Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

CONTEXT BASED MACHINE TRANSLATION WITH RECURRENT NEURAL NETWORK FOR ENGLISH - AMHARIC TRANSLATION

By

Yeabsira Asefa

February,2020

Addis Ababa

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

CONTEXT BASED MACHINE TRANSLATION WITH RECURRENT NEURAL NETWORK FOR ENGLISH - AMHARIC TRANSLATION

By

Yeabsira Asefa

A Thesis Submitted to the School of Graduate Studies in Partial Fulfillment of Masters of Science in Computer Engineering

Thesis Advisor	Thesis Evaluator
Dr. Surafel Lemma	Dr. Getachew Alemu
Chair of Department	Thesis Evaluator
Dr. Yalemzewed Negash	Menore Tekeba

CONTEXT BASED MACHINE TRANSLATION WITH RECURRENT NEURAL NETWORK FOR ENGLISH - AMHARIC TRANSLATION

By

Yeabsira Asefa

MSc. Thesis

Submitted in Partial Fulfillment of the Requirements
for Masters of Science in Computer Engineering
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering
February, 2020

Declaration

I, the undersigned, declare that this thesis is my original work, and it has not been presented for MSc. degree to this or any other universities by any other before me. I have cited the source of the works by others that have contributed for the research to the best of my knowledge.

Yeabsira Asefa

Signature_____

Date_____

This thesis document has been submitted for examination with my approval as the university advisor

Dr. Surafel Lemma

Signature_____

Date_____

Acknowledgement

First and foremost, I would like to reverently thank the one who is, was and will be; the compassionate and gracious, the long-suffering, and he abounding in kindness and faithfulness; the LORD GOD Almighty.

I would like to express my heartfelt gratitude to my advisor Dr. Surafel Lemma, for his guidance, continuous follow up and encouragement. This thesis would not have seen the light of day had he not done his part diligently, adequately and beyond. I would also like to thank Dr. Rosa for taking the time to read and give continues feedback on the document.

My deepest gratitude goes to my parents, Wro. Abebech and Ato Asefa; my brothers, Abenezer and Addis Hiwote and my sister Genet. You all made my life worth living and without your support, reaching here is unconceivable. And to all my relatives and friends who also played undismissable role in my life, I thank you immensely.

Finally yet importantly, I would like to thank all staff members of AAiT, especially of Electrical and Computer Engineering Department who have taught me both school and life lessons.

Abstract

The quote from Rev. Jesse Jackson, "A text without a context is a pretext", summarizes the reasoning behind this thesis. Capturing context in translating between two human languages using computing machines is challenging. It is more challenging when the languages differ greatly in grammar and have small parallel corpus like the English-Amharic pair. The current approaches for English-Amharic machine translation usually require large set of parallel corpus in order to achieve fluency as in the case of statistical machine translation (SMT) and example based machine translation (EBMT). The context awareness of phrase based machine translation (PBMT) approaches used for the pair so far are also questionable. This research develops a system that translates English text to Amharic text using a combination of context based machine translation (CBMT) and a recurrent neural network machine translation (RNNMT). We built a bilingual dictionary for the CBMT system to use along with a target corpus. The RNNMT model has then been provided with the output of the CBMT and a parallel corpus for training. The proposed approach is evaluated using the New Testament Bible as a corpus. The result shows that the combinational approach on English-Amharic language pair yields a performance improvement of 2.805 BLEU scores on average over basic neural machine translation(NMT).

Key words: Machine Translation, context based machine translation, English to Amharic translation, recurrent neural network machine translation, context based machine translation with neural network machine translation

Contents

List of Figures

List of Tables

Acronyms

BLEU	Bilingual Evaluation Understudy
CBMT	Context Based Machine Translation
EBMT	Example Based Machine Translation
GNMTS	Google Neural Machine Translation System
HPBMT	Hierarchical Phrase Based Machine Translation
MT	Machine Translation
NLP	Natural Language Processing
NMT	Neural Machine Translation
PBMT	Phrase Based Machine Translation
RBMT	Rule Based Machine Translation
SMT	Statistical Machine Translation

Chapter 1

Introduction

In these times where the world is advancing into becoming a village, translation from one language to another plays a crucial role in creating that village. Either humans or machines could do the translation of ideas from one language to another. The human translation approach produces a more accurate translation but it is very time consuming, hence the need for Machine translation aided by humans[Precup-Stiegelbauer, 2012].

Machine translation (MT) is the translation of a text or speech from a given language into another language using software in a machine. Although machine translation is not a new concept of today's age, there is room for improvement in terms of accuracy and fluency. Accuracy refers to the correct translation of words and phrases while fluency refers to a sturdy flow of the translated words and phrases from the initial language.

Different approaches are proposed to handle machine translation and so far, none of them have the accuracy of a human translator, even when well trained [Precup-Stiegelbauer, 2012]. Some of the proposed machine translation approaches are statistical machine translation (SMT) [Brown et al., 1990], rule based machine translation (RBMT) [Gasser, 2012], hybrid of SMT and RBMT [Yulianti et al., 2011], example based machine translation (EBMT)

[Gangadharaiah, 2011], neural machine translation (NMT)[Wu et al., 2016] and context based machine translation (CBMT)[Miller et al., 2006].

Except NMT and RBMT, the others mentioned above are mostly phrase based machine translations (PBMT), which translate phrases instead of a single word at a time. Translation requires more than word-for-word translation, the translated text needs to be coherent and loyal to the context of the text; the fluency and accuracy aspect mentioned in paragraph two of this chapter.

Context based machine translation (CBMT) is newer than other PBMT approaches proposed[Miller et al., 2006]. Unlike most PBMT approaches that rely on statistical occurrence of the phrases, CBMT works on the contextual occurrence of the phrases. CBMT uses bilingual dictionary as its main translator and produces phrases to be flooded into a large target corpus.

The CBMT approach addresses the problem of parallel corpus scarcity between language pairs. The parallel corpus set for English-Amharic language pair, for instance, composes of the Bible, the Ethiopian constitution and international documents. These sources use words specific to their domain and overlook phrases and words used by novels, news and similar literary documents. The CBMT uses synonyms of words in place of rare words and rely on large target corpus and a bilingual dictionary to help with data scarcity[Miller et al., 2006]. It is not dependent on large parallel corpus like most PBMT (e.g., SMT and EBMT). The CBMT, however, fails in fluently translating texts compared to the neural machine translation (NMT).

The NMT uses neural networks to learn how human's translate given the input of parallel source and target translated corpus, it attempts to learn the pattern and translate accordingly. Its translations are more fluent and accurate than all the rest so far when evaluated individually [Popovic, 2017]. However, NMT struggles to translate properly rare words and words not commonly used[Wu et al., 2016]. In addition, NMT requires large parallel corpus for training.

