2021-05-11

þÿ G E E Z - A M H A R I C M A C H I N E TRANSLATION USING DEEP LEARNING

GENET, WORKU

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF RESEARCH AND POSTGRADUATE STUDIES

SCHOOL OF COMPUTING

GE'EZ-AMHARIC MACHINE TRANSLATION USING DEEP LEARNING

GENET WORKU GEBEYEHU

BAHIR DAR, ETHIOPIA

May 11, 2021

GE'EZ-AMHARIC MACHINE TRANSLATION USING DEEP LEARNING

GENET WORKU GEBEYEHU

A Ge'ez-Amharic Machine Translation Using Deep Learning submitted to the school of Research and Graduate Studies of Bahir Dar Institute of Technology, BDU in partial fulfillment of the requirements for the Degree of Master of Science in the Computer science program in the faculty of computing.

Advisor Name: Dr. ESUBALEW ALEMNEH

Bahir Dar, Ethiopia

DECLARATION

Declaration

This is to certify that the thesis entitled "Ge'ez to Amharic Machine Translation using Deep Learning", submitted in partial fulfillment of the requirements for the degree of Master of Science in (**Computer science**) under **Faculty of Computing**, Bahir Dar Institute of Technology, is a record of original work carried out by me and has never been submitted to this or any other institution to get any other degree or certificates. The assistance and help I received during the course of this investigation have been duly acknowledged.

Genet Worku	_____	11/5/2021
Name of the candidate	signature	Date

May 11, 2021

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTING

Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student Genet Worku Signature _____ Date 5/11/21

As members of the board of examiners, we examined this thesis entitled "Ge'ez to Amharic Machine Translation using Deep Learning" by *Genet Worku.* We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of science in "Computer Science".

Board of Examiners

Name of Advisor	Signature	Date
Dr. Esubalew Alemneh		May 11, 2021
Name of External examiner	Signature	Date
Dr. Wondwossen Mulugeta		may 11, 2021
Name of Internal Examiner	Signature	Date
Dr. Mekonnen Wagaw		May 11, 2021
Name of Chairperson	Signature	Date
Dr. Gebeyehu Belay		may 11, 2021
Name of Chair Holder	Signature	Date
Haileyesus A.		may 11, 2021
Name of Faculty Dean	Signature	Date
Belete B.		may 11, 2021

Faculty Stamp

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my almighty **God** for always being there for me and giving me courage and patience in doing this work. It would not have been to this stage without the help of God.

Secondly, I would like to express my deepest gratitude to my thesis advisor **Dr. Esubalew Alemneh** for contributing his constructive comments, directions, ideas, and courage during the course of my work. Frankly speaking, this thesis would not have been accomplished without his perspective advice and observation.

I would also like to present my heartfelt admiration and gratitude to my husband Mr. Zewdie Habtie who contributed a lot from the beginning to the end of my work. I am also happy to thank my friends for contributing their suggestions and comments during the work of my thesis.

ABSTRACT

Neural machine translation (NMT) which has come to be the breakthrough in the field of machine translation is now greatly being used by many translation services such as google translate. This generic deep learning approach of machine translation (MT) with the help of attention mechanism is used as the core method of our Ge'ez-Amharic translation. Despite their low accuracy, low speed of translation, and involvement of linguistic professionals, other methods of Ge'ez-Amharic translation such as using Statistical Machine Translation (SMT), morpheme-based have already been researched. The complex nature of design the model for these approaches is a limitation. We used a unidirectional model which only translates from Ge'ez to Amharic. We designed an NMT encoder-decoder translation model based on attention mechanism that contains two Long Short-Term Memory (LSTM) layers with 500 hidden units both in the encoder and decoder parts. The model takes source sentence as input in the encoder side and generates a target sentence as output in the decoder side, generating a single word at a time. We used attention mechanism to handle long term dependencies in long sentences by paying attention to the parts of the input sentence which contain relevant information in generating a single word in the target sentence. We have collected 50k Ge'ez-Amharic parallel sentences and used different portions of this data for the different experiments. We used an OpenNMT for developing our model and Bilingual Evaluation Under Study (BLEU) is used in evaluating the translation quality of our model. We trained our model for the different experiments on Colab and we found the best performing translation with BLEU score of 15.4%. Despite the hungry nature of NMT models for data and the costly available data for the corpus, our model has performed well with the amount of corpus collected.

Keywords: Geez, Amharic, Translation, NMT, attention mechanism

TABLE OF CONTENTS

LIST OF ACRONYMS

ALPAC- Automatic Language Processing Advisory Committee

ANN- Artificial Neural Network

BLEU- Bilingual Evaluation Under Study

CAT- Computer Aided Machine Translation

CBMT- Corpus Based Machine Translation

DMT- Direct Machine Translation

DNN- Deep Neural Network

EBMT- Example Based Machine Translation

EOTC- Ethiopian Orthodox Tewahdo Church

GNMT - Google Neural Machine Translation

HAMT- Human Aided Machine Translation

LSTM- Long-Short-Term Memory

MAHT- Man Aided Human Translation

MT- Machine Translation

NLP - Natural Language Processing

NMT- Neural Machine Translation

PBMT- Phrase Based Machine Translation

RBMT- Rule Based Machine Translation

RNN- Recurrent Neural Network

SL- Source Language

SMT- Statistical Machine Translation

SOTA- State-of-the-art

SVO- Subject Verb Object

TL- Target Language

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

1. INTRODUCTION

Communication is the transfer of information from one to the other and language is the core element of communication. Nowadays, it is estimated that more than 6,500 languages are spoken around the globe. These languages are different in syntax, word order. This causes a problem in achieving a good global communication system. Having a central focal point where these different languages are translated from one another solves the problem of the international communication barrier. The state-of-the-art 'central focal point' is a machine translation. Many different and/or similar machine translation models are developed which translate from one natural language to the other. However, there are languages that are not yet included in the state-of-the-art machine translation systems one of which is Ge'ez.

Machine translation (MT), one branch of Natural Language Processing (NLP), helps in communication between machine-to-machine (like question answering) and human-to-human(Young et al., 2018). NLP is a combination of computational methods, which are theory-driven, used for human language representation and analysis. Deep learning is a recently used approach for one of the NLP categories, machine translation. Unlike traditional machine translation, neural machine translation is a better choice for more accurate translation and it also provides better performance (Singh et al., 2017). Deep learning which uses neural networks, is a type of learning in which researchers are attracted to the field of machine translation (Koehn & Knowles, 2017). This is a machine translation that is referred to as Neural Machine Translation (NMT). The basic motive behind NMT is to create a trained model for a system that will work as a translator by having learned from previous experiences and antiquity without the need to use linguistic rules.

1.1. Background

The idea of using machine translation as a complementary tool to improve the translation process has existed since its creation, with the exception of a few attempts to replace human translators.

The idea of translating human languages by machine was first imagined in the 17th century and become into action in the 20th. Computer programs were developed for

producing translations but those were not perfect translations, at that century the translators were used for the translation of technical manuals, commercial documents, medical reports, and scientific documents. The history of machine translation was discovered by inventors and early systems in the 1950s and 1960s. According to a report on the influence of the Automatic Language Processing Advisory Committee (ALPAC), the mid-1960s, the revival of the 1970s, the advent of commercial and operational systems in the 1980s, research in the 1980s, new research progress in the 1990s, and the usage of the system is growing up in the past years(J. W. Hutchins, 2015).

MT is an automatic translation in the field of artificial intelligence. Even though there is uncertainty in clearly identifying the difference between Human-aided machine translation (HAMT) and Machine-aided machine translation (MAMT), the term computer-aided translation (CAT) can include both concepts, automation of the overall translation task being the core of MT(J. Hutchins & Somers, 1992).

Translation in its simple definition includes: defining the source text and re-encoding this translation into the target language. The translator fully decodes the meaning of the original text. This means translators need to interpret and analyze all features of the text. At the same time, translators require the same detailed knowledge to re-encode the meaning of the target language.

The main aim of MT is to provide a system that translates text from a source language into a target language. The translated text should reflect exactly the same meaning as the source language text for a pair of languages (bilingual systems) or for more than a single pair of languages (multilingual systems). Linguistic systems that are designed to work in only one direction are called unidirectional bilingual systems whereas those that work in both directions are called bidirectional bilingual systems. Unidirectional systems do translation only from the source language to a target language, not in the reverse direction. On the other hand, bidirectional systems do translate both from the source language to the target language and vice versa. Multilingual systems are bidirectional; most bilingual systems are unidirectional.

Nowadays, machine translation has become a great concern in relation to natural language processing. The improvements in technology, increasing digital collections, the technical facility, and the continuing interest of Interlingua resources sharing have

necessitated the development of MT. Especially languages with limited resources are to benefit from the translation of the other digital corpus languages.

The major MT approaches are Rule-based and corpus-based. Rule-based machine translation (RBMT) approaches are performed by linguistic professionals to define the set of rules for the translation process. These machine translation approaches work on the morphology, syntax, and semantic of both languages. Therefore, we required the syntax analysis, semantic analysis of source text. Syntax and semantic generation are needed to generate a text in the target language. We also need the bilingual dictionary of source and target languages(Harper, 2018). The sub approaches of Rule-based approaches are Direct, Transfer based and Interlingua approaches.

In the Corpus-based approach, rules are automatically extracted and it is actually data-driven machine translation. It was introduced as an alternative approach to the rule-based approach. In this approach, the bi-language parallel corpus is used to extract the translation for new sentences. The sub approaches of corpus-based approaches are Statistical machine translation (SMT), Example-based machine translations (EBMT), and NMT.

Any generic human translation first encodes the source sentence and then decodes it into a target sentence word by word, looking back to the source sentence each time a single word is generated. In the following sections of the document, we are going to describe a neural machine translation approach for a pair of Ethiopian languages, Ge'ez and Amharic.

Many pieces of evidence witness that Ethiopia used to have its own educational and writing systems. Captions written with Ge'ez, Saba, and Greek language on Axum obelisks can be a good example.

Ge'ez is an ancient language used in the Horn of Africa, Ethiopia, and Eritrea. It is a "pure" Semitic language compared to Amharic and other Ethio-Semitic languages. It was a major language and linguistic king until the beginning of the twentieth century. Thereafter, it gradually started to disappear and eventually replaced with Tigrigna in northern Ethiopia and with Amharic in Central.
Ge'ez script is an alpha syllabary script also called "Abugida", in which a character represents a consonant and a vowel combination. This is different from an alphabetic

script where a character represents one sound either a consonant or a vowel. The alphabet of the Amharic script is unique scripts acquired from Ge'ez and use an alpha syllabary writing system where the consonant and vowel are combined to form a single symbol. Thus, once a person knows all the alphabets, he/she can easily read and write both Ge'ez and Amharic. Scripts in Ge'ez include 26 basic alphabets called 'Fidel' whereas there are 34 alphabets in Amharic scripts (Harper, 2018).

Amharic is a Semitic language spoken in Ethiopia. It originally came from the northern part of Ethiopia and is spoken by the people who live in an area known as Amhara. Later on, however, the language became an official language that is spoken all over the country. Thus, it was declared as a working language across the country, Ethiopia. It is also the second most spoken Semitic language (25 million) in the world after Arabic(Argaw & Asker, 2007). The Amharic alphabets are written from left to right. The language is spoken by 2.7 million people outside Ethiopia.

Amharic has its own writing system. It contains 34 alphabets and 26 of them are derived from the Ge'ez alphabets. The remaining eight alphabets are modifications of the Ge'ez alphabets which are; ሰ to ሸ ፣ ተ to ቸ ፣ ነ to ኘ ፣ ከ to ኸ ፣ ዘ to ዠ ፣ ደ to ጀ ፣ ጠ to ጨ and ፀ to ፘ. These 34 consonant symbols, with seven variations, and variations according to the vowel that is paired with a consonant. Each letter or symbol usually represents the entire syllable(Desalegn, 2015) (Kassa, 2018).

1.2. Problem statement

Nowadays, there are a few numbers of speakers of the Ge'ez language in Ethiopia. But there is a lot of literature written in the Ge'ez language which contains ancient but yet very important data. The vast literature written in Ge'ez derives expertise in different varieties of our life. Religious texts such as the Bible, theological, and magical texts, stories saints, religious poetry, and angels are some of the literatures which includes important ideas and philosophies. Such as Metsihafe Henock, which includes both medical and magical information, Ṣalota Nabiyat, AnqaṢa Berhān, Ayena Wareq, Māhléta Ṣegé, Seyefa Śelāssé, Qeddāsé Māryām, Dersāna Mikā'él, Amestu A'emāda Mesṭir.

Concepts that are important to get the idea of creativity, philosophy, and tradition of our country Ethiopia are written in Ge'ez. Therefore, the main problem here is understanding the very important data or information in this literature. The Ge'ez

Amharic MT is supportive for reading and communication purposes, as well as for language education in cases where the Geez language is taught and, in the meanwhile, there are unclear meaning of words, phrases or sentences. Translating Ge'ez text documents into Amharic text is currently very difficult. Different approaches have been proposed to address this difficulty. However, MT using deep learning is a state-of-the-art (SOTA) method with an improved quality of translation, fluency, speed, and accuracy. The previous approaches which require a linguistic professional in the translation process(Kassa, 2018)(Okpor, 2014). In addition to this the model design process is more complex which needs independent design of components included in the translation. Currently there are two translation works on Ge'ez to Amharic by Dawit Mulugeta who uses SMT and Tadesse Kassa who uses unsupervised segmentation and rule-based segmentation (Kassa, 2018; Mulugeta, 2015).

1.3. Research Question

1) What is the effect of maximizing the number of sentences to train the machine on translator's performance?
2) Is it possible to get generally understandable translation for the language pairs (Ge'ez- Amharic)?
3) Does attention mechanism help to improve the translation result?

1.4. The objective of the study

1.4.1. General objective

The general objective of this thesis is to design and implement a model for Machine Translation from Ge'ez to Amharic using deep learning.

1.4.2. Specific objective

To achieve the general objective, the following specific objectives will be addressed.

- ✓ To review the literature on natural language processing which are about MT and similar to MT, writing system of both Ge'ez and Amharic;
- ✓ To assess different techniques and approaches employed so far in MT and identify which approach is better for this bilingual translation;
- ✓ To develop a Geez-Amharic NMT model
- ✓ To prepare training and test data set;
- ✓ To train an MT system using the selected Machine Learning algorithm;

- ✓ To test the performance of the system;
- ✓ To conclude and forward recommendations

1.5. Scope of the study

Ge'ez to Amharic, machine translation designed to translate sentences written in Ge'ez text into Amharic text. The limitation of our study is, since there is a scarcity of resources or parallel corpus of both Amharic and Ge'ez languages, we collect the data from the religious domain specifically from Ethiopian Orthodox Tewahdo's Church (EOTC). such data are from the holy bible (New and Old Testament), wudasie Maryam, Drsane Gabriel, Drsane Michael, Kidasie, Liton.

The translator does not accept speech as input nor it translates the speech into the text of the source language. It is only text-to-text translation work. And our model is a uni-directional machine translation, it translates only from Ge'ez to Amharic not Amharic to Ge'ez. Our intension is to translate the available Ge'ez documents which have different information about the background of our country into our national language Amharic, there is no need of converting Amharic documents into Ge'ez since it is official language.

As we have mentioned in the introduction part, there are different types of MT, like Rule-Based, Example-based, statistical MT, NMT, and hybrid MT. Among these approaches, Rule-Based MT needs linguistic professionals since it is Knowledge-based MT and it has better performance when we compare with other approaches. But in order to use this approach, there is a scarcity of linguistic professionals for Ge'ez and Amharic languages. Instead, we are using the recent MT approach which is NMT, to design the translation model. It doesn't need professionals it only needs a large amount of parallel corpus to have a good translation result. And our research doesn't include punctuation marks as well as numbers.

1.6. Significance of the study

Nowadays, an MT began to gain attraction in the early 1950s and has come a long way. At present, the value of the MT market is estimated to be between $130-400 million, and as technology improves, more and more companies are using MT to help translators and optimize localization processes.

The results of the study can be used to develop machine translation software for Ge'ez to Amharic, which will be used to translate huge litterateurs in Ge'ez to Amharic. Researchers who need to take part in achieving the goal of developing an efficient NLP system for the Ge'ez language can greatly benefit from this work. Since NMT is much faster and accurate than a human translator is, an NMT service translates the available Ge'ez contents into the Amharic language quicker than ever before and to inspire students for learning the Ge'ez language. Basically, most of the ancient Ethiopian history is written in Ge'ez; so, understanding this is not only essential for the Ethiopians but also the rest of the world.

