress is on the penultimate. The third and the fourth class nouns are nouns ending in uwa and –iyu, respectively, in the absolute case as shown in Table 2.7 [48, 49].

Table 2.7: The Third and the Fourth class of Wolaita Nouns

Noun Class	Ending	Examples
First	-a	Aw**a**/ጸሃይ/sunrise፣ xum**a**/ጨለማ/darkness፣ ?aaw**a**/አባት/father
Second	-iya	Bitann**iya**/ወንድ ያገባ/married man፣ayf**iya**/ዓይን/eye፣kussh**iya**/እጅ/hand
Third	-uwa	Kaw**uwa**/መንግስት/government፣ met**uwa**/ችግር/'trouble፣
Fourth	-iyu	Kan**iyu**/ሴት ዉሻ/dog feminine፣bolloti**yu**/ሴት አማች/'mother-in-law

Wolaita nouns are marked for gender, number, and case [14, 49].

Gender: As indicated in [49] Wolaita, like other languages, exhibits two types of genders, i.e., masculine and feminine. According to the author, nouns that belong to the fourth class are feminine and the rest (first, second and third class) belong to masculine. Feminine and masculine differ from each other by their endings. While feminine ends in –iyo, masculine ends in –an in absolute case.

Example: Dorss-a /ወንድ በግ, masculine vs. dorss-iyo/በግ, feminine. Deeshsh-a /ወንድ ፍየል, masculine vs. deessh-iyo ፍየል, feminine. Addiy-a/አዉራ ዶሮ, masculine indd-iyo/ ዶሮ, feminine.

Number: According to [49], Wolaita noun contains singular and plural. The plural noun is formed by using suffixes and singular contains the basic form of the word. Example: Dorss-a በግ, singular vs. dorssa-ta/በጎች, plural. Deeshsh-a /ወንድ ፍየል, singular vs. deessha-ta ፍየሎች, plural. Addiy-a/አዉራ ዶሮ, singular adde-ta/ ዶሮዎች, plural.

Those of the 2nd class, instead, exhibit the ending –e-ta.

Noun (Singular) Noun (plural)

har-iya (donkey) ➔ har-e-ta 'donkeys'

?org-iya (he-goat) ➔?org-e-ta 'he-goats'

laagg-iya (friend) ➔ laagg-e-ta. 'friends

Those of the 3rd class are characterized by the ending –o-ta

Noun (Singular) Noun (plural)

Word-uwa (lie) ➔ word-o-ta 'lie pl.')

The nouns of the 4th class which consist of terms for female beings assume the plural form of heir masculine counterpart.

Noun (Singular) Noun (plural)

?imatt-iyu 'female guest'➔ ?imatt-a-ta 'female guests'

boogaaanc-iyu 'female robber' ➔ boogaanc-a-ta 'female robbers'

laagg-iyu 'female friend' ➔ laagg-e-ta 'female friends'

If the feminine noun does not have any masculine counterpart, then it normally agrees with the nouns of 2nd class and ends in –e-ta.

Noun (Singular) Noun (plural)

macc-iyu (wife) ➔ macc-e-ta 'wives'

misshir-iyu (married woman) ➔ misshir-e-ta 'married women'

Case: The noun inflection takes place by the suffixation of case endings to the noun stem or to the absolutive case form [50]. Accordingly, the absolutive case is characterized, as we have already seen above, by the ending –a (1st class and plural), -iya (2nd class), -uwa (3rd class) and –(i)yu (4th class) respectively, while the subject case ends in –y (first three classes), -i (plural), and –(i)ya (4^{th} class). The genitive is represented either by the noun stem alone or is more often characterized by the lengthening of the final vowel of the absolutive form. The object case of the noun inflection agrees with the respective absolute case [14, 49, 50]. Table 2.8 [48, 49] details Wolaita nouns case marker.

Table 2.8: Wolaita Nouns Case Marker

Case marker (morphemes)	Function	Example
-ssi, -w or –yoo	dative and benefactive case	Garssassi**ssi**/ለውስጠኛው
-kko or -mati	directive case	Garssakko/ወደ ዉስጥ
-ni	locative	Garsani/በዉስጥ በኩል
-ppe	ablative	Garsaappe/ከዉስጥ
-ra	comitative case	Garssara /ከታችኛው ጋር

Adjectives

According to [49], adjectives in Wolaita language are used to qualify nouns and they come before the noun they qualify. They are also defined as words that modify nouns by expressing their qualities, colour, size, etc. Adjectives in Wolaita language end in -a, -iy or –uw as shown in Table 2.9 [48, 49].

Table 2.9: Adjectives in Wolaita

Adjectives ending in -a:	Adjectives ending in -iya:	Adjectives ending in -uwa:
geessh-a /ንጹህ	mal-iya /ጣፋጭ	Lo7-uwa /ጥሩ
cinc-a /ብልጥ	haankett-iya /ልዩ	yuush-uwa /ዙሪያ
qant-a /አጭር	iiti-iya /መጥፎ	Im-uwa /መስጠት

Adjectives in Wolaita language precede the noun they modify and remain unchanged when used in attributive position because they do not have to agree in Wolaita with their governing noun either in gender or in number or in case [26, 49]. But most adjectives ending in –**uwa** and a few in –**iya** are replaced by the endings **-o** and **-e** respectively.

Example: Lo7-uwa /ጥሩ➔_lo7-o asa /ጥሩ ሰው

haah-uwa /ሩቅ➔_ haah-o sohuwa /ሩቅ ቦታ

Lamberti and Sottile [49] also reported that when adjectives are used in predicative position, **-uwa** will be changed to -o. For example: Lo7-uwa /ጥሩነት➔_ lo7-o/ጥሩ: he bitane lo7-o ➔ያ ሰውየው ጥሩ ነው. But every word that comes before a noun is not necessarily an adjective. For example, in the sentence **Ha dorsai taga "ይህ በግ የኔ ነው** "here the word **Ha** 'ይህ' has the role of an adjective but it is demonstrative determiner [26].

Verbs

As pointed in [26] verbs of Wolaita, like most Ethiopian languages, are very complex. Wolaita verbs usually have a consonant-vowel-consonant sequence. For example, uya-ጠጣ, gela/ ግባ. Some of Wolaita verbs are borrowed from the Amharic language. Eamples: azzaza-እዘዝ, nabbaba-ንባብ, kassas- ክሰስ.

Verbs in Wolaita language exhibit a very complex inflection system depending on mood, tense, kind of action and aspect. For example, if we take verb 7imm-a (ስጥ), it has the following inflection for past tense:

imm-a:si ➔ሰጥቸዋለሁ/ እኔ ሰጥቸዋለሁ, imm-adasa ➔ሰጥተሃል, imm-a:su ➔ሰጥተሻል

imm-i:si ➔ሰጥቱዋል, imm-ida ➔ሰጥተናል, imm-ideta ➔ሰጥታቹዋል, imm-idosona ➔ሰጥተዋሉ….etc.

We can produce many inflected verbs for future and present tense in the same way as we do for past tense. For example imm-a:si➔ ሰጥቸዋለሁ/ እኔ ሰጥቸዋለሁ(past tense)

imm-a-is➔እየሰጠሁኝ ነው(present tense)

imm-a-na➔እሰጠዋለሁ(future tense)

Just like that of Amharic language, Wolaita verbs are found at the end of the sentence and suffix bound morphemes which help to indicate the subject of the sentence in the sentences shown below.

Na7-ya mayuwa shama-su /ልጅቱዋ ልብስ ገዛች

neeni ne kawuwa ma-dasa /እራትክን በልተሃል

tani 7osuwa wursa-si /ስራዬን ጨርሸዋለሁ

In the above three sentences, shama-su, ma-dasa, wursa-si are verbs. The bound morphemes {-su},{-dasa},{-si} show 3rd person, 2nd person and 1st person pronouns that are used as subjects of the sentence. Verbs in Wolaita language change their shape for a person, gender, number and time by attaching suffixes [14, 49, 50].

B. Derivational Morphology in Wolaita

Nouns: Wolaita nouns are formed by the class suffixes -a and -uwa can refer either to abstract terms or to very concrete objects and they also serve to express action nouns as shown in Table 2.10 [14, 49, 50].

Table 2.10: Wolaita Noun Derivation

Noun	**Suffix**	**Derived noun**
hassay- 'speak	-a	Hassaya/conversation
harg-‘sick’	-ya	harg-iya ‘sickness’
gulba-‘knee’	-ta	gulba-ta ‘knee’
wurse- 'end	-tta	wurse-tta 'end
eeyya- 'stupidity'	-tetta	eeyya-tetta 'stupidity'
kaawo-tetta 'kingdom'	-tetta	kaawo-tetta 'kingdom'

Verbs: Like other Cushitic and Semitic languages of Ethiopia, Wolaytta makes use of some morphemes in order to derive a further stem from a root or a stem. This derivation procedure is concretely applied in Wolaytta by suffixing one or more morphemes to verbal stem [14, 49, 50]. They further indicated that it is possible to form three different kinds of secondarily derived verbal stems: iterative (or intensives), causatives and passives (or reflexives) [49, 50]. Iteratives and intensives stems are expressed in Wolaytta by means of the same morpheme –erett-, which is regularly suffixed to its verbal stem. Example: -ment- 'to break' ➔ ment-erett- 'to break many times or in many pieces'. shissh- 'to collect' ➔ shissh-erett- 'to collect many times or in many things'. The Wolaytta language possesses only productive causative morpheme, i.e –is-s.

gel- 'enter'➔ gel-iss- 'let someone enter/put into’.

Verbs which have their primary stem ending in -y- or -y-y- form their causative by replacing all existence of -y- by -sh-.

Example: uy-y- 'drink' ➔ ush-sh- 'let someone drink.

yuuy-y- 'turn, intr'➔ yuush-sh- 'turn,tr

Adjectives: Wolaita adjectives can be derived from verbal roots by suffixing the morphemes like –ta. For example, imma+ -ta ➔imota. It can be also derived from nouns, stems by suffixing bound morphemes, and compound words.

C. Affixation

According to [49] there are three types of affixes (prefix, infix and suffix), Wolaita uses suffixation to form words and prefixes and infixes are not used as word formation in Wolaita language. We already discussed inflectional and derivational affixation of Wolaita language in the previous sections.

2.2.5 Wolaita Phrases

In Wolaita, phrases are categorized into five categories, namely noun phrase (NP), verb phrase (VP), adjectival phrase (AdjP), adverbial phrase (AdvP) and prepositional phrase (PP) [26, 49].

A. Noun Phrases

A noun phrase (NP) is a phrase that has a noun as its head. In this phrase construction, the head of the phrase is always found at the end of the phrase. This type of phrase can be made from a single noun or combination of a noun with either other word classes including noun word class or phrases. A NP can be simple or complex [49]. The simple NP construction consists of a single noun or pronoun. A complex NP can consist of a noun with other constituents (specifiers, modifiers and complements) but the phrase must contain at least one sentence [49]. Example: - Itiyoophiya yeddido koyro satalaytiya*(the satellite that Ethiopia launched for first time)* here Itiyoophiya yeddido (*that Ethiopia launched*) is a sentence which is a modifier, whereas koyro satalaytiya (the first satellite) is the single word that is a complement.

B. Verb Phrases

A verb phrase (VP) is composed of a verb as a head, which is found at the end of the phrase, and other constituents such as complements modifiers and specifiers [4, 26, 49]. ካሳ [ወደ ቤተ ክርስቲያን] ሄደ *"kaasi woosa keettaa biisi"* [ወደ ቤተ ክርስቲያን] is prepositional phrase modifying the verb ሄደ 'biisi from place point of view. In general, the structural rule of Wolaita verb phrases can be formulated as VP => PP V|V|AdjP V|NP VP|NP PPVP|AdvP PP VP.

C. Adjectival Phrases

Wolaita Adverbial phrases (AdvP) are made up of one adverb as head word and one or more other lexical categories including adverbs itself as modifiers. The head of the AdvP also found at the end. Unlike other phrases, AdvPs do not take complements. Most of the time, the modifiers of AdvPs are PPs that comes always before adverbs [26, 46, 49]. Example: - kaasi [ba aawagadan] keehippe harggiis/ ካሳ [እንደ አባቱ ክፉኛ] ታመመ. Here the phrase in the bracket is adverbial phrase and the head word is ክፉኛ (keehippe). The modifier that is found in the AdvP is እንደ አባቱ (ba aawaadan) which is comparative PP. Generally, the structural rule for Wolaita languages can be formulated as follows: AdjP=>Adj|Spec Adv Adj|PP Adj|NP Adj.

D. Prepositional Phrase

Wolaita prepositional phrase (PP) is made up of a preposition head and other constituents such as nouns, noun phrases, prepositional phrase, etc[4, 26, 49]. In PP, if the complements are nouns or NPs, the position of prepositions are in front of the complements. In general, the structural rule for Amharic can be written as: PP => PP PP |PP NP|PP NN |PPV|N PP

E. Adverbial Phrases

Wolaita Adverbial phrases (AdvP) are made up of one adverb as head word and one or more other lexical categories including adverbs itself as modifiers [4, 26, 49]. The head of the AdvP also found at the end. Unlike other phrases, AdvPs do not take complements. Most of the time, the modifiers of AdvPs are PPs that comes always before adverbs [4, 26, 49]. Example: - ካሳ [እንደ አባቱ ክፉኛ] ታመመ *(kaasi ba awwagadan keehippe harggiis)*. Here the phrase in the bracket is adverbial phrase and the head word is ክፉኛ (keehippe). The modifier that is found in the AdvP is እንደ አባቱ (*ba awwagadan*) which is comparative PP.

The general structural rule for adverbial phrase is: AdvP => Adv|Adv Adv

2.2.6 Wolaita Sentences

Like Amharic, Wolaita language has also two kinds of sentences: simple and complex sentences.

A. Simple Sentences

Simple sentences a contain only one verb. A simple sentence can be constructed from NP followed by VP which only contain single verb [4, 43, 44]. For example: - Abebe kuwaasiya kaa7iis/Abebe played football. Here the sentence contains only one verb kaa7iis /played.

B. Complex Sentences

Complex sentences are sentences that contain at least one complex NP or complex VP or both complex NP and complex VP [4, 43, 44]. Complex NPs are phrases that contain at least one embedded sentence in the phrase construction. The embedded sentence can be complements. See the following examples. For example: - Abebe kuwaasiya kaa7i simmidi so oosuwa oottis/ Abebe did his homework after playing football. Here the sentence contains two verbs kaa7i/ played and oottis/ did.

2.3 Machine Translation

A language is used for conveying information or broadcasting the information. Stepping into the modern digital age, language as the information carrier has become the most significant means for a human to communicate. But it has been considered as the barrier of communications between people from different countries and between peoples who speak different languages within the same country. The problem of converting a language into another quickly and efficiently has become a problem of common concern for humanity [51]. Due to the advent of the computer and the Internet the world is becoming together to one [52]. Thus, the knowledge, culture, tradition, history, religious, philosophy documents of one country language can be translated into another language and the rest of the world via MT. To create a paperless working environment, translation plays a great role and to make accessible the document of one language in another language. Sharing of knowledge is also possible besides facilitating easy communication.

MT is an automatic translation of one language into one or more languages (in case of multi-lingual) by means of a computer. High-quality translation requires a thorough understanding of the source text and its intended function as well as good knowledge of the target language. Translation itself is a challenging task for humans and is no less challenging for computers because it deals with natural languages [53]. This section provides background information regarding the field of MT, its history in general and NMT in detail, and various MT approaches.

2.4.1 History of Machine Translation

Although there are some disputes about who first had the idea of translating automatically between human languages, the actual development of MT system can be traced back to the late forties after World War II [45]. In this time, MT was constrained by several factors: limitation of hardware particularly, inadequacy of memories and slow access and unavailability of high-level programming language. The linguistic study was not correlated with MT research. So, researchers relied on the dictionary-based approach and the application of statistical methods [54].

Researchers of that time were faced with a lot of technical constraints and realized that there could be no perfect high-quality translation, and suggested the involvement of humans in the process. They also proposed the development of controlled languages and restriction of systems to specific domains. Criteria concerning the success and failure of MT were set in its first 50 years of research and development. These criteria are the conceptual, engineering, operational, commercial and communicative criteria [55].

An American mathematician and scientist named Warren Weaver in 1947 had a belief that a computer is capable of translating one NL to another by using logic, cryptography, frequencies of letter combinations, and linguistic patterns [52]. He published a memorandum that outlines his belief. In the 1950s a research program at Georgetown University teamed up with IBM to perform research on MT. Later in 1954, they demonstrated a system that translates a few phrases from Russian to English. The research resulted in wide acceptance and interest in the field [52].

In 1966 the US sponsors of MT research committee called Automatic Language Processing Advisory Committee (ALPAC) published an influential report which concluded that MT was slower, less accurate and twice as expensive as human translation. However, in the following decade, MT research took place largely outside the United States, in Canada and in Western Europe and work continued to some extent [45].

In the 1970s and 80s researchers shifted their focus to assisting MT rather than replacing human translators. It has three main strands: first, the development of advanced transfer systems building upon experience with earlier Interlingua systems; secondly, the development of new kinds of Interlingua systems; and thirdly, the investigation of AI techniques and approaches. That resulted in the development of translation memory and many computer assisted tools (CAT) for MT. At the end of the 1980s, MT entered a period of innovation in methodology which has changed the framework of research. In 1981 came the first translation software for the newly introduced personal computers, and gradually MT came into more widespread use [45].

During the late 1980s, MT advanced rapidly on many fronts. The dominance of the rule-based approach waned in the late 1980s with the emergence of new methods called `corpus-based' approaches, which did not require any syntactic or semantic rules in text analysis or selection of lexical equivalents. The major reason for this change has been a paradigm shift away from linguistic/rule-based methods towards empirical/data-driven methods in MT. This has been made possible by the availability of large amounts of training data and large computational resources.

In the 1990s, the use of MT and translation aids by large corporations has grown rapidly. Particularly impressive increase is seen in the area of software localization (i.e., the adaptation and translation of equipment and documentation for new markets). On the research front, the principal areas of growth are seen in example-based and statistical machine translation approaches, in the development of speech translation for specific domains, and in the integration of translation with other language technologies.

In the 2000s, the research state was moved toward combining the rule-based and SMT paradigms. It was an approach developed by taking advantage of both statistical and rule-based approaches. After the late 2000s the state of MT shifted to NMT. Recently, it has gained popularity in the field of MT. NMT is starting to displace its corpus-based predecessor, SMT. It has proven to have better efficiency in the MT domain with the goal of improving the quality of the output as well as the performance of the system. With the emergence of the Internet and cheap and powerful computers accelerated the progress of MT. Nowadays researches are focused on improving the quality and performance of MT systems [52].

2.4.2 Approaches to Machine Translation

A machine translation system first analyses the source language input and creates an internal representation. This representation is manipulated and transferred to a form suitable for the target language. Then at last output is generated in the target language [56]. Based on the degree of dependence of internal representation on the source and target languages, MT can be classified into three approaches [6]: Rule-Based Machine Translation (RBMT), Corpus-based Machine Translation Approach (Corpus based MT), and Hybrid Machine Translation (Hybrid MT).

A. Rule-Based Machine Translation (RBMT)

RBMT is also known as Knowledge-Based Machine Translation or Classical Approach of MT. It is a general term that denotes machine translation systems based on linguistic information about the source and target languages basically retrieved from (bilingual) dictionaries and a collection of rules called grammar rules covering the main semantic, morphological, and syntactic regularities of each language respectively [25]. In this approach, human experts specify a set of rules to describe the translation process, so that an enormous amount of input from human experts is required [25]. It consists of a bilingual or multilingual lexicon, and software programs to process the rules. The rules play a major role in various stages of translation such as syntactic processing, semantic interpretation, and contextual processing of language [56].

The basic principle of RBMT methodologies is to apply a set of linguistic rules in three different phases [57]: analysis, transfer and generation. The core process (transfer) is mediated by bilingual dictionaries. RBMT systems parse the source text and produce an intermediate representation as shown in Figure 2.3 [24]. Rules for transforming source language structures into target language structures is from dictionaries and rules for deriving (intermediary representations) from which output can be produced. The preceding stage (analysis) interprets input source language strings into a suitable translation unit. The succeeding stage of synthesis (generation) derives target language output text from the target language structures or representations generated by the transfer process [2, 56].

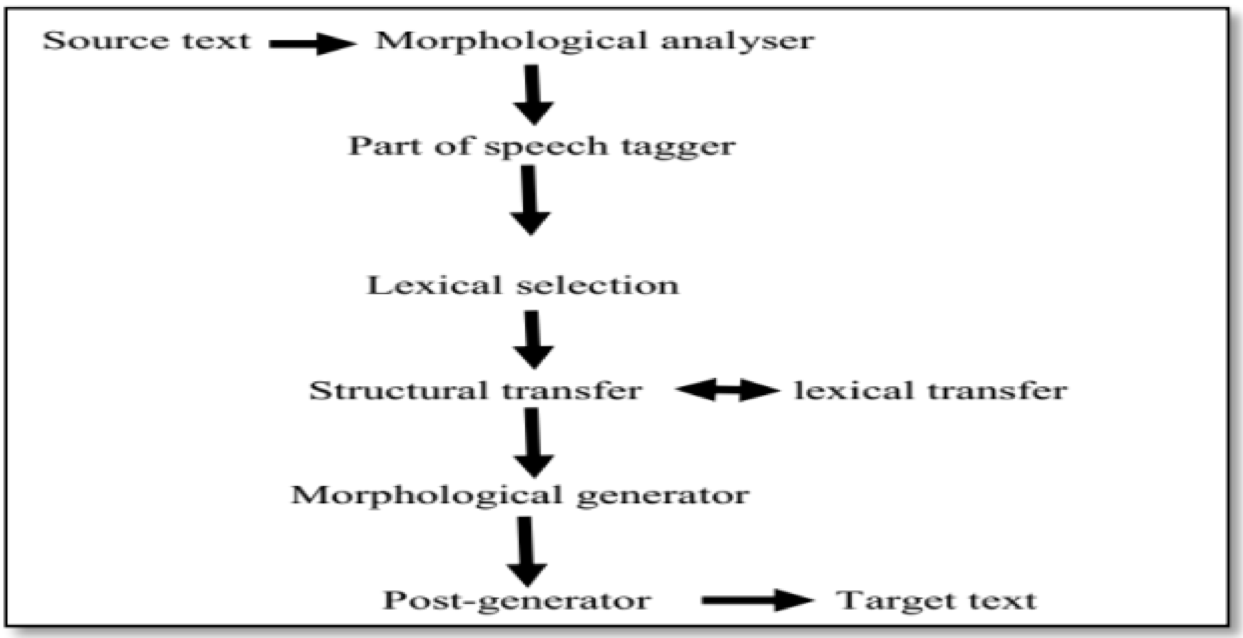


Figure 2.3: Architecture of RBMT Approaches

Based on the intermediate representation used this approach is further classified into the following approaches [56, 58]: Direct, Interlingua and Transfer-Based MT approaches.

i. Direct Machine Translation

Direct MT approach is historically the earliest and known as the first generation of MT systems employed around the 1950s to 60s when a need for MT was mounting. [45]. MT systems that use this approach can translate a source language directly to target language. Words of the source language are translated without passing through an additional/intermediary representation as shown in Figure 2.4 [24]. No complex architecture will be involved. It carries out word by word translation with the help of a bilingual dictionary usually followed by some syntactic rearrangement. Due to this direct mapping, such systems are highly dependent on both the source and target languages [59]. It needs only a little syntactic and semantic analysis and it is basically bilingual and uni-directional.

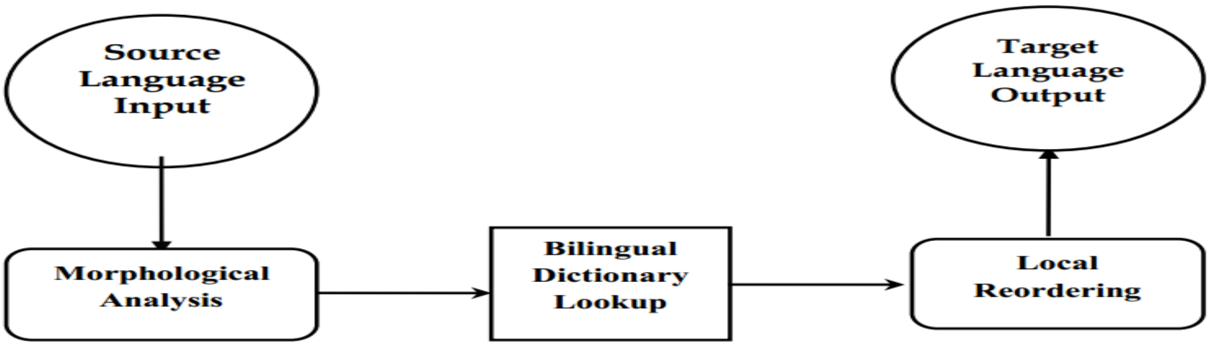


Figure 2.4: Major Tasks in Direct Machine Translation Approach

ii. Interlingua Approach

Interlingua approach to MT mainly aims at transforming the texts in the source language to a common representation which is applicable to many languages. Using this representation, the translation of the text to the target language is performed and it should be possible to translate to every language from the same Interlingua representation with the right rules [34]. In this approach, the source language is transformed into the Interlingua language representation that is independent of any language. The target language is then generated out of the Interlingua [57]. In short, the translation in this approach is a two-stage process, i.e., analysis and synthesis [51]. The first stage is particular to the source language and doesn't require any knowledge about the target language whereas the second stage is particular to the target language and doesn't require any knowledge of the source language. The main advantage of interlingua approach is that it creates an economical multilingual environment that requires 2n translation systems to translate among n languages wherein the other case, the direct approach requires n(n-1) translation systems [34].

iii. Transfer-based Machine Translation

The core idea of both transfer-based and interlingua-based MT is the same: to make a translation it is necessary to have an intermediate representation that captures the "meaning" of the original sentence in order to generate the correct translation. The difference is in interlingua-based MT this intermediate representation must be independent of source and target languages, whereas, in transfer-based MT, it has some dependence on the language pair involved [34]. Thus, the transfer-based approach is preferred over interlingua-based approach on the basis when there is known structural differences between the source and target languages.