The aim of this research is to build a system by combining the CBMT with the NMT for English to Amharic translation. The combination of PBMT and NMT is the future and most promising than the individual approaches themselves [Popovic, 2017]. The approach this thesis proposes, therefore, will be utilizing the strength of each method. The CBMT's ability to address rare words and the NMT's ability to produce fluent translation along with their context aware characteristics makes them complementary couples.

The combination is done by providing the NMT with two inputs, one from the source language and the other from the output of the CBMT to produces the final target sentence. In this paper, we show that this approach utilizes the strength of each method to achieve a significant translation performance improvement over simple NMT. The improvement is mostly dependent on the performance of the CBMT and mostly on the bilingual dictionary of the CBMT.

1.1 Motivation

Google and other major names in the field of MT used to have SMT as their first methodology while working with machine translation [Lan, 2018]. Now they have changed their methods to NMT [Wu et al., 2016]. Their accuracy has become acceptable in the eyes of most commoners for most languages[Lan, 2018]. Most of these language pairs are European or their decent. These are the ones with larger amount of parallel corpora, as Google and its counterparts use the United Nations document and European parliament document as their resource.

However, languages with limited amount of resource of parallel corpus do not become as much beneficiary of such systems as those with large translated resources. Parallel corpus refers to a document well translated into two or more languages by a human translator. Among the languages with less translated documents are Ethiopian languages such as Amharic.

Creating an inclusive system for less translated languages would create a better platform for the world to see and understand their civilization. They can write their science, their findings and their cultural knowledge with their own language and share it to the global world. We, from countries of fewer parallel corpora, shall focus on producing a system that does not require greater amount of parallel corpus yet do a fair translation. Fair, in-terms of accuracy and fluency, refers to having it be context aware and translate the message with the target languages proper grammar.

The proposed system may not compete with world renowned translation system with relative robustness; having limited resource both in corpus and hardware requirement, but it will bring forth a system that would improve translation of languages with little amount of parallel corpus. The dependency of the system is in finding a bilingual dictionary; with that met, it will produce a competitive translation.

1.2 Problem statement

PBMT approaches are mostly used to translate English to Amharic like SMT by [Tadesse and Mekuria, 2000], [Teshome, 2000], [Taye et al., 2015] and [Besacier et al., 2000], RBMT by [Gasser, 2012], hybrid of SMT and RBMT by [Zewgneh, 2017] .

PBMT approaches such as SMT, EBMT and hybrids involving one of these, like the hybrid of SMT and RBMT, require large parallel corpus to produce competent translation. The English-Amharic pair do not have as large a parallel corpus encompassing all domains. In order for this pair to have fairly translating system, an approach which requires a smaller parallel corpus is needed.

Most PBMT (e.g., SMT and hybrid of SMT and RBMT) are not context aware as they rely on statistical analysis than context based analysis.

Meanwhile, the languages English and Amharic have grammar and word use differences whose translation can be made accurate and fluent by the use of context. For instance Amharic has 'tebko-lalto' principle where a single word has two different meanings depending on the pronunciation of a syllable in the word. In addition, both English and Amharic have diverse vocabulary by which one word has different translation in the other language. For example the English words 'test', 'trial', 'exam' and 'temptation' translate to a single Amharic word 'fetena'. On the other hand, the Amharic words 'ante', 'anchi' and 'enante' translate to the English word 'you' and depending on the words surrounding it the Amharic words would morph to 'antem','beante','anten',leante','beantem','leantem', 'keante' and so forth.

CBMT unlike other PBMT approaches addresses parallel corpus scarcity and context awareness by using bilingual dictionary, large target corpus and synonyms of words in place of rare words[Miller et al., 2006]. CBMT, a PBMT, fails like the other PBMT approaches in fluency compared to NMT. NMT's translations are more fluent and accurate than all the rest seen so far [Popovic, 2017]. However, NMT not only requires large parallel corpus but it also struggles to translate rare words[Wu et al., 2016].

Therefore, this research tries to address the above shortcomings; the problem of not having a fluent and accurate translating system which is context aware and can handle rare words especially for language pair English-Amharic with smaller set of parallel corpus. A fairly good translation with a closer accuracy and fluency to that of humans can be produced by taking the strengths of PBMT combined with NMT[Popovic, 2017]. In this research CBMT and NMT are combined and it asks the following research questions:

RQ1 [**Performance**] Will the combination of CBMT and NMT be complimentary and perform better than the approaches themselves?

RQ2 [**CBMT impact**] Will the error introduced by the CBMT in the combinational system of NMT and CBMT affect the performance of the system significantly?

1.3 Objective

1.3.1 General objective

The main objective of this research is to combine a recurrent neural network machine translator and context based machine translator for translating English text to Amharic text.

1.3.2 Specific objective

- Prepare a parallel corpus of English - Amharic language pair
- Develop a bilingual dictionary
- Develop the proposed approach and the baseline approaches, CBMT and NMT
- Evaluate the performance of CBMT, NMT and the proposed approach in BLEU score and drive a conclusion
- Assess the impact of CBMT on the proposed approach in BLEU score and drive a conclusion

1.4 Scope

The focus of this research is in designing and developing a context based machine translator with neural network to translate from English to Amharic. Although the system can be trained for Amharic to English translation, the current implementation is for English to Amharic translation.

1.5 Significance of the study

Adwa is a good example as to why a context based machine translation is needed. The battle of Adwa was fought because a treaty signed between Ethiopia and Italy had varying translation for the Amharic and Italian versions of the document for one of the articles, article 37. Hence, unbiased context aware system for less resourced languages is quite important. Below is the list of some of the uses this system could provide.

- To have a proper translation of technical documents in a lesser time than a human translator would take
- To have a fairly good translation between language pairs with lesser amount of parallel corpus
- To have a research contribution for other less resourced languages worldwide
- To have a building block for speech translation and communication with machines

1.6 Research Methodology

The methodology this thesis uses is the combination of context based machine translation and neural network machine translation. This approach addresses the limitation of context unawareness of some PBMT approaches like SMT and the need for large parallel corpus of simple NMT.

The context based machine translation (CBMT) uses a bilingual dictionary for initial translation, a large target corpus to find phrases of the translation, a latent semantic indexing to sort the large target corpus based on the search.

The neural network machine translator (NMT) has an encoder to encode the source text and a decoder aided with an attention model to decode to the target text.

In our approach, the source sentence in English and the translation output of the CBMT in Amharic has been fed to the NMT's encoder-decoder model as shown in Figure 1.1. The NMT model then produces the final Amharic translation based on its training.