Therefore, the society that is capable of understanding Amharic will be able to acquire resources written such as history, philosophy, laws, traditions, religion, etc. and it is important to tackle the language barrier.

The new generation will have a better understanding of their culture, language, norms, and values instead of being confused with foreign languages. One can easily be able to understand the ancient Ge'ez literature which is found in many churches and monasteries. And it plays a great role in the development of the Ethiopian tourism sector. The huge amount of indigenous knowledge that has been accumulated in the country is written in the Ge'ez language. So, translating Geez to Amharic will unroll a huge knowledge to society. It helps design information retrieval and extraction across languages for translating the document the user is searching for and/or the user's query mode(Desalegn, 2015). It also has an academic value in the motivation of the researchers when conducting MT among local language searches and NMT is a better choice.

1.7. Organization of the thesis

The thesis is organized in five chapters which constitute introduction, literature review, methodology, experiment, result and discussion, conclusion and recommendation, the first chapter contains the general synopsis of the entire thesis. It includes the background of the thesis, problem of statement, research questions, the objective, scope, and significance of the study. The second chapter is literature review which briefly discuss the approaches of machine translation, such as rule based, corpus based,

neural and hybrid machine translations. Additionally, in the related works section machine translation works on local language and international languages are reviewed.

The third chapter is methodology which discusses how NMT can be implemented, the Ge'ez-Amharic model, data collection and the tools, software tools and hardware environments, we used during our experiment. The fourth chapter is experiment, result and discussion, which briefly discusses the experimental setups, number of the experiments we do, the parameters we consider during the experiment, the result of each experiments and it discusses about the results. The final chapter, chapter five entitled with conclusion and recommendations, which concludes the whole idea of the thesis and forward some recommendations.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. Geez and Amharic Languages

Geez language

The language Ge'ez (ግዕዝ) is also called traditional Ethiopic language, which is the ancient southern Semitic of Ethio-Semitic language branch. It was official language of the Kingdom of Aksum and Ethiopian imperial court.

Nowadays, Ge'ez is used only as of the main sacred language of the Ethiopian and Eritrean Orthodox Tewahdo Church, the Ethiopian Catholic Church and the Catholic Church Eritrea and the Jewish Community Beta Israel(O.Lambdin, 2014).

The most related Ethiopian local languages to Ge'ez are Tigre and Tigrinya, Ge'ez became a separate language early on from another hypothetical unattested common language. Because Ge'ez is no longer spoken in daily life by large communities, the early pronunciation of some consonants is not completely certain (Mulugeta, 2015).

Ge'ez is studied and taught in Ethiopia. Europe and United States University. Holy Trinity Spiritual College Ethiopia teaches Ge'ez at the diploma level. It is also taught by the Ethiopians Orthodox Tewahed Church School. Since the origin of the Ge'ez language is the Ethiopian Orthodox Tewahdo Church, A church that teaches Ge'ez at traditional schools at home and abroad(Kassa, 2018)(Mariam, 1963).

Geez, the script is an alpha syllable script also known as "Abugida", and the writing direction is from left to right in a horizontal line, in which a character represents a combination of consonants and vowels. It has different forms in the form of a letter. A sound represents a consonant or a vowel. The alphabet of Amharic writing is unique it uses Ge'ez scripts and an alpha syllabary writing system where consonants and vowels are combined to form a single sign.

The original Ge'ez script was abjad and no vowels were written, but the current script is classified as abugida. Each symbol represents a CV syllable, but vowels are not unique to consonants. The original Ethiopic scripts contained 182 characters, but the basic (unmarked) consonant is only 26 characters. The script has been extended to other languages and currently contains over 500 symbols. Each of the first order consonants can be combined with one of seven vowels, to produce a syllograph i.e., independent

sounds and other linguistic units with seven syllographic categories. The resulting sets of syllograph are known as the second, third, fourth, fifth, sixth, and seventh orders.

Morphology of Geez language:

Ge'ez distinguishes between two genders, male and female. These genders are marked with the suffix -t in certain words. These are not as strongly distinct as the other Semitic languages. There are two numbers in Geez, singular and plural. The plural can be constructed either by suffixing -at (ኦት) to a word, or by internal plural. **Plural using suffix:** semay(ሰማይ)-semayat(ሰማያት) sky(s), mezemr(መዘምር) mezemran(መዘምራን) Singer(s). **Internal plural**: debr(ደብር)-adbar(አድባር) church(s), res(ርእስ)-ariest(አርእስት) title (s)(O.Lambdin, 2014).

Amharic Language

In Amharic, there are 34 basic alphabets or Fidel of which 26 is derived from Ge'ez. The remaining 8 of them were by modifying 8 Ge'ez Fidel's; namely, ሰ to ሸ ፣ ተ to ቸ ፣ ነ to ኘ ፣ ከ to ኸ ፣ ዘ to ዠ ፣ ደ to ጀ ፣ ጠ to ጨ and ፀ to ፈ. As it is described in the above paragraph, to modify the character they were using -, and o. Also, Amharic has taken the entire derived alphabet from Ge'ez(Gated et al., 1963).

Syntax:

Natural languages follow different syntax rule during sentence formation. In Geez language, it follows, Subject + verb + object (SVO), object + verb + subject (OVS) and Verb + subject + object (VSO).

However, the common Amharic sequence of words is Subject-Object-Verb (SOV). However, if the object is that tropical, it can precede the subject (OSV).

Nominal phrases are head-end with adjectives and other modifiers preceding their nouns. Prepositions, postpositions or a combination of both are used to indicate syntactic relations, revealing the mixture of Semitic and Cushitic traits(Kassa, 2018).

2.2. Machine Translation

Machine translation is also called automated translation. It is one of the research areas under computational linguistics and language engineering. As we have discussed in the introduction part, it uses software to translate text or speech from one language to another. In other words, it is a method of converting an original sentence from one natural language to another natural language using computerized systems, without human assistance.

MT Systems are designed either for one particular pair of languages (bilingual systems) or for more than two languages (multilingual systems), either in one direction only (uni-directional systems) or in both directions (bi-directional systems)(J. Hutchins & Somers, 1992).

2.3. Approaches to machine translation

According to the core methodology they use, MT systems can be categorized into two main paradigms namely, the rule-based approach and the corpus-based approach. Linguistic professionals are involved to draft a set of rules which describe the translation process in the rule-based approach thereby requiring a large amount of input from human experts.

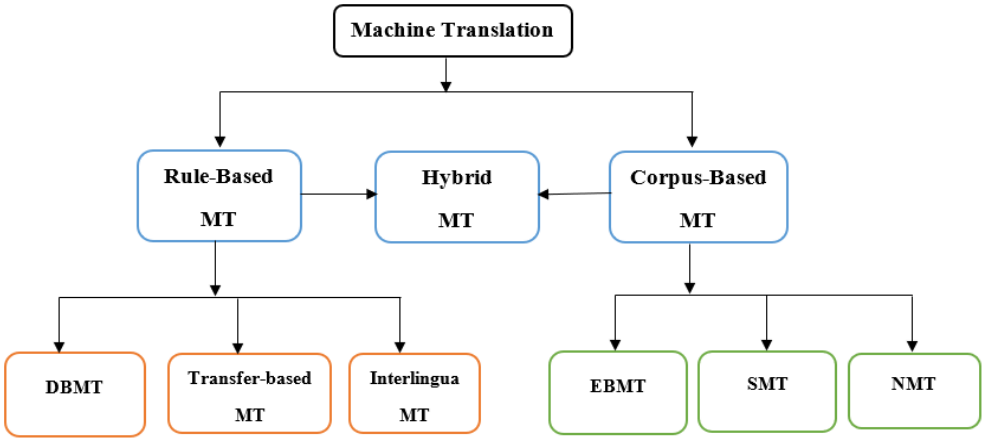


Figure 2:1 Machine translation approaches

On the other hand, automatic knowledge extraction by looking at translation examples from a parallel corpus provided by human experts is possible in the corpus-based approach. The exploitation of combined features of the rule-based approach and the corpus-based approach resulted in a new approach called Hybrid Machine Translation (HMT) approach (Chéragui, 2012)(Okpor, 2014).

2.3.1. Rule-Based Machine Translation Approach

Around the 1970s, the story of MT was started with an RBMT solution. RBMT is also known as a Knowledge-Based MT or Traditional MT Approach. A set of grammatical rules are created for the source and target languages and a type of conversion based on these rule sets is the machine translation which basically depends on the dictionary and grammar retrieved linguistic information of both languages. The semantic, syntactic,

and morphological information of the languages is stored in the dictionaries and grammar.

RBMT is a system that takes sentences from the source language as input and generates an output sentence in the target language. The translation task basically involves semantic, syntactic, and morphological analysis on both the source and the target languages(Harper, 2018) (Okpor, 2014). Although this concept works well for generic content, adding new content and new language pairs, and maintaining a set of rules can be very time consuming and very expensive.

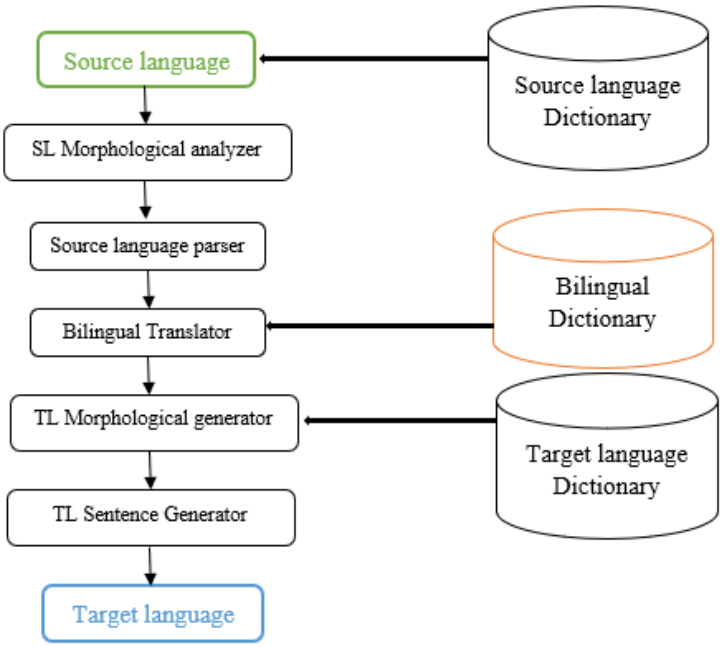


Figure 2:2 Rule-Based Machine Translation approach

Analysis, transfer, and generation are the three stages that an RBMT system uses to apply the set of grammar rules. In the analysis stage, the syntactic representation of the source sentence is produced by a parser. In the transfer stage, target language-oriented representation is produced from the source language representation created in the analysis stage. In the generation stage, a morphological analyzer of the target language is used to generate the sentence in the target language.

Advantages and disadvantages of the RBMT Approach

Advantages

- No need for parallel corpus and translation is possible for languages that don't have common text.
- Easy to debug and control because all the rules are manually written.
- Linguistic rules are not domain-specific which work in every possible domain.
- It is quite a generic approach where the input sentence analysis and output sentence generation can be shared across many translation systems.

Disadvantages

- Good dictionary preparation is very difficult.
- Sometimes it is required to manually prepare linguistic information.
- Effectively handling the interactions among the rules in a big system is hard.
- Changes made for rules to adapt to new domains are costly(Okpor, 2014).

RBMT sub approaches

There are three sub approaches of RBMT(Chéragui, 2012). These are Direct, Transfer-Based, and Interlingua Machine Translation Approaches. However, they are all related to RBMT and differ in the depth of source language analysis and the way they seek to obtain a language-independent representation of meaning between the source language and the target language.

Direct Machine Translation

The direct MT system is the first MT generation and it is often built in a single language pair. It is also known as word-for-word translation. This works by translating words separately regardless of how they are used in the sentence.

The only resource used directly by MT is a bilingual dictionary, so it is also called *dictionary-based MT*. The direct translation system is basically one-way bilingual and this approach requires only a bit of syntax and semantic analysis.

The meaning of each word is discovered later, and the word's position in the string matches the word order in the target language. This includes Subject-Verb-Object (SVO) supports, among others.

Source language analysis is oriented specifically to the production of representations appropriate for one particular target language. DMT is a word-for-word translation with some simple grammar modifications(Okpor, 2014)(Prasad & Muthukumaran, 2013).

Of course, this system has serious limitations. This is not an effect of either the parsing or semantic analysis that is done. As we all know, most words can convey different phrases when used in different contexts.

Direct MT holds sentences as word strings, inter-dependencies and semantic groupings between words may be lost, and there is a very high probability of choosing incorrect interpretations for a given word. A direct translation is linguistically weak. In other words, the word structure and the lack of connections between words. Direct translation removes the sense of nature(Prasad & Muthukumaran, 2013). This can be illustrated as a word-by-word translation with some local word order adjustments. It has provided a very affordable bilingual dictionary and translation quality that you would expect from someone with the most basic knowledge of target language grammar. It is a generally inappropriate syntactic structure that is often misunderstood at the lexical level and reflected very closely to the source language.

This approach is suitable for closely related language pairs. In closely related language pairs, with grammar, the vocabulary in general. Therefore, the effort required to develop a translation system decrease between related languages. Hence, the development of a machine translation system with closely related languages is easier to develop than machine translation systems language pairs that are not related to each other.

Figure 2:3 Direct Based Machine translation

Interlingua Machine Translation Approach

Interlingua machine translation is one example of RBMT approaches. It is based on the argument that MT should go beyond purely linguistic. Information (syntax and semantics) and its meaning is "understanding" of the contents of texts. Any interlingua-based translation contains two monolingual components. The first monolingual component performs source sentence analysis and converts it into a language-independent representation of meaning. The second monolingual component generates target meaning using syntactic construction and lexical units of target language(Alansary, 2014)(Dave et al., 2001).

The strong belief that all languages share a common deep structure despite their difference in their surface structure is the motivation behind the design of interlingua. Hence the idea of creating a scientific representation capable of conveying this deep structure while enjoying natural regularity and predictability languages is missing.

The advantage of the interlingual approach is that one can use interlingual expressions in NLP, a system for multilingual information retrieval, summarization, and other multilingual applications such as Question-and-answer sessions.

The only interlingua machine translation system that is made operational commercially, however, is the KANT (Knowledge-based, Accurate Natural-Language Translation) system, which translates Caterpillar Technical English (CTE) into other languages. The inter-language approach is clearly more interesting for multilingual systems. Each

analysis module can be independent, both from all other analysis modules and from all generation modules(Okpor, 2014)(Hakkani et al., 1998).

The interlingua machine translation approach presents many challenges, the first being that there are complications in the definition of an interlingua, even for closely related languages, secondly, the semantic differentiation is specific to the target language and making such distinctions is comparable to the transfer lexical not all the distinctions necessary for translation and the other is that extracting the meaning of the texts in the source language to produce language independent representation is difficult. The successful motivation behind this method is that given this conceptual representation, a natural language sentence can be generated using a generation module between the representation language and the target language. To include an additional language to a translator of this type merely add an analysis module and a generation module for the new language to be represented. This offers the advantage that the system grows linearly 2n, where n is the number of languages.

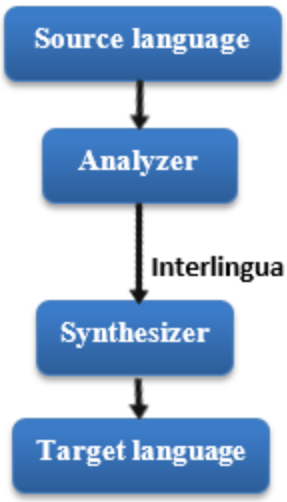


Figure 2:4 Interlingua machine translation approach

Transfer-based Machine Translation Approach

The transfer approach is used on the basis of the known structural differences between the source and target language. These translation systems have three stages. The first phase analyzes the source text and transforms it into an abstract representation. The second stage transforms them into equivalent target language-oriented representations. The third produces the final destination text. The expression is unique to each language

pair. Here, the source language text is translated into language-specific expressions and the same level of abstraction is created using grammar rules and bilingual dictionaries(Ballabh & Chandra Jaiswal, 2015)(Shilon et al., 2011).

In the transfer approach of translation divergence, there is the rule to convert a sentence from the source language (SL) to the target language (TL) through lexical and structural manipulation. **Mantra,** funded by the Government of India, is a transfer-based tool. In contrast to interlingua MT, the transfer-based depends on the language pair involved in the translation(Okpor, 2014)(Ballabh & Chandra Jaiswal, 2015).