Transfer based system can be broken down into three different stages: analysis, transfer and generation. In the first stage, the source language parser is used to produce the syntactic representation of the source language sentence (Internal representation). In the next stage, the result of the first stage is converted into equivalent target language representation (another internal representation). Finally, a target language morphological analyzer is used to generate the target language text [60]. Transfer based need rules for syntactic transfer, semantic transfer, and lexical transfer [57,61]. Syntactic transfer rules will tell us how to modify the source parse tree to resemble the target parse tree. Semantic transfer uses semantic role labelling. Lexical transfer rules are based on a bilingual dictionary. The dictionary can be used to deal with lexical ambiguity.

B. Corpus-based Machine Translation Approach

As of 1990s, MT research moved from classical rule-based approach to empirical or corpus-based systems. Empirical systems are data-driven as opposed to rule-driven. This approach uses a large amount of raw data in the form of parallel corpora. This raw data contains texts, dictionaries, grammars, etc. and their translations. These corpora are used for acquiring translation knowledge [58]. In recent years there is an increased interest in corpus-based MT systems because it needs less effort from the language/linguistic experts and less human effort is required [59]. In recent classification, Corpus-based approach has three varieties, namely, Example-Based Machine Translation (EBMT), Statistical Machine Translation (SMT), and Neural Machine Translation. The following subsections briefly explain these approaches.

i. Example-Based Machine Translation (EBMT)

EBMT is a translation method that retrieves similar examples (pairs of source phrases, sentences, or texts and their translations) from a database of examples adapting the examples to translate new input [2]. The system maintains an example-base (EB) consisting of translation examples. When a source language sentence is given to the system, the system retrieves a similar source language sentence from the EB with its translation. Then it adapts the example to generate the target language sentence for the input sentence. The basic premise is that, if a previously translated phrase occurs again, the same translation is likely to be correct again. Thus**,** the EBMT system rests on the idea that similar sentences will have similar translations. The system has two main modules 1) retrieval and 2) adaption [59]. There are three tasks in EBMT: Matching fragments against existing examples, transferring (identifying the corresponding translation fragments), and recombining the fragments to give the target text [2].

ii. Statistical Machine Translation (SMT Approach)

SMT is a method for translating text from one natural language to another based on the knowledge and statistical models extracted from bilingual corpora. A supervised or unsupervised statistical machine learning algorithm is used to build statistical tables from the corpora. This process is called learning or training. The statistical tables consist of statistical information such as the characteristics of well-formed sentences and the correlation between the languages. During translation, the collected statistical information is used to find the best translation for the input sentences. This translation step is called the decoding process [60].

In SMT, the core process (transfer) includes a translation model which takes as input source language words or word sequences (phrases) and produces target language words or word sequences as an output. The following stage includes a language model which synthesizes the sets of source language words in meaningful strings which are meant to be equivalent to the input sentences. The preceding (analysis) phase is represented by the conventional process of matching individual words or word sequences of input source language text against entries in the translation model [51]. The translation accuracy of these systems mainly depends on the parallel corpus regarding its domain, quantity and quality. So, in order to have a good translation quality, the data must be preprocessed consistently [24].

iii. Neural Machine Translation (NMT) approach

NMT is a new breed of corpus-based MT (also called data-driven or, less often, corpus-driven machine translation), which is beginning to displace its corpus-based predecessor, SMT [9, 10, 11, 12]. It is a newly emerging approach to MT, recently proposed by Kalchbrenner and Blunsom [11], Sutskever *et al.* [9] and Cho *et al*. [12]. Unlike the traditional phrase-based translation system, which consists of many small sub-components that are tuned separately, NMT attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation. It is trained on huge corpora of pairs of source language segments (usually sentences) and their translations, that is, basically from huge translation memories containing hundreds of thousands or even millions of translation units.

Deep neural networks (DNN) have shown great success in handwriting recognition [62, 63], speech recognition [64, 65] and in natural language process such as language modelling [66], paraphrase detection [67] and word embedding extraction [68]. Furthermore, in the field of MT, DNN is a newly emerging approach and proved to achieve excellent performance [9, 10, 11, 12]. During the last wave of neural network research in the 1980s and 1990s, MT was in the sight of researchers exploring these methods [69]. In fact, the models proposed by Forcada and Ñeco [70] and Castaño *et al.* [71] are striking similar to the current dominant NMT approaches. However, none of these models trained on data sizes large enough to produce reasonable results for anything. The computational complexity involved by far exceeded the computational resources of that era, and hence the idea was abandoned for almost two decades. During this hibernation period, data-driven approaches such as phrase-based SMT rose from obscurity to dominance and made MT a useful tool for many applications, from information gisting to increasing the productivity of professional translators.

The modern resurrection of neural methods in MT started with the integration of NLMs into traditional SMT systems. The pioneering work by Schwenk [72] showed large improvements in public evaluation campaigns. However, these ideas were only slowly adopted, mainly due to computational concerns. The use of Graphical Processing Unit (GPU) for training also posed a challenge for many research groups that simply lacked such hardware or the experience to exploit it. Moving beyond the use in language models, neural network methods crept into other components of traditional SMT, such as providing additional scores or extending translation tables by Schwenk [73]; Lu *et al.* [74], reordering by Kanouchi *et al.* [75]; Li *et al.* [76] and pre-ordering models by de Gispert *et al.* [77], and so on. For instance, the joint translation and language model by Devlin *et al.* [78] was influential since it showed large quality improvements on top of a very competitive SMT system.

More ambitious efforts aimed at pure NMT, abandoning existing statistical approaches completely. Early steps were the use of convolutional models proposed by Kalchbrenner and Blunsom and sequence-to-sequence models by Sutskever *et al.* [9] and Cho *et al.* [79]. These were able to produce reasonable translations for short sentences but fell apart with increasing sentence length. The addition of the attention mechanism finally yielded competitive results [8, 80]. With a few more refinements, such as byte pair encoding and back-translation of target-side monolingual data, NMT became the new state of the art. Within a year or two, the entire research field of machine translation went neural. To give some indication of the speed of change: At the shared task for MT organized by the Conference on Machine Translation (WMT), only one pure NMT system was submitted in 2015. It was competitive but outperformed by traditional statistical systems. A year later, in 2016, a NMT system won in almost all language pairs. Since 2017, almost all submissions were NMT systems.

Hybrid Machine Translation Approach (Hybrid MT)

Hybrid machine translation approach is developed taking advantage of both statistical and rule based translation methodologies. It has proven to have better efficiency in the area of MT systems at the time of its discoveries [23]. The hybrid approach can be used in many ways. In some cases, translations are performed in the first stage using a rule-based approach followed by adjusting or correcting the output using statistical information. In the other way, rules are used to preprocess the input data as well as post-process the statistical output of a statistical-based translation system. This technique has become better example-based MT and has more power, flexibility, and control in translation since the emergence of current-state of Machine Translation called NMT.

2.5 System Modelling and Language Modelling

2.5.1 System Modelling

Artificial Intelligence (AI) and machine learning (ML) are the cornerstones of the next revolution in computing [95]. These technologies hinge on the ability to recognize patterns then, based on data observed in the past, predict future outcomes. Deep learning (DL) can be considered as a subset of ML [88]. It is a field that is based on learning and improving on its own by examining computer algorithms. It can be used to solve any pattern recognition problem without human intervention. State of the art surveys on the data-driven methods and ML algorithms indicate that DL, along with ML methods is the future of data science. Sequence to Sequence (Seq2Seq) models are DL models that have achieved a lot of success in tasks like machine translation, text summarization, and image captioning [89]. A Seq2Seq model is a model that takes a sequence of items (words, letters, features of images, etc.) and outputs another sequence of items [90]. Its models vary in terms of the exact architectures to use. A natural choice for sequential data is the Recurrent Neural Network (RNN), used by most of the recent NMT work and for both the Encoder and Decoder. The use of RNN models, however, differ in terms of (a) directionality – unidirectional or bidirectional; (b) depth – single or deep multi-layer; and (c) type – often either a vanilla RNN, an LSTM, or a Gated Recurrent Unit (GRU). In general, for the Encoder, almost any architecture can be used since we have fully observed the source sentence. For example, the researchers in [91] used a convolutional neural network (CNN) for encoding the source. Choices on the Decoder side are more limited since we need to be able to generate a translation.

The Encoder-Decoder architecture with RNNs has become an effective and standard approach for both NMT and Seq2Seq prediction in general [9]. In case of MT, a model takes a sequence of source language texts (Amharic texts) as input and produces a sequence of target language texts (Wolaita) as output as shown in Figure 2.5 [91]. Seq2Seq was first proposed by Cho *et al*. [12] to model variable-length source input with temporal dependencies which is a drawback of N-gram model [10]. The work in [12] is one of the frontier studies investigating NMT with sequences [91]. Google translate started using such a model in production in late 2016 [27]. These models are also explained in the two pioneering papers [9, 12]. The standard Seq2Seq model is generally unable to accurately process long input sequences since only the last hidden state of the Encoder RNN is used as the context vector for the Decoder [81, 92]. On the other hand, the Attention Mechanism directly addresses this issue as it retains and utilizes all the hidden states of the input sequence during the decoding process. It does this by creating a unique mapping between each time step of the Decoder output to all the Encoder hidden states. This means that for each output that the Decoder makes, it has access to the entire input sequence and can selectively pick out specific elements from that sequence to produce the output [93].

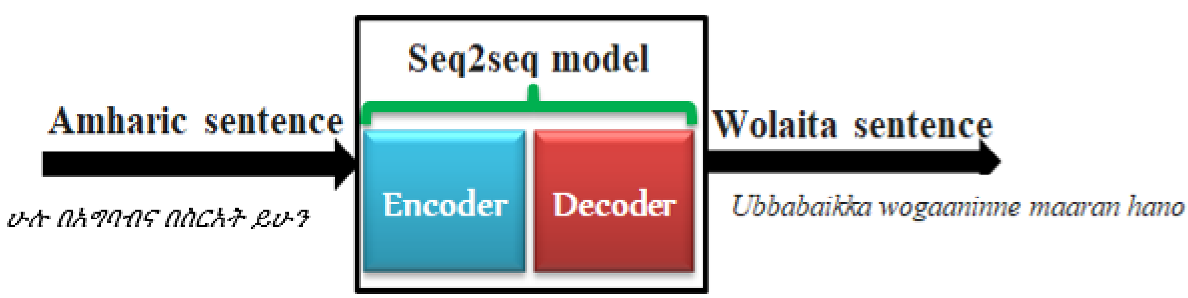


Figure 2.5: System Modelling

2.5.2 Language Modelling

After a few years, different authors outperformed SMT by using deep neural network (DNN) based approaches. Nowadays, rather than using tedious steps like preparation of language modelling, preparation of translation modelling, tuning and decoding steps of SMT, using Encoder-Decoder based language modelling became preferred approaches [8]. Language modelling (LM) is the task of assigning a probability to sentences in a language [95]. In recent time, the Encoder-Decoder end-to-end LM is becoming attractive LM for MT task. It is LM which is based on a deep learning algorithm. According to the work of different researchers, in the early emergence of NMT, the RNN was used to generate word co-occurrence probabilities for MT tasks. Besides assigning a probability to each sequence of words, LM also assigns a probability for the likelihood of a given word (or a sequence of words) to follow a sequence of words [96]. So that one can judge if a sequence of words is more likely or “fluent” than another [97].

To model these conditional probabilities, traditional n-gram LMs can only handle short contexts of about 4 to 6 words and do not generalize well to unseen n-grams [98, 99]. Thus, the neural language model is first proposed by [65] to solve the aforementioned problem in n-gram. Developing better language models often results in models that perform better on their intended NLP task. This is the basis for developing better and more accurate language models. Often training better language models improves the underlying metrics of the downstream task (such as BLEU score for translation), which makes the task of training better LMs valuable by itself [102]. Therefore, Neural Language Models (NLMs) became the better choice for the simplicity of the steps of modelling. In NLM, the steps are interconnected.

A. Neural Language Models

The use of neural networks (NNs) in LM is often called Neural Language Modelling or NLM for short. NN approaches are achieving better results than classical methods both on standalone LMs and when models are incorporated into larger models on challenging tasks like machine translation [97]. NN based LM involves the building of an end-to-end NN based LM that NN is trained to map aligned bilingual texts from source sentence to target sentences without additional external linguistic information [8, 9]. After the first proposal of NLMs in [10], it was enhanced by other researchers [104, 105, 106]. As a natural development, subsequent MT systems in [72, 106, 107], started adopting NLMs alongside with traditional n-gram LMs and generally obtain sizable improvements in terms of translation quality [85]. To make NLMs even more powerful, recent works in [73, 108, 109] proposed to condition on source words as well as the target context to lower uncertainty in predicting next words. More recently, RNN and then networks with a long-term memory like the Long Short-Term Memory network (LSTM), Gated Recurrent Unit (GRU) and bidirectional RNN (bi-RNN) allow the models to learn the relevant context over much longer input sequences than the simpler feed-forward networks.

B. Recurrent Neural Network (RNN) Language Model

For any of NLP task, a basic layered NN is used when there are a set of distinct inputs and we expect that all inputs are independent of each other [111]. This is a preferred approach for modelling languages for translation purpose because language is a sequence of words and each next word is dependent on the words that come before it. If we want to predict the next word in a sentence, we should know what the previous words were. This is analogues to the fact that the human brain does not start thinking from scratch for every word we say. The heart of the Seq2Seq modelling is RNN [9] and its structure is expressed in Figure 2.6 [16]. it is a powerful and expressive architecture that can handle sequential data and has been extensively used for LM [16].

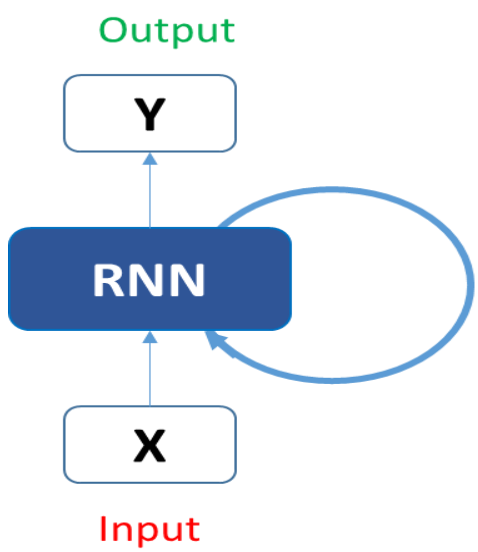


Figure 2.6: Structure of Recurrent Neural Network

An example of the RLM is illustrated in Figure 2.7 [16]. It can be noticed that the cell-state output (Cn) at a current time-step is used as input for next time-step.

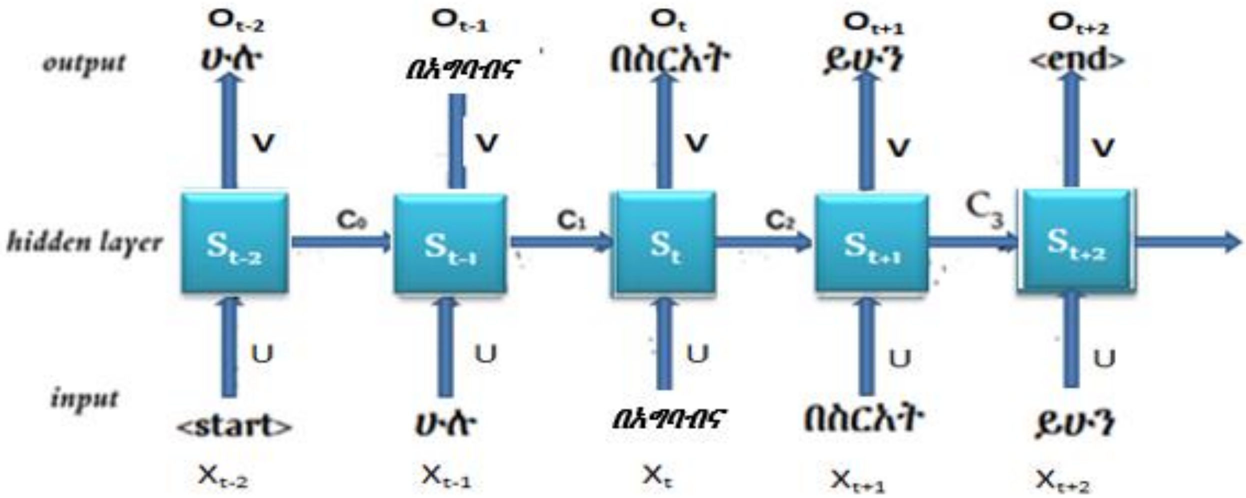
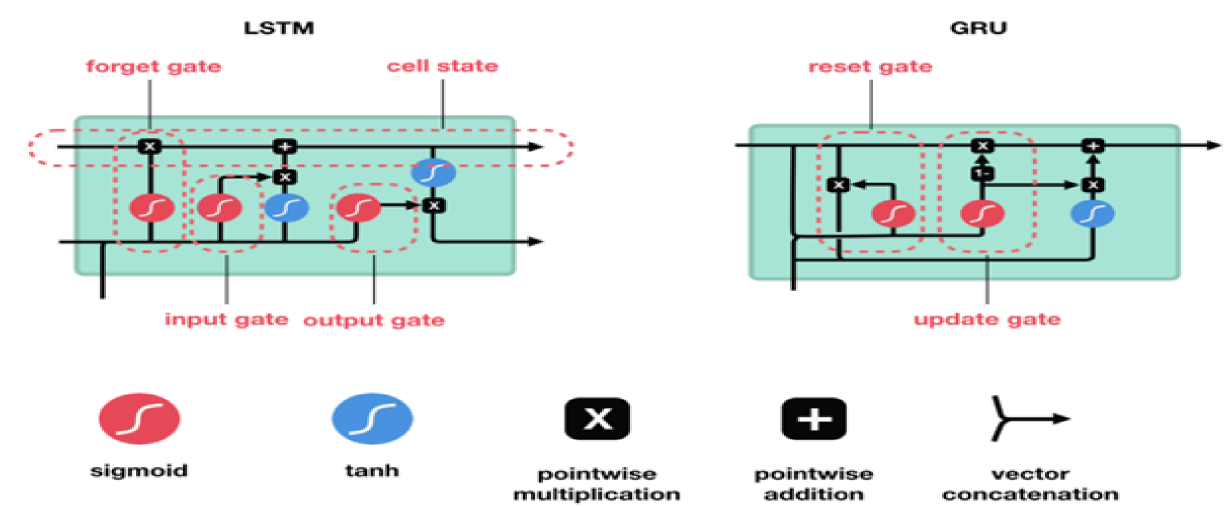


Figure 2.7: Example of a RLM that Processes an Input Sentence

Here, x_t is input to the network at time step *t.* x_1 is the input at index 1 in the sequence. S_t *is a* hidden state of the network at time *t*. It is the memory of the network. This corresponds to the weights in a normal NN which is learned at the time of training. Here is the advantage of using NLM as language modelling. RNN LMs can be trained on large amounts of data, and outperform n-gram models [110]. Here is an interesting example “I had a good time in **Wolaita**. I also learned to speak some _________”. If we want to predict what word will go in the blank, we have to go all the way to the word **Wolaita** and then conclude that the most likely word will be **Wolaiteygna**. In other words, we have to have some memory of our previous outputs and calculate new outputs based on our past outputs [110]. Another advantage of using RNNs for LM is that due to memory constraints, they are limited to remembering only a few steps back [110]. This is ideal because the context of the word can be captured in the five to ten words before it. We don’t have to remember fifty to a hundred words of context. RNN can also be used for arbitrarily long sequences if we have memory at our disposal. But it has also some drawbacks. As the context length increases, i.e., our dependencies become longer; layers in the unrolled RNN also increase. As the network becomes deeper, the gradients flowing back in the back-propagation step become smaller. As a result, the learning rate becomes slow and makes it infeasible to expect long term dependencies of the language. We used the GRU unit both in Encoder and Decoder part as the solution to this issue.

C. Gated Recurrent Unit (GRU)

Despite the advantage of RNN, it suffers from two major drawbacks: Exploding gradient and vanishing gradient problem. The exploding gradient refers to the phenomenon when gradient values become exponentially large as we perform backpropagate through time (BPTT). The exploding gradient problem is solved by clipping the gradient after it reaches a certain threshold value. The vanishing gradient, on the other hand, is a challenging task and it occurs when gradient values start approaching zero as we perform BPTT. There are certain ways to solve the vanishing gradient problem, such as specific leaky generators [46], regularization [113], and GRU [27], etc. The GRU by [27] is the most widely used solution for gradient vanishing problem. GRU networks are just an advanced version of plain RNNs that we discussed above. These networks are designed to remember information for long periods without having to deal with the vanishing gradient problem. In the GRU cell, the vanishing gradient problem is solved by writing the current state as a memory of the network. This writing process is regulated reset gates. If we want to remember something, we write it down. Similarly, GRUs also has the same intuition. The difference between GRU and LSTM architectures is that GRU has no output gate, while LSTM has an output gate as shown Figure 2.8. Therefore, GRU is LSTM that has no output gate, so writes the contents from its memory cell to the larger network at each time step [27, 114]. LSTM cells add a large number of additional parameters. For each gate alone, multiple weight matrices are added. More parameters lead to longer training times and risk over-fitting [115].



Source: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Figure 2.8: Comparison of LSTM Vs GRU Structure

Chapter 3: Related Work

3.1 Introduction

There are different studies conducted on MT approaches, strategies, techniques, and implementations that have been documented. In Ethiopia, some MT systems have been tried to be developed and documented as research work. This chapter discusses previous works related to our study in MT domain. We classified the chapter into three sections as MT for Ethiopian language pairs, MT for a foreign language to Ethiopian language pair and MT for foreign language pairs and we will discuss all in detail in the next sub sections.

3.2 Machine Translation for non-Ethiopian Language Pairs

Choudhary *et al.* [81] conducted research in Neural Machine Translation for English-Tamil. In this research, the basic objective of the researchers was to localize all the information available in the English language to the local language by using efficient MT approach. They used the datasets obtained from EnTam V2.05 and Opus. The sentences are taken from various domains like news, bible, cinema, movie subtitles and combined to build their final parallel dataset. Their final dataset contains 183,451 training corpus, 1,000 validation and 2,000 test corpus from English to Tamil. The data used is encoded in UTF-8 format. They used a BLEU score method to evaluate the performance of the system. Their model outperformed Google translator with a margin of 4.58 BLEU points.

Utiyama *et al.* [31] conducted research in Machine translation from Japanese and French to Vietnamese. This study was done by using the SMT approach. They conducted the experiments on parallel corpora collected from TED talks. They used phrase-based and tree-to-string models and have shown that the SMT system trained on French to Vietnamese obtains better results than the system of Japanese to Vietnamese because French and Vietnamese have more similarities in the structures of sentences than between Japanese and Vietnamese.

Brour *et al.* [84] developed Arabic text language into Arabic sign language NMT. Their system is based on previously developed two approaches with the same source and target language: rule-based Interlingua and example-based approaches. They stated the limitations of previous studies as the requirement of linguistic knowledge necessary to develop the rules. In addition to this, they stated notable results achieved by neural machine translation by listing known companies: including Google [27] who currently joined NMT. They implemented the system using a feedforward back-propagation ANN model. The system is trained using a dataset containing about 9,715 sentences, and it is evaluated with 73 simple sentences. The system is evaluated using BLEU score, and it is compared to the first version ATLASLang MTS, the 4-gram average BLEU score obtained for ATLASLang NMT is 0.79.

Matsumura *et al.* [82] conducted research in English-Japanese NMT with Encoder-Decoder-Reconstructor. In [8] researchers conducted research on Chinese-English NMT by using an Encoder-Decoder-Reconstructor framework for back-translation. Encoder-Decoder Reconstructor framework for back-translation is first time proposed by Tu *et al*. [83] to address the problem NMT suffers, which is the repeating or missing words in the translation. In this method, they selected the best forward translation model in the same manner as Bahdanau *et al.* [8] and then trained a bi-directional translation model as fine-tuning. Their experiments show that it offers a significant improvement in BLEU scores in Chinese-English translation task. The researchers in [82] performed the same approaches and in English-Japanese task too in addition to evaluating the effectiveness of pre-training by comparing it with a jointly-trained model of forward translation and back-translation. They used two parallel corpora: Asian Scientific Paper Excerpt Corpus (ASPEC) and NTCIR PatentMT Parallel Corpus for training and testing result. They used only the first 1 million sentences sorted by sentence-alignment similarity but they excluded sentences with more than 40 words from the training data. Their experiments had 512 hidden units, 512 embedding units, 30,000 vocabulary size and 64 batch size. They used Adagrad (initial learning rate 0.01) for optimizing model parameters. They trained their model on GeForce GTX TITAN X GPU. By using ASPEC corpus in Reconstructor (Jointly-Training) model, they obtained a 26.04 BLEU score in 174 hours for English-Japanese translation. By using NTCIR corpus in Reconstructor (Jointly-Training) model, they obtained 29.04 BLEU score in 252 hours for English-Japanese translation. By using ASPEC corpus in Reconstructor (Jointly-Training) model, they obtained 16.29 BLEU score for Japanese- English translation. By using NTCIR corpus in Reconstructor (Jointly-Training) model, they obtained 28.95 BLEU score for Japanese- English translation. But finally, they concluded that the system does not significantly improve translation accuracy in Japanese-English translation. In addition, it is proved that the encoder-decoder-reconstructor without pre-training worsens rather than improves translation accuracy.