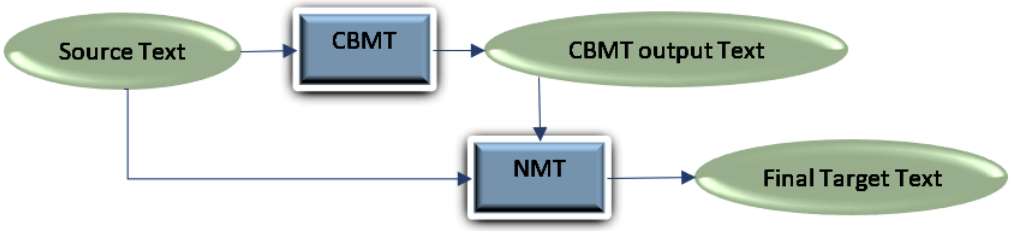


Figure 1.1: The proposed methodology

The below list shows the general flow of the methodology:

- Phase 1: Preparing parallel corpus of English and Amharic language pair.
- Phase 2: Develop a context based machine translator system for the language pair and measure its performance
- Phase 3: Develop neural machine translator system for the language pair and measure its performance
- Phase 4: Develop NMT and CBMT combination system for the language pair and measure its performance
- Phase 5: Compare the result of the individual systems with the combination of the systems

1.7 Thesis outline

The second chapter gives the detailed literature review of the project. Chapter 3 discusses the methodology used by this project. The procedure taken to implement the methodology and experiments conducted are discussed in Chapter 4. In Chapter 5, the results found from the experiments conducted are discussed. The last chapter, Chapter 6, concludes the findings of the research.

Chapter 2

Literature review

Machine translation has been researched since the 1950's and 1960's but it became computer based from 1980's to 1990's[John Hutchins, 1995].

There are different approaches proposed to produce a tolerable translation in terms of accuracy and fluency to contend with human translation. Some of these are example based machine translation (EBMT)[Gangadharaiah, 2011], rule based machine translation (RBMT)[Gasser, 2012], statistical machine translation (SMT)[Brown et al., 1990], hybrid of statistical machine translation (SMT) and rule based machine translation (RBMT)[Yulianti et al., 2011], context based machine translation (CBMT)[Miller et al., 2006] and neural machine translation (NMT)[Wu et al., 2016].

In this chapter the EBMT, RBMT, SMT, and hybrid of SMT and RBMT approaches are briefly explained introducing the systems. CBMT and NMT will also be discussed in depth along with the combination of NMT and PBMT approach.

2.1 Statistical machine translation (SMT)

The statistical machine translation (SMT) approach uses a parallel corpus of source and target language to translate a text. The source text is initially split into phrases to be translated using the parallel text. After these phrases are prepared from the source corpus, their translation is then looked for in the target corpus, which humans had translated. A phrase may be translated differently in different texts, so the different combinations of these translated phrases are looked for in the target corpus and given a statistical value based on their frequency. The one with the highest probability of occurrence will be chosen as the translation.

Warren Weaver[John Hutchins, 1995], a pioneer in machine translation, first introduced SMT and multiple researchers are researching into it until today. The basic structure of the method is as depicted in Figure 2.1.

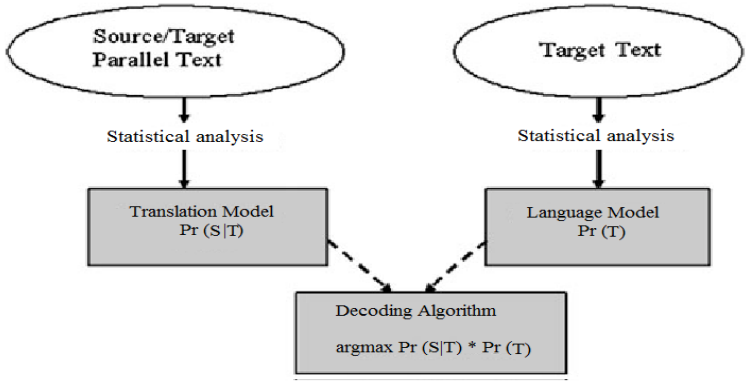


Figure 2.1: Basic statistical machine translation[Arora et al., 2013]

One of the initial researches is by the researchers at Thomas J. Watson research center [Brown et al., 1990] who properly formulated the formula to calculate the probability computation of the SMT system. According to their paper[Brown et al., 1990], $p_r(S|T)$ is the probability that a source sentence S will be translated from target sentence T and $p_r(T)$ is the probability of occurrence of T (target sentence). The $p_r(S|T)$ is calculated by using Bayer's theorem as; the probability of source ($p_r(S)$) times the probability of target

sentence T produced from source S ($p_r(T|S)$) divided by the probability of T ($p_r(T)$).

$$p_r(S|T) = \frac{p_r(S)p_r(T|S)}{p_r(T)} \tag{2.1}$$

Once their probability is calculated then the decoder calculates the actual translation. Given a sentence T in the target language, the decoder chooses a viable translation by selecting that sentence in the source language for which the probability $p_r(S|T)$ is maximum[Brown et al., 1990].

The implementation of the system is shown in Figure 2.2 where the phrase እኔ እንዴት እርባኖል is searched for in the parallel corpus and the different translated word combinations are produced for it. From the list of phrases the one most frequent is then selected; 'I' is found in all, 'am' and 'so' are found in two sentences and 'hungry' is found in all. Hence, the max frequent words combined will give ' I am so hungry'.

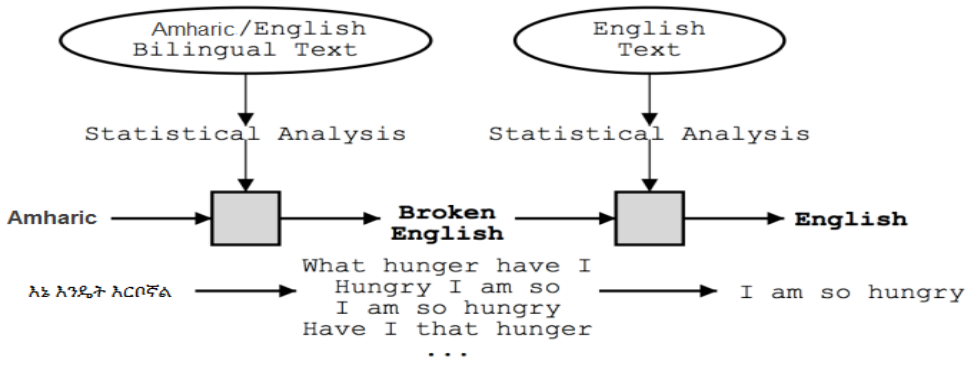


Figure 2.2: SMT implementation example

The statistical machine translation translates all words in the source sentence given to it, which makes it adequate but its context knowledge is pretty shallow[Oladosu et al., 2016].

Ambaye and Yared have done English-Amharic translation using SMT. Their paper has achieved an accuracy of 18.74 in BLEU score while using SMT [Tadesse and Mekuria, 2000]. Another paper which used SMT pub-

lished by Eleni Teshome achieved an accuracy of 82.22% for simple sentence and 73.38% for complex sentence [Teshome, 2000]. The later paper however used the test data that was similar to the training data. Hence, it is hard to take the results to show translation accuracy but it may be taken to show the performance of SMT on a fixed data set.