Transfer based machine translation also has its own challenges some of those are the following:

- Rules are applied to the source language analysis phase, transfer phase, and generation phase.
- Doing extensive work in the reusable analysis and synthesis modules is difficult.
- Keeping the transfer module as simple as possible is also difficult.

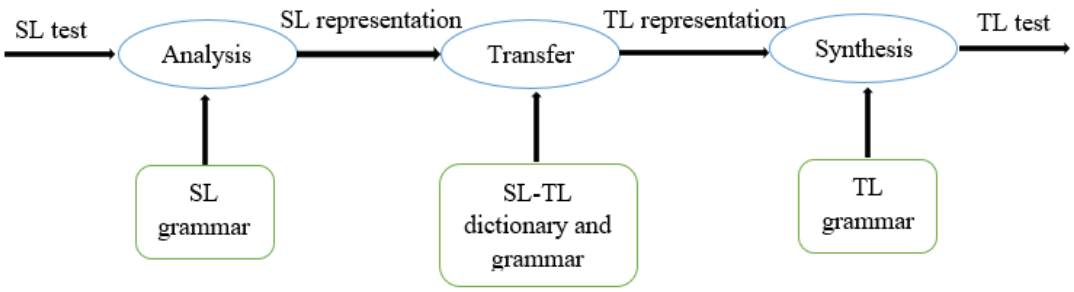


Figure 2:5 Transfer-based machine translation

2.3.2. Corpus-based Machine Translation

Corpus-based machine translation (CBMT) is one of the main methods of machine translation because this method achieves a high level of accuracy at the time of translation and is also called **data-driven** machine translation. It is an alternative method of machine translation to overcome the problem of knowledge acquisition in rule-based machine translation. As its name implies, a parallel bilingual corpus to obtain knowledge of the new translation that arrives. This approach uses a large amount of raw data in the form of a parallel corpus. This raw data includes the text and its translation. These corpora are used to gain translation knowledge(Shilon et al., 2011).

Large volumes of translations are presented after the development of a corpus-based system used in various computer-assisted translation applications. Example-based MT, Statistical MT, Neural MT are three different sub approaches of corpus-based MT.

Statistical Machine Translation (SMT)

SMT was begun in the late 1980s and early 1990. SMT systems create statistical models by analyzing the source sentence. The SMT system analyzes the tuned source and target language data (training set) to create statistical models and uses them to create translations. In recent times, statistical data analysis has been used to automatically gather machine translation knowledge, from the parallel bilingual text. These methods are extremely important in providing methods to deal with the knowledge acquisition problem that is encountered by all NLP applications (Ney, 2014).

Statistical models are applied in this method to create translated results with the help of bilingual corpora. The concept of statistical machine translation derives from information theory. The fact that no linguistic experts are required for customization work is an important feature that helps the tool to learn translation methods by only taking the statistical analysis of the parallel corpus.

The process of building such a system is based on a parallel corpus, which is a set of sentences in the first language and their corresponding translations in the other. The training of an SMT system produces a series of probabilistic models that have been automatically induced by the parallel corpus. These templates are then used by a decoder to perform the actual translation of documents never seen before (Vandeghinste & Van Eynde, 2012).

Among the models learned during the training process are the Translation Model (TM), which determines the match between words and phrases in both languages or stores different translations of the sentences, and the Language Model (LM), which measures fluent language output or stores the probability of a series of sentences on the target side.

The quality of the SMT system is commonly measured by the results of the bilingual evaluation under study (BLEU). Language models have their own common evaluation

metric, perplexity because language models are also used in other fields, such as Automatic Speech Recognition (ASR).

The idea behind statistical MT is the following:

"Given a sentence T in the target language, we seek the sentence S from which the translator produced T. We know that our chance of error is minimized by choosing that sentence S that is most probable given T. Thus, we wish to choose S so as to maximize Pr(S|T)" (Brown et al., 2002).

The maximization problem can be converted to the product of *Pr(S)* and *Pr (T | S)* using *Bayes' theorem*. Where Pr(S) is the language model probability of S (S is the correct sentence for that location). Pr (T | S) is the probability of translation of T given S. In other words, we are looking for the most likely translation, considering how accurate the candidate translation is and how well it fits the context.

$$Pr(S|T) = \frac{Pr(S)Pr(T|S)}{Pr(T)} \quad 2.1$$

SMT desires the following three steps:

1) Language Model (what is the correct word in its context?);
2) Translation Model (what is the best translation for a particular word);
3) A method of finding the correct word order(Aadil & Asger, 2017; Brown et al., 2002).

Generally, the basic idea in SMT is to learn all parameters from parallel data. And the Major types of SMT are word-based, phrase-based, and morpheme-based. Its strength is easy to build, and it requires no human knowledge, good performance when a large amount of training data is available.

The advantage of SMT is that it is an automated learning process and can be adapted relatively easily by altering or expanding the training set. An SMT is suitable for more language pairs, translations outside the dictionary: with the right language model, the translation is smoother when we compare with previous approaches. Its weakness is how to express linguistic generalization or the training set itself, not suitable for

language pairs with large differences in word order and specific errors are difficult to correct.

Creating a usable engine requires a large database of source and target segments. Moreover, SMT is language-agnostic in the sense that it is very sensitive to language combinations and very difficult to handle grammatically rich languages. Furthermore, SMT is language independent in the sense that it is very sensitive.

Here, NMT starts flashing. This allows us to see the entire sentence and make connections between phrases at greater distances within the sentence. The result is forcing fluidity and better grammatical accuracy than SMT. The architecture of SMT is shown in Figure 2:6 Architecture of SMT.

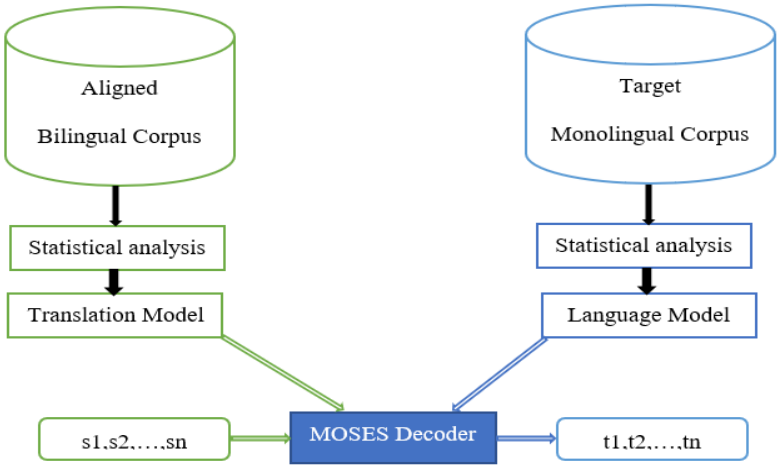


Figure 2:6 Architecture of SMT

Table 2:1 A comparison of SMT and RBMT

SMT	RBMT
The translation quality couldn't be predicted	Steady and predictable quality
Doesn't need a lot of labor	Labor-intensive
Doesn't know linguistic rules	Knows linguistic rules
It needs more memory space	High performance and robustness.
Good fluency	Lack of fluency
Suitable for exception handling	Difficult to handle rule of exceptions
It needs less time	Time-consuming

Example-Based Machine Translation (EBMT)

A bilingual set which contains bilingual translations as a basic knowledge is used to characterize Example-based machine translation. An EBMT system takes a series of sentences in the source language (to be translated from) and corresponding translations of each sentence into the target language with point-to-point mapping. These examples are used to translate similar types of sentences in the source language into the target language.

The basic idea of translation used in example-based machine translation is a translation by analogy. The principle of analogy translation is encoded in the machine translation based on examples through the example translations used to train such a system(Okpor, 2014). Example acquisition, Example base and management, Example application, and synthesis are the Basic stages followed in EBMT.

The first task is example acquisition, which describes how to get an instance from an available bilingual parallel corpus. The required example could consist of bilingual dictionaries at the word level, the corpus at the multi-word level, and the sub-sentential levels including idioms, collocations, and multi-word terms and phrases.

The alignment of the text is a necessary step to obtain examples at various levels. Text alignment methods can be further classified into two types, i.e., resource-poor approaches that rely primarily on phrase length statistics, co-occurrence statistics, and some limited lexical information, and resource-rich approaches that use any useful and available bilingual lexicon and glossary.

The second task of EBMT is Example base and management, which focuses on how the examples can be maintained and archived. This function is the most important module in the EBMT system because it handles the storage, edition, and retrieval of instances. Therefore, to be efficient the EBMT needs to be capable with a sufficiently high speed of finding large-scale instances from both source and target languages.

The third task is an example application that looks at how existing examples can perform translation, which involves getting examples from decomposing the original input sentence and converting the decomposed original sentence to a target sentence.

The fourth task is the so-called synthesis and smoothing of sentences, which consists of composing the target sentence by placing the transformed examples in an easy-to-read order, aimed at improving the readability of the target sentence after the transformation.

After decomposing the sentence and transferring the example, we now have a sequence of translated fragments. The next task is to combine these translated pieces into a well-formed and highly readable sentence in the target language.

Since different languages have different syntax rules for controlling the sentential structures and word order, in most cases it won't work if we simply link the translated parts in the same order as in the source language(Mulugeta, 2015)(John Hutchins, 2005) .

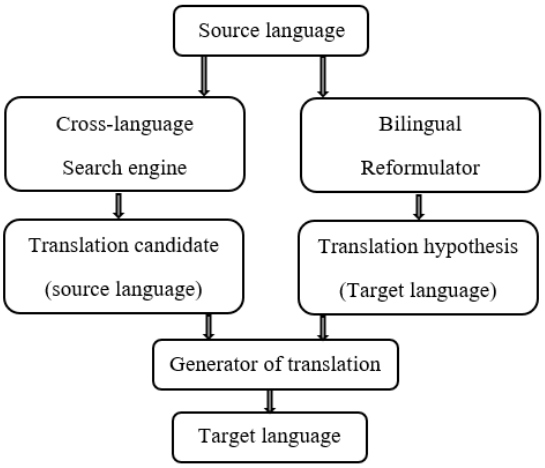


Figure 2:7 Example-Based Machine Translation

There are three components of EBMT:

- Matching the input to a database of real examples
- Identifying related translation fragments
- Reconnect pieces or fragments to the target text

The main advantage of this model is that it works well with a small set of data and it is possible to generate output more quickly by training a translation.

Two totally unrelated languages can be translated one to/from the other using the Example-based method. One of the main drawbacks of an Example-based engine is that applying deep linguistic analysis is not possible. PanEBMT is an example of an EBMT tool (Brown et al., 2002).

Table 2:2 A comparison of SMT and EBMT

SMT	EBMT
Corpus-based	Corpus-based
uses statistical data such as parameters and probabilities	An analogy is the main idea (uses the bitext as its primary data source) Data preprocessing is optional and the same transformation occurs if the input is in the set of examples.
Needs a large amount of data	Can work with a small set of data
It takes time to train	It can train and decode quickly

2.3.3. Neural Machine Translation

In this section of the document, we are going to comprehensively describe how machine translation using a deep learning approach with the use of neural networks. This approach is simply known as neural machine translation. Before we describe neural machine translation, we provide a brief introduction to deep learning.

Introduction to deep learning

Deep learning (DL) is an exciting and powerful field of machine learning. It uses an algorithmic layer to process data, understand human speech, and visually perceive objects. The flow of information across the layers is made possible by passing the output of a previous layer to the next layer as input. The first layer of the network is called the input layer and the last layer is called the output layer. All layers between the two are called hidden layers.

Each layer is usually a simple, uniform algorithm containing one type of activation function. Feature extraction is another aspect of deep learning. Feature extraction uses

algorithms to automatically build meaningful "features" of the data for training, learning, and understanding.

Deep learning is the name of an approach to artificial intelligence known as neural networks which have been going out of fashion for more than 70 years. The history of deep learning can be traced back to 1943 when the two researchers of the University of Chicago Walter Pitts and Warren McCulloch created a computer model based on the neural networks of the human brain.

A paper on how neurons work is written by Warren McCulloch, and Walter Pitts. They showed a simple neural network in electrical circuits. In the 1950s, Nathanial Rochester from the IBM research labs made the first efforts to simulate a neural network.

The network is usually done using electronic components or software on a digital computer. Those are a way of doing machine learning, where the computer learns to perform some tasks by analyzing training examples(Bishop, 2015).

Neural networks are massively parallel processors with natural processing power and availability. It is similar to the brain in two ways a) the network acquires knowledge of the environment by the learning process and b) the acquired knowledge is stored between the neurons as an inter-neuron connection strength or a synaptic weight.

Artificial Neural Network (ANN) is a neural network model that works based on the neural structure which learns to perform tasks by looking at examples. The tasks include decision-making, prediction, classification, and more. It is a collection of artificial neurons organized in three interconnected layers known as input, hidden, and output. More than one layer may be included in the hidden layer. One type of ANN that includes many hidden layers between the input and the output layers is known as Deep Neural Network (DNN).

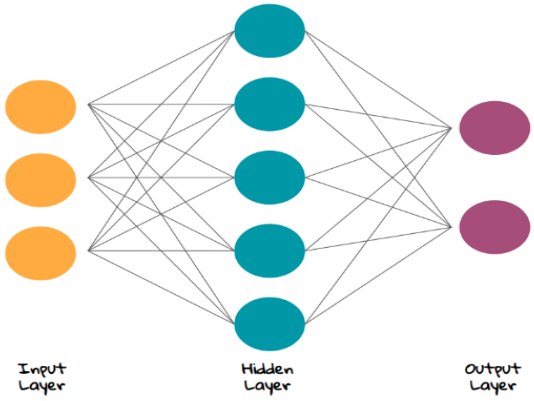


Figure 2:8 Architecture of ANN

In many ways, ANN is not much different from other machine learning methods but has distinct strengths, in which the computer learns to perform some tasks by analyzing the trained examples. Commonly, the examples have been hand-labeled in advance. For example, more than thousands of organized images of trees, animals, mountains, and others, it would find the pattern of these visual images that consistently correlate with the particular labels called an object recognition system.

Artificial neural networks have two main parameters that control the network's architecture or topology, it depends on the number of layers and the number of nodes in each hidden layer(Techopedia, 2020). A given ANN can be a single layer or multilayer. The architecture of ANN is shown in Figure 2:8.

Single-layer neural network

A single-layer neural network is the simplest form of the neural network, there is one layer of input nodes and the receiving nodes takes the weighted inputs from the previous layers, or in some cases, to a receiving node. This single-layer design was part of the basis of more complex systems. The information flow in a single-layer neural network is only in one direction directly from the input to the output. So, it is one of the types of feedforward neural networks.

Multiple layer neural networks

Multilayer neural networks consist of multiple layers of artificial neurons or nodes. They differ greatly in design. While single-layer neural networks were useful early in

the development of AI, it is important to note that most networks in use today have multilayer models.

Multilayer neural networks can be configured in several ways. They usually have at least one input layer, which sends the weighted inputs to a series of hidden layers and, finally, the output layer. These more sophisticated configurations are also associated with nonlinear constructions that use sigmoid and other functions to direct the firing and activation of artificial neurons. Some of these systems can be built physically using physical materials, but most are made with software resources that model neural activity.

Input layer

The input layer of a neural network is made up of artificial input neurons and brings the initial data into the system by artificial lunar layers for subsequent processing. The input layer is the beginning of the workflow of artificial neural networks.

One of the distinguishing features of the input layer is that artificial neurons have a different role in the input layer, experts interpret it as a layer formed as “passive” neurons that do not take information from the previous layers because they are the first layer of the network.

In general, artificial neurons tend to have a set of weighted inputs and functions based on these weighted inputs, although, in principle, an input layer can be made up of artificial neurons that do not contain weighted inputs, or where weight is calculated differently, for example at randomly, because the information is entering the system for the first time. What is common in neural network models is that the input layer sends data to subsequent layers, which contains the weighted input from the neurons.

Hidden layers

The hidden layer in an artificial neural network is a layer between the input layers and the output layers, where artificial neurons receive a series of weighted inputs and produce outputs through an activation function.
It is a specific part of nearly all neural networks where engineers simulate the type of activity going on in the human brain. There are many different ways to set up hidden neural network layers. In some cases, weighted inputs are assigned randomly. In other

cases, they are adjusted and calibrated through a process called backpropagation. In any case, the artificial neuron in the hidden layer functions as a biological neuron in the brain: it receives its input probabilistic signals, processes them, and produces the input signals into an output signal corresponding to the axon of the biological neuron.
Many analyses of machine learning models focus on building hidden layers of neural networks. There are different ways to configure these hidden layers to produce different results. For example, homogeneous neural networks that focus on image processing, recursive neural networks with memory, and simple feedforward neural networks. Define the elements that work directly in the training data.