Almahairi *et al.* [17] proposed NMT for the task of Arabic translation in both directions (Arabic-English and English-Arabic) and compared a Vanilla Attention-based NMT system against a Vanilla Phrase-based system. Preprocessing Arabic texts can increase the performance of the system, especially normalization, but the model consumes much time for training.

Sutskever *et al.* [9] conducted research in English to French Machine Translation using Sequence to Sequence Learning with Neural Networks. The study was carried out by using Neural Machine Translation approach. They applied Deep Neural Network (DNN) approach on the previous study carried out in phrase-based SMT approach [87]. The model they used is the Recurrent Neural Network language model. They used a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. They used WMT'14 English to French dataset. They trained their models on a subset of 12M sentences consisting of 348M French words and 304M English words, which is a clean "selected" subset from [87]. They chose this translation task and this specific training set subset because of the public availability of a tokenized training and for comparing the performance from the baseline SMT [87]. They evaluated their models using the standard BLEU score metric. On the WMT'14 English to French translation task, they obtained a BLEU score of 34.81 using a simple left-to-right beam-search decoder. For comparison, a phrase-based SMT system achieves a BLEU score of 33.3 on the same dataset [87]. This result shows that a neural network architecture outperforms a phrase-based SMT system. When they reverse the order of the words in all source, the BLEU score increased to 36.5. Finally, they found that reversing the order of the words in all source sentences (but not target sentences) improved the LSTM's performance markedly, because doing so introduced many short term dependencies between the source and the target sentence which made the optimization problem easier. Additionally, they confirmed LSTM did not have difficulty in long sentences.

Norouzi *et al.* [27] conducted research in Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. The study was done by using NMT approach for the purpose of overcoming many of the weaknesses of previous Google's conventional phrase-based translation systems. In addition to this, they improved NMT robustness problem (particularly when input sentences contain rare words) which is stated as a problem by many previous researchers [9, 86]. Their model is sequence-to-sequence learning framework with attention. It has three components: an encoder network, a decoder network, and an attention network. It consists of a deep LSTM network with 8 encoder and 8 decoder layers. The encoder transforms a source sentence into a list of vectors, one vector per input symbol. Given this list of vectors, the decoder produces one symbol at a time, until the special end-of-sentence symbol (EOS) is produced. A decoder is implemented as a combination of an RNN network and a softmax layer. The encoder and decoder are connected through an attention module which allows the decoder to focus on different regions of the source sentence during the course of decoding. To improve parallelism and therefore decrease training time, their attention mechanism connects the bottom layer of the decoder to the top layer of the encoder. To improve the handling of rare words, they divided words into a limited set of common sub-word units ("word pieces") for both input and output. They followed a beam search technique for implementation purpose. For testing their system, they used WMT'14 English-to-French and English-to-German benchmarks as a dataset. They evaluated their models using the standard BLEU score metric. Specifically, on WMT'14 English-to-French, their single model scores 38.95 BLEU, an improvement of 7.5 BLEU from a single model without an external alignment model reported in [85] and an improvement of 1.2 BLEU from a single model without an external alignment model reported in [86]. Likewise, on WMT'14 English-to-German, their single model scores 24.17 BLEU, which is 3.4 BLEU better than a previous competitive baseline [86]. They also reported as, on production data, their implementation is even more effective. Finally, they reported as human evaluations show that their system has reduced translation errors by 60% compared to their previous phrase-based system on many pairs of languages: English↔French, English↔Spanish, and English↔Chinese. Additionally, their experiments suggest the quality of the resulting translation system gets closer to that of average human translators.

3.3 Machine Translation involving Ethiopian languages

Solomon Teferra *et al.* [3] conducted research in Parallel Corpora for bi-directional Statistical Machine Translation for Seven Ethiopian Language Pairs. The researchers presented some Ethiopian language researches conducted by graduate students and mainly raised the unavailability of linguistic resources stated by students which in turn affects the results that they obtain. They attempted towards the development of parallel corpora for English and Ethiopian Languages, such as Amharic, Tigrigna and Ge'ez from the Semitic, Afan-Oromo from the Cushitic and Wolaytta from Omotic language families. But they prepared a corpus by using religious resources other domains. For example: - the Tigrigna-English and Afan Oromo-English corpora are in legal and religious (both bible and other religious collections) domains. The Wolaytta-English and Ge'ez-English language pairs are from the religious domain only. However, the Ge'ez-English corpus is only from the Bible while the Wolaytta-English consists of the Bible and other religious collections. They tried to study the nature of different language pairs. They used the corpora they developed for conducting a bi-directional SMT experiment. In the experimental setup, they used Moses with GIZA++ alignment tool for aligning words and phrases. SRILM toolkit was used to develop language models using semi-automatically prepared corpora from the training and tuning corpora of target languages. They achieved a BLEU score of 13.31 for English-Amharic translation while the Amharic-English has a 22.68. Similarly, the English-Tigrigna and Tigrigna-English have BLEU scores of 17.89 and 27.53, respectively. Likewise, English-Afaan Oromo has a 14.68 BLEU and Afan Oromo-English has 18.88 BLEU score. In a similar way, the English-Wolaytta translation has BLEU of 10.49 while Wolaytta-English has 17.39. Finally, The English-Ge'ez and Ge'ez-English translation has a BLEU score of 6.67 and 18.01, respectively. Finally, they concluded that the English-Ethiopian languages SMT systems have less BLEU scores than that of Ethiopian languages-English ones. The reason they raised is the fact that when the Ethiopian languages are used as a target language, the translation from English as a source language is challenged by many-to-one alignment.

Dawit Mulugeta [45] conducted research in Ge'ez to Amharic automatic MT using SMT approach. As a research methodology, the author used qualitative experimental method to investigate the effect of variables such as normalization, corpus and test split options on the SMT result. The data used for the research experiment were found from both online and manually prepared. Totally 12, 860 parallel sentences were used for both languages. Regarding the organization of the data, out of the bilingual data, 90% for training and 10% for testing were used for the experiment. Moses decoder, IRSTLM, GIZA++ and BLEU were used to build translation model, language model, Word alignment and evaluation of the Ge'ez to Amharic MT system respectively. The parallel corpus used for the experiment was sentence level aligned. The average translation result was 8.26 BLEU score.

Mulu Gebreegziabher and Besacier conducted preliminary experiments on English-Amharic SMT [21]. The main objective of the research was the need to begin empirical researches towards developing English-to-Amharic SMT. The major problem they stated was the rule-based approach yet not recommended to be used for under-resourced languages like Amharic due to the different linguistic knowledge, rules and resources required. To meet their goal, the total corpus size of 632 parliamentary corpora of which 115 had been used for the experiment. The experiment had been conducted using 18,432 English-Amharic sentence pairs extracted from these corpora. Out of the total 90% randomly selected sentence pairs had been used for training while the remaining 10% sentence pairs were used for testing. There were different software resources used for the experiment in general integrated with MOSES like SR Luthra Institute of Management (SRLIM) to build the language model, Giza++ for building translation model and BLEU metric for evaluating the performance of the MT system. When the researchers evaluate their MT system for English-to-Amharic SMT the baseline phrase-based BLEU score results 35.32% translation had been achieved. The preliminary experiment result shows that the EASMT can translate the basic meaning of the English sentence when translating into Amharic sentence. However, there are some strong as well as weak points in the performance of the EASMT. Keeping the storing side, to address problems like non-translated words, wrongly translated, insertion, deletion, alignment problem, preposition usage, and morphological errors they had used word segmentation on the target side is vital. According to these results, more experimentation and research is required to further improve the translation accuracy of the EASMT. The experiment done so far is encouraging as the translation is done from less inflected English language to a morphologically rich language Amharic.

Eleni Teshome developed bi-directional English-Amharic machine translation using a constrained corpus [4]. The objective of this study was developing a bi-directional English-Amharic MT system by using the SMT approach. In this paper, two different corpora prepared: the first corpus (Corpus I) was made of 1,020 simple sentences that have been prepared manually. The second is made of 1,951 complex sentences (Corpus II) from sources, one from the Bible and the other from the public procurement directive of the Ministry of Finance and Economic Development. From total of 2,971 sentences, 10% were taken for the testing process and the remaining 90% is used for training. Since the translation is bi-directional, two language models were developed, one for Amharic and the other for English and translation models were also built. Two different experiments were conducted and the evaluation was performed by using two different methodologies. The first experiment was performed using simple sentences and the accuracy of Amharic to English translation is 94% and English to Amharic translation and 90.59% for Amharic to English translation. The second experiment was performed by using the manual questionnaire method, the accuracy of English to Amharic translation is 91% and the accuracy of Amharic to English translation is 97%. For complex sentences, the first methodology, the accuracy of the translation from English to Amharic was 73.38% and from Amharic to English translation was 84.12%. The second experiment was performed by using the manual questionnaire method was 87% for the English to Amharic translation and 89% for Amharic to English translation. The study shows Amharic to English translation has better accuracy than English to Amharic translation.

Sisay Adugna developed English–Afaan Oromo machine translation using a statistical approach [6]. This study had two main goals: the first is to test how far one can go with the available limited parallel corpus for English-Afaan Oromo language pair and the applicability of existing SMTsystems on these language pairs. The second one is to analyze the output of the system with the objective of identifying the challenges that need to be addressed. In this study, the architecture includes four basic components of statistical machine translation, which are language modelling, translation modelling, decoding, and evaluation. By using a corpus of about 20, 000 bilingual sentences, an author achieved the translation accuracy of 17.74%.

Jabesa Daba and Yaregal Assabie developed bi-directional English-Afaan Oromo MT using hybrid approach [23]. The research work is implemented using a hybrid of rule-based and statistical approaches. Corpus was collected from different domains. They collected 3,000 parallel sentences.

Out of those sentences 90% are used for training and the remaining 10% are used for testing. Since the system is bidirectional, two language models are developed, one for English and the other for Afaan Oromo. The study was carried out with two experiments that are conducted by using two different approaches and their results are recorded. The first experiment is carried out by using a statistical approach. The result obtained from the experiment has a BLEU score of 32.39% for English to Afaan Oromo translation and 41.50% for Afaan Oromo to English translation. The second experiment is carried out by using a hybrid approach and the result obtained has a BLEU score of 37.41% for English to Afaan Oromo translation and 52.02% for Afaan Oromo to English translation. From the result, we can see that the hybrid approach is better than the statistical approach for the language pair and a better translation is acquired when Afaan Oromo is used as a source language and English is used as a target language.

Yitayew Solomon conducted bidirectional English-Afaan Oromo MT Systems using SMT approach [29]. For the researcher to have such an objective was, the research done by Sisay Adugna [6] and Jabesa Daba [23] score poor performance with the BLEU score of 17% and 37% respectively. This is due to the alignment quality of the prepared data and the unavailability of a well-prepared corpus for the MT task for English to Afaan Oromo MT. To build the translation model, 6400 parallel sentences and 19300 and 12200 sentences, to build a language model for both English and Afaan Oromo languages were used respectively. Randomly, for training 90% and 10 % testing of corpus size were used. 700 simple and 5700 complex sentences with a total of 6400 sentences was used. Moses for Mere Mortal is used for SMT and integrates different toolkits which are used for translation purpose such as IRSTLM for language model, Decoder for translation, MGIZA++ for word alignment. Hunalign, Anymalign and MGIZA++ where software tools, used for sentence, phrase and word level alignment, respectively. BLEU score was used to evaluate the MT system. Preprocessing tasks sentence splitting, margining and true casing are used to make ready the corpus for the experimentation. Six experiments were done by the researcher to select the optimal alignment quality for English to Afaan Oromo where, Experiments I and II for word-level alignment, Experiments III and IV for phase level alignment and Experiments V and VI for sentence-level alignment. Word level alignment when the max phrase length is 4 and min is 1which record 21% and 42% BLEU score from English-Afaan Oromo and Afaan Oromo-English respectively. Phrase level alignment when the max phrase length is 16 and min is 4 which record

27% and 47% BLEU score from English-Afaan Oromo and Afaan Oromo-English respectively. Sentence level alignment when the max phrase length is 30 and min is 20 which record 18% and 35% BLEU score from English-Afaan Oromo and Afaan Oromo-English respectively. Optimal alignment is phrase-level alignment when the max phrase length is 16 and min is 4 which record 27% and 47% BLEU score from English-Afaan Oromo and Afaan Oromo-English respectively. Finally, the researcher recommends, better results can be achieved by using the corpus with proper alignment used for training the system. So, by increasing the size of the training data set that properly aligned at phrase level one can develop a better bi-directional English-Afaan Oromo machine translation.

3.4 Summary

DNN is a newly emerging approach and proved to achieve excellent performance. Unlike the traditional SMT, the NMT aims at building a single neural network that can be jointly tuned to maximize the translation performance. After a few pioneer works in exploring neural features in SMT systems [28], NMT quickly become the dominant approach for MT. The researchers in [10] and [11] first propose to use the encoder-decoder architecture to do sequence to sequence mapping. At the same time, the other study in [8] applies end-to-end MT. In [9] the attention mechanism proposed to dynamically attend to different source words when generating different target words, which becomes the default component of current NMT systems. In general, as we have seen in the above related works, most of the papers done for different language pairs in recent years are based on attention-based NMT and they resulted in promising results when replacing SMT by NMT [9, 10, 11, 12]. It is the current SOTA MT technology and also many researchers are joining this approach today. Speakers of Wolaita language who are unable to speak and understand Amharic cannot communicate and interact with Amharic speakers without finding translators. Thus, non-Amharic speakers of Wolaita language speakers face problems of lack information. If documents and news articles automatically translated into Wolaita language, from Amharic language, it solves the problem. As the researcher knowledge, no prior study has carried out in Amharic to Wolaita NMT. Thus, we apply the attention-based NMT approach for Amharic to Wolaita neural machine translation.

Chapter 4: Design of the Proposed System

4.1 Introduction

This Chapter covers the architectural design of Amharic-to-Wolaita NMT both in attention-based and non-attention-based mechanism. The proposed system is based on Seq2Seq learning model with Recurrent Neural Network based Encoder-Decoder architecture. The Seq2Seq model with RNN based Encoder-Decoder architecture is the SOTA technique in NMT. Encoder and Decoder help the model gain a deeper understanding of the two languages, i.e., source and destination languages. The Encoder encodes the complete information of the source sequence, which is the source language, Amharic, into a single real-valued vector using word-embedding, also known as the context vector. This context vector is passed to the Decoder to produce an output sequence, which is the target language, Wolaita. We used the GRU unit to train the system. The system based on an attention mechanism solves the bottleneck of basic Encoder-Decoder non-attention-based NMT.

4.2 System design

4.2.1 Language Model Training

To train our system model, we used a forward pass and back-propagation algorithms which uses a deep multi-layer GRU architecture as we detailed in Algorithm 4.1 and Algorithm 4.2, respectively. For each new input x_i at the time i, GRU unit updates its memory to produce a hidden state h_i which one can think of as a representation for the partial sequence $x_1, x_2, x_3 \ldots\ldots x_i$. Mathematically it can be represented as:

$$h_t = f(x_t, h_{t-1}) \quad (1)$$

In the above formula, f is an abstract function that computes a new hidden state given the current input x_t and the previous hidden state h_{t-1}. The starting state h_0 is often set to 0 though it can take any value. A popular choice of f is tanh as being a non-linear function.

$$h_t = tanh(W_{xh}x_t + W_{hh}h_{t-1}) \quad (2)$$

Since the Encoder and Decoder share many operations in common in forward pass algorithm, we combine the Amharic sentence x (length m_x), the Wolaita sentence y (length m_y), and the end of sentence markers “<end>” together to form an input sequence s as shown in Line 1 of Algorithm 1. We first start with the Encoder weights and initial states set to zero (lines 2-3). The algorithm switches to the Decoder mode at time m_x+ 1 (line 5). The same GRU codebase (lines 8-11) is used for both the Encoder and Decoder in which embedding is first looked up for the input s_t; after that, hidden states as well as GRU cell memories are built from the bottom layer to the top one (the L^{th} layer). In Line 10, GRU refers to the entire formulation in which anyone who interested to work with other hidden units such as bi-RNN and LSTM can easily replace it. Lastly, on the Decoder side, the top hidden state is used to predict the next symbol s_{t+1} (line 13); then, a loss value l_t and a probability distribution p_t computed according to Equation 3 and Equation 4.

$$St = W_{hy}.h_t \text{)} \quad (3)$$

$$p_t = softmax\ (s_t) \text{)} \quad (4)$$

Here, $W_{hy} \in R^{|Y| \times d}$, with d being the dimension of the GRU hidden state, to compute a score vector S_t. Computing the total loss for (x, y) during the forward pass is written in Equation 5.

$$-1/n\ [log\ (y_1\ (word_1) + log\ (y_2\ (word_2) + log\ (y_n\ (\text{"<end>"})) \quad (5)$$

The back-propagation algorithm reveals many similarities compared to the forward pass algorithm except that we have reversed the procedure. First, we initialize gradients of the GRU layers at the final time step (line 1) as well the model weights on the Decoder size (line 2) to zero. At time m_x, we switch to the Encoder mode by saving the currently accumulated GRU and embedding gradients for the Decoder (line 5) and starting to accumulate gradients for the Encoder weights (line 6). The back-propagation procedure presented earlier for GRU can simplify the core NMT gradient computation (lines 8-18) by making the following two referents: (a) Predict grad:- which computes gradients for the target-side losses with respect to the hidden states at the top layer and the softmax weights W_{hy} ; and (b) GRU grad:- which computes gradients for inputs to GRU and the GRU weights per layer $GRU^{(l)}$. It is important to note that in Lines 10 and 15 of Algorithm 2, we add the gradients (flowing vertically from either the loss or the upper GRU layer) to the gradient of the below layer (which already contains the gradient back-propagated horizontally) instead of overriding it. In Line 18, we perform sparse updates on the corresponding embedding matrix for participating words only.

Algorithm 4.1: RNN language model training algorithm – forward pass.
Input: source sentence x of length m_x, target sentence y of length $m_{y.}$
Parameters: encoder $W^{encoder}$, $GRU^{encoder}$; decoder $W^{decoder}$, $GRU^{decoder}$
Output: loss *l* and other intermediate variables for back-propagation.

1. s $\Leftarrow$ [x,<end> , y,<end>] ; // Length of s is $m_x + 1 + m_y + 1$
2. We, $GRU^{encoder\ (1\ldots L)} \Leftarrow W^{encoder}$, $GRU^{encoder}$; // Encoder weights
3. $h_0^{(1..L)}$, $c_0^{(1..L)} \Leftarrow 0$; // Zero initialization
4. **for** $t = 1 \Rightarrow (m_x + 1 + m_y)$ **do**
   // Decoder transition
5. if $t == (m_x + 1)$ then
   We, $GRU^{encoder\ (1\ldots L)} \Leftarrow W^{decoder}$, $GRU^{decoder}$;
6. end
   // Multi-layer GRU
7. $h_t^{(0)} \leftarrow$ Emb LookUp(s_t, We) ;
8. for $l = 1 \Rightarrow L$ do
9. $h_t^{(l)}$, $c_t^{(l)} \Leftarrow GRU(h_{t-1}^{(l)}, c_{t-1}^{(l)}), h_t^{(l-1)}, GRU^{(l)}$ // GRU hidden unit
10. end
    // Target-side prediction
11. if $t \geq (m_x + 1)$ then
12. l_t, $p_t \Leftarrow$ Predict(s_{t+1}, $h_t^{(L)}$, W_{hy}) ;
13. end
14. end

Algorithm 4.2: RNN language model training algorithm – back-propagation pass

1. $dh_{mx+1+my,}{}^{(1..L)}$, $dc_{mx+1+my,}{}^{(1..L)} \Leftarrow 0$; / / Cell and state gradients
2. $dGRU^{(1\ldots L)}$, dWe, $dW_{hy} \Leftarrow 0$; // Model weight gradients
3. for $t = (m_x + 1 + m_y) \Rightarrow 1$ do // Encoder transition
4. if $t == m_x$ then
5. $dWe^{decoder}$, $dGRU^{decoder} \Leftarrow dWe$, $dGRU^{(1..L)}$; // Save decoder gradients
6. dWe, $dGRU^{(1..L)} \Leftarrow 0$
7. end
8. if $t \geq (m_x + 1)$ then // Target-side prediction
9. dh, $dW \Leftarrow$ Predict_grad(s_{t+1}, p_t, $h_t^{(L)}$);
10. $dh_t^{(L)} \Leftarrow dh_t^{(L)} + dh$; // Vertical gradients
11. $dW_{hy} \Leftarrow dW_{hy} + dW$;
12. end
    // Multi-layer GRU
13. for $l = L \Rightarrow 1$ do // Recurrent gradients
14. $dh_{t-1}^{(l)}$, $dc_{t-1}^{(l)}$, dx, dT $\Leftarrow$ GRU_grad ($dh_t^{(l)}$, $dc_t^{(l)}$, $h_{t-1}^{(l)}$, $c_{t-1}^{(l)}$, $h_t^{(l-1)}$)
15. $dh_t^{(l-1)} \Leftarrow dh_t^{(l-1)} + dx$; // vertical gradients
16. $dGRU^{(l)} \Leftarrow dGRU^{(l)} + dT$;
17. end
18. $dWe \Leftarrow$ Emb_grad_update(s_t, $dh_t^{(0)}$, dWe) ;
19. end
20. $dWe^{encoder}$, $dGRU^{encoder} \Leftarrow dWe^{encoder}$, $dGRU^{(1..L)}$ // Save encoder gradients

When we are training the Decoder, we don't take the output of Decoder in previous step we actually give the correct output from the previous step. That is called teacher forcing. In other word we may want to produce the correct Amharic sentence "ሁሉ በአግባብና በሥርዓት ይሁን" equivalent in Wolaita sentence like "ubbababaikka wogaaninne maaran hano". But the system may produce different word instead of "ubbababaikka" in Wolaita. Teacher forcing handles this issue by making the system to produce correct translation. Thus, teacher forcing feeds the target as the next input.

4.2.2 Language Model Testing

After training our system model, we need to be able to use it to translate, or decode, unseen Amharic sentences. The strategy we used to translate an Amharic sentence to Wolaita sentence is performing beam searching mechanism which outperforms the greedy searching mechanism by solving its garden-path problem [85, 104]. The idea is simple: we first encode preprocessed Amharic sentence, "ሁሉ በአግባብና በሥርዓት ይሁን" and the decoding process is started as soon as an end-of-sentence marker "<end>" for the Amharic sentence is fed as an input to the Encoder. For performing beam searching mechanism we performed the following steps. (a) At each timestep on the Decoder side, we keep track of the top K (the beam size) best translations together with their corresponding hidden states. (b) Then we select the top K most likely words. (c) Given K previous best translation ×K best words, we select a new set of K best translations for the current timestep based on the combined scores (previous translation scores + current word translation scores). When predicting the first word of the output sentence, we keep a beam of the top K most likely word choices. They are scored by their probability. Then, we use each of these words in the beam in the conditioning context for the next word. Due to this conditioning, we make different word predictions for each. We now multiply the score for the partial translation (at this point just the probability for the first word), and the probabilities from its word predictions. We select the highest-scoring word pairs for the next beam. This process continues until the end of sentence token <end> is produced. At each time step, we accumulate word translation probabilities, giving us scores for each hypothesis. At this point, we remove the completed hypothesis from the beam and reduce beam size by 1. The search terminates when no hypotheses are left in the beam.

The search produces a graph of hypotheses, as shown in Figure 4.1. It starts with the start of sentence symbol <start> and its paths terminate with the end of sentence symbol <end>. Given the complete graph; the resulting translations can be obtained by following the back-pointers. The complete hypothesis (i.e., one that ended with a <end> symbol) with the highest score (i.e. the path with bold lines in Figure 4.1) points to the best translation. When choosing among the best paths, we score each with the product of its word prediction probabilities. During the search, all translations in a beam have the same length, so the normalization would make no difference.

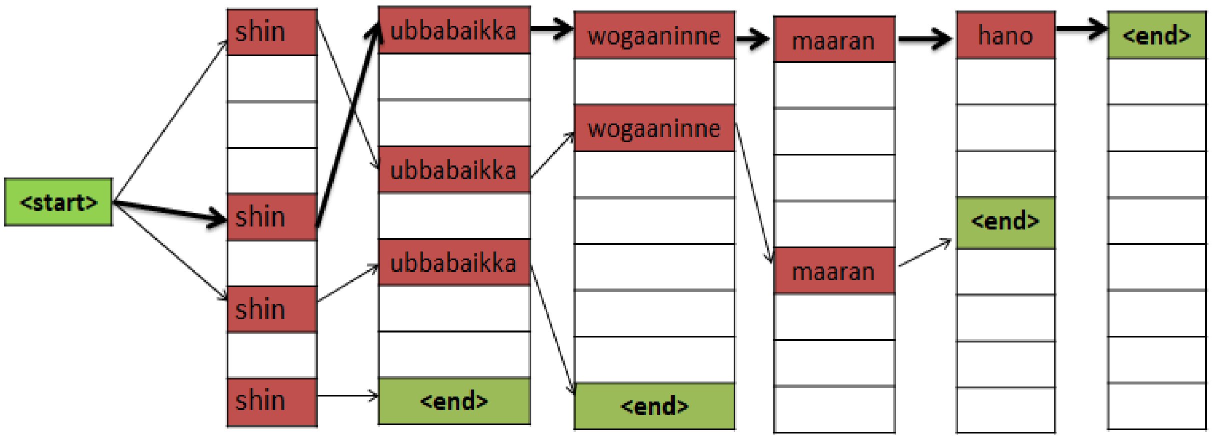


Figure 4.1: Language Model Testing in the Beam Search Graph

Here n = 5 best partial translations (called hypotheses) are selected. An output sentence is complete when the end of sentence token <end> is predicted. We reduce the beam after that and terminate when n full-sentence translations are completed. Following the back-pointers from the end of sentence tokens allows us to read them off. Empty boxes represent hypotheses that are not part of any complete path.