Besides text translation, different papers have used SMT for speech translation as well. One of such papers is by Mulu [Taye et al., 2015]. The paper focused on improving the quality of the translated word rather than improving the quality of the translation itself. The paper achieved around 37% accuracy while it had only 35% accuracy when it was not using phoneme based approach. Although it is a well-researched paper, it is not as much a translation-focused paper but more of performance enhancement of translated document.

Another paper is by [Besacier et al., 2000] which has achieved an accuracy of 76-77% for a word-by-word translation which increases as we go down to the level of character and morph. However, not only the domain restricted but also the focus was in speech recognition than translation.

2.2 Rule based machine translation (RBMT)

In the rule based machine translation (RBMT) approach the language rules of the target and source language are coded to create a well-organized and grammatically correct translation. Since the syntactic and semantic grammar rules are quite complicated and bunch in number, this approach not only requires language expertise but also an intense coding on the part of the programmer.

Figure 2.3 shows the steps RBMT takes to translate a sentence in English to Amharic. The first phase is tagging of the words with their respective part-of-speech (POS). In phase two, the RBMT analyzes the verb to determine

its syntactic information. The third phase parses the source sentence, which is usually done in part to get the syntactic structure of the sentence. The translation is done on the fourth step with word-by-word translation. The final phase will rearrange the translated words according to their POS tagging in the target language; it maps the translated words into their changed word form (i.e. change in tense, gender or number).

	A	**Student**	**Failed**	**An**	**Exam**
Phase 1 (POS tagging)	Indefinite article	noun	verb	Indefinite article	noun
Phase 2 (analyze the verb)			Np-failed-Np Simple Past, 3rd Person, Active Voice		
Phase 3 (parsing)					Np-an exam The object of 'failed'
Phase 4 (translate)	አንድ	ተማሪ	ወደቀ	አንድ	ፈተና
Phase 5 (mapping)	አንድ	ተማሪ	አንድ	ፈተና	ወደቀ

Figure 2.3: RBMT implementation steps

RBMT, therefore, has the following components: morphological analyzer, parser, translator and a dictionary for both the target and source language. Figure 2.4 shows the flow chart of the components.

A paper published, which solely uses this method to translate English to Amharic [Gasser, 2012] states that RBMT can surpass SMT, if the content domain is narrowed and the translation is meant to keep publication standard. Although their first publication did not publish the final result; in their second publication of 2012, the proposed framework was clearly explained and offered for the public.

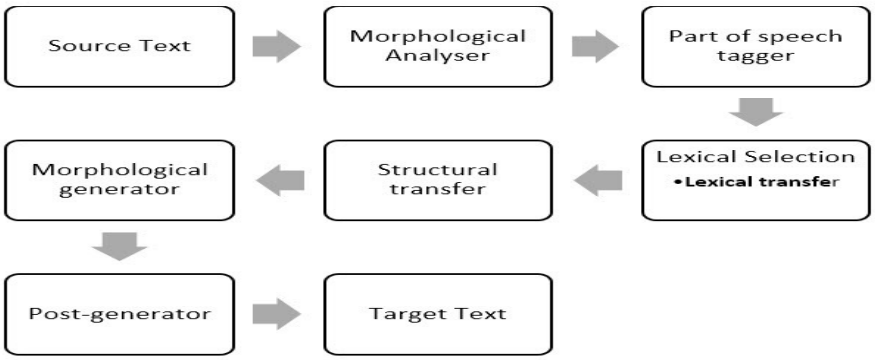


Figure 2.4: RBMT flow chart [Irfan, 2017]

2.3 Hybrid of SMT and RBMT approach

The hybrid approach is proposed to fix the flaws of SMT and RBMT by building a combination of the two approaches. The failure of SMT to bring forth a grammatically correct sentence is fixed by the RBMT and the relatively context aware SMT will help the RBMT approach to produce a more statistically probable result. The RBMT merged with the SMT will not require intensive coding as before since the rules to be coded are general.

The hybridization can be done in different orders; that is, the output of RBMT can be given to the SMT or the SMT's output can be fed to the RBMT or the SMT can be sandwiched by the RBMT and any other combination the researcher sees fit.

In the first case, the RBMT does the morphological analysis and translation and the SMT does the editing and decoding of the RBMT output. This was implemented by an Indonesian team [Yulianti et al., 2011] and their implementation is shown in Figure 2.5.

The second case, which is common, uses the SMT to perform the translation and then depends on the RBMT to correctly present the grammatical order of the target sentence[Labaka et al., 2014] . This can be seen from the implementation shown in Figure 2.6.

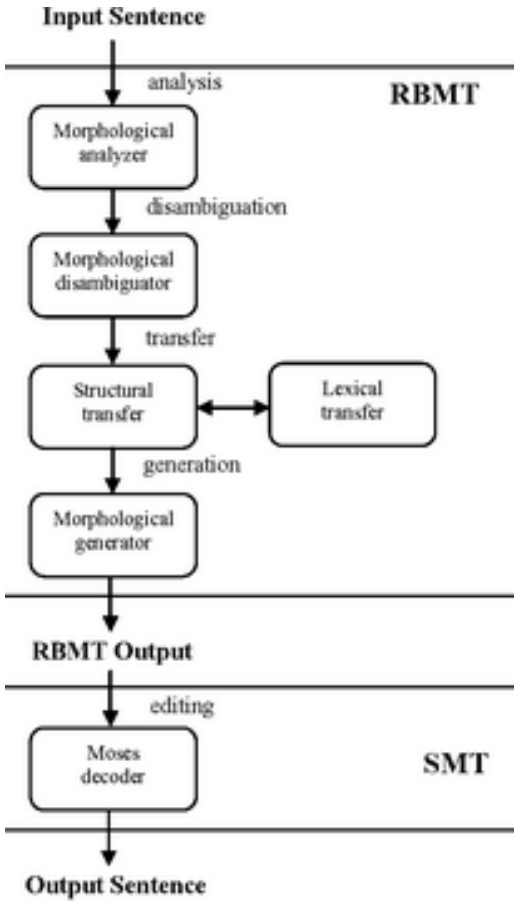


Figure 2.5: RBMT and SMT hybrid with the RBMT doing the translation [Yulianti et al., 2011]

The recent paper implementing the hybrid approach for English-Amharic language pair is by Smarawit [Zewgneh, 2017]. This paper used pre-processing where the rule based machine translation is applied to the source language then it is feed into the SMT. Translation is conducted by the SMT and then post-processing was applied where the output of the SMT is given to the rule based system for processing. It has achieved an accuracy of 15% and 20% improvement for simple and complex sentences respectively over normal SMT.