Output layer

The output layer of an artificial neural network is the last layer of neurons that produce a certain output for a program. They are made very similar to other artificial neurons in a neural network, but since they are the last "actor" nodes in the network, neurons in the output layer can be constructed or observed in different ways.

A distinctive traditional neural network has three layers. One or more input layers, one or more hidden layers, one or more output layers. Simple forward neural networks with three separate layers provide basic, easy-to-understand models. More sophisticated, innovative neural networks can have several types of layers of any type and, as mentioned, each type of layer can be built differently. A traditional artificial neuron consists of some weighted input, a transformation function, and an activation function corresponding to the axon of the biological neuron. However, to streamline and improve the iterative process, the output layer neurons can be designed differently.

Machine learning has many different variants of neural networks. RNN is one of these different types which we are going to use to develop our model of the Ge'ez to Amharic neural machine translation.

Neural Machine Translation (NMT) is a type of the corpus-based MT approach and it is also called sequence to sequence model, data-driven or corpus-driven machine translation(Forcada, 2017). It is a deep neural network model that happened to be the state-of-the-art machine translation nowadays. The statistical models of the machine translation are learned by the use of neural network models. Google Translate, Baidu

Translate are well-known examples of NMT that are made available to the public on the Internet.

As we can see in Figure 2:9, neural machine translation technology is currently the cutting-edge technology in machine translation and offers the highest quality translation.

The main benefit to using NMT is that the system is trained on the source and target text directly without having to leverage any specialized systems which SMT systems used. According to (Bahdanau et al., 2015)and (Krenker et al., 2011), neural machine translation builds and trains a single, large neural network that reads a sentence as input and produces a correct translation as output in an end-to-end fashion (mapping from input sentence to its corresponding output sentence). Separate models such as the language model, translation model, and reordering model are not needed, it is just a large single sequence model (neural network) that predicts one word at a time.

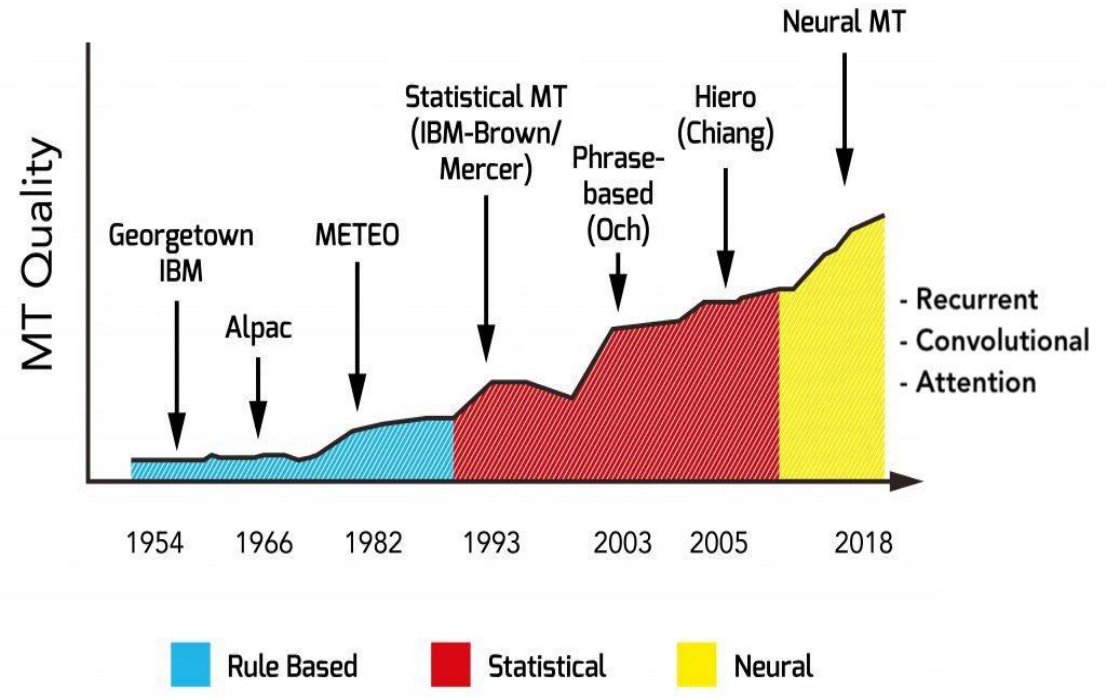


Figure 2:9 Progress of machine translation over time(Iconic, n.d.)

When producing translations, NMT thinks more like the human brain unlike its predecessor, RBMT, and SMT. In NMT, Words or even word breaks are transformed

into “word vectors”. This means that "dogs" not only represent the letters d, o, g, but can also contain information related to training data(Brown et al., 2002).

In the training stage, the NMT system attempts to assign weights to the neural network based on source-target translation values. A similar word vector will be generated for words appearing in a similar context. As a result, a neural network that can process source segments and transmit them to target segments will be created.

In NMT, instead of translating word for word, or looking at shorter strings of words in parallel, it can take whole sentences and more into context at once. The results of this are more fluent, accurate, and usable translations than ever before. NMT doesn’t care about the different meanings of letters with the same phonetics, Ge’ez language, for example, has three letters with the same phonetics ሀ፡ ሐ and ኀ, since it is context based. Word’s sound has not been a concern. The tool for our translation model, OpenNMT, uses a preprocessing algorithm that is not language specific, rather generic. Our translation process doesn’t yet consider the word sounds, the word context is instead thoroughly considered using a method called attention mechanism. Since the dataset we used is religious data, we believe that the issue of “same Amharic sound but different Geez meaning” is resolved with a well prepared and credible dataset, which ours’ is. More generally, this issue will have a greater effect on studies that focus on text to speech or speech to text translation.

2.3.4. Hybrid machine translation

Hybrid machine translation (HMT) combines the basics of existing methods that include, rule-based MT, statistical-based MT, and example-based MT, and creates the flaws of the individual MT method. The failure of previous translation systems to achieve a satisfactory level of translation accuracy is the motivation for developing HMT.

A hybrid machine translation engine is made to include the rule-based translation module and the statistical translation module being capable of converting source sentences into the target sentences according to the rules in the models. The goal is to combine the best features of each type of MT approach(Almansor & Al-Ani, 2018) (Xuan et al., 2012).

2.4. Related Works

Several works related to the field of machine translation have already been proposed and developed previously. In the next sections of the document, we are going to briefly provide the previous related works which have been developed both for local languages, in Ethiopia, and foreign languages.

2.4.1. Local language MTs

English to Amharic SMT

English to Amharic statistical machine translation (EASMT) was conducted by (Gebregziabher, 2012). The main purpose of the study is to develop English to Amharic machine translation using SMT. In the EASMT system, the experiment was conducted in the training corpus of both languages based on the words contained in the parallel documents.

To measure the accuracy of the targeted translation system, an experiment had been done using 18,432 English-Amharic parallel sentences extracted from the prepared corpora. 90% of the collected parallel corpora randomly selected have been used for training and the remaining 10% parallel sentences are used for testing and validating the system. In addition to the parallel corpora, the researchers used 254,649 monolingual corpora. The monolingual corpus is used for Language Modeling (LM). Accordingly, the baseline phrase-based BLEU score result was 35.32%.

In the Preliminary experiment, the results show that EASMT can translate the basic meaning of English sentences to translate to Amharic sentences. However, there are some pros and cons. EASMT performance. Retain the strong aspects to resolve issues such as untranslated words, translation errors, they use insertion, deletion, alignment problems, preposition usage, and morphological errors the word segmentation at the target is very important.

The researchers recommended in (Gebregziabher, 2012) recommended that more experiments and research are needed to further improve EASMT translation accuracy. The experiment done so far is as promotive as the translation is made from less fluent English to morphologically rich Amharic.

Bidirectional English-Amharic MT

English to Amharic machine translation using SMT approach with constrained corpus is conducted by (Teshome, 2013). An experiment using two different corpora was made with the first corpus containing simple sentences and the second containing complex sentences. A bidirectional translation model that assigns a probability that a given source sentence generates a target sentence was built. A decoder and an expectation-maximization algorithm were used to search for the shortest path and to align words in their correct order respectively.

According to (Teshome, 2013), an experiment was carried out for the first corpus and the second corpus separately. The BLEU result of the experiment conducted using the first corpus for English to Amharic translation was 82.22% and 90.59% for Amharic to English translation was recorded. The result of the experiment using the second corpus was approximately 73.38% and 84.12% for English to Amharic and for Amharic to English respectively. Finally, (Teshome, 2013)concluded that with a large amount of dataset available, a good translation performance could be achieved as there would be more words in the corpus which will maximize the probability that a word precedes another.

English – Afaan Oromo MT

English to Afaan Oromo MT was conducted by(Sisay Adugna, 2009). A statistical approach to machine translation was applied regarding the design of a machine translation system from English to Afaan Oromo.

The dataset for the experiment was gathered from both governmental and non-governmental documents. The corpus used in the experiment was preprocessed using the Perl script. Apostrophes, sentence alignment, tokenization, lowercase, these scripts truncated long phrases that prevented the alignment from being optimized.

A 32-bit Linux machine was used as an operating system to train his model and the language modeling tool he has used was the SRILM toolkit, for word-alignment, the state-of-the-art method is GIZA++.

The size of the monolingual which is Afaan Oromo 62,300 sentences and a bilingual corpus of 20,000 were used for conducting the experiment of which 90% and 10% used for training and testing the MT system respectively.

Statistical machine translation has been tried for English to Afaan Oromo a score of 17.74% was found. Since Afaan Oromo is one of the resource scarcity languages in the world,
The result of this experiment t shows that the amount of data available can be used as a good starting point for building an English to Afaan Oromo machine translation.

According to the recommendation of (Sisay Adugna, 2009) doing a lot more translation between two languages make the accuracy of the translation genuine. They suggested a lot of translation between the two languages helps to make the accuracy of the translation genuine.

Amharic to Tigrigna MT

Amharic to Tigrigna MT is done by(Woldeyohannis & Meshesha, 2018) using a statistical machine translation approach. They used 27,470 parallel Amharic and Tigrigna sentences for training, the selected corpus has been preprocessed and analyzed morphologically using confessor.

Researchers suggested that since Amharic and Tigrigna are morphologically rich and complex languages; therefore, it is important to conduct the experiment using units of words and morphemes. The software they have used for data alignment was multi-threaded Giza (MGIZA).

The translation unit used in this work is words or morphemes. A sentence with a pair of a maximum of 80 words per sentence is selected without considering special characters that should not be translated. Colon, exclamation marks are not removed before translating the sentence. Finally, the result obtained in this work is promising and serves as proof that it is possible to have an SMT system for implementing a translation system for a pair of local languages.

The BLEU score result of the translation from Tigrigna-Amharic and Amharic-Tigrigna is 6.65 and 8.25 respectively. The translation result is increased to be 13.49 and 12.93 for Amharic-Tigrigna and Tigrigna-Amharic respectively using morphemes as units for

Amharic and Tigrigna. Those researchers said that they found a promising result on their experiment and they are working on post-editing to enhance the performance of the bi-lingual Amharic-Tigrigna translator.

Amharic-to-Tigrigna MT Using a Hybrid Approach

In this work, (Gebremariam, 2017) used a hybrid approach i.e., combination of SMT and RBMT. A syntactical reordering approach is proposed to align word order in the source language so that a similar order of words in the target language is achieved. A unidirectional language model is developed in the research to assign a probability that a given input sentence in the source language generates an output sentence in the target language. Two major experiments carried out by (Gebremariam, 2017) came out with BLEU results of 7.02% using the statistical machine translation approach and 17.47% using the hybrid machine translation approach.

Bidirectional Tigrigna – English SMT

This work was conducted in 2017 by(Mulubrhan, 2017), which aimed to investigate the development of the Tigrigna - English bidirectional machine translation system using a statistical approach. The main aim of the work was to design a bidirectional Tigrigna to English machine translation. So, to achieve their aim, they prepared a corpus collected from different domains and classified into five sets of corpora and arranged a format appropriate for use in the development process.

He has conducted three sets of experiments: baseline (phrase-based machine translation system), morph-based (based on morphemes obtained using the unsupervised method), and post-processed segmented systems (based on morphemes obtained by post-processing the output of the unsupervised segmenter). A free statistical machine translation framework called MOSES which allows automatically training translation model using parallel corpus was used.

The translation evaluation metric used was BLEU, according to their experimental result, the BLEU score of 53.35 % for Tigrigna – English and 22.46 % for English – Tigrigna translations.

Finally, (Mulubrhan, 2017) recommended that “future research should focus to further improve the BLEU score by applying semi-supervised segmentation to include the remaining linguistic information”.

Ge'ez to Amharic MT

Ge'ez to Amharic MT was done by(Mulugeta, 2015), using a statistical machine translation approach. The research came with the aim of addressing Amharic speakers getting knowledge decoded in Ge'ez using automated translation techniques is mandatory. The methodology used was a qualitative experimental method to examine the effect of variables such as normalization, corpus, and test segmentation options of SMT results.

The data used for the experiment were found both from the internet and prepared manually. The data collected by(Mulugeta, 2015) was in a different format so to make those different formats suitable for the experiment, the researcher merges all documents to Ms.-Word format and aligns to verse/sentence level, cleaned for noisy characters, and converted to plain text in UTF-8 format.

As described by (Mulugeta, 2015), "*the bilingual data used for the experiment includes the Old Testament Holy Bible, Genesis, Exodus, Leviticus, Numbers, Deuteronomy, Joshua, Judith, Ruth and Psalms and some religious books like Wedase Mariam and Arganon were used*". The number of parallel sentences, 12, 860 for both Ge'ez and Amharic languages.

As for data organization, bilinguals, 90% for training, and 10% for the testing were used for the experiment. GIZA ++, IRSTLM, Moses decoder, and BLEU were used to it building a translation model, a linguistic model, an alignment of words, and a Ge'ez evaluation for the Amharic MT system, respectively. The parallel corpus used for the experiment was aligned at the sentence level.

As stated by the researcher, the translation result was high when test data was taken from psalm as a complete low when the testing data contains sentences from the praise of Saint Mary and part of the Bible using 10-fold cross-validation. The results show inconsistency.

After experimenting, had an average accuracy of translating the BLEU score of 8.26. Using a sufficiently large parallel Ge'ez-Amharic corpus language and collection synthesis tool, it is possible to develop a better translation system for language pairs.

Finally, they recommended that Ge'ez and Amharic are related, but morphologically complex and limited research were performed in the morphological segmentation and

in the synthesis of the two languages. The development of language morphological synthesizers and segmentation tools can help improve performance. The researcher recommends the extension of this research using the different morphological and synthesis mechanisms. Effectively handling the morphological complexity of both Ge'ez and Amharic languages at the same is a drawback. Application of NMT approach will resolve this complexity by simply training a single end-to-end system with a sufficient amount of parallel corpus of the source language and the target language.

Morpheme-Based Bi-directional Ge'ez to Amharic MT

Bidirectional Ge'ez to Amharic MT using Morpheme-Based approach was done by (Kassa, 2018). It is the second Ge'ez to Amharic MT next to(Mulugeta, 2015). As stated by (Kassa, 2018), the aim of his research was to address the problem observed in the manual translation are time-consuming, resource-intensive, and linguistic knowledge language is required. Machine translation, while having its challenges, can improve performance and reduce costs. Although there are advances in the application of MT for different language pairs, is still in its initial stage for local languages. He also reviewed different approaches of MT, the morphological structure of both Ge'ez and Amharic languages.

The tools used are MGIZA++, which is a software-based on the famous word-alignment software GIZA++, and morfessor for word-level and BLEU for evaluating the result.

Two experiments were conducted in his study, using unsupervised segmentation and rule-based segmentation. The BLEU score of the dataset prepared by using unsupervised segmentation was 14.54% and 14.88% from Ge'ez to Amharic and from Amharic to Ge'ez respectively.

The BLEU score of the dataset prepared by using rule-based segmentation was 15.14 and 16.15 from Ge'ez to Amharic and from Amharic to Ge'ez. They recommended that the alignment of the Ge'ez to Amharic text is a challenging task because of many to many correspondences between words/morphemes of the two languages. Therefore, it is necessary to identify the ideal alignment for the Ge'ez to Amharic Machine translation. The alignment challenge caused by the many to many correspondences

between words of both languages is not an issue in applying an NMT approach because a contextual meaning of words in a sentence is extracted using an attention mechanism.