4.3 System Architecture

Attention-based Amharic-to-Wolaita NMT is a translation system where a given Amharic text is translated into equivalent Wolaita sentence through three layers: Encoder, Attention, and Decoder. An input Amharic sentence is preprocessed before performing one-hot representation. Word embedding can be formed from one-ot representation and it is given to Encoder layer as input. Encoder layer outputs attention weight and gives it to attention layer. Attention layer processes attention weight and finally outputs the single representation for input sentence. This representation is given to Decoder and it produces equivalent Wolaita translation based on input. The architecture of the system is shown in Figure 4.2.

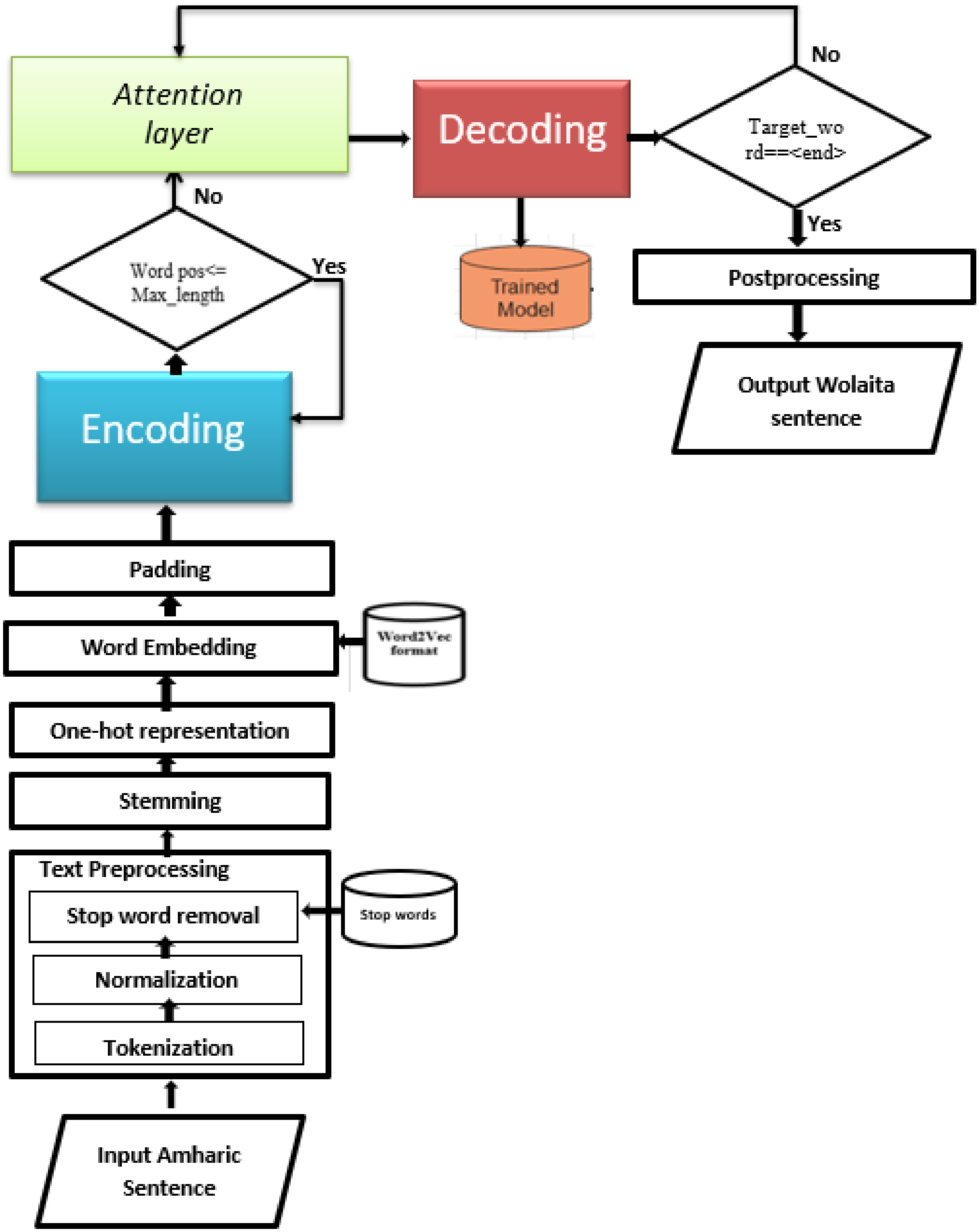


Figure 4.2: Architecture of Attention-based Amharic-to-Wolaita NMT

4.4 Text Preprocessing

For our study, the training and testing corpus is collected from different sources which contain parallel text (Amharic and Wolaita) in religious domain, i.e., bible texts. Preprocessing is involved in preparing the input sentence into a format that is suitable for the morphological analysis. The preprocessing stage consists of steps such as tokenization, normalization and stop-word removal.

4.4.1 Tokenization

The first step in the preprocessing of the sentence is tokenization, which is also known as lexical analysis. Tokenization is essentially splitting of a sentence into smaller units, such as individual words. Each of these smaller units is called token. Most of the time, tokenization of words in Amharic language is performed using Amharic punctuation marks as delimiter characters which are white space, '፡'(hulet netib), '።'(arat netib), '፣'(netela serez),'፤'dereb serez), '!' (kaleagano) and '?' (Question mark) [116]. We used white space (' ') delimiter for tokenization with the buit-in function *tf.keras.preprocessing.text.Tokenizer(filters=' ')* library from a Tokenizer class. In order to find the boundaries of the sentence, we added <end> tag as indicator of the end of a sentence so that the model knows as it reached the end of words in input sentence. For example:- When we perform tokenization in the input sentence ጳውሎስ እንዲህ በማለት ፅፏል ስለ ምንም ነገር አትጨነቁ ከዚህ ይልቅ ስለ ሁሉም ነገር በፀሎትና በምልጃ ከምስጋና ጋር ልመናችሁን ለአምላክ አቅርቡ, it results with the tokens ['<start> ','ጳውሎስ', 'እንዲህ', 'በማለት', ፅፏል', 'ስለ',' ምንም', 'ነገር',' አትጨነቁ', 'ከዚህ', 'ይልቅ', 'ስለ', 'ሁሉም', 'ነገር', 'በፀሎትና', 'በምልጃ', 'ከምስጋና', 'ጋር', 'ልመናችሁን', 'ለአምላክ', 'አቅርቡ', <end>'] as output. Tokenization in Amharic and Wolaita presents a problem because of the rich and complex morphology of the two languages. Thus, each token has to be normalized.

4.4.2 Normalization

Normalization is performed on the word tokens that result from text tokenization. We performed text normalization for both languages based on the properties of the languages. For performing Amharic sentence normalization, we used the algorithms adopted by Tessemma Mindaye Mengistu in [116]. The author in [116] discussed the two types of normalization issues that arise in the Amharic language. The first one is the identification and replacement of short hand representation of a word that is written using forward slash "/" or period "." as shown in Algorithm 4.1. An example is the replacement of "ት/ቤት" by "ትምህርት ቤት"/school/. The second normalization issue is the identification and replacement of Amharic alphabets that have the same pronunciation as shown in Algorithm 4.2. The replacement is made using a representative alphabet from a set of similar alphabets. For example, identifying words like ውሃ, ውሀ, ውሓ, ውሐ, ውኃ, ውኀ/ wɨhä/ which are pronounced the same to mean "Water" and replace by /wɨhä/. In addition to Tessema's normalization issues, we also performed normalizing of words with Labialized Amharic characters such as በልቱዋል or በልቱአል to በልቷል. And also, we performed removing quotes, removing spaces; adding a space between the word and the punctuations like "?"; cleaning digits; and removing of special characters for both Amharic and Wolaita sentences. We also performed Converting of text to lower case of the Wolaita sentence.

Algorithm 4.1: Amharic Word Expanding Algorithm from [116]

```
Read a character before '/' or '.'
Search the location of the characters in corpus
If found
    Return the corresponding expanded word
Else
    Return the original word
Endif
```

Algorithm 4.2: Character Normalization Algorithm from [116]

```
Read a character
If the character is one of ኀሐ orኸ replace them with ሀ (The same applies for the orders,
i.e. the orders of ሐ and ኀ will be replaced by the corresponding orders of ሀ).
    Return the replaced character
Else If the character is one of ኁሑ or ኹ replace them with ሁ.
    Return the replaced character
Else If the character is one of ኂሒ orኺ replace them with ሂ.
    Return the replaced character
Else If the character is one of ኃሓ or ኻ replace them with ሃ.
    Return the replaced character
Else If the character is one of ኄሔ or ኼ replace them with ሄ.
    Return the replaced character
Else If the character is one of ሕኅ or ኽ replace them with ህ.
    Return the replaced character
Else If the character is one of ኆሖ orኾ replace them with ሆ.
    Return the replaced character
Else If the character is ሠ replace it with ሰ (The same applies for ሡሢሣሤሥ).
    Return the replaced character
Else If the character is ዐ replace it with አ(The same applies for ዑዒዓዔዕዖ)
    Return the replaced character
Else If the character is ጸ replace it with ፀ (The same applies for ፁፂፃፄፅፆ)
    Return the replaced character
Endif
```

4.4.3 Stop Word Removal

In Amharic, common words such as pronouns, prepositions and conjunctions occur so frequently that they cannot give any useful information about the content and be discriminatory for a specific class. These words are called stop words. Stop words are low information bearing words such as "ነው" or "ና", typically appearing with high frequency as listed in Appendix II [32]. Stop words may be context dependent. There is not one definite list of stop words, which all tools use, and such a filter is not always used. Some tools specifically avoid removing them to support phrase search. Like other languages, Amharic has non-content bearing words. Usually words such as articles (e.g. 'ያኛው', 'ይሄ'), conjunctions ('ና', 'ነገርግን', 'ወይም') and prepositions (e.g. 'ውስጥ', 'ላይ') do not have a significant discriminating power in the meaning of ambiguous words. In this thesis, stop words like 'ነው', 'እስከ', 'እንደ', etc, are discarded from input texts as these words are meaningless to derive the "sense" of the particular sentence. Then, the text containing meaningful words (excluding the stop words) pass through stemming.

4.5 Stemming

Stemming is the process of removing affixes (i.e. prefixes, infixes and suffixes) that improve the accuracy and performance MT systems. It is the process of reducing inflected and/ or sometimes derived words to their word stem base or root form- generally a written word form. The stem need not be identical to the morphological root of the word. It is very important to reduce the various original forms of a word to a single root/ stem of these words as it reduces the dimensionality of the feature space. It will in turn deliver a significant improvement in the accuracy of MT system. In this thesis, we used Hornmorpho stemming algorithm developed by Gasser [47]. Table 4.1 shows an example of stemmed Amharic Words using Hornmorpho stemming algorithm.

Table 11: Stemmed Amharic Words

Amharic word	Root word	Attached pp	negated?	Is it plural?	Subj pron	Obj pron	Word tense
በሉ	ብልእ		0	0	እነሱ		ሀላፊ
እየበላ	ብልእ		0	0	እሱ		አሁን
ልትበላ	ብልእ		0	0	እሷ		ትንቢት
አልበላሁም	ብልእ		1	0	እኔ		
የበላሁት	ብልእ	የ	0	0	እኔ		

4.6 One-hot Representation

In NMT the original sequential data is not suitable to read for the neural network. It needs some form of representation in a suitable format. Words like "አትጨነቁ" in a corpus is a sequence of ASCII character encodings. Since a neural network is a series of multiplication and addition operations, the input data needs to be number(s). To change the data into this suitable format, each word of the sentence in the data must be identified and represented by a unique index and the process is called one-hot representation. It is sometimes called indexing. One-hot representation is a way of converting individual unique words into unique numbers. In our model, we used a built-in method index_word[index] methods to convert vocabulary words into a unique id representation. Here the first word in a vocabulary is represented with the first index zero (0) and the last word is represented with the index total number of unique words minus one. An index_word attribute is a word-to-index dictionary where words are the keys and the corresponding integers are the values. Thus, each word in a sentence corresponds to a vocabulary item in a vocabulary. Here, we have a vocabulary item of 1,734,130 characters without blanks which is from 254,328 words in 9280 lines. After normalizing, removing stop words and stemming, total number of a unique vocabulary word length is 22,075 words for Amharic vocabulary and 18,987 words for Wolaita vocabulary and each component in that vector corresponds to a specific word in the vocabulary. The neural network that operates on this vocabulary will not be able to do anything with any other word except the words listed in vector representation. So, we represent each word by simply putting 1(one) in the position corresponding to that words in the vector and 0(zero) everywhere else as shown in Table 4.2.

Table 12: Word Representation in Indexing

index	corpus	የኋለኛው	ጠላት	የሚሻረው	ሞት	ነው
0	.....	0	0	0	0	0
1	የኋለኛው	1	0	0	0	0
2	ጠላት	0	1	0	0	0
3	የሚሻረው	0	0	1	0	0
4	ሞት	0	0	0	1	0
5	ነው	0	0	0	0	1
6	....	0	0	0	0	0

4.7 Word Embedding

From sequences of vector representation (one-hot representation), word embedding will be formed. It is a way of representing words on a vector space where the words having the same meaning have similar vector representations [117] as shown in Table 4.3 and Figure 4.3. Thus, it focuses on the construction of a semantic representation of words based on the statistical distribution of word co-occurrence in the text corpus. One of the basic advantages of neural word embedding is the reduction of out-of-vocabulary impact. This is possible because words will not be completely unknown as far as they have feature vectors even if they may not be seen in the training dataset. Words can be encoded into limited vector spaces or neural word embeddings in one of the two methods, the continuous bag of words (CBOW) method and the skip-gram method (Word2Vec). CBOW takes the average of the possible contexts of a word in representing it in a limited dimensional vector space. For example, ‘Apple’ can refer to either the name of the company or the type of fruit. CBOW places the vector of the word ‘Apple’ to a medium position of the two contexts.

Skip-gram model was introduced by Mikolov *et al.* [117] which is an efficient method for learning high quality vector representations of words from large amounts of unstructured text data. The authors of Skip-gram model stated as it is efficient and more accurate than CBOW in that it does not involve dense matrix multiplications. Skip-gram method, on the other hand, can assign two vector values for the above two contexts of ‘Apple’. Based on the above preferences, we used Skip-gram Word2Vec algorithm to construct our semantic model in this thesis.

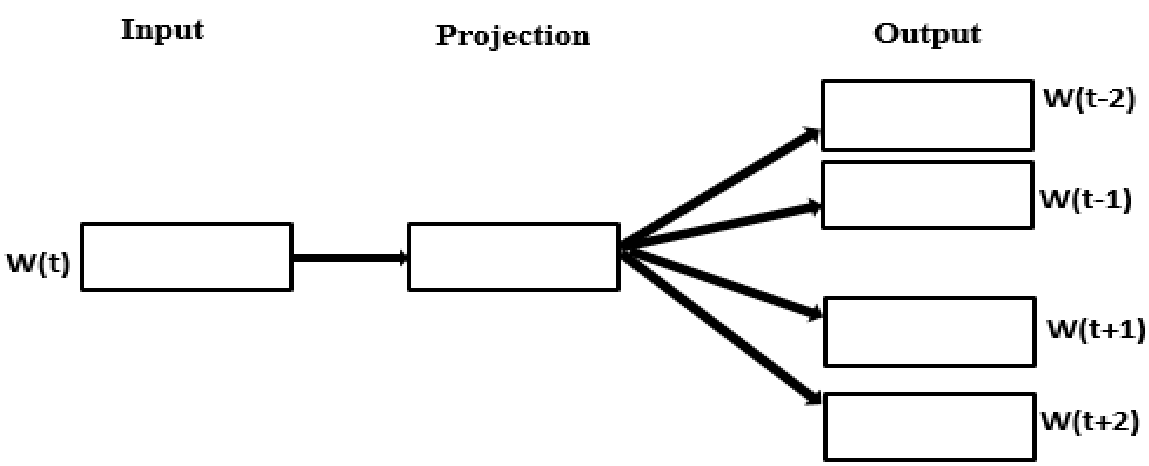


Figure 93: Skip Gram Model

Table 13: Semantic Relationship of Words Representation in Word-embedding

	አንበሳ (Lion)	ስጋ (Meat)	በላ (Eat)	ሰው (People)	በሬ (Ox)
ምግብ(food)	0.1	0.9	0.6	0.1	0.5
ግስ(verb)	0.3	0.3	0.9	0.4	0.4
ስጋ በዪ (carnivore)	0.9	0.0	0.0	0.9	0.1
ብዛት(size)	0.6	0.2	0.2	0.6	0.6
የዱር እንስሳ(rogue)	0.8	0.0	0.1	0.1	0.2

Here we can represent it as, meat is food (ስጋ ምግብ ነው) highly in 90%, human is rogue (ሰው የዱር-እንስሳ ነው) in 10%) and meat is neither carnivore nor rogue (ስጋ ስጋ-በዪም የዱር እንስሳም አይደለም). Thus, we can represent the remaining in this way. As we can see in the representation, አንበሳ (lion) and ሰው (human) are very similar. Thus, we can say both አንበሳ ስጋ በላ (lion eat meat) and ሰው ስጋ በላ (human eat meat). But we cannot say በሬ ስጋ በላ (ox eat meat), because it is semantically incorrect. Similar words will have a similar vector. The position of a word within the vector space is learned from text and is based on the words that surround the word when it is used. Thus, it stores each word close to the other word which is close to it in the meaning as shown in Figure 4.4 [117].

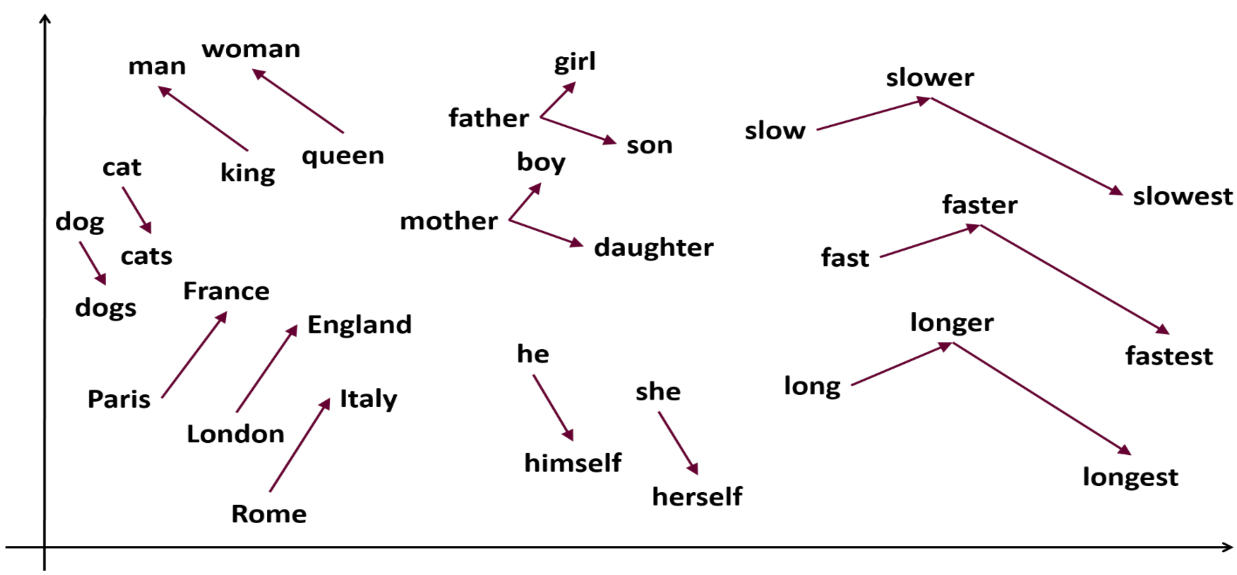


Figure 10 Word-embedding Example in Matrices

4.8 Padding

The next step after word embedding is applying padding to the sentences that are longer or shorter than a certain length. When batching the sequence of word ids together, each sequence needs to be the same length. Since sentences are dynamic in length, we can add padding to the end of the sequences to make them the same length. The Encoder part receives this embedded form of data and generates a contextual relation between words. The Decoder part receives the context generated by Encoder and searches for matching words within Wolaita language. When the sequence includes the longer sentence, the enhancement mechanism is needed to help to store of the longer context within the sentence. This enhancement mechanism is called padding. Thus, padding is a term that describes the act of taking a value and adding pad characters to the left or right (or both) sides of a string in order to fix the length of the string. We performed it by using a built-in function *tf.keras.preprocessing.sequence.pad_sequences ([inputs])* and it results in all the Amharic and Wolaita sequences have the same length by adding padding to the end of each sequence.

4.9 Encoding

Encoding is the process of converting the sequences of inputs into a single-valued context vector. In our Encoder-Decoder model, we have used a RNN based model to design Encoder for Seq2Seq model. The Encoder of a Seq2Seq network is a RNN that encodes the complete information of the source sequence which is the source language like Amharic into a single real-valued vector, also known as the context vector, which is passed to the Decoder to produce an output sequence, which is the target language like Wolaita in basic Encoder-Decoder model. To work on textual data by Encoder part of Encoder-Decoder LM, the data must be given in the embedded form of sequential data. If $X=\{x_1, x_2, x_3 \ldots x_n\}$ is the input sequence to the Encoder with n sequence length, the Encoder will produce $Y= \{y_1, y_2, y_3 \ldots y_m\}$ context in the sequence of input data. To drive this context, the selection of neural network architecture is necessary.

In the RNN based Encoder-Decoder model, Encoder can use either of DNN architecture. Some of these architectures are LSTM, GRU, gated linear unit (GLU) or bi-directional RNN [88]. From these architectures, GRU and LSTM are most familiar for Encoder part of NMT on text-based translation [117]. The difference between these two architectures is that GRU has no output gate, while LSTM has both output gate and a backward gate for storing the internal state of the network. Therefore, GRU is LSTM that has no output gate, so writes the contents from its memory cell to the larger network at each time step [27, 117]. We preferred the GRU unit as both Encoder and Decoder unit. However, we used single layer GRU in architecture Figure in 4.2, we have used deep-multilayer GRU. The GRU uses gate unit to control the flow of information. It uses the current input and its previous output, which can be considered as the current internal state of the network, to give current output [110].

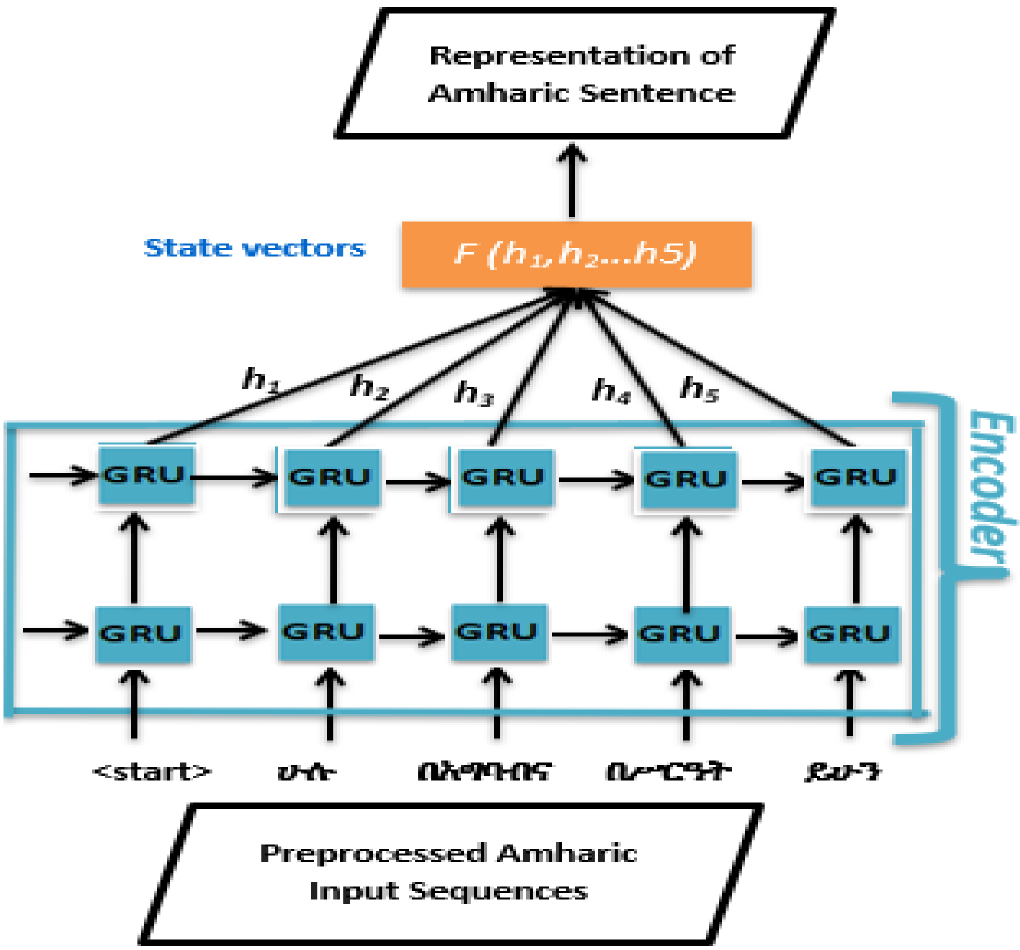


Figure 11: Encoder Architecture

GRU uses two parameters to drive the sentence context. These are the collection of currently inputted data and the previous output of the network [88]. Therefore, when using RNN the previous output is considered as the internal state of the neural network and if the current input is the starting word, then the input is marked by the sign of the start of token symbol <start> token indicator and given as input as we have explained in Equation. 4.2. Then Encoder outputs context vector as its final output from current input data and previous output of the system. This context vector will be fed into Decoder part either by using the enhancement mechanism which is attention-mechanism (we will see the detail in section 4.11) or without using an attention mechanism. If the attention mechanism is left, the last layer of Encoder network will be connected to the Decoder part of the network [88]. If we use the attention mechanism, the last layer of Encoder network will be connected to attention layer of the network [111].