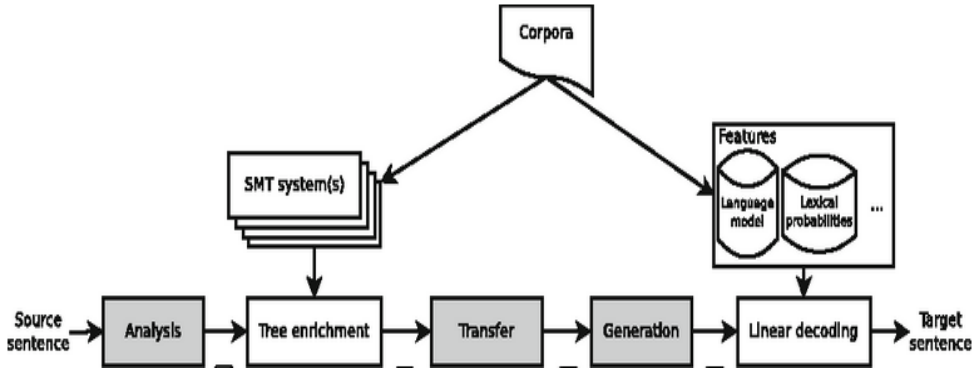


Figure 2.6: RBMT and SMT hybrid with the SMT doing the translation [Labaka et al., 2014]

2.4 Example based machine translation

This is a method usually merged with another machine translation method to produce a desired result. In this approach, the system takes in large sets of example sentences unlike the normal documents in SMT.

EBMT is also phrase based machine translation, which uses a parallel bilingual corpus like SMT, but it has a dictionary like impression. It uses analogy for translation using the parallel bilingual corpus. The bilingual corpus used by the EBMT is different from that of the SMT or any other that uses parallel corpus, in that, it should be of similar format in both the source and target. Figure 2.7 is an example of EBMT bilingual parallel corpus.

Amharic	English
ትንሹ ቦርሳ ስንት ነው	How much is that **small bag**?
ቀዩ ዣንጥላ ስንት ነው	How much is that **red umbrella**?

Figure 2.7: Example Based Machine Translation example

The EBMT will be trained with such sentences of similar sentence formats

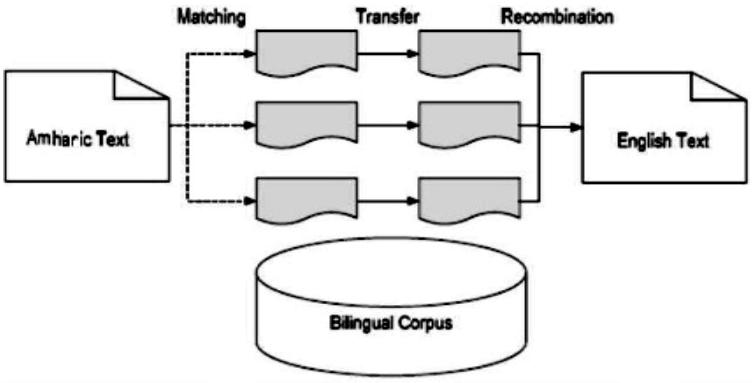


Figure 2.8: EBMT flow of translation

and when a sentence of similar structure (e.g., 'X ስንት ነው') is sought to be translated, it uses the format (how much is that X) and replace X's dictionary translation.

The EBMT will have the follow shown in Figure 2.8. Each sentence in the source text will be matched using a bilingual corpus to the target sentence of the same format. The translated sentences are then combined to form the text.

The search for EBMT applied to the language pair of English-Amharic did not produce any published papers. Therefore, to my knowledge, no one has used this approach for these language pairs. However, other language pairs have been using this approach and it is usually used as a support than main approach.

The paper, which is significant to our research using this approach, is by Rashmi[Gangadharaiah, 2011]. The paper uses EBMT and makes it general by using templates instead of sentences and then use SMT to decode the translation from English to Chinese and English to French. When it comes to words not in vocabulary or rare words, the paper uses the word itself or the direct translation from the cluster of word built using the template.

2.5 Context based machine translation

A group of researchers from the company "meaningful machine" proposed the context based machine translation (CBMT) in 2006. They stated that it outperforms RBMT, SMT and EBMT when it comes to languages with less parallel corpora[Miller et al., 2006]. It uses a bilingual dictionary, a large target corpus and smaller amount of source corpus, which is optional.

The basic working principle of CBMT is as follows: first, segments are prepared from the source sentence that would produce set of words or phrases. Then the words are translated using the bilingual dictionary directly. Afterwards the combination of this translated words made into n-grams are searched for in the target corpus for a match. The phrase with the longest n-gram and longest overlap is taken as the final translation of the combined words. The flow of translation they Proposed is shown in Figure 2.9.

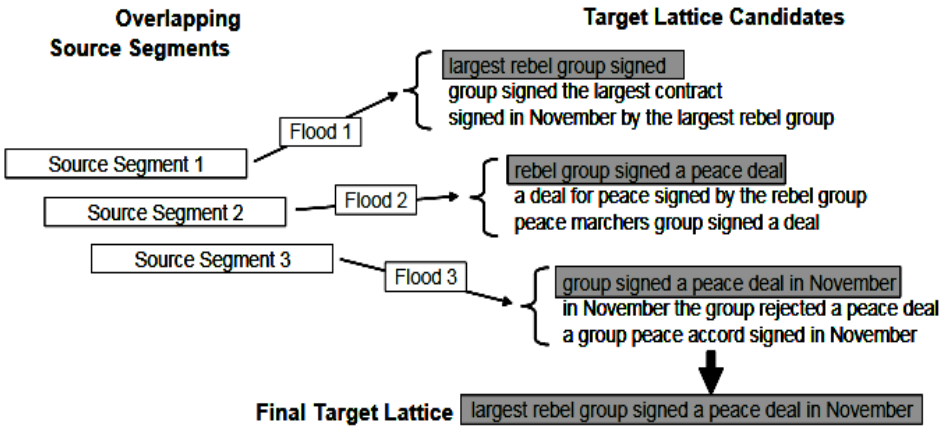


Figure 2.9: CBMT implementation [Miller et al., 2006]

As the Figure 2.9 shows the flooded outputs from different papers (three in the Figure) are looked for overlaps. The flooded output for the first phrase or n-gram and the one next to it are the ones to be checked for an overlap. Meaning the phrase 'largest rebel group signed' will be checked for overlap in all the three outputs of the second flooded phrase ('rebel group signed

a peace deal', 'a deal for peace signed by the rebel group', 'peace marchers group signed a deal') and not from the third flooded phrase. When it comes to rare words, this method resolves to searching for a synonym and replacing with a synonym. Figure 2.10 shows this implementation of the proposed method by the CBMT initial researchers.