2.4.2. International NMT works

Amharic – Arabic Neural machine translation

An Amharic to Arabic machine translation using NMT approach is done by (Gashaw, 2016), according to researchers said there are very few researches on Amharic to Arabic translation due to the luck of parallel corpus. They have developed attention-based encoder decoder NMT models, two Long-short-term memory (LSTM) based and Gated recurrent unit (GRU) based. Researchers used an OpenNMT to design the translation model, and they said that the translation done by LSTM out performs that of GRU and Google translate systems which is 12% BLEU.

Researchers have listed some of the challenges of Amharic to Arabic translation, a), since both languages are morphologically rich, lack of machine-readable lexicons. b), the absence of capitalization in Arabic and Amharic languages makes it hard to identify proper nouns, titles, acronyms, and abbreviations. Arabic sentences are usually long, and punctuation has no or little effect on the interpretation of the text. c) Standard preprocessing techniques such as capitalization, annotation, and normalization cannot be performed on Amharic and Arabic languages due to issues of orthography.

CHAPTER THREE

3. METHODOLOGY

This thesis follows experimental research which requires data preparation, tool selection for constructing a translation model, and evaluation of the performance of the model.

In this chapter of the document, we first briefly describe how our translation model with attention mechanism works with the generic encoder-decoder architecture of the NMT system (Figure 3:2 and Figure 3:4), and components included in the architecture are also discussed separately.

After the model description, we presented the data collection methodology, data preprocessing, the tools used for our experimental purpose using our proposed model, and finally, we evaluate the result of the experiment using an evaluation tool called BLEU.

3.1. Ge'ez-Amharic NMT Model with attention

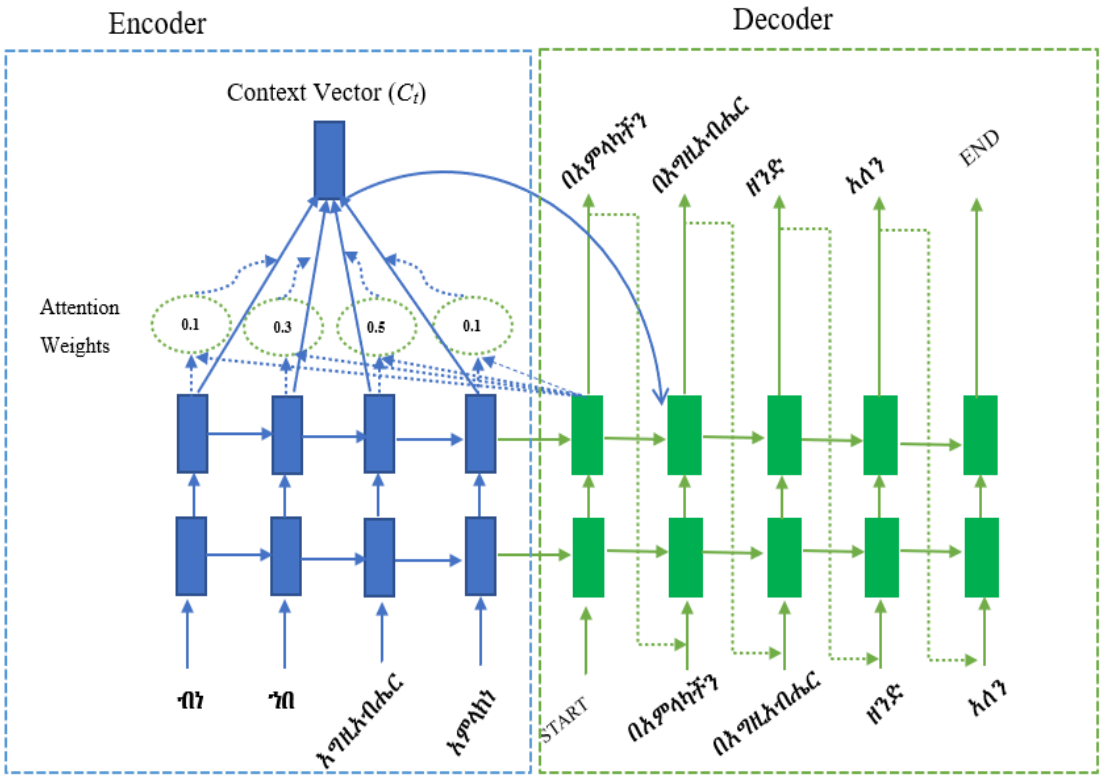


Figure 3:1 Model of Ge'ez-Amharic NMT

A brief description of the architecture of our neural network-based translation model is provided below. The translation model contains a 2-layer LSTM network with 500 hidden layers on both the encoder and the decoder side. This Ge'ez – Amharic model is different from other translation models proposed by previous Ge'ez-Amharic translations in that it is a single end-to-end model that is trained to translate input sentence into its corresponding target sentence without having to construct multiple independent components such as language model, translation model, and others.

As shown in Figure 3:1, our translation model contains two main components, the encoder, and the decoder. The Ge'ez sentence (ብነ ኀበ እግዚአብሔር አምላክነ) is fed into the LSTM based RNN (blue color) of the encoder as source sentence. Then the encoder RNN computes the representation of the input sentence in the form of hidden states (pool of source states) in the different layers of its network. The decoder RNN (green color) then generates the Amharic sentence or target sentence (በአምላካችን በእግዚአብሔር ዘንድ አለን) one word at a time by looking back into the pool of source states in the encoder every time it generates a word.

As will be described in the *attention mechanism* section of the document, a fixed-dimensional representation of the input sentence is a problem when translating longer sentences. On the other hand, using the attention mechanism will help us to make the entire pool of source states available to the decoder in the translation process.[1]

When translating any particular word, the decoder needs to work out which one, out of the pool, it wants to draw from. So effectively, the pool of source states is a random-access memory also known as LSTM which the neural network is then going to be able to retrieve as needed when it wants to do its translation. Attention for neural machine translation is one specific example of this. The attention model specifically tells which part of the source sentence to translate next. To generate the next word, we need to use the hidden states from the LSTM cells in the encoder (blue part of Figure 3:1) and do a comparison with the hidden state of the previously generated word in the decoder (green part of Figure 3:1). Upon comparison, each component of the LSTM is scored with an attention function shown in equation 3.3.

[1] Most commonly, the highest level of hidden state in the encoder RNN is considered for attention.

$$a_{t(s)} = \frac{e^{score(h_t, \bar{h}_s)}}{\sum_s e^{score(h_t, \bar{h}_s)}} \quad 3.1$$

We then do a combined representation of all the memories weighted by the score which comes to be the alignment weights or *attention weights*. A SoftMax function can easily do this for us (equation 3.1). At each time step t, the model computes a variable-length attention weight vector a_t from the previous hidden sate $\boldsymbol{h_{t-1}}$ and all hidden states of the source $\bar{\boldsymbol{h}}_{\boldsymbol{s}}$. A context vector c_t is computed as a weighted sum of all source states (see equation 3.2).

$$c_t = \sum_s a_t(s)\bar{\boldsymbol{h}}_{\boldsymbol{s}} \quad 3.2$$

To put a score for each LSTM in the encoder, we used a bilinear attention function shown in equation 3.3, where $\boldsymbol{h_t}$ and $\bar{\boldsymbol{h}}_{\boldsymbol{s}}$ are the decoder and encoder hidden states respectively and $\boldsymbol{W_a}$ is a matrix that determines how much weight we want to put on the **dot product** of the two vectors.

$$score(h_t, \bar{h}_s) = \boldsymbol{h_t}^{\boldsymbol{T}}\boldsymbol{W_a}\bar{\boldsymbol{h}}_{\boldsymbol{s}} \quad 3.3$$

Encoder-Decoder Model of NMT

At the very early stages of NMT, multilayer perceptron neural network models were used for translation where fixed-length input sentence is taken and output sentence of the same length is produced. These multilayer perceptron models have been hugely improved by using recurrent neural networks embedded into encoder-decoder architecture making it possible to use variable-length input and output sentences.

The architecture of NMT contains a network of encoders and decoders, where the encoder part of the network encodes a source sentence into a fixed-size vector from which different target translations can be made. The decoder part of the network takes the fixed-size vector of the encoder then generates a translation in the target language. Both the encoder and the decoder are jointly trained to maximize correct translation probability(Bahdanau et al., 2015).

To generate the fixed-size vector, the encoder RNN (Figure 3:2) reads the input sentence as a sequence of vectors $X = (x1, \cdots, x_{T_x})$ and convert it into a vector c such that:

$$h_t = f(x_t, h_{t-1}) \quad 3.4$$

And

$$c = q(\{h_1, \cdots, h_{T_x}\}) \quad 3.5$$

where h_t is a hidden state at time t and c is a vector generated from a sequence of hidden states which we use as an input sentence representation.

The decoder is then trained to predict the next word y_t, given the vector c and all the words predicted previously (equation *3.6*). The decoder, on the other hand, computes probability over translation y partitioning the joint probability into ordered conditionals.

$$p(y) = \prod_{t=1}^{T} p(y_t \mid \{y_t, \cdots, y_{t'-1}\}, c) \quad 3.6$$

The fixed-length vector generated by the encoder part of the neural network introduces an issue into the field of machine translation, being a bottleneck for the performance achievement of the encoder-decoder architecture. It makes it difficult for the neural network to deal with long sentences, mainly for sentences that are longer than sentences in the training data.

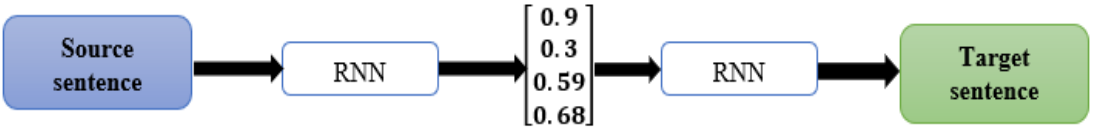


Figure 3:2 Architecture of NMT

An extension of this encoder-decoder architecture that learns to jointly translate and align (attend) was introduced by(Bahdanau et al., 2015) as a solution for the difficulty. This, extended, architecture of the encoder-decoder, searches for parts of the source sentence that provided relevant information while generating the target word at each time step. Generally, the issue of long-term dependencies among words of a sentence

is resolved with a mechanism known as the *attention mechanism* (discussed in the following sections of the document in detail).

The basic idea behind the NMT system is to predict the target language string Y = (y1, ..., yt) for a particular source language string X = (x1, ..., xs). It is a conditional distribution modeled with an RNN-based encoder-decoder architecture. The encoder extracts a variable-length sentence from the source language and converts it to a fixed-length vector. This constant vector is called the **sentence embedding** and contains the meaning of the inserted sentence. The decoder then takes care of recording that sentence and starts guessing the words in the output, taking into account the context of each word as input.

A simplified example of a Ge'ez to Amharic machine translation: "ዕንዘ አልቦ ነገር ዘ ይስአኖ ለ እግዚአብሔር "is encoded into numbers 251, 3245, 953, 2,552, 2388, 3. The numbers 251, 3245, 953, 2,552,2388, and 3 are input into a neural translation model and results in output 2241, 9242, 98, 6342, 2241 is then decoded into the Amharic translation "ለ እግዚአብሔር እሚሳነው ነገር የለም" (each number in the input and output represents a word in the Ge'ez and Amharic dictionary and are always encoded and decoded accordingly).

Recurrent Neural Networks

The state-of-the-art algorithm used by Google Voice Search and Apple Siri for sequential data is RNN[2]. It is the first algorithm that remembers the input due to internal memory, which is perfect for machine learning problems involving sequential data. It is one of the algorithms behind the remarkable results that have been observed in deep learning for the past few years. The most powerful and robust neural network algorithms which are guaranteed to be the only neural networks with internal memory are RNNs.

RNNs were first created in the 1980s, but only in recent years, we have seen their true potential. We have a lot of information that we have to deal with now and then. A long LSTM, developed in 1997, computing power brought RNNs to the forefront.

LSTM

Long-term memory is one of the topologies of recurrent artificial neural networks. Unlike basic repetitive artificial neural networks, it can learn from its experience to process, classify, and predict time series with very long delays of unknown magnitude between important events. Because of this, long-term memory outperforms other recurrent neural networks, hidden Markov models, and other sequencing methods. The artificial neural network with long-term memory consists of blocks of long-term memory that can memorize the value for a certain period of time. This is achieved with gates that determine when the input is meaningful enough to remember when to remember or forget, and when to display the value(Zakaria et al., 2014)(Rodvold et al., 2001).

Because of their internal memory, RNNs can remember important things about the input they received, this allows them to be very `accurate` in predicting what will happen next. RNNs also is known as recurrent because they perform the same task for every element of a sequence, with the output being dependent on the previous computations (equation.*3.4*).

These are networks in which information is transmitted from the previous information go to the next step. It may be considered multiple copies of the same network, each message transfers from one successor to the next. This chain-like structure implies that RNNs can be associated with sequences or lists and have thus found a variety of applications like speech recognition, language processing, and machine translation (sentiment classification, image caption, and language translation).

RNNs take sequence x as input and produces a sequence y as output. The critical factor is that the output's vector is influenced not only by the corresponding input but by the whole sequence of inputs that have been fed in. RNNs can be shaped as a form of repeating modules of a neural network. The repeating module contains a single simple structure as shown in Figure 3:3 ,(Agnihotri, 2019). [2]

[2] The architecture in Figure 3:2 can be modeled with other alternatives of RNN such as long Short-Term Memory (LSTM), Gate Recurrent Unit (GRU) or bidirectional RNN.

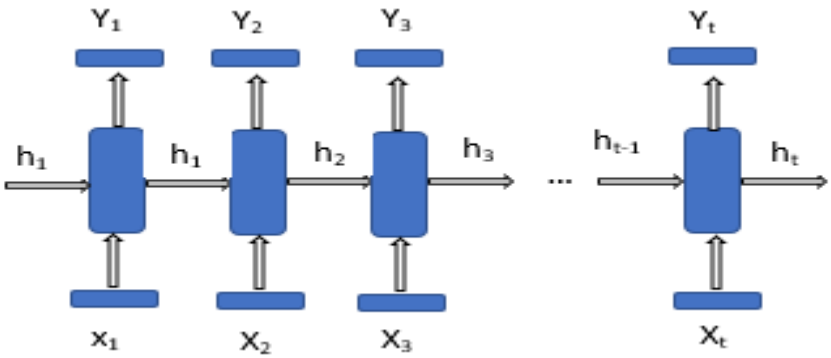


Figure 3:3 Architecture of RNN

where: $x_1, x_2, x_3, \ldots, x_t$ represent the input words from the text, $y_1, y_2, y_3, \ldots, y_t$ represent the predicted next words, and $h_0, h_1, h_2, h_3, \ldots, h_t$ hold the information for the previous input words. Basically, for our MT model, we use a sequence-to-sequence model consists of two recurrent neural networks: an encoder that processes the input and a decoder that produces the output.

A broader description of the encoder-decoder model

The encoder, as its name indicates, encodes the input sequence into a fixed-size vector. The decoder takes the vector as input and produces a translation after processing the vector(Agnihotri, 2019).

Encoder

The task of the encoder is to convert the given source sentence into numbers, more specifically vectors (Figure 3:2) and metrics which are just an assortment of numbers representing data since computers do not understand sentences as humans do. The input words are processed using RNNs resulting in hidden states (equation *3.4*). The hidden states encode each word with all the preceding words i.e., left context. An RNN that processes the words from right to left is also built to produce the right context, generally having two RNNs running in two different directions. This is called bidirectional RNNs.

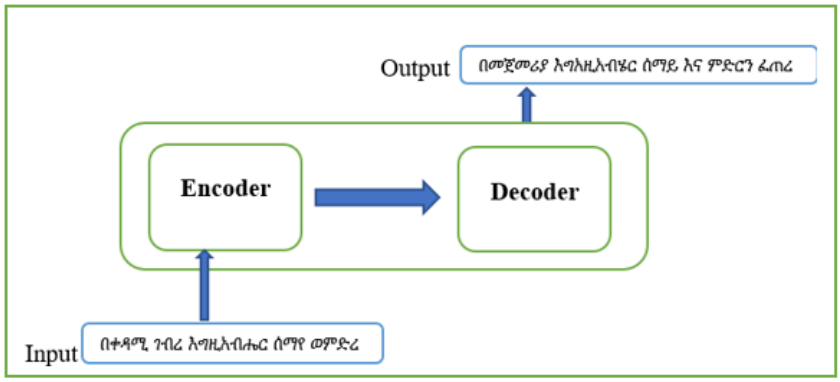


Figure 3:4 The encoder-decoder model

The encoder Architecture

The encoder architecture is built using multiple (deep) LSTM layers which can learn in multiple ways.