4.10 Decoding

Decoding is the process of generating the equivalent meaning to what encoded in Encoder [113]. In our Encoder-Decoder model, we have also used RNN based model to design Decoder for Seq2Seq model. There are several choices for the Decoder architecture that combines these inputs to generate the next hidden state: linear transforms with activation function, GRUs, LSTMs, etc. Typically, the choice here matches the Encoder. So, if we use GRUs for the Encoder, then we also use GRUs for the Decoder too. Since we used GRU architecture for Encoder part, the Decoder part of RNN based system is also designed by using GRU architecture. It takes some representation of the input context and the previous hidden state. Once the Decoder is set up with its context, a special token to signify the start of output generation (i.e., <end> token appended to the end of the input; there's also one at the end of the output Wolaita sentence) is passed to it. Then, all layers of GRU run one after the other, following up with a softmax on the final layer's output to generate the first output word. Then, we pass that word into the first layer and repeat the generation. This is how we get the GRUs to act as a language model. Until the end of text generation, Decoder outputs word prediction and generates a new hidden Decoder state and a new output word prediction [112]. The number of GRUs used is set as the maximum length of the sequence in the target sentence data (i.e., 5 GRU unit is used for <**start> ሁሉ በአግባብና በሥርዓት ይሁን** tokens) and when the shorter sentence entered into the network, the end of sentence marked with the end of string indicator symbol <end> and the other units are fed with padding which is used to change the variable-length vector into a fixed-length vector by completing the length of shorter sentence by using zero value.

In order to find the matching word, the Decoder uses the beam searching mechanism of best search type with the use of softmax function which searches for the best matching word. Therefore, the

Decoder generates one word at a time and repeats the searching until it encounters end of string indicator <end> which shows the last word of the sentence. See Figure 4.6 for an example of a Decoder network. Once we have the output sequence, we use the same learning strategy as usual.

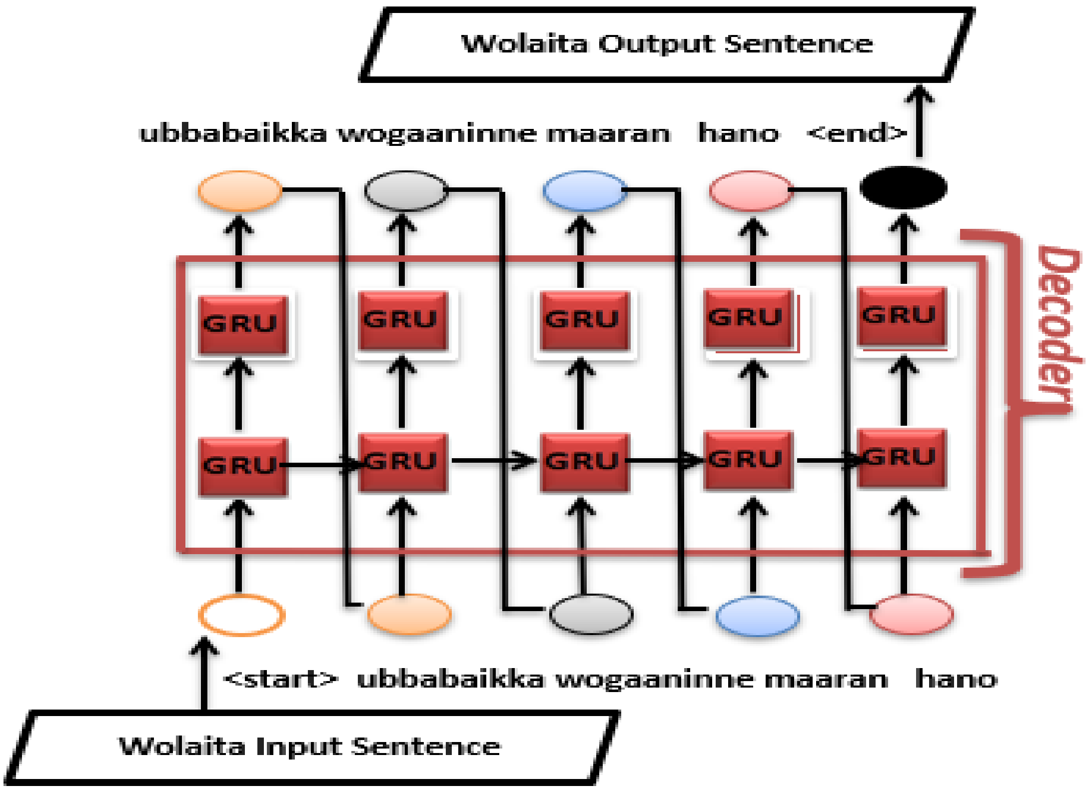


Figure 12: Decoder Architecture

Here is a recap that a Seq2Seq model with RNN based GRU unit does in order to translate the Amharic sentence "ሁሉ በአግባብና በሥርዓት ይሁን" into the Wolaita sentence "ubbabaikka wogaaninne maaran hano". First, we start with 5 one-hot vectors (i.e. one for each word including <start> token) for the input. Then, a *GRU* network reads the sequence and encodes it into a context vector. This context vector is a vector space representation of the notion of translating the Amharic text. It is used to initialize the first layer of another GRU. We run one step of each layer of this network, perform softmax on the last layer's output, and use that to select our first output word. This word is fed back into the network as input, and the rest of the sentence "ubbabaikka wogaaninne maaran hano" is decoded in this fashion. During back-propagation, the Encoder's GRU weights are updated so that it learns a better vector space representation for sentences, while the Decoder's GRU weights are trained to allow it to generate grammatically correct sentences that are relevant to the context vector [113].

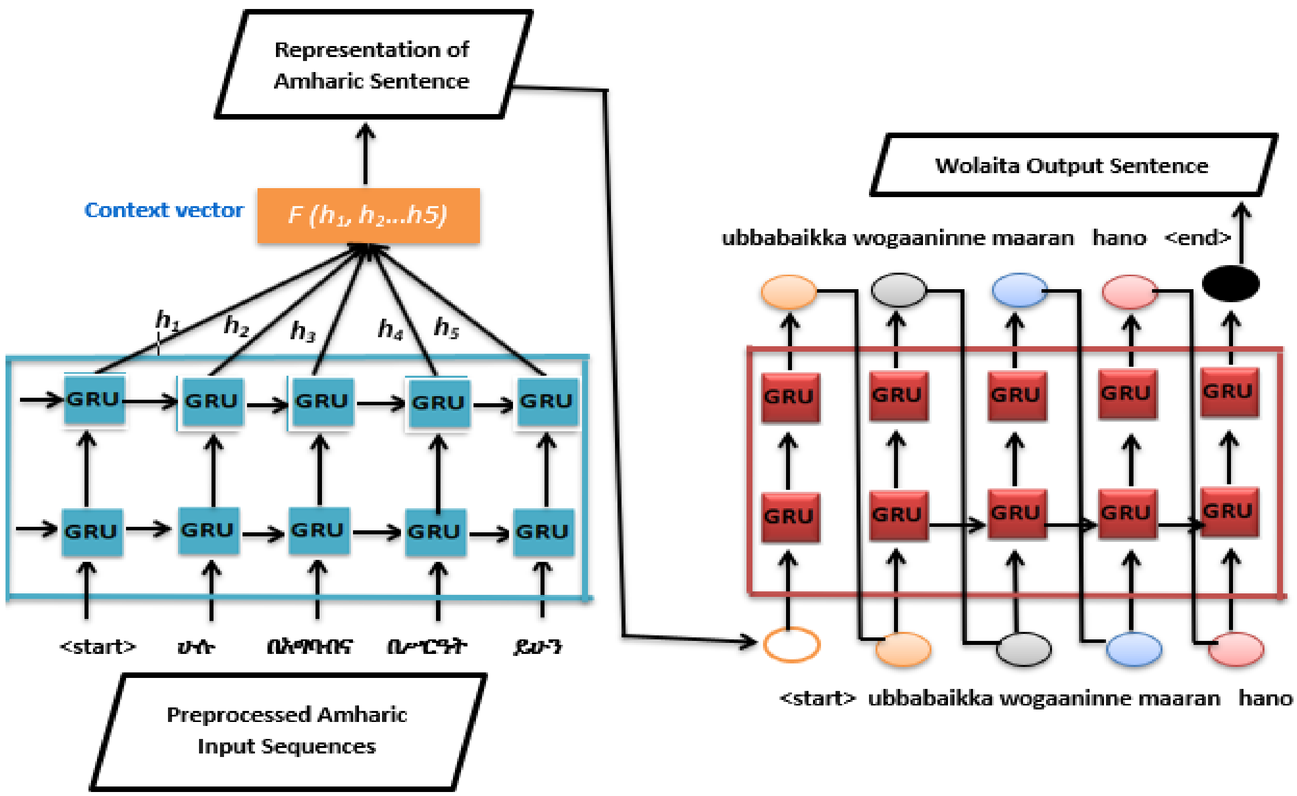


Figure 13: Basic Encoder-Decoder Model without Attention

4.11 Attention Mechanism

Encoder in basic Encoder-Decoder based RNN architecture encodes the complete information of the source sequence into a single real-valued vector, also known as the context vector, which is passed to the Decoder to produce an output sequence. Here, a context vector has the responsibility to summarize the entire input sequence into a single vector. Thus, the Decoder has access only to the output of context vector of the Encoder as shown in Figure 4.8.

Therefore, the entire input sequence representation in a single vector is inefficient; and unable to represent when the sentences are being long and the vocabulary size is becoming large. One effective way to address such a problem is through the attention mechanism, which has gained popularity recently in training neural networks. Unlike Encoder-Decoder architecture without attention, in which the source representation is only used once to initialize the Decoder hidden state; Encoder-Decoder architecture with Attention model predicts a target word based on the context vectors associated with the source position and the previously generated target words. In this way, we solved the basic Encoder-Decoder problem to handle long sentences by adding attention mechanism to it. In addition to this, we handled a larger number of vocabulary size problem of basic Encoder-Decoder based approach by adding attention mechanism to Encoder-Decoder LM which minimizes higher dimension vector into lower dimension vector [114]. It helps to pay attention to the most relevant information in the source sequence.

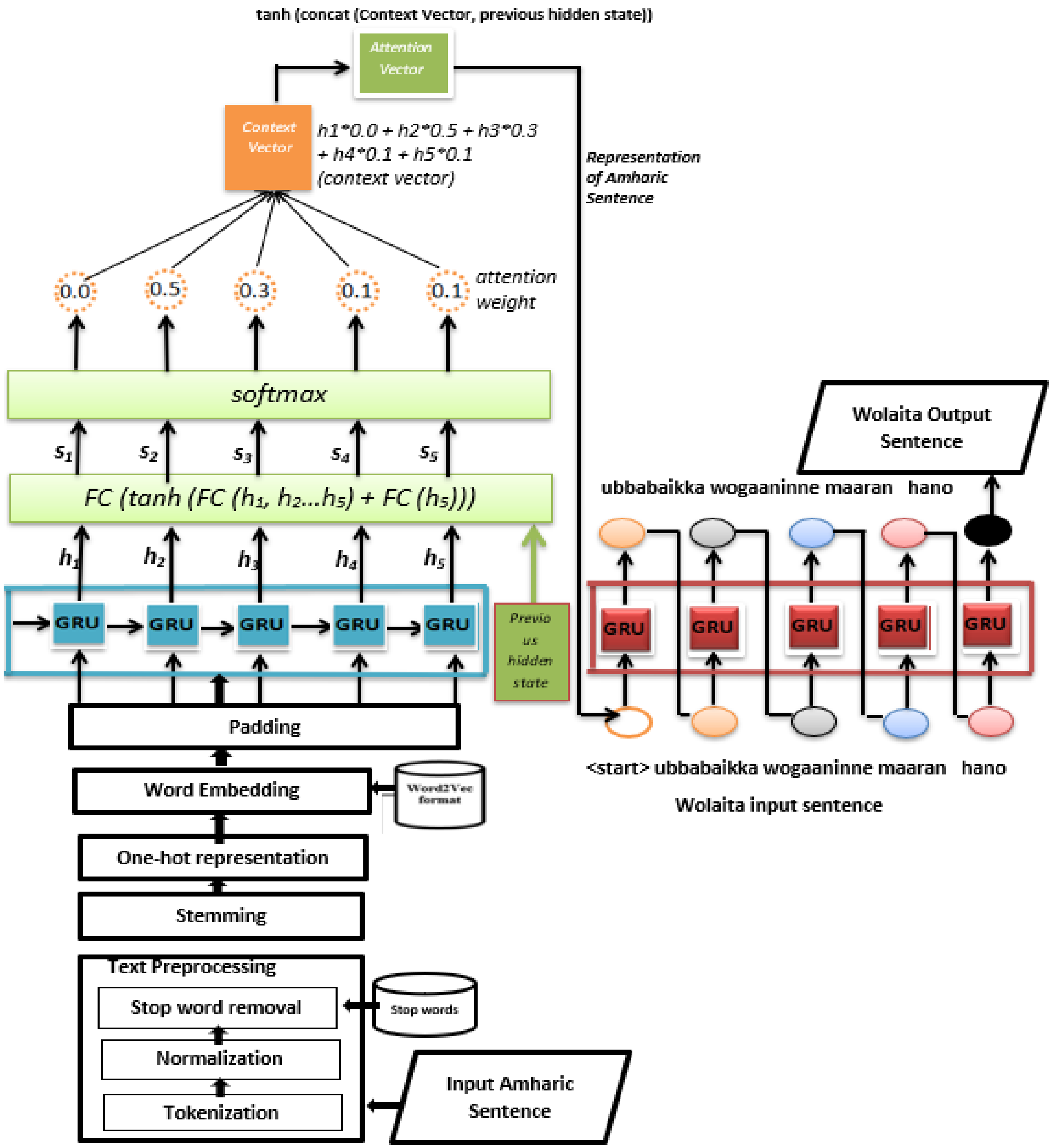


Figure 14: Attention-based Encoder-Decoder Architecture

Chapter 5: Experimentation and Evaluation

5.1 Introduction

Under this Chapter, the conducted experiments and discussion is presented in a way that states the evaluation and performance of the experiments. Section 5.2 details about data collection and preparation, Section 5.3 about system environment/tools used for system development, Section 5.4details about parameter optimization and training of the experimental systems and Section 5.5 details about the experimental results of attention-based and non-attention-based system in comparison. Finally, this chapter details about Evaluation metrics and discussion on the result of the study.

5.2 Data Collection and Preparation

For our study, we have collected datasets of total 3000 sentences of parallel Amharic-Wolaita corpus from different sources mostly from religious books. 6280 datasets are taken from [3]. Hee the datasets are available separately and we make it to Amharic-Wolaita parallel dataset. The total dataset we used for the system training and testing is 9280 parallel sentences. After we had finished collecting the datasets, we uploaded it into Google drive to load into Google collaboratory [115] since we have used GPU rather than CPU for the sake of fastening the training time. Then we aligned this data of parallel corpus along with its translation with the use of space tab (' ') separation. Therefore, this aligned data is stored in text format to feed into systems. Then, we shuffled the data elements before classifying into a training set and testing set. After shuffling the data element, we have classified the data elements into a training set and testing. For the classification, we followed one of the best techniques called Pareto principle (80/20) in which 80 percent of total data is used for the training set, while 20 percent of the total dataset is left for testing the system [41]. Thus, the training set consists of 7424 sentences of parallel corpus, while the test set consists of 1856 sentences. We used a built-in function *train_test_split(amharic_tensor, wola_tensor, test_size=0.2)* to split the total dataset into training and testing set.

5.3 System environment/tools used for development

To implement our system, choosing a programming language and preparing the required environment is necessary. A programming language chosen to implement the system is a python programming language. Python programming language supports a set of a freely available library in the deep learning algorithm. We used Keras library, TensorFlow library and NumPy library which is freely available. We have chosen colab as a baseline for implementation. Colab is the only tool which supports graphical processing unit (GPU) which speeds up the training time hundreds time faster than central processing unit (CPU).

5.4 Parameter Optimization and Training the Experimental Systems

In our experimental study, we have trained and tested the proposed system on our parallel corpus of 9280 sentences in two columns (one for Amharic and other for Wolaita) separated by space tab as we provided a sample in Appendix I. To get the desired result, we have done different experiments on different issues to adjust the parameters of the model. The first issue is the selection of neuron units to the number of dense layers and batch sizes. For training and testing our dataset, we can select one of 16, 32, 64, 128, 256, 512, 1024, 2048 neuron units with batch sizes based on the number of data size we are using. Batch size is several samples processed before the model is updated. Thus, the size of a batch must be more than or equal to one and less than or equal to the number of samples in the training dataset. For our 7424 training samples and with a batch size of 64, the algorithm takes the first 64 samples from the training dataset and trains the network. Next, it takes the second 64 samples and trains the network again. We can keep doing this procedure until we have propagated all samples through the network. For each epoch, it iterates for 116 times as detailed in Appendix IV. The number of batches in each epoch is inversely proportional to batch size. Training a system by minimum batch size results in less accurate estimate of the gradient and training a system by maximum batch sizes requires more memory and even networks train slower. Thus, we preferred 1024 neuron units for 2 dense layers with 64 batch sizes for our training dataset since our corpus is medium size.

The second issue after setting neuron units and batch size is the selection of word embedding size. For this purpose, we trained 1024 neuron units with 64 batch size in three different number of word embedding sizes 64, 128 and 256 respectively for 50 epochs. From these different word embedding sizes, we selected 1024 neuron units with 64 batch size and 256 word embedding dimension for 50 epochs because it minimizes the loss level when compared with 1024 neuron units with 128 batch size and 128 embedding dimension, 1024 neuron units with 256 batch size and 256 embedding dimension and 1024 neuron units with 64 batch size and 128 embedding dimension with the same number of epochs as shown in Figure 5.2. The number of epochs is the number of complete passes (number of looking back into a single dataset element) through the training dataset. The bigger size shows a sign of faster convergence at the beginning but starts to diverge very soon. Therefore, we choose the medium size of word embedding as 256.

After adjusting neuron units, batch size and word embedding, choosing a learning rate for training is important. In order to select optimum learning rate, we trained our system in 0.01 learning rate which takes 63, 844 secs and its loss level is 0.017, 0.001 learning rate which takes 6,612 secs and its loss level is 0.015 and 0.0001 learning rate which takes 25,684 secs and its loss level is 0.025. As shown in Figure 5.3 and Figure 5.4, Adam's optimizer (which defaults learning rate to 0.001) minimizes a training data's training time highly in optimum loss level in comparison to 0.01 learning rate and 0.0001 learning rate. Thus, we have initialized parameter by using *optimizer = tf.keras. optimizers.Adam()*, which defaults learning rate to 0.001 and we selected sigmoidal optimizer to optimize parameter. Therefore, the training of our architecture took 6,612 seconds (1 hour 50 minute and 12 secs). To our model, we have used batch training with a batch size of 64 and trained for 50 epochs on GPU based core i5 with 4GB RAM computer. As we have seen in Figure 5.1 the loss level is simultaneously decreased until reaching to the 30th epoch and then it decreased in rare. If we can use more dense layers with big dataset it may minimize the loss level with less than that of 50 epochs.

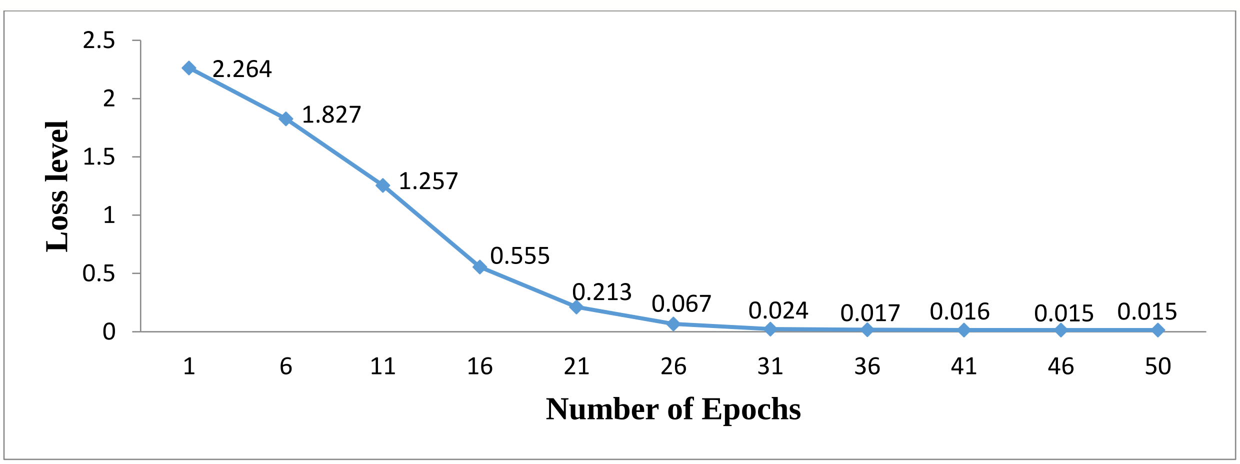


Figure 15: Loss Level Vs Number of Epochs

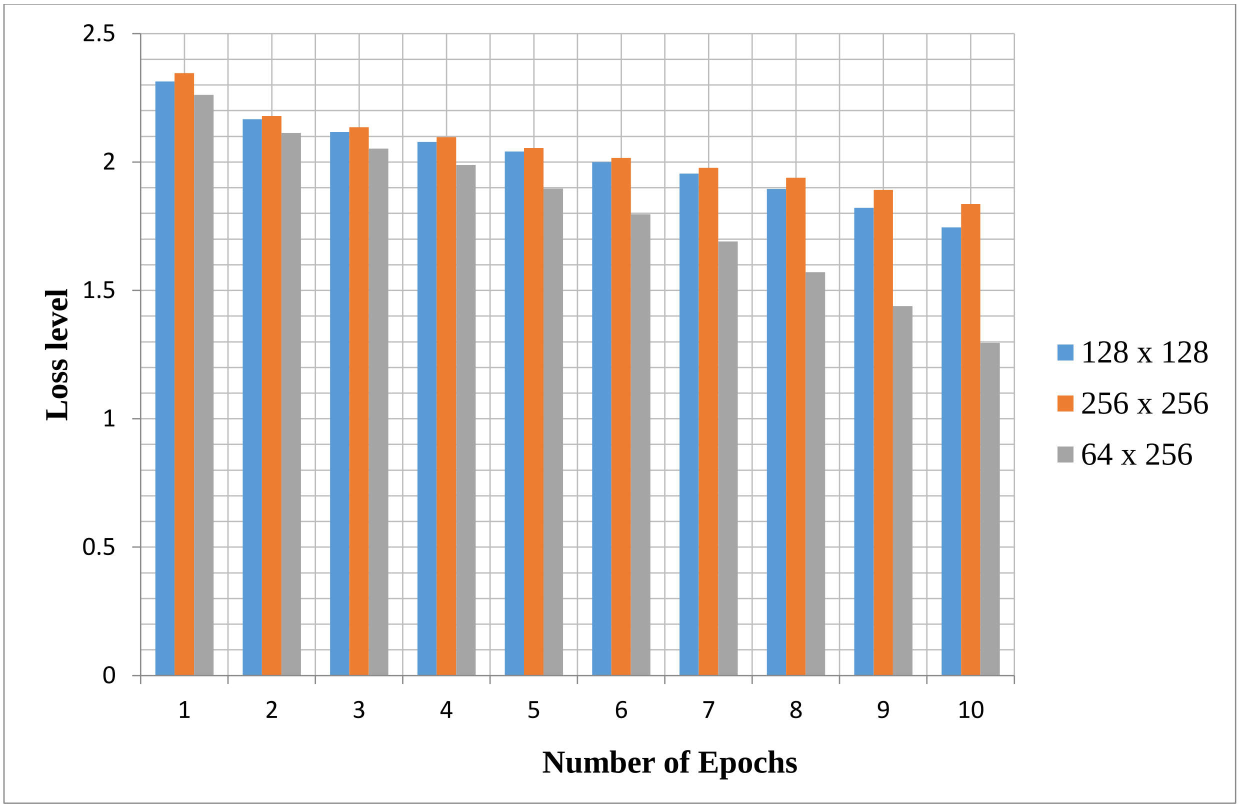


Figure 16: Loss Level for each Batch Size and Embedding Dimension for Number of Epochs