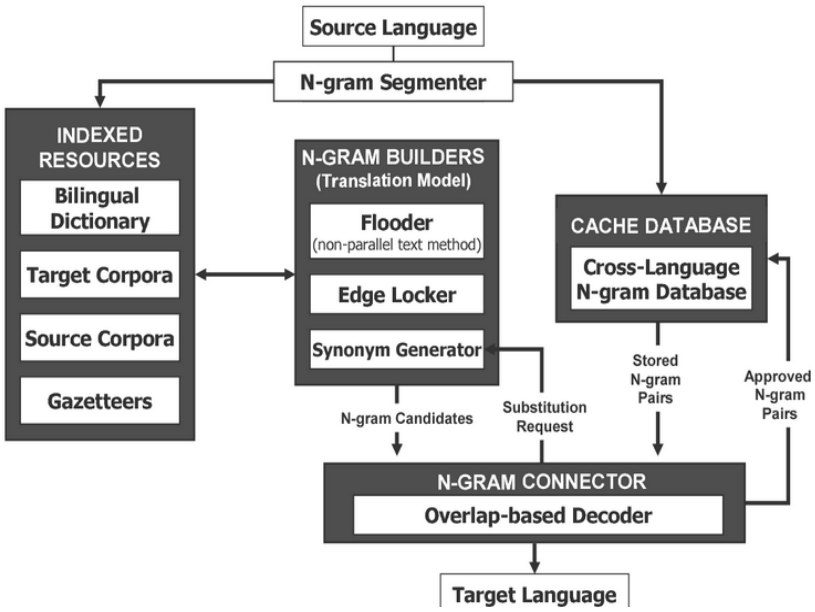


Figure 2.10: CBMT proposed methodology [Miller et al., 2006]

They generated the corpus of both the target and the source language from the web by crawling different sites. Since there is no need for parallel corpora in this approach, the sites need not be translated from one language to the other.

The paper[Miller et al., 2006] has implemented the method for the Spanish-English pair and reported to have achieved a BLEU score of 0.6462. The paper has not yet implemented the synonym generator proposed for rare words hence implementing that could improve the performance of the system. In our paper, we use synonym generator.

2.6 Neural machine translation

The neural machine translation (NMT) mimics the human neural network in training and decoding the words and sentences. The major translation software produced by Google[Wu et al., 2016] uses this approach.

The system has two main parts or phases: the encoding and the decoding with an attention algorithm. In the encoding phase, each word is converted to a vector, which will be modified into another vector based on the words next to it in order to make the system context aware. After repeated decoding based on the number of encoding layers, the final output is given to the attention algorithm, which will perform word-by-word vector definition based on the training of the neural network. This defined vector is translated and inputted back into the attention algorithm for refinement and this stage is the decoding. More like the encoding, it may have multiple layers or a single layer.

The Google translate uses recurrent neural network with 8 layers. According to their published paper[Wu et al., 2016], neural network in general has the following setbacks: it has slow training and inference speed, it is ineffective when it comes to handling rare words and it sometimes fails to translate all words in the source sentence.

To solve the above-mentioned problems of the NMT, the Google neural machine translation system (GNMTS) did the following: it implemented parallelism to improve the speed by connecting the attention algorithm from the bottom layer of the decoder network to the top layer of the encoder network. To improve inference speed, it used low-precision arithmetic, which is improved further by using a special hardware (Google's Tensor Processing Unit, or TPU). In order to handle rare words it used sub-word units called "word pieces" for both the inputs and the outputs. It sub divides the words and try to find meaning for the sub-words. In order to force the system to translate all words, they implemented beam search technique. This

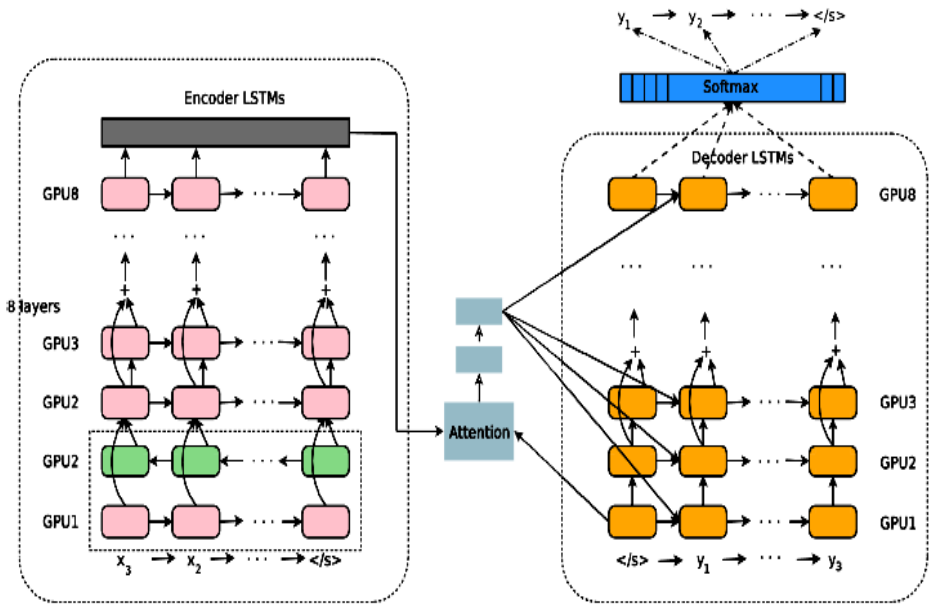


Figure 2.11: Google's neural network implementation [Wu et al., 2016]

technique includes a length normalization procedure and a coverage penalty to encourage the model. Figure 2.11 shows Google's implementation of the NMT.

This GNMTS system was tested for multiple language pairs. It has achieved an accuracy of 38.95 for English to French translation.

2.7 Combination of NMT and PBMT

Phrase based machine translators (PBMT) are those that translate a sentence phrase by phrase rather than the sentence as a whole. This category includes SMT, EBMT and CBMT.

The combination of CBMT and NMT aims at resolving the weakness of the approaches and brings forth a stronger system. The NMT has a bad reputation when it comes to rare words and leaving parts of sentences not

translated; while, PBMT would always translate the whole sentence and has better coping mechanism when it comes to rare words. On the other hand, PBMT is not fluent like NMT as its training mechanism is statistical and not well learnt.

Some papers have used the combination of these two approaches. One of those papers is by [Niehues et al., 2016] which proposed two approaches. The first approach is a pipeline method where the translated output of the PBMT is fed to the NMT. The second approach, which performed well, has two inputs for the NMT named mixed input. The source sentence is feed into the SMT and the output of it along with the source sentence is given to the NMT, which gives the final translation. They achieved a higher BLEU score for the later approach for their English-German pair language. In the test they conducted in 2016, they obtained a BLEU score of 30.67%. Figure 2.12 depicts their proposed combination mode.