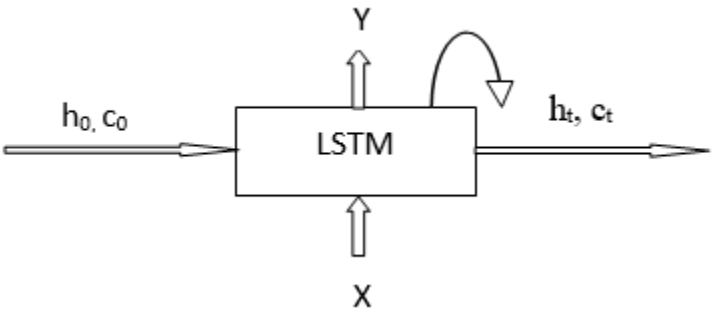


Figure 3:5 Single LSTM Unit

However, for simplicity purpose, let us describe how a single LSTM unit works. A single LSTM (Figure 3:5) unit receives three parameters as input and produces three outputs. X is a word from the input sentence, h0 and c0 are hidden state and vector state respectively.

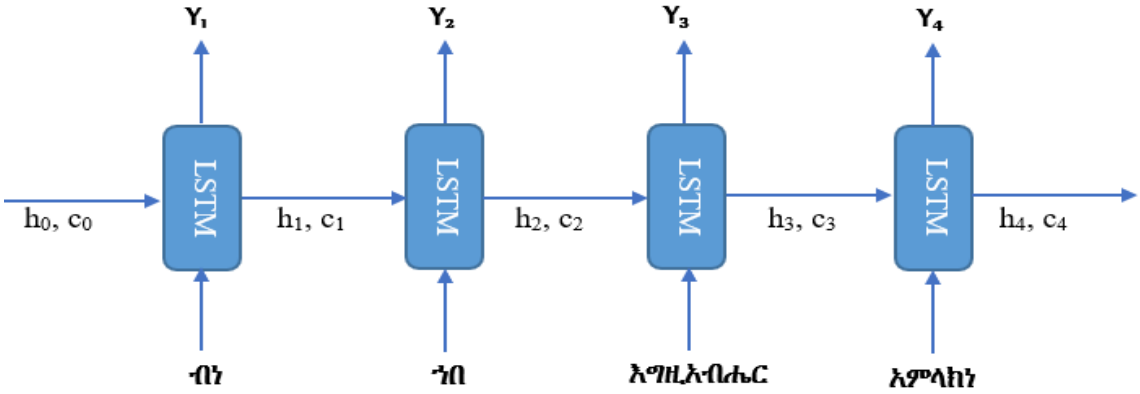


Figure 3:6 Model of encoder

The first input is a word from the sequence which we want to convert and the other two inputs are two vectors, cell state, and hidden state. Let us consider the sentence "**ብነ ጎበ እግዚአብሔር አምላክነ**" as our input sequence, and the encoder LSTM will process each word in the input sequence at every time step.

The word "**ብነ** ", is the input word which is represented by x1, the input state vector i.e. h_0, c_0 are randomly initialized at the beginning. The output vector y1, the state vectors h1, and c1 are the output of the given inputs. The state vectors, h1and c1 have the information of the previous work "**ብነ** " which is inserted at time step t_0. Then at time step t_1, the next LSTM receives the state vectors h1, c1, and the next word from the input sentence "**ብነ** " as input. And similarly, the next LSTM receives the state vectors h2, c2, and the next word from the input sentence "**ጎበ** " as input.

The third LSTM also takes the state vectors h3, c3, and the next word from the input sentence "**እግዚአብሔር** " as input. So, at the last time in step 5, the last state vectors, h4, and c4 have the information of the whole input sentence, "**ብነ ጎበ እግዚአብሔር አምላክነ**", so we don't need the vectors y1 to y5, we only need the output state vectors since these contain the information of the entire input sentence. So, there is no random input, instead, the encoder LSTM initializes the decoder with h4, and c4, these are the last vector states of the encoder. The reason behind this is the decoder should have the idea of the given input sentences to decode it.

Decoder

The decoder part of the NMT is also an RNN, which takes the contextual representation of the input sequence with the prediction of the previous hidden state and output word, and it generates the prediction of the new hidden state of decoder and output word.

The LSTM encoder and decoder of NMT has similar architecture. But they have different input and output(Koehn Philipp, 2017).

The decoder's h0 and c0 are not random input, rather they initialized with the final hidden states of the encoder these are h4 and c4. So, here in the decoder, we have to add the symbol _START_ at the beginning of the target word and the symbol _END_ at the end of the target sentence. Finally, the last row will be "_START_ **በአምላካችን በእግዚአብሔር ዘንድ አለን** _ END_".

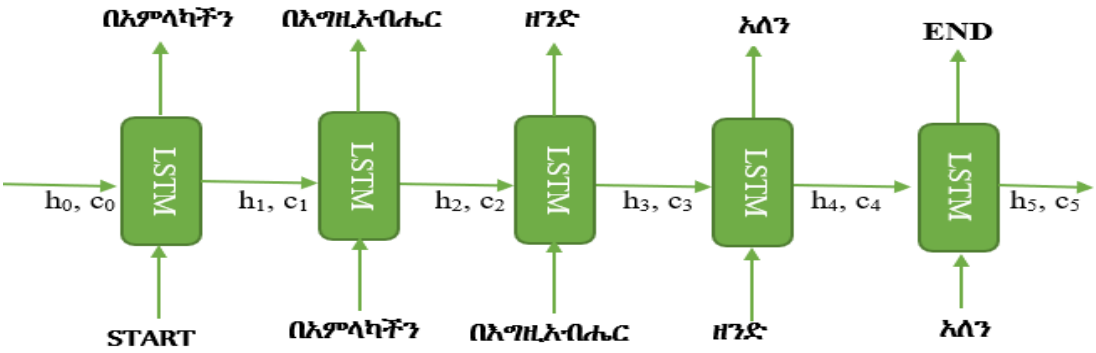


Figure 3:7 Model of decoder

Where h0 of the decoder is h4 of the encoder and c0 of the decoder is c4 of the decoder, x1 is the symbol _START_, and y1 is the first word in the target sequence. The state vectors h1 and c1 will be inserted into the LSTM decoder in the next time step. The output y1 is the essential certainty of the decoder. This task will be done until the model gets the symbol, _END_. Finally, at the last time step, we only need y's for output and we just rejected the final state vectors of the decoder. The training architecture of both the encoder and decoder is illustrated in Figure 3:8.

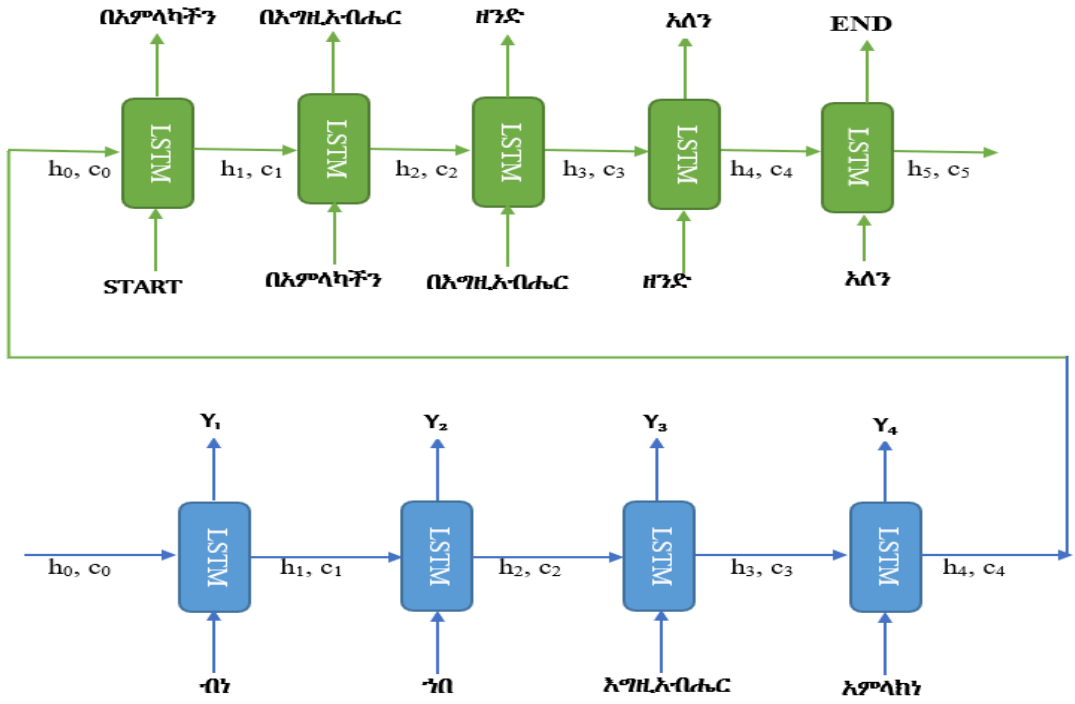


Figure 3:8 Encoder-Decoder architecture

Attention mechanism

The main disadvantage of the sequential encoder-decoder model for sequencing recurrent neural networks is that it only works with short sequences. The encoder model finds it difficult to remember long sentences and convert them into fixed-length vectors. Besides, the decoder receives only information that is the last hidden state of the encoder. Therefore, it is difficult for the decoder to summarize a large input sequence at once. So, to solve this problem, we have to use the attention mechanism.

Capturing semantic details of very long sentences using fixed-size vector is difficult. An effective approach that resolves this difficulty is reading the whole sentence at once and focusing on different parts of the source sentence that contain relevant information on the word being predicted. This architecture is called encoder-decoder RNN with attention. This architecture the core of Google Neural Machine Translation, or GMNT which they used in their "*google translate*" service.

The first attention mechanism was created by (Jia, 2019) in 2015. Attention is probably one of the most powerful concepts in the field of deep learning today. It is based on a common sensory intuition to which we "attend to" a certain part while processing a large amount of information.

The way the human mind concentrates on important facts when assessing a situation, the attention pattern focuses on the most relevant segments and speeds up the processing of large amounts of data. The attention model can extract the most important words, even if the sentence is long and complex. Most of the state-of-the-art neural translation systems employ attention mechanism.

Generally, an encoder which computes a representation *c* for each source sentence and a decoder which generates translation one word at a time work based on conditional probability as stated in Eq. *3.6*. This conditional probability equation can also be written as:

$$log\, p(y|x) = \sum_{t=1}^{T} log\, p(y_t|y < t, \boldsymbol{c}) \tag{3.7}$$

$$p(y_t|y < t, c) = \boldsymbol{softmax}\big(g(h_t)\big) \tag{3.8}$$

h_t is modeled as in Eq. (*3.4*).
Our training model is then described as:

$$J_t = \sum_{(x,y)\epsilon D} -log\, p(y|x) \tag{3.9}$$

D is parallel training corpus and *g* is a function that outputs a vocabulary size vector, *h* is RNN hidden unit, and *f* in equation 3.4 *c*omputes the current hidden state given the previous hidden states. The *SoftMax* function normalizes the vocabulary size vector into values between 0 and 1.

Eventually, NMT directly calculates:

$$p(y|x) = p(y_1|x)p(y_2|y_1, x)p(y_3|y_1, y_2, x) \cdots\ (y_T|y_1, \cdots, y_{T-1}, x) \tag{3.10}$$

There are generally two types of attention mechanisms, global and local attention. The global attention mechanism is the one in which at each translation step, all the source hidden states of used for context vector computation. Whereas the local attention mechanism is one in which only some of the hidden states of the encoder are used for context vector computation.

NMT has become so popular because:

- NMT achieved the most advanced performance on large-scale translation tasks, such as English to French / German.
- NMT requires a little knowledge in the field and is a simple concept.
- NMT does not store huge phase tables or language models, which results in a small memory footprint.
- NMT can generalize very long word sentences with the help of the attention mechanism.
- The design of the model is much simpler than the SMT

NMT produces results that significantly reduce the total editing performance on the best phrase-based machine translation (PBMT) system. It surpasses PBMT systems in all meaningful lengths, although production length deteriorates faster than its competitors. Additionally, its output contains fewer morphology errors, fewer lexical errors, and substantially fewer word order errors. In Table 3:1, a comparison of NMT and SMT is provided (Jia, 2019).

Table 3:1 Comparison of SMT and NMT

	SMT	NMT
Time	Consumes a lot of time	Takes less time
Memory	Needs more memory space	Need less memory space
Accuracy	Less accurate because sentence divided into subsentences in translation.	More accurate translation
Simplicity	Complex because two models, the language model, and the translation model are used.	Simple because a single sequence model is used.

3.2. Data collection

For experimental purposes, a corpus of 50k parallel sentences is collected from religious institutions and other related sources such as Ethiopian Orthodox Tewahdo church religious books[3] including the holly bible i.e., old, and new testaments, wudasie Maryam, Drsane Gebriel, Drsane Michael. We divided the collected data into training, validation, and testing set as shown in Table 3:2. Generally, proportion is selected to make the training data set relatively large for the different experiments we conducted. The rest of the proportion is used for validating and testing our model. The data used for validation is used to check whether the training converges after certain number steps or not. The amount of data used in the validation and testing is almost similar except in our second experiment which contained 2k sentences for validation and 1k sentences for testing.

Table 3:2 Size of the total data used in the study for the different experiments

Experiment	Training data		Validation data		Testing data		Total	
	Geez	Amharic	Geez	Amharic	Geez	Amharic	Geez	Amharic
1	46k	46k	2k	2k	2k	2k	50k	50k
2	47k	47k	2k	2k	1k	1k		
3	48k	48k	1k	1k	1k	1k		
4	48k	48k	1k	1k	1k	1k		
5	48k	48k	1k	1k	1k	1k		

3.3. Tools

There are a number of available licensed user-friendly tools and frameworks for NMT, some of which are openNMT, AMUNMT (neural Monkey). Specifically, we are using openNMT for our purpose.

OpenNMT

Neural machine translation uses an open-source toolset known as OpenNMT. Its two state-of-the-art deep learning foundations are OpenNMT-py (implemented in PyTorch), and OpenNMT.tf (implemented using TensorFlow). Due to its highly customizable model architectures and efficient training procedures, we prefer

[3] www.ethiopicbible.com

OpenNMT-py to OpenNMT-tf. This open-source translation toolkit also includes many modules such as CTranslate2 (inference engine), Tokenizer (C++ tokenization library), and a docker interface (for training and translating using docker containers) to facilitate the overall translation procedures. The fact that NMT systems take from days to weeks to train is a factor for considering the training efficiency in the design of OpenNMT-py.

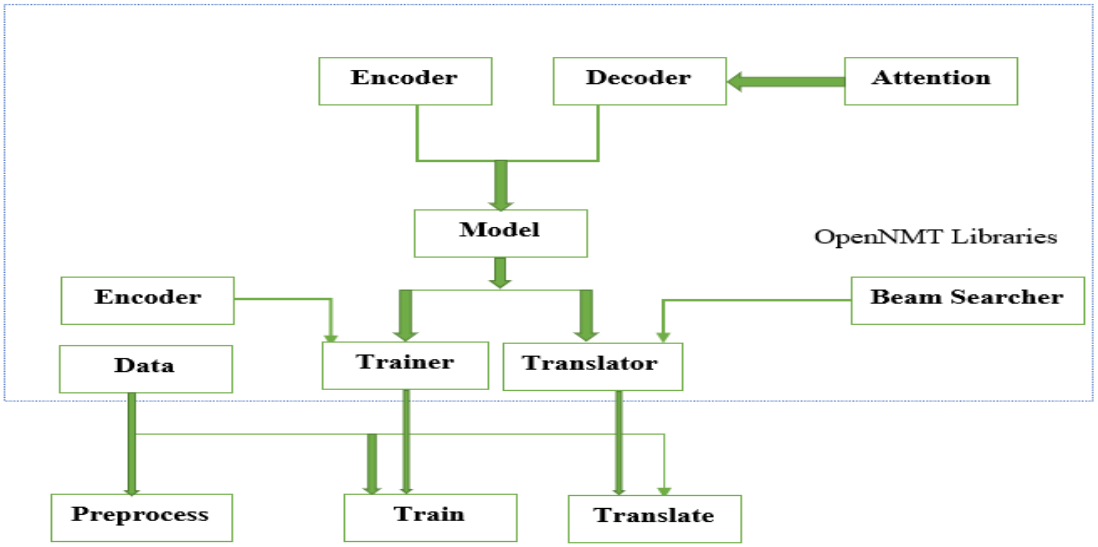


Figure 3:9 Schematic Architecture of OpenNMT-py

As it can be seen in Figure 3:9, simple interfaces such as preprocessing, translate, and train are provided so that any regular user can interact with the toolkit by simply providing source and target files as input through the interfaces. OpenNMT also implemented different types of attention functions like the one in equation 3.3 . Other attention functions such as dot and concatenation are also implemented.