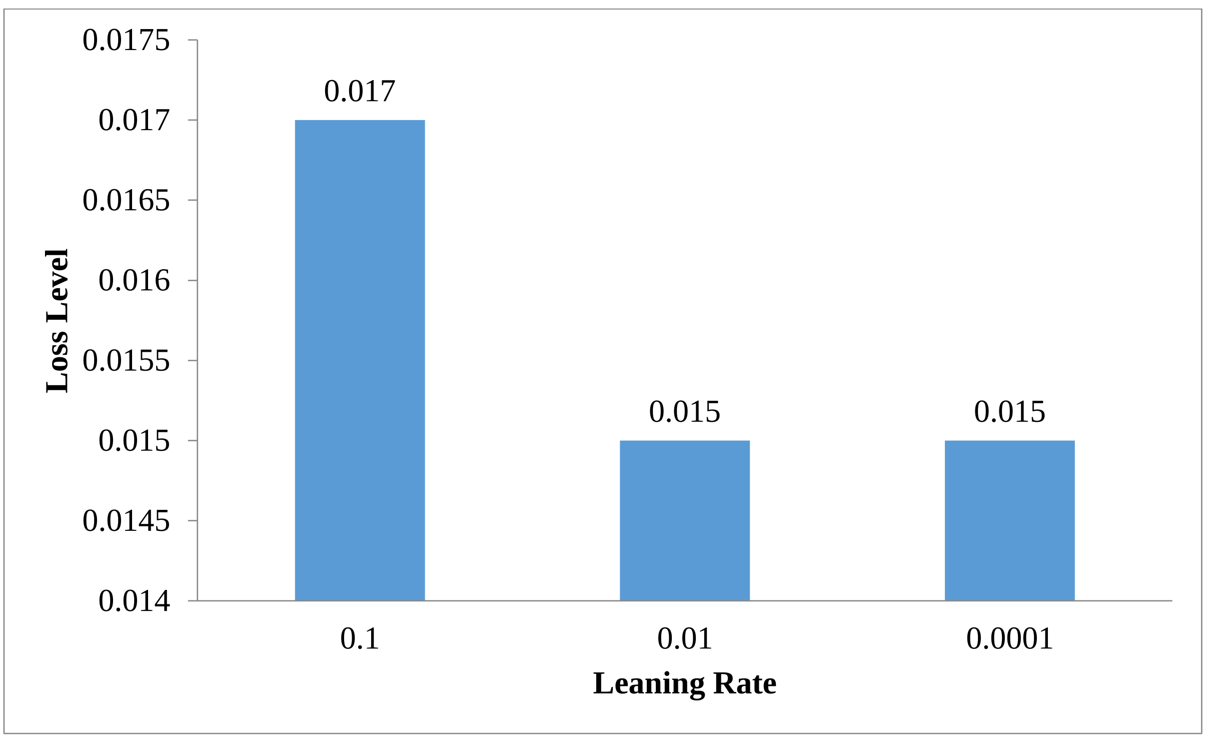


Figure 17: Loss Level Vs Learning Rate for Training

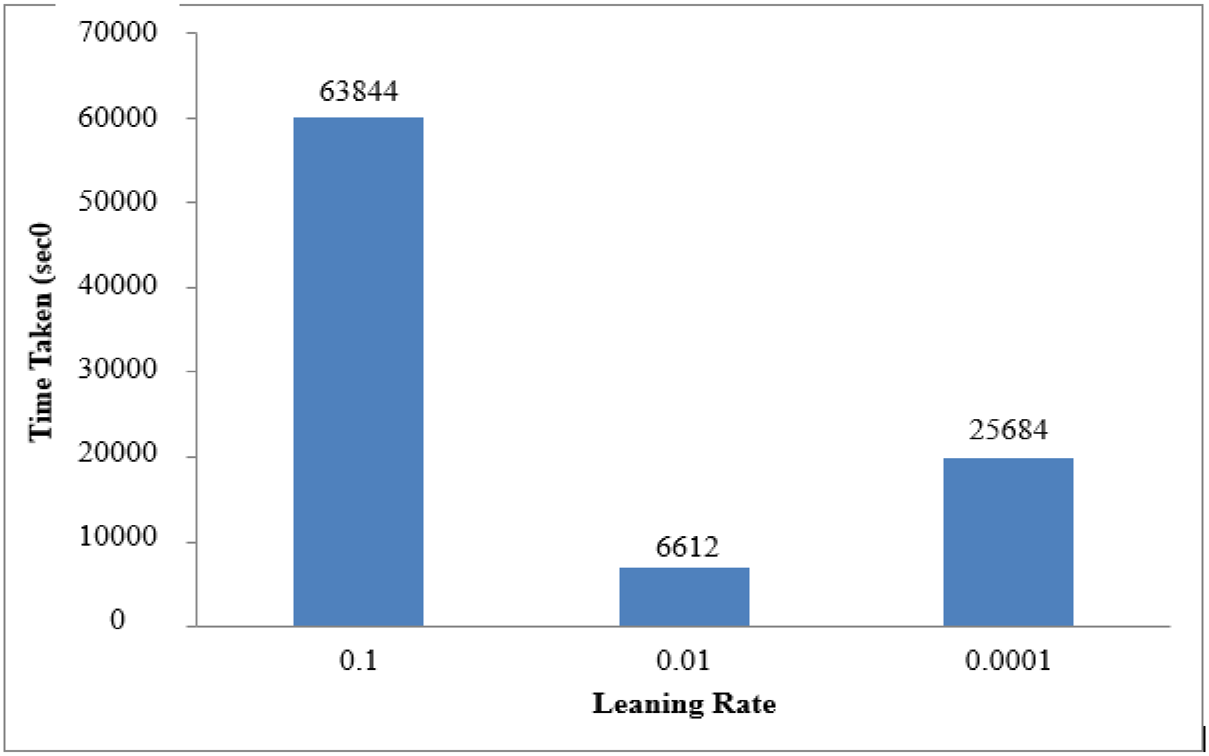


Figure 18: Time Taken Vs Learning Rate for Training

5.5 BLEU Evaluation Metrics

To evaluate the performance of any MT in terms of translation accuracy, various methods have been developed. One of such metrics is bi-lingual evaluation understudy (BLEU) that is used to measure translation accuracy by comparing the system's translation output against human translated reference sentences. A high-quality translation is the one which is closer to a professional human translation and BLEU's main idea is the measurements of this closeness. BLEU score value falls in the range between 0 and 1. Most of the time it is reported in percent from 0-100. The higher the BLEU score the more the translation resembles with the translation of a human translation.

5.6 Experimental Results

In our experimental study, we have implemented the two designed system of our study as Amharic to Wolaita NMT without attention and attention-based. Rather than adding attention mechanism for Attention-based system, we trained and tested the systems in a similar way to get a correct comparison of their performance. After training and testing the two systems, we have measured the result of testing the system by using BLEU score metrics to see the difference in their scores. In order to report the result of our testing, we have used the average BLEU score result of 50 epochs. For training our system in this epoch, attention-based system has taken 6,612 secs and non-attention-based system has taken 21,253 secs for the same training dataset in 0.01 learning rate.

When translating input sentence to output sentence, we use human translated reference sentence to get BLEU score of the translation. We calculated the average BLEU score from the translation as some of the translation is detailed in Appendix V. Therefore, this average result of Amharic-to-Wolaita NMT system without attention in BLEU score is 0.5960, and attention-based Amharic-to-Wolaita NMT system is 0.6258. Thus, as we have discussed in section 4.11 attention-based system gives us +0.02978 BLEU gain. In addition to BLEU score improvement, the training time for attention-based system is efficient than non-attention-based system.

5.7 Discussion on the Result of the Study

To accomplish the objective of this thesis work, we focused on the design and implementation of MT from Amharic to Wolaita language. We conducted two experiments using two different mechanisms, i.e., non-attention-based and attention-based approach with an expectation to get a better result in both BLEU score and time efficiency. As the length of the sentence increase, the inter-dependency of words at the beginning of the sentence and at the end of the sentence is loosely related. This architecture works quite well for short sentences and this may make it difficult for the neural network to cope with long sentences since the context within a sentence is derived as the inter-dependency of nearby words in a given sequences words in the sentence.

The second problem of Encoder-Decoder model is how to handle a larger number of vocabulary sizes available within the data. As each word in the sentence is visited, it must be assigned a new identity number in order to identify a word by a unique index at the time it encountered in the data. But when the length of the dictionary increases, the number used for word representation becomes higher and the dimension of word vector needed becomes higher. These two basic issues are solved by using an attention mechanism with a basic Encoder-Decoder architecture. To the best of our knowledge during the time of this work, there has not been any other work exploring the use of attention-based architectures for NMT.

One can observe that the result recorded from BLEU score shows the attention-based approach is better than the non-attention-based approach for Amharic-to-Wolaita translation. But the results recorded from both experiments are from less size of the corpus. As the size of corpus increases the accuracy also increases and similarly the BLEU score result can also increase proportionally.

According to the result of our experiments, non-attention-based Amharic-to-Wolaita NMT has shown interesting translation result as a BLEU score of 0.5960 with respect to our small datasets. Moreover, we have got great improvement by using attention-based system which has shown 0.02978 BLEU improvement when compared with non-attention-based system. Also, when we compared attention-based system with non-attention-based system in case of training time, attention-based system shows better time efficiency. Also, attention-based system can also store longer contextual relevancies of words found in longer sentence of datasets where we compare it to the non-attention-based system. Therefore, the comparison of BLEU score metrics for some range of word length shows that both of the systems have shown good translation result for translating shorter sentences than longer sentences. This feature has been described in Appendix-V, which compares both systems on the translation of the top 30 sentences from the parallel corpus. Therefore, as we can see from Appendix I when the length of the word in source sentence increases, the BLEU score level is decreases with an increase of sentence length for both attention-based and non-attention-based models. The results of the experiment confirmed that the attention-based model works better than the non-attention-based system with an increase of sentence length. Moreover, the other interesting feature seen on using attention-based translation with RNN model is that the RNN based model keeps contextual relatedness in which related words are used for translation rather than only using exact target word for translation. In that case, the BLEU score can be affected as it is dependent on the exact matching word of reference sentence rather than using related dictionary words. Therefore, the result of our model has been affected by this property of BLEU score metric and would have better than what is reported in Appendix V.

Chapter 6: Conclusion and Future work

This chapter discusses the conclusion driven from this research work and the Future work for any person or organization that interested to work on the machine translation between Amharic and Wolaita language pair (or any other language pair) or related task.

6.1 Conclusion

As the objective of the work is to improve the MT between the language pair by implementing RNN based architecture, two experiments were conducted to check the accuracy of the system using two different approaches. The first experiment is conducted by using the non-attention-based approach and it has a BLEU score of 0.5960. The second experiment is carried out by using an attention-based approach and it has a BLEU score of 0.6258. Therefore, the results of our experiments have shown that our attention-based system has shown better BLEU score in translations, which is 0.02978 improvements for translation and uses less training time in comparison with our non-attention-based system. More importantly the attention-based system shows the great result in case of translating longer sentence than the non-attention-based system. Also, the attention-based system is faster by comparing against the non-attention-based system. We have also seen as using attention-based translation with RNN model is that the RNN based model keeps contextual relatedness in which related words are used for translation rather than only using exact target word for translation.

6.2 Future Works

- ✓ Since our work is the first MT system developed from the Amharic language to Wolaita language pair as the translation is from Amharic to Wolaita language, we recommend anyone who interested in this work for making this system bi-directional in future work in order to easily distribute resources written in both languages.
- ✓ As the objective of our study is to implement MT only at the level of text to text translation, we recommend anyone to develop speech to text first and use our work for further study.

- ✓ Also, we recommend any interested one in our work to include a higher number of the dataset to further increase the translation quality since neural network needs more dataset in order to perform better and also, we recommend to try with more layers and more hidden units.
- ✓ For future we recommend any interested one to train our system for another language pairs.
- ✓ For future, we recommend one to use an autoencoder for the corpus separated by a space tab (like ሁሉ በአግባብና በስረዓት ይሁን \t ubbabaikka wogaaninne maaran hano). To do this, train as an auto-encoder, save only the Encoder network and train a new Decoder for translation from there.
- ✓ In order to make a system reliable, we recommend one to replace the position of untrained (unknown) words by some special references like UNK.

References

[1] https://www.omniglot.com/language/articles/MachineTranslation.htm last visited Oct 4,2019.

[2] Vani K, “Example-Based Machine Translation.” Internet: http://dspace.cusat.ac.in/jspui/bitstream/3623/1/EBMTorginal.htm Nov 10, 2017 [last visited: Oct 4, 2019].

[3] Solomon Teferra, Martha Yifru, Michael Melese, Million Meshesha, Solomon Atinafu, Yaregal Assabie, Biniyam Ephrem, Wondimagegnhue Tsegaye, Tsegaye Andargie, Wondwossen Mulugeta, Hafte Abera, Tewodros Abebe, Amanuel Lemma, Seifedin Shifaw “Parallel Corpora for bi-Directional SMT for Seven Ethiopian Language Pairs”, Proceedings of the First Workshop on Linguistic Resources for NLP, pp. 83–90, Santa Fe, New Mexico, USA, August 20, 2018.

[4] Eleni Teshome, "Bidirectional English-Amharic Machine Translation: using constrained corpus", unpublished masters thesis, Addis Ababa University, Ethiopia, 2013.

[5] Sloculn, Jonathan, "A survey of Machine Translation: its history, current status, and future prospects", 1985.

[6] Sisay Adugna. "English-Afaan Oromo Machine Translation: An Experiment using Statistical Approach", unpublished masters thesis, Addis Ababa University, Ethiopia, 2009.

[7] Microsoft.com,“Microsoft Translator Blog: Microsoft Translator launching Neural Network based translations for all its speech languages,” Online: https://www.microsoft.com/en-us/translator/blog/2016/11/15/microsoft-translator-launching neural-network-based-translations- for-all-its-speech-languages/. [last accessed: Oct. 12, 2019]

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. Proceedings of the 3rd International Conference on Learning Representations (ICLR), pp. 1–15. 2015.

[9] Sutskever, I., Vinyals, O., & Le, Q. V. “Sequence to sequence learning with neural networks”, In Advances in neural information processing systems pp. 3104-3112, 2014.

[10] H. Luong, K. Cho, and Ch. Manning, “Neural Machine Translation” - Tutorial ACL 2016.

[11] Nal Kalchbrenner and Phil Blunsom.. Recurrent continuous translation models. In Proceedings of the 2013 Conference on Empirical Methods in NLP. Association for Computational Linguistics, Seattle, Washington, USA, pp. 1700–1709, 2013.

[12] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. “Learning Phrase Representations using RNN Encoder-Decoder for SMT,” In Proceedings of the Conf. on EMNLP, pp. 1724-1734, 2014.

[13] “Neural Machine Translation - Tut ACL” Oct, 2019, https://sites.google.com/site/acl16nmt/

[14] Lemma Lessa, "Development of Stemming Algorithm for Wolaita Text", unpublished masters, Addis Ababa University, Ethiopia, 2003.

[15] “Constitution of the Federal Democratic Republic of Ethiopia”, www.wipo.int/edocs/ laws/en/ et/et007en.pdf.

[16] “NMT using keras” https://www.analyticsvidhya.com/blog/2019/01/neural-machine-translation-keras/ last accessed: oct 6, 2019.

[17] Almahairi, A., Cho, K., & Habash, N, “First Result on Arabic to English NMT”, Vol. 1, 2016.

[18] Takahashi S., Wada H., Tadenuma R., and Watanabe S. “English-Japanese Machine Translation”. Readings in Machine Translation. 2003

[19] Holger Schwenk, Jean-Baptiste Fouet, and Jean Senellart, “First Steps towards a general purpose French/English Statistical Machine Translation System”, In Proceedings of the 3rd workshop on statistical Machine Translation, pp. 119-122, Ohio, 2008.

[20] “Top ten most spoken language in the world”, listverse.com, last visited Oct 2, 2019

[21] Mulu Gebreegziabher Teshome and Laurent Besacier. "Preliminary experiments on English Amharic Statistical Machine Translation." In SLTU, pp. 36-41. 2012.

[22] Michael Gasser, “Toward a Rule-Based System for English-Amharic Translation”, In Proc of the 8th Int. 4th workshop on African LT, 2012.

[23] Jabesa Daba and Yaregal Assabie “A Hybrid Approach to the Development of Bidirectional English-Oromiffa MT”, In Proceedings of the 9th Int. Conference on NLP (PolTAL2014), Springer Lecture Notes in Artificial Intelligence (LNAI), Vol. 8686, pp. 228-235, Warsaw, Poland, 2014.

[24] Akubazgi Gebremariam, “Amharic-Tigrigna machine translation using hybrid approach”, unpublished masters thesis, Addis Ababa University, Ethiopia, 2017.

[25] Tadesse Kassa, “Morpheme-Based Bi-directional Ge’ez Amharic Machine Translation”, unpublished masters thesis, Addis Ababa University, Ethiopia, 2018.

[26] Wakasa Motomich. “A descriptive study of the modern Wolaytta language”. Unpublished PhD thesis, University of Tokyo, 2008.

[27] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, and Mohammad Norouzi, “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation” Oct 8 2016.

[28] J. Hirschberg and C. D. Manning. “Advances in NLP”, Retrieved October 4, 2019, from Science Mag: http://science.sciencemag.org/, Sept 12 2019.

[29] Yitayew Solomon and Million Meshesha, " Bi-directional Afaan Oromo-English Statistical Machine Translation", unpublished masters thesis, Addis Ababa University, Ethiopia, 2017.

[30] Rae Steinbach, “Neural machine translation: Now and into the future” https://www.banklesstimes.com/2018/02/24/NMT-now-future/ last accesed: Oct 7 2019.

[31] M. Utiyama, D. Do, and E. Sumita, “Machine translation from Japanese and French to Vietnamese, the difference among language families,” October 2015.

[32] Teshome Kassie.“Word Sense Disambiguation for Amharic Text Retrieval”, MSc thesis, Addis Ababa University, Ethiopia, 2018.

[33] Mulubrhan Hailegebreal, “A Bidirectional Tigrigna – English SMT”, unpublished masters thesis, Addis Ababa University, Ethiopia, 2017.

[34] Habtamu Fanta Alambo, “Speaker Dependent Speech Recognition for Wolaytta Language”, unpublished masters thesis, Addis Ababa University, Ethiopia, 2010.

[35] Biruk Abel, “Geez to Amharic Machine Translation using Hybrid approach”, unpublished masters thesis, Addis Ababa University, Ethiopia, 2018.

[36] Atelach Alemu Argaw, “Amharic Stemmer: Reducing Words to their Citation Forms”., Proceedings of the 5th Workshop on Important Unresolved Matters, pp. 104–110, 2007.

[37] Samuel Eyassu,. “Classifying Amharic News Text Using Self Organizing Maps”. Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, pp. 71–78, Ann Arbor. @2005 Association for Computational Linguistics. 2005.

[38] Bethelhem Mengistu. N-gram-Based Automatic Indexing for Amharic Text. unpublished masters thesis, Addis Ababa University, Ethiopia. 2002.

[39] Gochel, Daniel. “An integrated approach to automatic complex sentence parsing or Amharic text”. Diss. Addis Ababa University, 2003

[40] Tigist Tensou, “Word Sequence Prediction for Amharic Language” unpublished masters thesis, Addis Ababa University, Ethiopia, 2014.

[41] Nega Alemayehu and Peter Willett, “Stemming of Amharic Words for Information Retrieval”, Literary and Linguistic computing, 17(1): 1-17, 2002.

[42] ጌታሁን አማረ፣ የአማርኛ ሰዋስው በቀላል አቀራረብ, 1989.

[43] ባዬ ይማም፣ የአማርኛ ሰዋስው, Addis Ababa, Ethiopia: EMPDA Publications, 1995.

[44] ጌታሁን አማረ፣ ዘመናዊ የአማርኛ ሰዋስው በቀላል አቀራረብ, Addis Ababa, Ethiopia, 2010.

[45] Dawit Mulugeta, “Geez to Amharic Automatic Machine Translation: A Statistical Approach”, unpublished masters thesis, Addis Ababa University, Ethiopia, 2015.

[46] Abeba Ibrahim, “A hybrid approach to Amharic base phrase chunking and parsing”, unpublished masters thesis, Addis Ababa University, Ethiopia, 2013.

[47] Michael Gasser, *Hornmorpho User's Guide*, 2012.

[48] Berhanu Herano Ganta, “Part of Speech Tagging for Wolaita Language”, unpublished masters thesis, Addis Ababa University, Ethiopia, June 2015.

[49] Lamberti, Marcello and Roberto Sottile, “The Wolaytta Language”, In Studia linguarum Africae Orientalis, pp. 79–86. Cologne: Rüdiger Köppe, 1997.

[50] Demewoz Beldados, “Automatic Thesaurus Construction from Wolaytta Text”, unpublished masters thesis, Addis Ababa University, Ethiopia, 2013.

[51] Li Peng, “A Survey of Machine Translation Methods”, 2013, http://www.iaesjournal.com/online/index.php/telkomnika/article/viewFile/2780/ [last accessed: Dec 30, 2019].

[52] M. D. Okpor , "Machine Translation Approaches: Issues and Challenges," IJCSI International Journal of Computer Science Issues, Vol. 11, No. 5, pp. 159-165, 2014.

[53] Sainik Kumar Mahata, Dipankar Das, and Sivaji Bandyopadhyay “MTIL2017: Machine Translation Using RNN on Statistical Machine Translation” Journal of Intelligent Systems · May 2018.

[54] Adam Lopez,“Statistical Machine Translation”, ACM Computing Surveys, Vol. 40, Aug 2008.

[55] Mark Y. Liberman and Kenneth W., “Text analysis and word pronunciation in text to speech synthesis”, Advances in speech signal processing, pp. 791-831, Dekker, New York,1992.

[56] Suhad M. Kadhem, Yasir R. Nasir, “English to Arabic Example-based Machine Translation System” IJCCCE Vol.15, No.3 January 2015, DOI: 10.13140/RG.2.2.10922.88006.

[57] Daniel Jurafsky and James Martin, “An introduction to natural language processing, computational linguistics, and speech recognition”, USA: Prentice-Hall Inc, 2000.

[58] Harjinder Kaur, and Vijay Laxmi, “A Survey of Mt Approaches”, 2013.

[59] Jaganadh G, “Man to Machine: A tutorial on the art of Machine Translation”, 2010, http://www.slideshare.net/jaganadhg/a-tutorialon-machine-translation.

[60] Antony P. J., “Machine Translation Approaches and Survey for Indian Languages”, 2013, http://www.aclclp.org.tw/clclp/v18n1/v18n1a3.

[61] M. D. Okpor, "Machine Translation Approaches: Issues and Challenges," IJCSI International Journal of Computer Science Issues, Vol. 11, No. 5, pp. 159-165, 2014.

[62] Leon Bottou, Yoshua Bengio, and Yann Le Cun. “Global training of document processing systems using graph transformer networks”. In Computer Vision and Pattern Recognition, Proceedings, IEEE Computer Society Conference on, pp. 489-494, 1997.

[63] Alex Graves, Marcus Liwicki, Santiago Fernandez, Roman Bertolami, Horst Bunke, ´ & Jurgen Schmidhuber. 2009. “A novel connectionist system for unconstrained handwriting recognition”. Pattern Analysis and Machine Intelligence, IEEE Transactions.pp:855–868.

[64] Alex Graves, Santiago Fernandez, and Jurgen Schmidhuber. “Connectionist temporal classification: labelling unsegmented sequence data with RNNs”. In Proceedings of the 23rd international conference on Machine Learning, pp. 369–376, 2006.

[65] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. “Attention-based models for speech recognition”. In Advances in Neural Information Processing Systems, pp. 577–585, 2015.

[66] Yoshua Bengio and Rejean Ducharme, Pascal Vincent, and Christian Janvin. “A neural probabilistic language model”. Pp. 1137–1155, 2003.

[67] Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. “Dynamic pooling and unfolding recursive autoencoders for paraphrase detection”. In Advances in Neural Information Processing Systems,. 2011.

[68] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. “Distributed representations of words and phrases and their compositionality”. In Advances in Neural Information Processing Systems 26, pp. 3111–3119, 2013.

[69] A. Waibel, A. N. Jain, A. E. McNair, H. Saito, A.G. Hauptmann, and J. Tebelskis. Janus: "A speech-to-speech translation system using connectionist and symbolic processing strategies". In Proceedings of the 1991 International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 793–796, 1991.

[70] Mikel L Forcada and Ramón P Ñeco. "Recursive hetero-associative memories for translation". In Biological and Artificial Computation: From Neuroscience to Tech., Springer, pp 453–462, 1997.

[71] M. Asunción Castaño, Francisco Casacuberta, and Enrique Vidal. "MT using NNs and finite-state models". pp. 160–167, 1997.

[72] Holger Schwenk. "Continuous space LMs". Computer Speech and Language 3(21):492 518. https://wiki.inf.ed.ac.uk/twiki/pub/CSTR/ListenSemester2_2009_10/sdarticle.pdf, 2007.

[73] Holger Schwenk. "Continuous space translation models for phrase-based SMT". In Proceedings of COLING 2012: Posters. The COLING 2012 Organizing Committee, Mumbai, India, pp. 1071–1080, 2012.

[74] Shixiang Lu, Zhenbiao Chen, and Bo Xu. "Learning new semi-supervised deep auto encoder features for SMT". In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Baltimore, Maryland, pp. 122–132, 2014.

[75] Shin Kanouchi, Katsuhito Sudoh, and Mamoru Komachi. "Neural reordering model considering phrase translation and word alignment for phrase-based translation". In Proceedings of the 3rd Workshop on Asian Translation. The COLING 2016 Organizing Committee, Osaka, Japan, pp 94–103, 2016.

[76] Peng Li, Yang Liu, Maosong Sun, Tatsuya Izuha, and Dakun Zhang. "A neural reordering model for phrase-based translation". In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pp 1897-1907, 2014.

[77] Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. "Fast and accurate preordering for SMT using neural networks". In Proc of the 2015 Conference of the North American Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Denver, Colorado, pp 1012–1017, 2015.

[78] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. "Fast and robust neural network joint models for statistical machine translation". In

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Baltimore, Maryland, pp. 1370–1380, 2014.

[79] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. "On the properties of NMT: Encoder–decoder approaches". Association for Computational Linguistics, Doha, Qatar, pp. 103–111. 2014.

[80] Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. "NMT systems for wmt. In ˇ Proc of the 10th Workshop on SMT. Association for Computational Linguistics, Portugal, 2015.