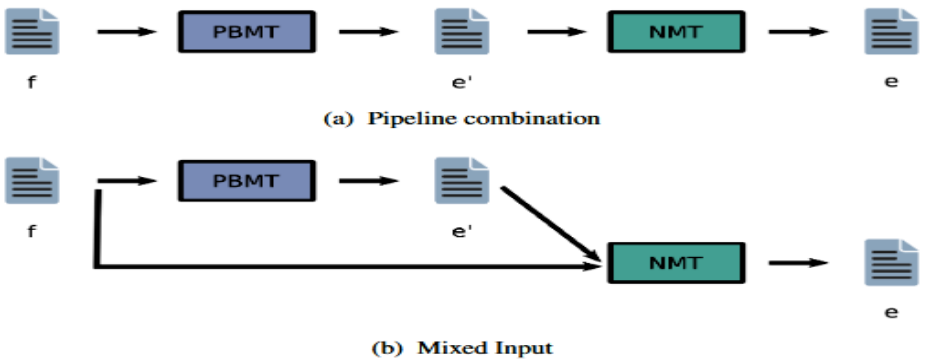


Figure 2.12: Proposed combination model for pre-translation for NMT [Niehues et al., 2016]

The other paper which used combination is by [Zhang et al., 2017] for the language pairs of English-Chinese. Their experiment was similar to that of the previous paper but they not only gave the phrase-based SMT output to the NMT but also a hierarchical PBMT (HPBMT) of the SMT, which in a normal case would outperform the simple PBMT. So the neural network not only received the source sentence as an input but also the outputs of both the

HPBMT and the simple PBMT of the SMT. They manage to outperform the previous system by attaining a Blue score of 43.44% average result. Their proposed combination model architecture is shown in Figure 2.13.

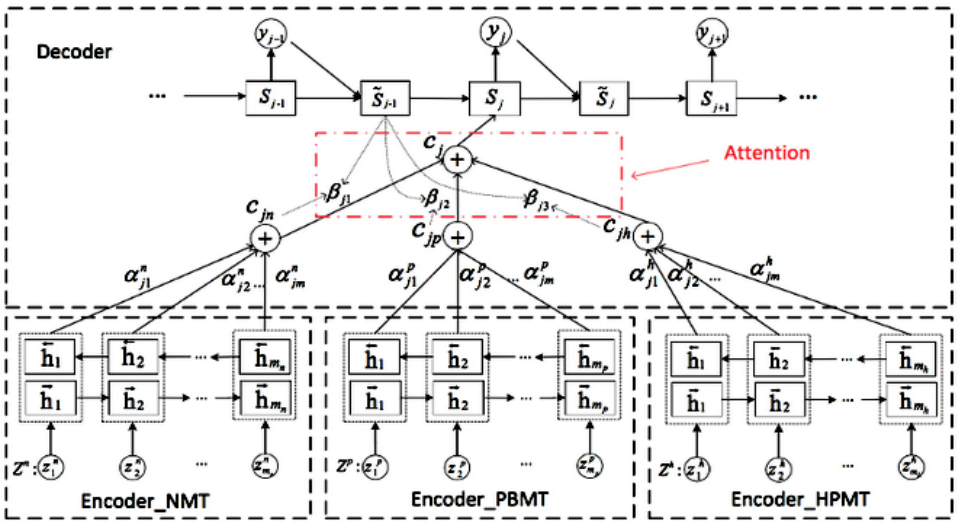


Figure 2.13: Proposed combinational model architecture [Zhang et al., 2017]

In summary, statistical approach has good accuracy in translating all the words but fails to be fluent [Oladosu et al., 2016]. The rule based approach is good in building initial system but it is not robust as language rules vary and meanings may be lost in translation[Oladosu et al., 2016]. RBMT is fluent for the rules it has but lacks when the knowledge is not there. Hybrid of RBMT and SMT gets fluency from RBMT and accuracy from SMT but for longer sentences, the reordering fails [Oladosu et al., 2016]. The example based, when using a template will become more general unlike when it had used real examples[Gangadharaiah, 2011]but it still cannot stand alone and need another supporting approach such as SMT[Gangadharaiah, 2011].

The CBMT outperforms SMT and EBMT in accuracy and fluency but when it comes to accuracy of less translated words, it is not competent enough[Miller et al., 2006]. The NMT, on the other hand, has accuracy and fluency but it fails to translate the whole sentence and also fails to perform well with rare words[Wu et al., 2016]. The combination of PBMT and NMT

has the better end than all[Popovic, 2017] in terms of accuracy and fluency. The CBMT performs better than other PBMT approaches, hence the combination of CBMT and NMT is done in this research.

The combination of the CBMT and the NMT follows the method of mixed approach proposed by [Niehues et al., 2016]. Their mixed approach feeds the NMT with the source sentence and the output of the PBMT. The NMT block will encode both the source sentence and the translation by the CBMT and use the decoding approach used by [Zhang et al., 2017]. The CBMT method is similar to that by [Miller et al., 2006] with the synonym finder aspect added to it.

Chapter 3

Methodology

This chapter explains the methods used and procedures taken to solve the problem mentioned in the problem statement. It provides the basic explanation of the different methodologies proposed earlier and provides the reasoning behind for selecting the methods chosen apart from their competitors.

This research focuses on building a translation system from source language English to target language Amharic based on context of the text with a good accuracy. Since the pair does not have large size of parallel corpus, the methods used need to be dependent on other available resources like bilingual dictionary than parallel corpus. In order to have a better accuracy, methods that tune the translation need to be used.

In the research, the methods used are context based machine translation and neural machine translation. We have selected these two for different reasons. The CBMT does not use parallel corpus and gives out a context based translation, which are full translations of the sentences. On the other hand, NMT has a better accuracy at translating than all the other systems[Wu et al., 2016]. In this thesis, the source sentence in English and the translation output of the CBMT in Amharic are fed to the NMT as shown in Figure 3.1.

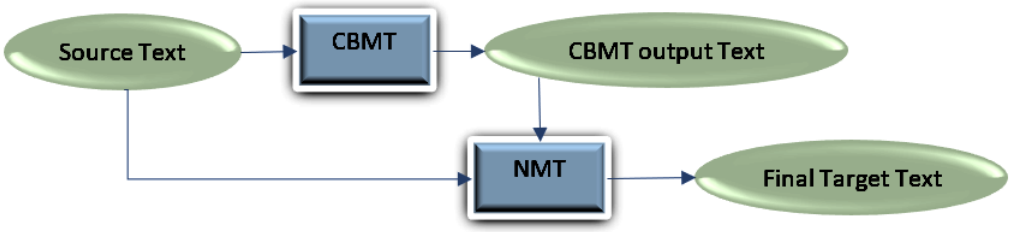


Figure 3.1: Combination method used by this paper

In our approach, the source sentence in English and the translation output of the CBMT in Amharic has been fed to the NMT's encoder-decoder model as shown in Figure 3.1. The NMT model then produces the final Amharic translation based on its training. The combination of the CBMT and the NMT follows the mixed approach proposed by [Niehues et al., 2016]. The research by [Zhang et al., 2017] also supports this way of combining different systems.