Programming language

We used python programming language because it is the language that one only must understand the technical nuances of the language. Besides, Python is also extremely efficient and it allows the developers can complete more work using fewer lines of code. Python code is also easily recognizable by humans, making it ideal for machine learning models.

Evaluation tool

OpenNMT provides the baseline performance of the output measurement parameters like BLEU, TER, DLRATIO. All measured values can be used as validation values during training or independently with tools. BLEU is the most used quality review at MT Systems. It is an evaluation tool that measures the quality of translation.

Quality is thought to be the similarity between a machine's output and a human. BLEU is still dominant among other parameters because the language is free, very quick, and has proven to be the best metric for tuning PBSMT models (Shterionov et al., 2018). BLEU measures the accuracy of the MT system calculated by the results of the system and by pure precision and normalized human meanings.

BLEU algorithm takes the n-gram (typically, n∈ {1, ..., 4}) precision of a candidate translation compared to a reference translation. Calculates the number of n-grams to match, and calculates the average weight. That is, the more n-gram matches between a translation and the reference, the higher the score. The higher values of n represent the word order and the lower values of n capture lexical coverage of the translation.

BLEU scores can be computed either at a document level or at a sentence level(Almansor & Al-Ani, 2018) (Papineni et al., 2015) . They range between 0 or 0% is the lowest quality and it is completely irrelevant to reference and 1 or 100% refers to the highest quality means the same as a reference.

Since BLEU score is the comparison of human translation and machine translation, to calculate the BLEU score we have to prepare two files: 1) Reference which is our target dataset, i.e., human translation and 2) System which is prediction, generated by the machine translation model for the source of the same test dataset used for “Reference”.

CHAPTER FOUR

4. EXPERIMENT, RESULTS AND DISCUSSION

4.1. Experiment

The open-source toolkit we used for our translation experiment is OpenNMT[4]. We used Colab[5] to train our model. Colaboratory, or "Colab", is a product of Google that helps users to write and execute python code using the browser and it is principally suited for data analysis, machine learning.

We conducted five experiments with different subset of the collected data used for each experiment with different number of epochs. As shown Table 3:2,our first experiment used 46k pair of sentences of the collected data a training set, 2k pair of sentences as a validation set, and 2k pair of sentences as a testing set. In this experiment, the model is trained on 10,000 training steps. Our second experiment also used 47k pair of sentences as a training set, 2k pair of sentences as a validation set, and 1k pair of sentences a testing set with 20000 epochs used in the training step. In our third experiment we used 48k pair of sentences as a training set, and the rest 2k pair of sentences are used as validation and testing sets 1k sentences for each. The number of training steps used in this experiment is 50,000. The data set up for our fourth experiment is the same as that of the third experiment but the number of training steps used is 100,000. But in the fifth experiment we try using three layers of LSTM, but at some time the system collapses, since it needs, more GPU. Basically, we have followed the following major steps during the experiment:

Cleaning the collected data

The collected data is sorted in such a way that the sentences in both languages are aligned and noisy symbols such as special characters are removed. The corpus is split into space-separated words or tokens.

[4] www.opennmt.net
[5] https://colab.research.google.com/

Tokenization

To hurriedly process the new training data, there are tokenization utilities which are accommodated by openNMT. The purpose of tokenization is to break-down the given source sentences into token strings. During the conversion process, there are two main operations will be performed:

- **Normalization:** the tasks in this operation are, applying some similar transformation to the source sequence to identify and protect a particular sequence like URL, simplify characters, for example, types of quotes, Unicode variants into unique representation to make the translation simpler.
- **Tokenization:** it converts the actual normalized sequence of sentences into space-separated token strings. Byte pair encoding is used for tokenization, which is a method that often divides the corpus so that the string is combined; It results in dividing the word surface forms by root word and affix.

Preprocess the text-data

Preprocessing is the first important step for creating a machine learning model, and if we want to work with data, we need to clean it up and place it in an organized way so, it is the process of preparing raw data and applying it to the machine learning model. After preprocessing the data will be converted and encoded, then it will be easy to parsed or interpreted by the machine.

In the preprocessing step, every token is assigned an index, and too long sentences are also filtered so that they are sorted by their length. By keeping the truck of token frequencies vocabulary list will be constructed. A new file is generated in the preprocessing phase from the tokenized corpus which will be used for training.

Train

To reduce the computational difficulty and enhance the speed for the training phase, we did our experiment on Google's cloud virtual machine Colaboratory (Colab) which provides free GPUs for computationally intensive tasks.

Terms used in training

Epoch: it shows the number of times or steps (forward and backward passes) the algorithm sees the entire data set. The input data will be split into batches if the dataset

is large in size. when an algorithm used in training seen all of the prepared samples in the dataset, an epoch will be completed.

There are several parameters to be taken into account when determining the number of epochs, the model needs to perform. Mostly, increasing the number of epochs does not improve accuracy. This is due to **overfitting** or **underfitting** and other issues that can significantly reduce accuracy. Overfitting happened when the model is fed with training data and noise in the training data insofar as it negatively affects the performance of the model. Underfitting means when a model cannot learn training data or generalize any new data provided to it (Afaq & Rao, 2020).

Validation: which is the process used to build our model, it is used to select parameters and avoid overfitting. At each step of the validation, a check is performed whether the training algorithms converge or not.

Translate

After successfully creating and training the model with the preprocessed data, we tested the model to translate 1k source sentences to target sentences. The quality of the translation is then evaluated using a translation evaluation tool BLEU.

Detokenization

The output after the translation differs from the actual sentence structure because it is still a segment after the translation is complete. When we run the detokenization process, it will be returned in the form of the actual statement.

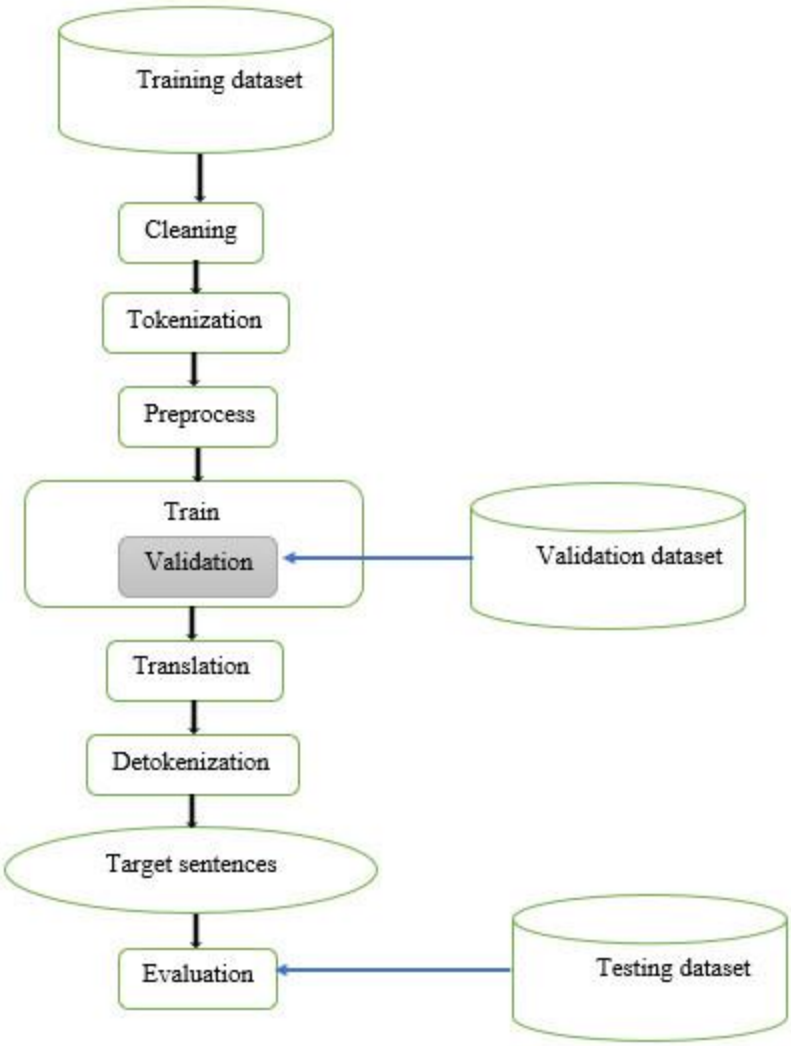


Figure 4:1 Experimental Steps

Evaluating

The quality of our translation model is evaluated by simply comparing the translated sentences and the reference sentences in the target language which are assumed to be the correct meaning of the input test sentences. This process of comparing the sentences is performed using BLEU[6]. The overall steps of our experiments are depicted in Figure 4:1.

[6] BLEU compares the n-gram of the candidate translation with n-gram of the reference translation to count the number of matches

As described in the previous sections of the document, our model was designed to include an LSTM encoder and decoder with an attention mechanism to be able to translate longer sentences and paying attention to context.

The model was trained with four independent experiments with different proportion of the collected and different training parameters. Each of these experiments resulted in unique translation results due to the difference in the training parameters and proportion of the dataset. In the first experiment, the model was trained on 46K pair of sentences of both Geez and Amharic with the rest 2k pair of sentences for validation and the other 2k pair of sentences for testing. The number of training steps (epochs) used in this experiment is 10,000 which took up almost two and half hours to complete training the model. The translation result is 6.06 BLEU. See Appendix D for the translation quality of this experiment. A 5-fold cross validation technique is applied to select the testing data in this experiment.

In our second experiment, we trained the model with dataset that is 1k greater than the first experiment and with the training steps doubled. The validation dataset used for this experiment is the same as the first experiment. The time it took to complete training the model is three and half hours. The BLEU of the translation result came to be as better as 7.71 as compared to the result of the first experiment. See Appendix C for the translation quality of this experiment. A 5-fold cross validation technique is applied to select the testing data in this experiment.

In the third experiment, the training dataset is increased to be 48k while the validation and testing dataset are 1k each. The model is trained with 50,000 epochs which took up approximately four hours to complete training the model. This experiment achieved a translation result of 12.3 BLEU. See Appendix B for the translation quality of this translation. The fourth experiment is quite similar the third experiments except the number of training epoch is doubled to be 100,000. The BLEU of the translation result is 15.4. See Appendix A for the translation quality. A 10-fold cross validation technique is applied to select the testing data in this experiment.

As it can be seen from each of the above experiments, the BLEU of the translation result improves on increasing the number of training epochs. However, this doesn't increase indefinitely. Increasing the number of training epochs indefinitely will never be a remedy for a translation result of lower BLEU. To this end, we performed one

more final experiment with 120,000 epochs and the BLEU of the translation came to be lower than the previous experiment, which is a sign of overfitting.

According to our experiments, the best BLEU score of the translation result obtained for the model trained with different subsets of the 50k Ge'ez – Amharic parallel corpus in OpenNMT is 15.4%. Despite the hungry nature of NMT models for data and the costly available data for the corpus, our model has performed well with this limited amount of dataset. For comparison, the same model was trained with another English and Amharic parallel corpus which is largely and freely available and the translation result obtained exceeded our language pair translation. This shows that the model will efficiently translate Geez to Amharic text provided that it is trained with a sufficient amount of training corpus.

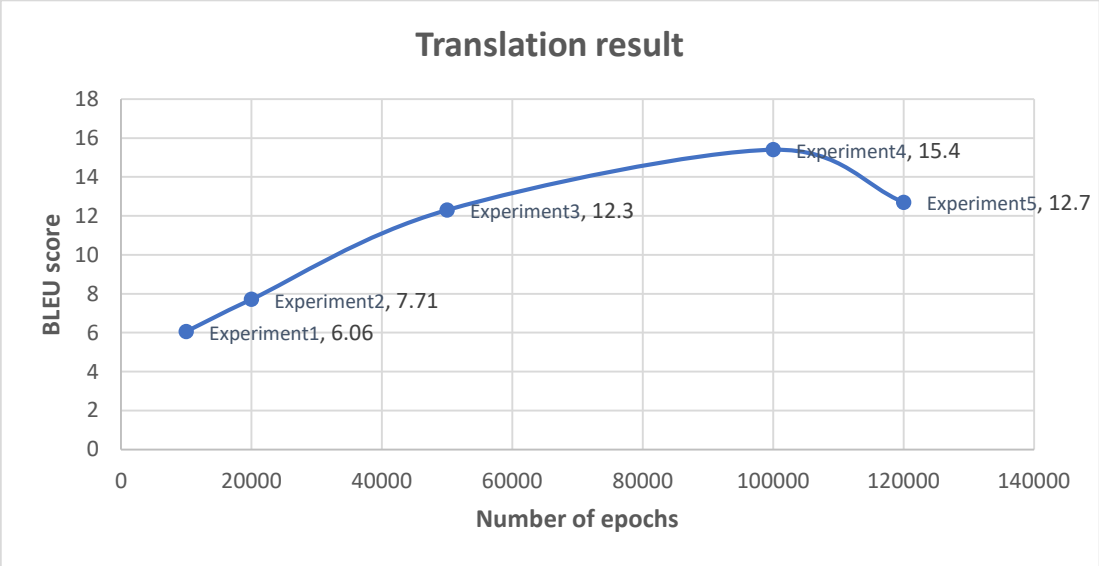


Figure 4:2 Translation result of experiments

Table 4:1 Experimental result analysis of our model

Experiment No.	Training epochs	Training dataset	Validation dataset	Testing dataset	BLEU score(%)	Cross validation fold	LSTM layers
1	10,000	46k	2k	2k	6.06	5	2
2	20,000	47k	2k	1k	7.71	5	2
3	50,000	48k	1k	1k	12.3	5	2
4	100,000	48k	1k	1k	15.4	10	2
5	120000	48k	1k	1k	12.7	5	2

CHAPTER FIVE

5. CONCLUSION AND RECOMMENDATION

5.1. Conclusion

The main concern of this study for applying NMT with attention to translate Ge'ez documents into their corresponding Amharic meaning. Mainly, we used an encoder-decoder architecture with LSTM cells on both the encoder and decoder sides to support longer sentence translation. NMT is the state-of-the-art machine translation with better performance and less memory usage than its predecessors such as RBMT and SMT. And it needs a huge number of parallel corpora of both the source language Ge'ez and the target language Amharic. In addition to the huge amount of data it requires, wisely choosing the training parameters such as training epochs, number of LTSM layers, the size of RNN cells in the hidden layer also makes a great difference at the cost of computational resources. Due to the fact that we are able to access only one GPU, all of our experiments are performed using only two layers of LSTM with 500 hidden units. The number of training steps in training a given NMT model is also a determinant factor. Using too large number for the training step will cause overfitting that lowers the translation quality while using too small number will also result in low quality of translation. Our last experiment used three layers of LSTM with 512 hidden layers to train the model with 100,000 training steps which caused the system to total crash. So, choosing optimized values for the parameters used in training the model is required.

5.2. Recommendations

Our thesis is a machine translation from Ge'ez to Amharic using NMT, So, according to our exploration we would like to recommend the following future works:

- ✓ The accuracy we get in this system is to much lower, the main challenge we faced in this work is finding a parallel corpus for the languages Ge'ez and Amharic so, it makes the result lower. Thus, researchers should work on preparing a huge amount of standardized training datasets for both languages.
- ✓ Due to computational cost, we used only two layers to train our model, it has an impact on translation accuracy to be lower. The researcher recommends use a stack of LSTM layers or increasing the vertical layers of LSTM with highspeed preprocessors like GPU, will improve the performance of the translation.
- ✓ There are a bunch of scanned copies of Ge'ez and Amharic documents which are available on the internet. If most of them are converted into an available format, there will be a better improvement in translation results.
- ✓ The neural network we used for training our data is LSTM, which processes the given input sentence sequentially, that means one word at a time, and backpropagation should be done to catch errors done in forwarding propagation, there are deep learning models under development like a transformer, it is better to overcome these problems. Because transformer process the whole sentence simultaneously, which reduces training time and it computes similarity scores between words in a sentence by itself means self-attention.