[81] Himanshu Choudhary, Aditya Kumar Pathak, Rajiv Ratn Shah, and Ponnurangam Kumaraguru, "Neural Machine Translation for English-Tamil" 2019.

[82] Yukio Matsumura, Takayuki Sato, and Mamoru Komachi, "English-Japanese Neural Machine Translation with Encoder-Decoder-Reconstructor" Tokyo, Japan, 2018.

[83] Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li." Neural Machine Translation with Reconstruction". Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI), pp. 3097–3103, 2017.

[84] Mourad Brour, and Abderrahim Benabbou, "Arabic text language into Arabic sign language neural machine translation", Journal of King Saud University, 2019.

[85] Luong M., Sutskever I., and Zaremba W. "Addressing the rare word problem in NMT". In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on NLP, 2015.

[86] Buck C., Heafield K., and Van Ooyen B. "N-gram counts and language models from the common crawl". In LREC, Vol 2, pp. 4, 2014.

[87] H. Schwenk. http://www-lium.univlemans.fr/˜schwenk/cslm_joint_paper/ 2014. last acc: Jan 4 2020].

[88] Kate Reyes. "What is Deep Learning and How Does Deep Learning Work?" simplilearn.com https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deep learning. Last acc: Aug 12 2020.

[89] Utkarsh Dixit, "Neural Machine translation and the need for Attention Mechanism" medium.com https://medium.com/analytics-vidhya/neural-machine-translation-and-the-need-for-attention-mechanism-60f9a39da9a. Last acc: Aug 2020.

[90] Jason Brownlee. “Encoder-Decoder Recurrent Neural Network Models for Neural Machine Translation”. https://machinelearningmastery.com/Encoder-Decoder-recurrent-neural-network-models-neural-machine-translation/. Last acc: Aug 2020.

[91] Jason Brownlee. “Encoder-Decoder Long Short-Term Memory Networks”. https://machine learningmastery.com/Encoder-Decoder-long-short-term-memory-networks/ [acc Aug 2020].

[92] Adrian Rosebrock, “Keras vs. TensorFlow – Which one is better and which one should I learn?” https://www.pyimagesearch.com/2018/10/08/keras-vs-tensorflow-which-one-is-better-and-which-one-should-i-learn/. Last acc: Aug 2020.

[93] Ronald Rosenfeld. “Two decades of statistical language modelling: Where do we go from here?”, Volume 88, pp. 1270–1278. 2000.

[94] Yoav Goldberg and Bar Ilan. “Neural Network Methods in Natural Language Processing [lecture notes]. Available: https://www.morganclaypoolpublishers.com/catalog_Orig/ samples/9781627052955_sample.pdf.

[95] Andreas Stolcke. “SRILM – an extensible language modelling toolkit”. In ICSLP, 2002.

[96] Yee Teh. “A hierarchical Bayesian language model based on Pitman-Yor processes”. 2006.

[97] Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. “IRSTLM: an open source toolkit for handling large scale language models”. In Interspeech, 2008.

[98] Kenneth Heafield.” KenLM: faster and smaller language model queries”. In WMT. 2011.

[99] Rafal Jozefowicz, Oriol Vinyals and Mike Schuster. “ Exploring the Limits of Language Modelling”, 2016.

[100] Frederic Morin and Yoshua Bengio. “Hierarchical probabilistic neural network language model”. In AISTATS, 2005.

[101] Andriy Mnih and Geoffrey Hinton.“A scalable hierarchical distributed language model”. 2009.

[102] Andriy Mnih and Yee Whye Teh. “A fast and simple algorithm for training neural probabilistic language models”. In ICML. 2012.

[103] Minh-Thang Luong, Michael Kayser, and Christopher D. Manning. “Deep neural language models for machine translation”. 2015a.

[104] Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. “Joint language and translation modelling with Recurrent Neural Networks”. 2013.

[105] Jacob Devlin, Richard Schwartz, and John Makhoul. “Fast and robust neural network joint models for statistical machine translation”. 2014.

[106] Yoon Kim, Yacine Jernite, and Alexander M. Rush. "Character-Aware Neural Language Model, 2015.

[107] T. Pawar, "Language modeling using Recurrent Neural Networks Part - 1" medium.com, https://medium.com/praemineo/language-modelling-using-recurrent-neural-networks part-1-427b165576c2. [Accessed Aug. 2, 2020].

[108] Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan Cernock, and Sanjeev Khudanpur.. "Extensions of Recurrent Neural Network language model". 2011.

[109] Tom´aˇs Mikolov and Geoffrey Zweig. "Context dependent Recurrent Neural Network language model". 2012.

[110] H. Jaeger, M. Lukosevicius, D. Popovici and U. Siewert. "Optimization and applications of echo state networks with leaky-integrator neurons", pp 335–352. 2007.

[111] https://hackernoon.com/attention-mechanism

[112] Lecun, Y. Bengio, and G. Hinton, "Deep learning," 2015.

[113] G. Genthial, L. Liu, B. Oshri, and K. Ranjan. Natural Language Processing with Deep Learning [Lecture notes]. Available: https://web.stanford.edu/class/cs224n/ readings/cs224n-2019-notes06-NMT_Seq2Seq_attention.pdf. 2019.

[114] Y. Liu, L. Ji, R. Huang, T. Ming, and C. Gao, "An attention-gated convolutional neural network for sentence classification," pp. 1–19.

[115] https://colab.research.google.com/drive

[116] Tessemma Mindaye Mengistu, "Design and Implementation of Amharic Search Engine" , AAU, Addis Ababa, 2007.

[117] An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/ Last Accessed: June 04 2020.

Appendix I: Sample of parallel corpus

Amharic	**Wolaita**
ነገር ግን ሁሉ በአግባብና በሥርዓት ይሁን።	SHin ubbabaikka wogaaninne maaran hano.
በዓለም ምናልባት ቁጥር የሌለው የቋንቋ ዓይነት ይኖራል ቋንቋም የሌለው ሕዝብ የለም፤	Ha sa7an daro qaalai de7ennan aggenna; qassi birshshetti bainna qaali issoinne baawa.
ምድርም ሁሉ በአንድ ቋንቋና በአንድ ንግግር ነበረች።	kase biittiya hanna issi qaalaninne issi haasayan daasu.
በበሩ የሚገባ ግን የበጎች እረኛ ነው።	SHin penggeera geliyaagee dorssaa heemmiya asa.
አንተማ መልካም ታመሰግናለህ፥ ሌላው ግን አይታነጽበትም።	Neeni Xoossaa lo77o galataasa; shin hegee hinkko bitaniyaa maaddenna.
የኋለኛው ጠላት የሚሻረው ሞት ነው፤	Ubbaappe wurssettan xayana morkkee haiquwaa.
ንቁ፥ በሃይማኖት ቁሙ፥ ጎልምሱ ጠንክሩ።	Tishshi giite. Ammanuwan minnite. Xalite. Minnite.
በእናንተ ዘንድ ሁሉ በፍቅር ይሁን።	Ooso ubbaa siiquwan oottite.
ማንም የማያውቅ ቢኖር ግን አይወቅ።	SHin ooninne akeekana xayikko, akeekoppo, aggi bayo.
እኔ ጳውሎስ ይህን ሰላምታ በገዛ እጄ ጽፌአለሁ።	Taani PHauloosi ha sarotaa ta kushiyan xaafaas.
እንግዲህ እኔን የምትመስሉ ሁኑ ብዬ እለምናችኋለሁ።	Simmi tanadan hanite yaagada inttena woossais.
በዋጋ ተገዝታችኋል፤ የሰው ባሪያዎች አትሁኑ።	Xoossai inttena waagan shammiis. Asa aille gidoppite.
ጌታም አለ። ዓመፀኛው ዳኛ ያለውን ስሙ።	Qassikka Godai, “He makkala daannai giidoogaa siyite.
ይህን ብታውቁ፥ ብታደርጉትም ብፁዓን ናችሁ።	Intte hegaa eridi oottiyaabaa gidikko anjjettidaageeta.
ወደ ቤትም ስትገቡ ሰላምታ ስጡ፤	Qassi soo geliiddi sarotite.
ሁሉንም ተወ፤ ተነሥቶም ተከተለው።	Yaagin Leewi ubbabaa aggi bayidi, denddi eqqidi, a kaalliis.
ከሁሉም በኋላ እንደ ጭንጋፍ ለምሆን ለእኔ ደግሞ ታየኝ።	Qassi ubbaappe wurssettan, wodee gakkennan yelettida taassikka qoncciis.
ወንድሞች ሆይ፥ በአእምሮ ሕፃናት አትሁኑ፤ ለክፋት ነገር ሕፃናት ሁኑ እንጂ በአእምሮ የበሰሉ ሁኑ።	Ta ishatoo, intte iita yohuwan yiira mala gidanaappe attin, qofan na7a mala gidoppite; intte wozanaa qofan wozannaamata gidite.
እንዲያማ ካልሆነ፥ አንተ በመንፈስ ብትባርክ ባልተማሩት ስፍራ የተቀመጠው የምትለውን ካላወቀ እንዴት አድርጎ ለምስጋናህ አሜን ይላል?	Hegaa gidana xayikko, neeni ne ayyaanaa xalaalan Xoossaa galatiyaabaa gidikko, he sohuwan neeni giyoogaa erenna asi neeni Xoossaa galatiyoogaa erennan, neeni galatidoogan woigidi, “Amin77i” gaanee?
በዚች ሕይወት ብቻ ክርስቶስን ተስፋ ያደረግን ከሆነ፥ ከሰው ሁሉ ይልቅ ምስኪኖች ነን።	Nuuni ha de7uwaa xalaalaassi Kiristtoosa yainnidabaa gidikko, ha sa7an de7iya asa ubbaappekka nubai pala.
ፍቅርን ተከታተሉ፥ መንፈሳዊ ስጦታንም ይልቁንም ትንቢት መናገርን በብርቱ ፈልጉ።	Siiquwaa kaallite. Ayyaanaa imuwaakka minttidi koyite. SHin ubbaappe aattidi, hananabaa yootiyoogaakka koyite.
መጽሐፍም እንደሚል በሦስተኛው ቀን ተነሣ፥	Haiqqidi moogettiis. Xoossaa maxaafai giyoogaadankka, heezzantto gallassi haiquwaappe denddiis.
ለኬፋም ታየ በኋላም ለአሥራ ሁለቱ፤	Denddidi, PHeexiroosassi qoncciis. Hegaappe guyyiyan, ba kiittido tammanne naa77atussi qoncciis.

ከዚያም በኋላ ከአምስት መቶ ለሚበዙ ወንድሞች በአንድ ጊዜ ታየ፤ ከእነርሱም የሚበዙቱ እስከ አሁን አሉ አንዳንዶች ግን አንቀላፍተዋል፤	Hegaappe guyyiyan, ichchashu xeetaappe dariya ishatussi issitoo qoncciis. Etappe dariya baggai hanno gakkanaassikka paxa de7ees; shin amaridaageeti haiqqidosona.
ከዚያም በኋላ ለያዕቆብ ኋላም ለሐዋርያት ሁሉ ታየ፤	Hegaappe guyyiyan, Yaaqoobassi qoncciis; guyyeppe ba kiittido ubbatussi qoncciis.
እኔ ከሐዋርያት ሁሉ የማንስ ነኝና፥ የእግዚአብሔርን ቤተ ክርስቲያን ስላሳደድሁ ሐዋርያ ተብዬ ልጠራ የማይገባኝ፤	Aissi giikko, Yesuusi kiittido ubbaappe taani laafa; taani Xoossaa woosa keettaa waissido gishshau, Yesuusi kiittidoogaa geetettada xeesettanaukka bessikke.
ነገር ግን በእግዚአብሔር ጸጋ የሆንሁ እኔ ነኝ፤ ለእኔም የተሰጠኝ ጸጋው ከንቱ አልነበረም ከሁላቸው ይልቅ ግን ደከምሁ፥ ዳሩ ግን ከእኔ ጋር ያለው የእግዚአብሔር ጸጋ ነው እንጂ እኔ አይደለሁም።	SHin Xoossaa aaro kehatettan taani tanakka. Qassi taassi i immido aaro kehatettai hada gidibeenna; shin taani eta ubbaappe aattada oottaas. SHin tanaara de7iya Xoossaa aaro kehatettai oottiisippe attin, tana gidikke.
እንግዲህስ እኔ ብሆን እነርሱም ቢሆኑ እንዲሁ እንሰብካለን እንዲሁም አመናችሁ።	Simmi tana gidikkokka, woikko eta gidikkokka, nuuni ubbai yootiyoogee hagaa; inttekka hagaa ammanideta.
ክርስቶስ ከሙታን እንደ ተነሣ የሚሰበክ ከሆነ ግን ከእናንተ አንዳንዶቹ። ትንሣኤ ሙታን የለም እንዴት ይላሉ?	Kiristtoosi haiquwaappe denddidoogaa yootiyoobaa gidikko, intteppe issi issi asati yaatin woigidi, “Haiqqidabai haiquwaappe denddenna” yaagiyoonaa?
አሁን ግን፥ ወንድሞች ሆይ፥ ወደ እናንተ መጥቼ በልሳኖች ብናገር፥ በመግለጥ ወይም በእውቀት ወይም በትንቢት ወይም በትምህርት ካልነገርኋችሁ ምን እጠቅማችኋለሁ?	SHin ta ishatoo, ha77i taani inttekko baada, dumma dumma qaalan haasayaidda, ajjuutan woi eratettan woi hananabaa yootiyoogan woi tamaarissiyoogan inttessi yootana xayikko, taani inttena ai go77iyaanaa?
ነፍስ የሌለበት ነገር እንኳ ዋሽንትም ክራርም ቢሆን ድምፅ ሲሰጥ የድምፁን ልዩነት ባይገልጥ በዋሽንት የሚነፋው ወይስ በክራር የሚመታው መዝሙር እንዴት ይታወቃል?	Harai atto, shemppoi bainnabai pulaale gidin woikko diitta gidin giiriiddi, ba giirettaa dummatettaa erissana xayikko, pulaale punniyaakkonne, woi diitta diixxiyaakko, asi waatidi eranee?
ደግሞም መለከት የማይገለጥን ድምፅ ቢሰጥ ለጦርነት ማን ይዘጋጃል?	Qassi malkkataa waasoi geeyidi erettana xayikko, olaassi oonee giigettanai?
እንዲሁ እናንተ ደግሞ የተገለጠውን ቃል በአንደበት ባትናገሩ ሰዎች የምትናገሩትን እንዴት አድርገው ያስተውሉታል? ለነፋስ የምትናገሩ ትሆናላችሁና።	Hegaadankka, qassi dumma dumma qaalan intte haasayiyoogee qoncce gidennan xayikko, intte haasayiyoogaa asi waatidi akeekanee? Aissi giikko, carkkuwaassi intte haasayeeta.
እንግዲህ የቋንቋውን ፍች ባላውቅ ለሚናገረው እንግዳ እሆናለሁ፥ የሚናገረውም ለእኔ እንግዳ ይሆናል።	SHin issi asi haasayiyo qaalaa taani erana xayikko, he haasayiyaagee taassi imatta gidees; taanikka assi imatta gidais.
እንዲሁ ደግሞ እናንተ መንፈሳዊ ስጦታን በብርቱ የምትፈልጉ ከሆናችሁ ቤተ ክርስቲያንን ለማነጽ እንዲበዛላችሁ ፈልጉ።	Hegaadankka, qassi Geeshsha Ayyaanaa imuwaa intte ekkanau minttidi koyiyaageeta gidiyo gishshau, woosa keettaa dichchanau maaddiyaabaa ubbabaappe aattidi koyite.
ነገር ግን ሌሎችን ደግሞ አስተምር ዘንድ በማኅበር እልፍ ቃላት በልሳን ከመናገር ይልቅ አምስት ቃላት በአእምሮዬ ልናገር እወዳለሁ።	SHin taani woosa keettan harata tamaarissanau dumma dumma qaalan tammu sha7u qaalata haasayiyoogaappe ichchashu qaalata ta wozanaa qofan haasayanau dosais.
ነቢያትም ሁለት ወይም ሦስት ሆነው ይናገሩ ሌሎችም ይለዩአቸው፤	Hananabaa yootiyaageetikka naa77u gididi woi heezzu gididi haasayona. Qassi harati eti giyoobaa pirddona.

Appendix II: Common Amharic stop words.

በኋላ	እኔ	ታወሰ	ይፋ	ተፈጸመ	እንግዲህ	ወጣ	ምን
ናት	እኛ	ዛሬ	መሰረት	ነገሰ	ጥቃት	ጊዜ	ፈጠረ
ይናገራል	እነሱ	ልዩ	ታወቀ	የሚባሉ	አስተዋወ	ተበረከተ	እርግጥ
ያንጻል	እሱ	ይሆናሉ	አስታወቀ	ንዴት	አስፈጸመ	እንዲያ	ሄደ
እንዲሁ	እሷ	ተናገረ	ተስማማ	አነጋገር	ልብ	ሰነባበተ	ተካሄደ
ግን	አንተ	ተባለ	ወሰነ	ተወሰነ	ተቆጠበ	ላከ	ይሄን
ደግሞ	እናንተ	ርእይ	ጠየቀ	ተነጋገረ	ሆይ	ናደ	አቀፈፈ
ሁሉም	እና	ዋለ	ተጠየቀ	አረጋገጠ	ታመነ	አወሰተ	ንዴት
ይበል	ወይ	አመነ	ተጠናቀ	ሰበር	ዘለለ	አይስተዋ	ተጫወተ
ማንም	በላይ	ተሳተፈ	አገኘ	ጠበቀ	ተመለሰ	ል	አላግባብ
ነገርግን	ወዘተ	ተጋለጸ	ምንም	አሸነፈ	አከዋያ	ተጻፈ	ወይ
ዘንድ	አቶ	ተናገርኸ	ወይንም	ከለከለ	ተቋረጠ	ወዲያው	ጠቄመ
እላለሁ	ተለያየ	ተለመደ	አይደለም	ወሰደ	አስተዋወ	ን	ተጠቄመ
ነው	እስከ	ቀረ	ያህል	ረገድ	ተለየ	ተሰማ	ጠቀሰ
ፊት	ግን	ተደረገ	ነኝ	ተጫወተ	አነሰ	ኤፍቢሲ	ተካተተ
አለ	መጣ	ተጠቀመ	ተጠቃ	አሁን	ነውና	ነን	ውስጥ
ሆነ	አመጣ	ተያያዘ	ተሻለ	አሳሰበ	አኳኋን	ፖሊሲ	ማረተ
በ-ሆላ	ወይም	ለውጥ	ደረገ	ቀጠለ	ወጥ	ተገቢ	ጀመረ
ስለ	እንደ	ተቆጠረ	ተረጋገጠ	ተመረጠ	ምክንያት	ራስ	አስፈጻ
እንጂ	ሁኔታው	አደረ	ሁልጊዜ	ሀላፊነት	አሳሰበ	ገባ	ሚ
ደገመ	የት	በአል	ተናግሮብ	ወዲ	ተደነገገ	አጋለጠ	ደነገገ
ከ	መቼ	ተነገረ	ታየ	ምንድነ	በአል	ጨመረ	ነች
በ	ኋላ	ተፈጠረ	ሞከረ	ምላሽ	ቻለ	ቀደም	እንኳን
ናቸው	ሁኔታ	እነ	ተረዳ	ራስ	ተቻለ	ሳልቫ	አወጣ
ጥቂት	ሁል	ተገመተ	መጠን	ተለየ	አስቻለ	ተደመጠ	ወይስ
በርካታ	ቢቢሲ	ለም	ቅርብ	ቆየ	ሰራ	አካል	እንግዲህ
ብቻ	ብዛት	ዋደደ	እዚህ	ተወገደ	እንኳ	ምንድን	ወዲያ
ሌላ	ቦታ	ነገረ	ነህ	አጠቃላ	ናችሁ	ያሉት	ይልዋል
ሁሉ	በጣም	ተወሰደ	ተከላከለ	ይ	ቀረበ	አስመዘገ	ዳረገ
አንዳንድ	በተለይ	ተከሰተ	አነሳ	ወደ	አቀረበ	ተቀመጠ	ፈለገ
ማን	ተመለከ	ጠቀመ	አስቸኳይ	ወደቀ	ተናጋረ	እንዴት	አን
ባክዎ	ተመሳሰለ	አሳየ	ደረሰ	አስቸገረ	ጭምር	በለጠ	በኩል
ባክ	ተገልጹል	ወጣ	ዘነጋ	ጠቀሰ	ቴአትር	አልያ	ከበደ
ተጨማሪ	ችግር	ገናማ	አስከተለ	አስቀመ	አስመለከ	ተጠቀሰ	ሙሉ
ሰአት	አስታወሰ	ድረግ	መልክ	ጠ	ድረ	ሄዱከ	ደላ
ዉጪ	አሳሳበ	ይሄ	ፈጸመ	ድጋሚ	አደረገ	ግልጽ	ፓርቲ
ናት	ስፈላጊ	ይህ	ያዘ	ተከናወነ	ዋነኛ	ተደመጠ	ደነገገ
ያ	አስገነዘበ	ይህንን	አመጥ	ኢጋድ	አንድ	ተጠረጠ	መከረ
ወይዘሮ	አበራራ	ይኸ	እነሆ	ገለጸ	አስወገደ	ርእሰ	ፌስ
ወይዘሪት	አስረዳ	ስነ	ተከራከረ	ተገለጸ	ተሰጠ	አወቀ	ተገቢ
ታች	አንጻር	ተናገር	ዘንድ	ይልቅ	ሰጠ	ቶሎ	ቃለ
ከተተ	እንኳ	አመለከተ	እንዲ	አከናወነ	ፈለገ	ላይ	አቃጠለ
መካከል	ገና	እጅግ	ውድ	አከለ	ከፊል	ነበር	ጠፋ
ሰሞን	ወቅት	ከዛ	ላቀ	አካተተ	ማንኛው	ጠራ	አጠናቀቀ
ትናንት	ዋና	ተጠበቀ	ተገኘ	ተደጋጋ	ኖረ	አስቄጠረ	ተነሳ
ትናንትና	ወጭ	ተከተለ	ተራዘመ	ተመከረ	መካነ	ወጣ	ጋር
ሆነ	ጋራ	ቀነሰ	አይ	አመመ	ተጠቀሰ	አገለገለ	መሰለ

Appendix III: Each epoch's loss level and time taken for training the system.

Epoch 1 Loss 2.264173 Time 153.21986 sec
Epoch 2 Loss 2.111294 Time 90.3907 sec
Epoch 3 Loss 2.054009 Time 90.9183 sec
Epoch 4 Loss 1.997920 Time 90.9472 sec
Epoch 5 Loss 1.920717 Time 93.0655 sec
Epoch 6 Loss 1.827197 Time 91.3056 sec
Epoch 7 Loss 1.735980 Time 91.2038 sec
Epoch 8 Loss 1.633517 Time 91.3208 sec
Epoch 9 Loss 1.520991 Time 91.4875 sec
Epoch 10 Loss 1.395341 Time 93.1666 sec
Epoch 11 Loss 1.257370 Time 91.3188 sec
Epoch 12 Loss 1.109244 Time 91.3559 sec
Epoch 13 Loss 0.954131 Time 91.0642 sec
Epoch 14 Loss 0.803845 Time 91.2477 sec
Epoch 15 Loss 0.668424 Time 92.7495 sec
Epoch 16 Loss 0.555440 Time 91.3822 sec
Epoch 17 Loss 0.462380 Time 91.2049 sec
Epoch 18 Loss 0.383793 Time 91.1040 sec
Epoch 19 Loss 0.316827 Time 91.0404 sec
Epoch 20 Loss 0.260945 Time 92.8468 sec
Epoch 21 Loss 0.213393 Time 91.2266 sec
Epoch 22 Loss 0.171829 Time 90.9922 sec
Epoch 23 Loss 0.136427 Time 91.1438 sec
Epoch 24 Loss 0.108045 Time 91.2057 sec
Epoch 25 Loss 0.085063 Time 93.0783 sec
Epoch 26 Loss 0.067203 Time 91.1177 sec
Epoch 27 Loss 0.051981 Time 91.0183 sec
Epoch 28 Loss 0.041225 Time 90.9578 sec
Epoch 29 Loss 0.033504 Time 91.0428 sec
Epoch 30 Loss 0.028677 Time 92.7335 sec
Epoch 31 Loss 0.024998 Time 91.2618 sec
Epoch 32 Loss 0.021703 Time 90.8723 sec
Epoch 33 Loss 0.019390 Time 91.0475 sec
Epoch 34 Loss 0.017768 Time 91.0090 sec
Epoch 35 Loss 0.016498 Time 92.5059 sec
Epoch 36 Loss 0.015730 Time 90.9708 sec
Epoch 37 Loss 0.014732 Time 91.0016 sec
Epoch 38 Loss 0.014117 Time 91.0047 sec
Epoch 39 Loss 0.014345 Time 90.9625 sec
Epoch 40 Loss 0.014517 Time 92.8075 sec
Epoch 41 Loss 0.014911 Time 91.0141 sec
Epoch 42 Loss 0.015411 Time 90.8811 sec
Epoch 43 Loss 0.016670 Time 90.8737 sec
Epoch 44 Loss 0.016608 Time 91.0245 sec
Epoch 45 Loss 0.016925 Time 92.8748 sec
Epoch 46 Loss 0.017245 Time 91.0994 sec
Epoch 47 Loss 0.017937 Time 91.0599 sec
Epoch 48 Loss 0.017741 Time 91.0709 sec
Epoch 49 Loss 0.016927 Time 90.8503 sec
Epoch 50 Loss 0.016383 Time 92.7557 sec

Appendix IV: The last epoch results with loss level and a time to taken for 116 batches.

Epoch 50 Batch 0 Loss 0.013544
Epoch 50 Batch 1 Loss 0.017937
Epoch 50 Batch 2 Loss 0.025752
Epoch 50 Batch 3 Loss 0.009433
Epoch 50 Batch 4 Loss 0.017541
Epoch 50 Batch 5 Loss 0.015115
Epoch 50 Batch 6 Loss 0.010091
Epoch 50 Batch 7 Loss 0.027269
Epoch 50 Batch 8 Loss 0.011607
Epoch 50 Batch 9 Loss 0.011041
Epoch 50 Batch 10 Loss 0.014244
Epoch 50 Batch 11 Loss 0.015845
Epoch 50 Batch 12 Loss 0.021646
Epoch 50 Batch 13 Loss 0.023589
Epoch 50 Batch 14 Loss 0.010957
Epoch 50 Batch 15 Loss 0.021193
Epoch 50 Batch 16 Loss 0.023150
Epoch 50 Batch 17 Loss 0.022084
Epoch 50 Batch 18 Loss 0.014431
Epoch 50 Batch 19 Loss 0.013112
Epoch 50 Batch 20 Loss 0.021191
Epoch 50 Batch 21 Loss 0.012547
Epoch 50 Batch 22 Loss 0.025012
Epoch 50 Batch 23 Loss 0.016545
Epoch 50 Batch 24 Loss 0.013147
Epoch 50 Batch 25 Loss 0.010812
Epoch 50 Batch 26 Loss 0.014996
Epoch 50 Batch 27 Loss 0.015847
Epoch 50 Batch 28 Loss 0.012309
Epoch 50 Batch 29 Loss 0.022938
Epoch 50 Batch 30 Loss 0.023369
Epoch 50 Batch 31 Loss 0.017606
Epoch 50 Batch 32 Loss 0.017034
Epoch 50 Batch 33 Loss 0.020166
Epoch 50 Batch 34 Loss 0.022465
Epoch 50 Batch 35 Loss 0.015764
Epoch 50 Batch 36 Loss 0.013880
Epoch 50 Batch 37 Loss 0.014561
Epoch 50 Batch 38 Loss 0.011305
Epoch 50 Batch 39 Loss 0.015690
Epoch 50 Batch 40 Loss 0.018793
Epoch 50 Batch 41 Loss 0.015226
Epoch 50 Batch 42 Loss 0.019245
Epoch 50 Batch 43 Loss 0.013860
Epoch 50 Batch 44 Loss 0.015987
Epoch 50 Batch 45 Loss 0.010803
Epoch 50 Batch 46 Loss 0.016218
Epoch 50 Batch 47 Loss 0.009233
Epoch 50 Batch 48 Loss 0.022518
Epoch 50 Batch 49 Loss 0.017496
Epoch 50 Batch 50 Loss 0.010595
Epoch 50 Batch 51 Loss 0.013958
Epoch 50 Batch 52 Loss 0.017162
Epoch 50 Batch 53 Loss 0.017755
Epoch 50 Batch 54 Loss 0.023228
Epoch 50 Batch 55 Loss 0.012986

Epoch 50 Batch 56 Loss 0.011647
Epoch 50 Batch 57 Loss 0.009399
Epoch 50 Batch 58 Loss 0.014679
Epoch 50 Batch 59 Loss 0.021417
Epoch 50 Batch 60 Loss 0.018727
Epoch 50 Batch 61 Loss 0.025010
Epoch 50 Batch 62 Loss 0.011620
Epoch 50 Batch 63 Loss 0.025125
Epoch 50 Batch 64 Loss 0.010695
Epoch 50 Batch 65 Loss 0.011838
Epoch 50 Batch 66 Loss 0.019401
Epoch 50 Batch 67 Loss 0.013548
Epoch 50 Batch 68 Loss 0.021070
Epoch 50 Batch 69 Loss 0.011381
Epoch 50 Batch 70 Loss 0.013061
Epoch 50 Batch 71 Loss 0.019269
Epoch 50 Batch 72 Loss 0.013836
Epoch 50 Batch 73 Loss 0.013661
Epoch 50 Batch 74 Loss 0.016355
Epoch 50 Batch 75 Loss 0.010690
Epoch 50 Batch 76 Loss 0.013935
Epoch 50 Batch 77 Loss 0.011561
Epoch 50 Batch 78 Loss 0.017078
Epoch 50 Batch 79 Loss 0.022206
Epoch 50 Batch 80 Loss 0.016161
Epoch 50 Batch 81 Loss 0.013388
Epoch 50 Batch 82 Loss 0.017339
Epoch 50 Batch 83 Loss 0.012918
Epoch 50 Batch 84 Loss 0.012886
Epoch 50 Batch 85 Loss 0.015998
Epoch 50 Batch 86 Loss 0.010652
Epoch 50 Batch 87 Loss 0.015663
Epoch 50 Batch 88 Loss 0.019641
Epoch 50 Batch 89 Loss 0.020596
Epoch 50 Batch 90 Loss 0.022625
Epoch 50 Batch 91 Loss 0.012869
Epoch 50 Batch 92 Loss 0.022783
Epoch 50 Batch 93 Loss 0.017098
Epoch 50 Batch 94 Loss 0.011240
Epoch 50 Batch 95 Loss 0.026242
Epoch 50 Batch 96 Loss 0.016545
Epoch 50 Batch 97 Loss 0.017733
Epoch 50 Batch 98 Loss 0.013144
Epoch 50 Batch 99 Loss 0.017378
Epoch 50 Batch 100 Loss 0.011456
Epoch 50 Batch 101 Loss 0.015737
Epoch 50 Batch 102 Loss 0.025300
Epoch 50 Batch 103 Loss 0.014221
Epoch 50 Batch 104 Loss 0.020778
Epoch 50 Batch 105 Loss 0.022121
Epoch 50 Batch 106 Loss 0.018971
Epoch 50 Batch 107 Loss 0.013128
Epoch 50 Batch 108 Loss 0.013268
Epoch 50 Batch 109 Loss 0.010633
Epoch 50 Batch 110 Loss 0.016226
Epoch 50 Batch 111 Loss 0.014788
Epoch 50 Batch 112 Loss 0.015013
Epoch 50 Batch 113 Loss 0.015009
Epoch 50 Batch 114 Loss 0.014251
Epoch 50 Batch 115 Loss 0.017589
Epoch 50 Loss 0.016383
Time 92.7558 sec

Appendix V: Sample output

Translating the same sentence	Input	ነገር ግን ሁሉ በአግባብና በሥርዓት ይሁን
	Reference	SHin ubbabaikka wogaaninne maaran hano.
A2W NMT	Translation	shin ubbabaikka wogaaninne maaran hano.
	BLEU score	0.665076
Attention-based A2W NMT	Translation	shin ubbabaikka wogaaninne maaran hano.
	BLEU score	0.665076
Translating the same sentence	Input	በዓለም ምናልባት ቁጥር የሌለው የቋንቋ ዓይነት ይኖራል ቋንቋም የሌለው ሕዝብ የለም፤'
	Reference	Ha sa7an daro qaalai de7ennan aggenna; qassi birshshetti bainna qaali issoinne baawa.
A2W NMT	Translation	ha sa7an daro qaalai de7ennan aggenna qassi birshshetti bainna qaali issoinne baawa.
	BLEU score	0.578965
Attention-based A2W NMT	Translation	ha sa7an daro qaalai de7ennan aggenna qassi birshshetti bainna qaali issoinne baawa.
	BLEU score	0.575757
Translating the same sentence	Input	ምድርም ሁሉ በአንድ ቋንቋና በአንድ ንግግር ነበረች።
	Reference	kase biittiya hanna issi qaalaninne issi haasayan daasu.
A2W NMT	Translation	kase biittiya hanna issi qaalaninne issi haasayan daasu.
	BLEU score	0.614788
Attention-based A2W NMT	Translation	kase biittiya hanna issi qaalaninne issi haasayan daasu.
	BLEU score	0.614788
Translating the same sentence	Input	በበሩ የሚገባ ግን የበጎች እረኛ ነው።
	Reference	SHin penggeera geliyaagee dorssaa heemmiya asa.
A2W NMT	Translation	shin penggeera geliyaagee dorssaa heemmiya asa.
	BLEU score	0.638943
Attention-based A2W NMT	Translation	shin penggeera geliyaagee dorssaa heemmiya asa.
	BLEU score	0.638943
Translating the same sentence	Input	አንተማ መልካም ታመሰግናለህ፥ ሌላው ግን አይታነጽበትም።
	Reference	Neeni Xoossaa lo77o galataasa; shin hegee hinkko bitaniyaa maaddenna.
A2W NMT	Translation	aissi giikko, eti ainne hanennan de7iiddi, he urai issi asati a bolli misimaariyan xishettidi i haa yeddido gishshau, a bolli pirddanau yiidoogaappe attin, issi asati a bolli misimaariyan xishettidi i haa yeddido gishshau, a bolli pirddanau yiidoogaappe attin, issi asati a bolli
	BLEU score	0.379671
Attention-based A2W NMT	Translation	neeni xoossaa lo77o galataasa shin hegee hinkko bitaniyaa maaddenna.
	BLEU score	0.588566
Translating the same sentence	Input	የኋለኛው ጠላት የሚሻረው ሞት ነው፤
	Reference	Ubbaappe wurssettan xayana morkkee haiquwaa.
A2W NMT	Translation	ubbaappe wurssettan xayana morkkee haiquwaa.

	BLEU score	0.665438
Attention-based A2W NMT	Translation	ubbaappe wurssettan xayana morkkee haiquwaa.
	BLEU score	0.665438
Translating the same sentence	Input	ንቁ፥ በሃይማኖት ቁሙ፥ ጎልምሱ ጠንክሩ።
	Reference	Tishshi giite. Ammanuwan minnite. Xalite. Minnite.
A2W NMT	Translation	tishshi giite. ammanuwan minnite.
	BLEU score	0.688725
Attention-based A2W NMT	Translation	tishshi giite. ammanuwan minnite. xalite. minnite.
	BLEU score	0.630365
Translating the same sentence	Input	በእናንተ ዘንድ ሁሉ በፍቅር ይሁን።
	Reference	Ooso ubbaa siiquwan oottite.
A2W NMT	Translation	ooso ubbaa siiquwan oottite.
	BLEU score	0.712104
Attention-based A2W NMT	Translation	ooso ubbaa siiquwan oottite.
	BLEU score	0.712104
Translating the same sentence	Input	ማንም የማያውቅ ቢኖር ግን አይወቅ።'
	Reference	SHin ooninne akeekana xayikko, akeekoppo, aggi bayo.
A2W NMT	Translation	shin ooninne akeekana xayikko, akeekoppo, aggi bayo.
	BLEU score	0.606819
Attention-based A2W NMT	Translation	shin ooninne akeekana xayikko, akeekoppo, aggi bayo.
	BLEU score	0.606819
Translating the same sentence	Input	እኔ ጳውሎስ ይህን ሰላምታ በገዛ እጄ ጽፌአለሁ።
	Reference	Taani PHauloosi ha sarotaa ta kushiyan xaafaas.
A2W NMT	Translation	taani phauloosi ha sarotaa ta kushiyan xaafaas.
	BLEU score	0.655997
Attention-based A2W NMT	Translation	taani phauloosi ha sarotaa ta kushiyan xaafaas.
	BLEU score	0.655997
Translating the same sentence	Input	እንግዲህ እኔን የምትመስሉ ሁኑ ብዬ እለምናችኋለሁ።
	Reference	Simmi tanadan hanite yaagada inttena woossais.
A2W NMT	Translation	taani intteyyo yootidoogee tuma intte nagaran haiqqanaagaa intteyyo yootidoogee tuma intte nagaran haiqqanaagaa intteyyo yootidoogee tuma intte nagaran haiqqanaagaa intteyyo yootidoogee tuma intte nagaran haiqqanaagaa intteyyo yootidoogee
	BLEU score	0.343295
Attention-based A2W NMT	Translation	simmi tanadan hanite yaagada inttena woossais.
	BLEU score	0.641936
Translating the same sentence	Input	በዋጋ ተገዝታችኋል፤ የሰው ባሪያዎች አትሁኑ።
	Reference	Xoossai inttena waagan shammiis. Asa aille gidoppite.
A2W NMT	Translation	xoossai inttena waagan shammiis. asa aille gidoppite.
	BLEU score	0.622333
Attention-based A2W NMT	Translation	xoossai inttena waagan shammiis. asa aille gidoppite.
	BLEU score	0.622333
	Input	ጌታም አለ። ዓመፀኛው ዳኛ ያለውን ስሙ።

Translating the same sentence	Reference	Qassikka Godai, “He makkala daannai giidoogaa siyite.
A2W NMT	Translation	qassikka godai, he makkala daannai giidoogaa siyite.
	BLEU score	0.624953
Attention-based A2W NMT	Translation	qassikka godai, he makkala daannai giidoogaa siyite.
	BLEU score	0.624953
Translating the same sentence	Input	ይህን ብታውቁ፥ ብታደርጉትም ብፁዓን ናችሁ።
	Reference	Intte hegaa eridi oottiyaabaa gidikko anjjettidaageeta.
A2W NMT	Translation	intte hegaa eridi oottiyaabaa gidikko anjjettidaageeta.
	BLEU score	0.617252
Attention-based A2W NMT	Translation	intte hegaa eridi oottiyaabaa gidikko anjjettidaageeta.
	BLEU score	0.617252
Translating the same sentence	Input	ወደ ቤትም ስትገቡ ሰላምታ ስጡ፤
	Reference	Qassi soo geliiddi sarotite.
A2W NMT	Translation	qassi soo geliiddi sarotite.
	BLEU score	0.731111
Attention-based A2W NMT	Translation	qassi soo geliiddi sarotite.
	BLEU score	0.731111
Translating the same sentence	Input	ሁሉንም ተወ፤ ተነሥቶም ተከተለው።
	Reference	Yaagin Leewi ubbabaa aggi bayidi, denddi eqqidi, a kaalliis.
A2W NMT	Translation	yaagin leewi ubbabaa aggi bayidi, denddi eqqidi, a kaalliis.
	BLEU score	0.587832
Attention-based A2W NMT	Translation	yaagin leewi ubbabaa aggi bayidi, denddi eqqidi, a kaalliis.
	BLEU score	0.587832
Translating the same sentence	Input	ከሁሉም በኋላ እንደ ጭንጋፍ ለምሆን ለእኔ ደግሞ ታየኝ።
	Reference	Qassi ubbaappe wurssettan, wodee gakkennan yelettida taassikka qoncciis.
A2W NMT	Translation	qassi ubbaappe wurssettan, wodee gakkennan yelettida taassikka qoncciis.
	BLEU score	0.596476
Attention-based A2W NMT	Translation	qassi ubbaappe wurssettan, wodee gakkennan yelettida taassikka qoncciis.
	BLEU score	0.596476
Translating the same sentence	Input	ወንድሞች ሆይ በአእምሮ ሕፃናት አትሁኑ ለክፋት ነገር ሕፃናት ሁኑ እንጂ በአእምሮ የበሰሉ ሁኑ'
	Reference	Ta ishatoo, intte iita yohuwan yiira mala gidanaappe attin, qofan na7a mala gidoppite; intte wozanaa qofan wozannaamata gidite.
A2W NMT	Translation	ta ishatoo intte iita yohuwan yiira mala gidanaappe attin, qofan naa mala gidoppite intte wozanaa qofan wozannaamata gidite
	BLEU score	0.524634
Attention-based A2W NMT	Translation	ta ishatoo intte iita yohuwan yiira mala gidanaappe attin, qofan na7a mala gidoppite intte wozanaa qofan wozannaamata gidite.
	BLEU score	0.523645

Translating the same sentence	Input	እንዲያማ ካልሆነ፥ አንተ በመንፈስ ብትባርክ ባልተማሩት ስፍራ የተቀመጠው የምትለውን ካላወቀ እንዴት አድርጎ ለምስጋናህ አሜን ይላል?
	Reference	Hegaa gidana xayikko, neeni ne ayyaanaa xalaalan Xoossaa galatiyaabaa gidikko, he sohuwan neeni giyoogaa erenna asi neeni Xoossaa galatiyoogaa erennan, neeni galatidoogan woigidi, “Amin77i” gaanee?
A2W NMT	Translation	aissi giikko, xoossaa maxaafai attin, cashsha asa woi asaa, ai oottiyoonaa? haiqqidaageeti a kiitanchchaa gelidi, ubbai xoossaa kiitanchchaa gelidi, ubbai xoossaa kiitanchchaa gelidi, ubbai xoossaa kiitanchchaa gelidi, ubbai xoossaa kiitanchchaa gelidi, ubbai xoossaa kiitanchchaa gelidi, ubbai xoossaa
	BLEU score	0.347871
Attention-based A2W NMT	Translation	hegaa gidana xayikko, neeni galatidoogan woigidi, amin77i gaanee? <end>
	BLEU score	0.577351
Translating the same sentence	Input	በዚች ሕይወት ብቻ ክርስቶስን ተስፋ ያደረግን ከሆነ፥ ከሰው ሁሉ ይልቅ ምስኪኖች ነን።
	Reference	Nuuni ha de7uwaa xalaalaassi Kiristtoosa yainnidabaa gidikko, ha sa7an de7iya asa ubbaappekka nubai pala.
A2W NMT	Translation	nuuni kiristtoosaara nuuyyo ashshiya xoossai koyiyoogaadan gidenna.
	BLEU score	0.566641
Attention-based A2W NMT	Translation	Nuuni ha de7uwaa xalaalaassi Kiristtoosa yainnidabaa gidikko, ha sa7an de7iya asa ubbaappekka nubai pala.
	BLEU score	0.606306
Translating the same sentence	Input	ፍቅርን ተከታተሉ፥ መንፈሳዊ ስጦታንም ይልቁንም ትንቢት መናገርን በብርቱ ፈልጉ።'
	Reference	Siiquwaa kaallite. Ayyaanaa imuwaakka minttidi koyite. SHin ubbaappe aattidi, hananabaa yootiyoogaakka koyite.
A2W NMT	Translation	xoossaa sunttaa gishshau, intte huuphiyan intte godaa yesuus kiristtoosa sunttaa gishshau, intte huuphiyan intte godaa yesuus kiristtoosa sunttaa gishshau, intte huuphiyan intte godaa yesuus kiristtoosa sunttaa gishshau, intte huuphiyan intte godaa yesuus kiristtoosa
	BLEU score	0.362558
Attention-based A2W NMT	Translation	siiquwaa kaallite. ayyaanaa imuwaakka minttidi koyite. shin ubbaappe aattidi, hananabaa yootiyoogaakka koyite.
	BLEU score	0.526641
Translating the same sentence	Input	መጽሐፍም እንደሚል በሦስተኛው ቀን ተነሣ፥
	Reference	Haiqqidi moogettiis. Xoossaa maxaafai giyoogaadankka, heezzantto gallassi haiquwaappe denddiis.
A2W NMT	Translation	haiqqidi moogettiis. xoossaa maxaafai giyoogaadankka, heezzantto gallassi haiquwaappe denddiis.
	BLEU score	0.545018

Attention-based A2W NMT	Translation	haiqqidi moogettiis. xoossaa maxaafai giyoogaadankka, heezzantto gallassi haiquwaappe denddiis.
	BLEU score	0.545018
Translating the same sentence	Input	ከዚያም በኋላ ከአምስት መቶ ለሚበዙ ወንድሞች በአንድ ጊዜ ታየ፤ ከእነርሱም የሚበዙቱ እስከ አሁን አሉ አንዳንዶች ግን አንቀላፍተዋል፤
	Reference	Hegaappe guyyiyan, ichchashu xeetaappe dariya ishatussi issitoo qoncciis. Etappe dariya baggai hanno gakkanaassikka paxa de7ees; shin amaridaageeti haiqqidosona.
A2W NMT	Translation	hegaappe guyyiyan, ichchashu xeetaappe dariya ishatussi issitoo qoncciis. etappe dariya baggai hanno gakkanaassikka paxa deees shin amaridaageeti haiqqidosona.
	BLEU score	0.495419
Attention-based A2W NMT	Translation	hegaappe guyyiyan, ichchashu xeetaappe dariya ishatussi issitoo qoncciis. etappe dariya baggai hanno gakkanaassikka paxa de7ees shin amaridaageeti haiqqidosona.
	BLEU score	0.494676
Translating the same sentence	Input	ከዚያም በኋላ ለያዕቆብ ኋላም ለሐዋርያት ሁሉ ታየ፤
	Reference	Hegaappe guyyiyan, Yaaqoobassi qoncciis; guyyeppe ba kiittido ubbatussi qoncciis.
A2W NMT	Translation	hegaappe guyyiyan, yaaqoobassi qoncciis guyyeppe ba kiittido ubbatussi qoncciis.
	BLEU score	0.567128
Attention-based A2W NMT	Translation	hegaappe guyyiyan, yaaqoobassi qoncciis guyyeppe ba kiittido ubbatussi qoncciis.
	BLEU score	0.567128
Translating the same sentence	Input	ነገር ግን በእግዚአብሔር ጸጋ የሆንሁ እኔ ነኝ፤ ለእኔም የተሰጠኝ ጸጋው ከንቱ አልነበረም ከሁላቸው ይልቅ ግን ደከምሁ፥ ዳሩ ግን ከእኔ ጋር ያለው የእግዚአብሔር ጸጋ ነው እንጂ እኔ አይደለሁም።
	Reference	SHin Xoossaa aaro kehatettan taani tanakka. Qassi taassi i immido aaro kehatettai hada gidibeenna; shin taani eta ubbaappe aattada oottaas. SHin tanaara de7iya Xoossaa aaro kehatettai oottiisippe attin, tana gidikke.
A2W NMT	Translation	shin taani inttenaara de7iyo gishshau, taani ha77i taani ha77i taani ha77i taani ha77i taani ha77i taani ha77i taani ha77i taani ha77i taani ha77i taani ha77i taani ha77i
	BLEU score	0.388549
Attention-based A2W NMT	Translation	shin xoossaa aaro kehatettan taani tanakka. qassi taassi i immido aaro kehatettai oottiisippe attin, tana gidikke. <end>
	BLEU score	0.536172
Translating the same sentence	Input	እንግዲህስ እኔ ብሆን እነርሱም ቢሆኑ እንዲሁ እንሰብካለን እንዲሁም አመናችሁ
	Reference	Simmi tana gidikkokka, woikko eta gidikkokka, nuuni ubbai yootiyoogee hagaa; inttekka hagaa ammanideta.
A2W NMT	Translation	simmi tana gidikkokka woikko eta gidikkokka nuuni ubbai yootiyoogee hagaa ammanideta.
	BLEU score	0.488549

Attention-based A2W NMT	Translation	simmi tana gidikkokka woikko eta gidikkokka nuuni ubbai yootiyoogee hagaa inttekka hagaa ammanideta.
	BLEU score	0.531831
Translating the same sentence	Input	ክርስቶስ ከሙታን እንደ ተነሳ የሚሰበክ ከሆነ ግን ከእናንተ አንዳንዶቹ ትንሳኤ ሙታን የለም እንዴት ይላሉ
	Reference	Kiristtoosi haiquwaappe denddidoogaa yootiyoobaa gidikko, intteppe issi issi asati yaatin woigidi, “Haiqqidabai haiquwaappe denddenna” yaagiyoonaa?
A2W NMT	Translation	nuuni eroos pirddettenna shin taani intteyyo odikke yaagiis.
	BLEU score	0.421257
Attention-based A2W NMT	Translation	kiristtoosi haiquwaappe denddidoogaa yootiyoobaa gidikko intteppe issi issi asati yaatin woigidi haiqqidabai haiquwaappe denddenna yaagiyoonaa
	BLEU score	0.621557
Translating the same sentence	Input	በትህትና ሁሉና በየዋህነት በትእግስትም እርስ በርሳችሁ በፍቅር ታገሱ
	Reference	leemisuwawu, nuuni intteyyo aayye7ana.
A2W NMT	Translation	leemisuwawu nuuni intteyyo aayye7ana.
	BLEU score	0.668791
Attention-based A2W NMT	Translation	leemisuwawu nuuni intteyyo aayye7ana.
	BLEU score	0.668791
Translating the same sentence	Input	አንድ ጌታ አንድ ሃይማኖት አንዲት ጥምቀት
	Reference	Issi Godai, issi ammanoinne issi xinqqatee de7ees.
A2W NMT	Translation	issi godai issi ammanoinne issi xinqqatee de7ees.
	BLEU score	0.788549
Attention-based A2W NMT	Translation	issi godai issi ammanoinne issi xinqqatee de7ees.
	BLEU score	0.788549
Translating the same sentence	Input	ከሁሉ በላይ የሚሆን በሁሉም የሚሠራ በሁሉም የሚኖር አንድ አምላክ የሁሉም አባት አለ።
	Reference	Ubbaa Godai, ubbaa baggaara oottiyaagee, ubbaa giddon de7iyaagee, issi Xoossai, ubbaa Aawai de7ees.
A2W NMT	Translation	ubbaa godai ubbaa giddon de7iyaagee issi xoossai ubbaa aawai de7ees.
	BLEU score	0.468549
Attention-based A2W NMT	Translation	ubbaa godai ubbaa baggaara oottiyaagee ubbaa giddon de7iyaagee issi xoossai ubbaa aawai de7ees.
	BLEU score	0.636172
*Average BLEU score of A2W NMT is **0.5960*** *Average BLEU score of Attention-based A2W NMT is **0.6258***		

Declaration

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

Declared by:

Name: _____Workineh Wogaso Gaga ______________________________________

Signature: ____

Date: _______October 27, 2020 ____

Confirmed by advisor:

Name: _____Yaregal Assabie (PhD) _______________________________________

Signature: ____

Date: ____