REFERENCE

Aadil, M., & Asger, M. (2017). An Overview of Statistical Machine Translation Tools. *International Journal of Advanced Research in Computer Science and Software Engineering*, *7*(7), 289. https://doi.org/10.23956/ijarcsse/v7i7/0201

Afaq, S., & Rao, S. (2020). Significance Of Epochs On Training A Neural Network. *International Journal of Scientific and Technology Research*, *19*(6), 485–488. www.ijstr.org

Agnihotri, S. (2019). Hyperparameter Optimization on Neural Machine Translation. *Creative Components*, *124*. https://lib.dr.iastate.edu/creativecomponents/124/

Alansary, S. (2014). Interlingua-based Machine Translation Systems: UNL versus Other Interlinguas. *The Egyptian Journal of Language Engineering*, *1*(1), 42–54. https://doi.org/10.21608/ejle.2014.59863

Almansor, E. H., & Al-Ani, A. (2018). A hybrid neural machine translation technique for translating low resource languages. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10935 LNAI*(July), 347–356. https://doi.org/10.1007/978-3-319-96133-0_26

Argaw, A. A., & Asker, L. (2007). *An Amharic Stemmer : Reducing Words to their Citation Forms*. *June*, 104–110.

Bahdanau, D., Cho, K. H., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Ballabh, A., & Chandra Jaiswal, U. (2015). a Study of Machine Translation Methods and Their Challenges. *International Journal of Advance Research In Science And Engineering IJARSE*, *8354*(4), 423–429. https://www.ijarse.com/images/fullpdf/320.pdf

Bishop, J. M. (2015). History and philosophy of neural networks. *Computational Intelligence - Volume 1*, *1*(February), 400.

Brown, P. F., Cocke, J., Pietra, S. A. Della, Pietra, V. J. Della, Jelinek, F., Lafferty, J. D., Mercer, R. L., & Roossin, P. S. (2002). *A STATISTICAL APPROACH TO MACHINE TRANSLATION*. *July*.

Chéragui, M. A. (2012). Theoretical overview of machine translation. *CEUR Workshop Proceedings*, *867*, 160–169.

Dave, S., Parikh, J., & Bhattacharyya, P. (2001). Interlingua-based English-Hindi Machine Translation and Language Divergence. *Machine Translation*, *16*(4), 251–304. https://doi.org/10.1023/A:1021902704523

Desalegn. (2015). *Desalegn Asfawwesen*.

Forcada, M. L. (2017). Making sense of neural machine translation. *Translation Spaces*, *6*(2), 291–309. https://doi.org/10.1075/ts.6.2.06for

Gashaw, I. (2016). A -a n m t. *AMHARIC-ARABIC NEURAL MACHINE TRANSLATION*.

Gated, T., Ababa, I. A., & Admassie, Y. (1963). Institute of Ethiopian Studies. *Journal of Ethiopian Studies*, *1*(1), 3–7.

Gebregziabher, M. (2012). *ENGLISH-AMHARIC STATISTICAL*

MACHINE TRANSLATION (1) Addis Ababa , Ethiopia : IT Doctoral Program , Addis Ababa University (2) Grenoble , France : University Joseph Fourier.

Gebremariam, A. (2017). *Amharic-to-Tigrigna Machine Translation Using Hybrid Approach*.

Hakkani, D. Z., Tür, G., Oflazer, K., Mitamura, T., & Nyberg, E. H. (1998). An English-to-Turkish interlingual MT system. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *1529*, 83–94. https://doi.org/10.1007/3-540-49478-2_8

Harper, K. E. (2018). Machine translation. *Soviet and East European Linguistics*, *October*, 133–142. https://doi.org/10.4324/9781315678481-27

Hutchins, J., & Somers, H. (1992). General introduction and brief history. *An Introduction to Machine Translation*, 1–9.

Hutchins, J. W. (2015). Machine translation: History of Research and Applications. *Routledge Encyclopedia of Translation Technology*. http://www.hutchinsweb.me.uk/Routledge-2014.pdf

Hutchins, John. (2005). Example-based machine translation: A review and commentary. *Machine Translation*, *19*(3–4), 197–211. https://doi.org/10.1007/s10590-006-9003-9

Iconic. (n.d.). *Neural Machine Translation | Iconic Translation Machines*. 2018. Retrieved February 1, 2021, from https://iconictranslation.com/what-we-do/neural-machine-translation/

Jia, Y. (2019). Attention Mechanism in Machine Translation. *Journal of*

Physics: Conference Series, *1314*(1). https://doi.org/10.1088/1742-6596/1314/1/012186

Kassa, T. (2018). *Morpheme-Based Bi-Directional Ge'ez -Amharic Machine Translation*. http://etd.aau.edu.et/handle/123456789/18688

Koehn, P., & Knowles, R. (2017). Six challenges for neural machine translation. *ArXiv*, 28–39. https://doi.org/10.18653/v1/w17-3204

Koehn Philipp. (2017). Neural Machine Translation. *Tradumàtica: Tecnologies de La Traducció*, *15*, 66. https://doi.org/10.5565/rev/tradumatica.203

Krenker, A., Bester, J., & Kos, A. (2011). Introduction to the Artificial Neural Networks. *Artificial Neural Networks - Methodological Advances and Biomedical Applications*. https://doi.org/10.5772/15751

Mariam, K. W. (1963). Institute of Ethiopian Studies, Addis Ababa. *The Journal of Modern African Studies*, *1*(3), 383–384. https://doi.org/10.1017/S0022278X00001762

Mulubrhan. (2017). *College of Natural and Computional Sciences School of Information Science*.

Mulugeta, D. (2015). *Geez to Amharic Automatic Machine Translation : A Statistical Approach SCHOOL OF GRADUATE STUDIES COLLEGE OF NATURAL SCIENCES SCHOOL OF INFORMATION SCIENCE GEEZ TO AMHARI*.

Ney, H. (2014). *Éõê*. *June 2001*.

O.Lambdin, T. (2014). *Muhammad , the Qurʾan , and Islam by N . A . Newman Review by : A . Rippin Published by : American Oriental*

Society American Oriental Society is collaborating with JSTOR to digitize , preserve and extend access to Journal of the American Oriental Society. *118*(3), 408–409.

Okpor, M. D. (2014). Machine Translation Approaches: Issues and Challenges. *International Journal of Computer Science Issues*, *11*(5), 159–165. www.IJCSI.org

Papineni, K., Roukos, S., Ward, T., & Zhu, W. (2015). *BLEU : a Method for Automatic Evaluation of Machine Translation B LEU : a Method for Automatic Evaluation of Machine Translation*. *October 2002*. https://doi.org/10.3115/1073083.1073135

Prasad, T. V., & Muthukumaran, G. M. (2013). Telugu to English Translation using Direct Machine Translation Approach. *International Journal of Science and Engineering Investigations*, *2*(12), 25–32.

Rodvold, D. M., McLeod, D. G., Brandt, J. M., Snow, P. B., & Murphy, G. P. (2001). Introduction to artificial neural networks for physicians: Taking the lid off the black box. *Prostate*, *46*(1), 39–44. https://doi.org/10.1002/1097-0045(200101)46:1<39::AID-PROS1006>3.0.CO;2-M

Shilon, R., Wintner, S., Science, C., & Landman, F. (2011). Transfer-based Machine Translation between morphologically-rich and resource-poor languages : The case of Hebrew and Arabic MA thesis submitted by. *Computer*, *March*.

Shterionov, D., Superbo, R., Nagle, P., Casanellas, L., O'Dowd, T., & Way, A. (2018). Human versus automatic quality evaluation of NMT and PBSMT. *Machine Translation*, *32*(3), 217–235. https://doi.org/10.1007/s10590-018-9220-z

Singh, S. P., Kumar, A., Darbari, H., Singh, L., Rastogi, A., & Jain, S. (2017). Machine translation using deep learning: An overview. *2017 International Conference on Computer, Communications and Electronics, COMPTELIX 2017*, *July*, 162–167. https://doi.org/10.1109/COMPTELIX.2017.8003957

Sisay Adugna. (2009). *English – Afaan Oromoo Machine Translation: An Experiment Using Statistical Approach a thesis submitted to the school of graduate studies of Addis Ababa University.*

Techopedia. (2020). *What is an Artificial Neural Network (ANN)? - Definition from Techopedia*. https://www.techopedia.com/definition/5967/artificial-neural-network-ann

Teshome, E. (2013). *School of Graduate Studies College of Natural Science Department of Computer Science Addis Ababa University School of Graduate Studies*. *March*.

Vandeghinste, V., & Van Eynde, F. (2012). Philipp, Koehn. 2010. Statistical Machine Translation. *Target. International Journal of Translation StudiesTarget / International Journal of Translation StudiesTarget*, *24*(1), 157–159. https://doi.org/10.1075/target.24.1.12van

Woldeyohannis, M. M., & Meshesha, M. (2018). Experimenting statistical machine translation for ethiopic semitic languages: The case of Amharic-Tigrigna. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST* (Vol. 244, Issue July). Springer International Publishing. https://doi.org/10.1007/978-3-319-95153-9_13

Xuan, H. W., Li, W., & Tang, G. Y. (2012). An advanced review of hybrid machine translation (HMT). *Procedia Engineering*, *29*, 3017–3022. https://doi.org/10.1016/j.proeng.2012.01.432

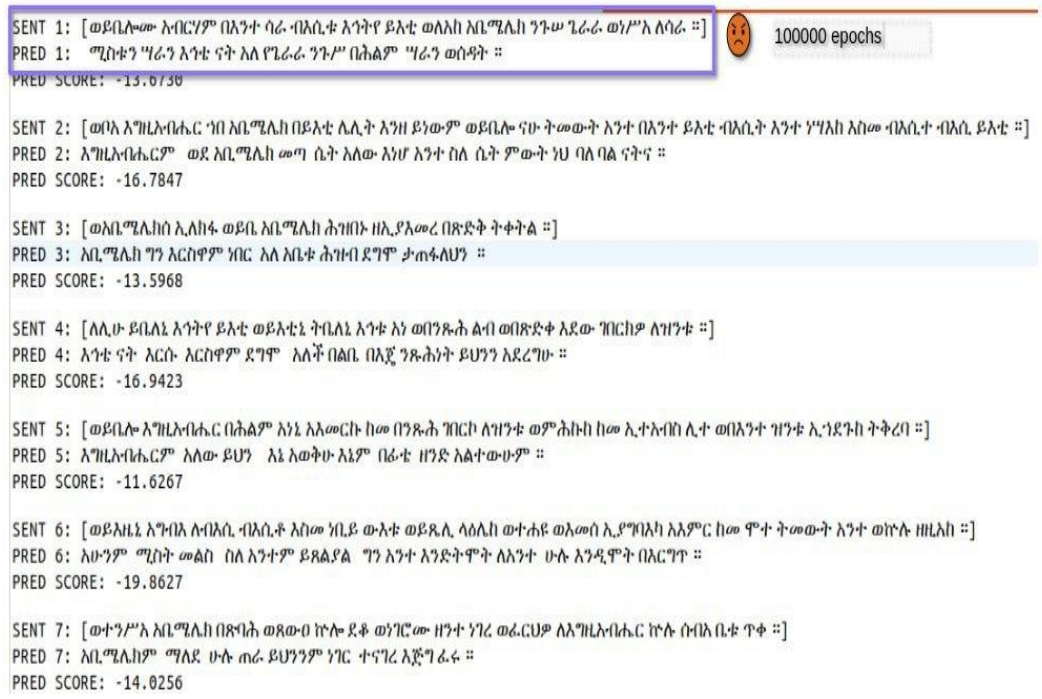
Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing [Review Article]. *IEEE Computational Intelligence Magazine*, *13*(3), 55–75. https://doi.org/10.1109/MCI.2018.2840738

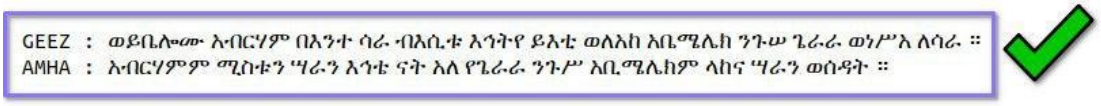
Zakaria, M., Al-Shebany, M., & Sarhan, S. (2014). Artificial Neural Network : A Brief Overview. *Int J Eng Res Appl*, *4*(2), 7–12.

APPENDIX

Appendix A

Translation with 100k training steps





Appendix B

Translation with 50k training steps

Appendix C

Translation with 20k training steps

Appendix D

Translation with 10k training steps

Appendix E

Algorithms for cleaning a dataset

```
Initialize: cleaned

define: regular_expression_for_filtering

remove punctuations

for pair in lines

        define: clean_pair

        for line in pair

                normalize unicode characters

                tokenize on white space

                remove punctuation from each token

                remove non-printable chars form each token

                remove tokens with numbers in them

                store as string

        append to clean_pair

return array(cleaned)
```

Appendix F

Algorithms for tokenization of dataset

```
Define: list of unwanted_character

Define: sentence_string


for i=1 to the number_of_character in sentence_string

        if(sentence_string[i]==unwanted_character)

                remove sentence_string[i];

        end if

end for

string_split(sentece_string)
```

Appendix F

Training Verbose

```
[2021-01-23 15:40:06,214 INFO] Loading ParallelCorpus(NMT/src-train.txt, NMT/tgt-train.txt, align=None)...
[2021-01-23 15:40:10,896 INFO] Step 49200/50000; acc:  69.56; ppl:  3.49; xent: 1.25; lr: 1.00000; 4821/4770 tok/s;  15059 sec
[2021-01-23 15:40:24,131 INFO] Step 49250/50000; acc:  75.05; ppl:  2.67; xent: 0.98; lr: 1.00000; 5377/5054 tok/s;  15073 sec
[2021-01-23 15:40:37,268 INFO] Step 49300/50000; acc:  74.43; ppl:  2.69; xent: 0.99; lr: 1.00000; 5755/5335 tok/s;  15086 sec
[2021-01-23 15:40:51,139 INFO] Step 49350/50000; acc:  74.80; ppl:  2.68; xent: 0.99; lr: 1.00000; 5388/4909 tok/s;  15100 sec
[2021-01-23 15:41:05,728 INFO] Step 49400/50000; acc:  72.53; ppl:  2.97; xent: 1.09; lr: 1.00000; 5388/4908 tok/s;  15114 sec
[2021-01-23 15:41:22,840 INFO] Step 49450/50000; acc:  69.20; ppl:  3.51; xent: 1.26; lr: 1.00000; 5072/5056 tok/s;  15131 sec
[2021-01-23 15:41:35,770 INFO] Step 49500/50000; acc:  73.44; ppl:  2.87; xent: 1.05; lr: 1.00000; 5708/5761 tok/s;  15144 sec
[2021-01-23 15:41:49,976 INFO] Step 49550/50000; acc:  72.70; ppl:  2.99; xent: 1.10; lr: 1.00000; 5611/5623 tok/s;  15158 sec
[2021-01-23 15:42:03,959 INFO] Step 49600/50000; acc:  74.28; ppl:  2.78; xent: 1.02; lr: 1.00000; 5587/5617 tok/s;  15172 sec
[2021-01-23 15:42:18,229 INFO] Step 49650/50000; acc:  76.44; ppl:  2.53; xent: 0.93; lr: 1.00000; 5795/5716 tok/s;  15187 sec
[2021-01-23 15:42:33,849 INFO] Step 49700/50000; acc:  74.44; ppl:  2.74; xent: 1.01; lr: 1.00000; 5388/5411 tok/s;  15202 sec
[2021-01-23 15:42:51,152 INFO] Step 49750/50000; acc:  70.17; ppl:  3.31; xent: 1.20; lr: 1.00000; 5553/5368 tok/s;  15220 sec
[2021-01-23 15:43:06,302 INFO] Step 49800/50000; acc:  71.18; ppl:  3.17; xent: 1.15; lr: 1.00000; 5785/5415 tok/s;  15235 sec
[2021-01-23 15:43:22,729 INFO] Step 49850/50000; acc:  73.89; ppl:  2.82; xent: 1.04; lr: 1.00000; 5146/5073 tok/s;  15251 sec
[2021-01-23 15:43:41,964 INFO] Step 49900/50000; acc:  70.99; ppl:  3.25; xent: 1.18; lr: 1.00000; 5288/5151 tok/s;  15270 sec
```