3.1 The CBMT system

The CBMT outperforms SMT and EBMT when it comes to languages with less parallel corpora[Miller et al., 2006]. It uses a bilingual dictionary, a large target corpus and smaller amount of source corpus, which is optional. In the context based machine translation, there are different components working together to produce the translation. Figure 3.2 shows the flow of data in these different components.

The source sentence is converted into N-gram phrases and then it is translated using bilingual dictionary. CBMT's performance is mostly dependent on the efficiency of the dictionary.

We have manually built a phrase based dictionary aided by Google translate. A synonym finder helps the dictionary's search and in this paper we have used WordNet[Soergel, 1998].

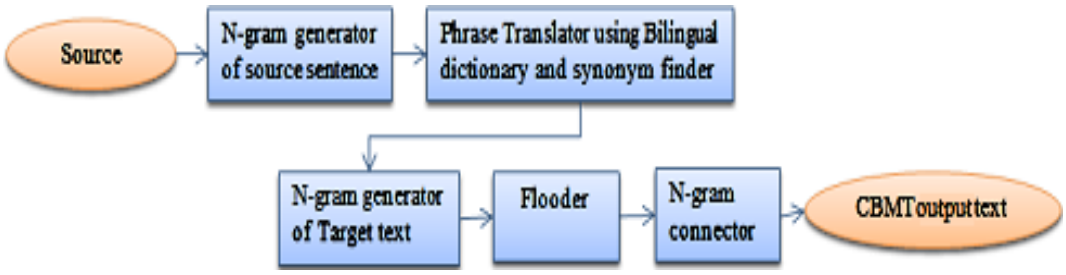


Figure 3.2: Overview of the CBMT system

WordNet is a library with large lexical database of English words. In our research WordNet is used to provide synonyms of the English words whose Amharic translations are not found in the manually built dictionary. These synonym words are then searched for in the dictionary to be used in place of the original words.

The English Sentence in Example 3.1.1 is used as a running example to explain the translation from English to Amharic using CBMT. The pseudo codes explain the different steps it goes through and the results obtained at the end of each step. The algorithms are based on the paper which initially proposed CBMT[Miller et al., 2006].

Example 3.1.1 ***"everyone who calls on the name of the lord will be saved"***

3.1.1 Dictionary translation

As shown in Algorithm 1 the first step in CBMT has been to generate n-grams for the sentence. These N-grams are then translated using the Bilingual dictionary and the synonym finder, WordNet in the case of this paper.

The translation steps start from step 3 of Algorithm 1 which is the for loop that loops through all the N-grams. The Maximum N-gram length for example 3.1.1 is four and they are listed in Output 1.

Algorithm 1: Translate Source Sentence

Input: *source_sentence* **and** *Max_N_gram_length*
Output: list of translated words with their translation
Data: dictionary

```
ngrams = source_ngrams/* list of ngrams made of source_sentence of
    Max_N_gram_length                                              */
Translation_list = []                  // holds the translation and the word
for each N_gram in ngrams do
    translation = search for the N_gram in dictionary
    if translation is NULL and N_gram is a single word then
        Stem the word                              // remove 'ing',ed', 's' ..
        translation = search for the stemmed word in dictionary
        if translation is NULL then
            search for its synonym in WordNet
            if synonym is not found then
                add word itself to Translation_list
            else
                translation = search for synonym in dictionary
    else if word is not a single word then
        continue
    if translation is Not NULL then
        add translation and N_gram to Translation_list
return Translation_list
```

Output 1: The Output of the first step of Algorithm 1

ngrams=*['everyone', 'everyone who', 'everyone who calls', 'everyone who calls on', 'who', 'who calls', 'who calls on', 'who calls on name', 'calls', 'calls on', 'calls on name', 'calls on name of', 'on', 'on name', 'on name of', 'on name of lord', 'name', 'name of', 'name of lord', 'name of lord will', 'of', 'of lord', 'of lord will', 'of lord will be', 'lord', 'lord will', 'lord will be', 'lord will be saved', 'will', 'will be', 'will be saved', 'be', 'be saved', 'saved']*

The output of Output 1 are translated to produce Output 2.

Output 2: The translated output of the for loop of Algorithm 1

```
[[['everyone', 'ሁሉም ሰው'], ['everyone', 'ሁሉ'], ['everyone', 'ለሁሉም'],
['everyone', 'ማንኛውም ሰው'], ['everyone', 'ለእያንዳንዱ']],
[['who', 'ማን']],
[['calls', 'ጥሪዎች'], ['calls', 'የሚጠራ'], ['calls', 'ከጠሪው']],
[['on', 'ላይ']],
[['name', 'ስም'], ['name', 'ስሙ'], ['name', 'ስሚም'], ['name', 'ለስምህም']],
[['of', 'የ']],
[['Lord', 'ጌታ'], ['Lord', 'የጌታን'], ['Lord', 'ጌታችን'], ['Lord', 'ከጌታም'],
['Lord', 'በጌታችን'], ['Lord', 'ጌታን'], ['Lord', 'ለጌታው'], ['Lord', 'ለጌታ'],
['Lord', 'የጌታ'], ['Lord', 'በጌታ'], ['Lord', 'የጌታችንን'], ['Lord', 'ጌታችንን'],
['Lord', 'የጌታችን']],
[['will', 'ፈቃድ']],
[['will be saved', 'ይድናልና']],
[['be', 'መሆን']],
[['saved', 'ይድናልና'], ['saved', 'እንድንም'], ['saved', 'መዳናችን'], ['saved',
'ድነናል'], ['saved', 'ይድናል'], ['saved', 'እንዲድኑ'], ['saved', 'ትድናለህ'], ['saved',
'የምትድነውም'], ['saved', 'ይድናሉ']]]
```

As in Output 2 some of the words may have multiple translations while others have only one and some are phrases like "will be saved" while others are single words.

3.1.2 Flooder

The flooder is then responsible for searching the translated phrases in the target corpus and finding the longest N-gram match. For each phrase to be flooded, it selects a phrase in the target corpus with the most translated words and least in-between words amongst the words matched.

The first thing to do before flooding is to combine the translated words into sentences as shown by Algorithm 2 steps 1 through 7. The for loop from steps 8 through 14 of Algorithm 2 then converts the sentences into N-grams of the target language.

If the length of the total word count is greater than 4, the total count of words is halved plus one to get the length of the N-gram to be formed.

Algorithm 2: Form N-grams of Target sentences to be flooded

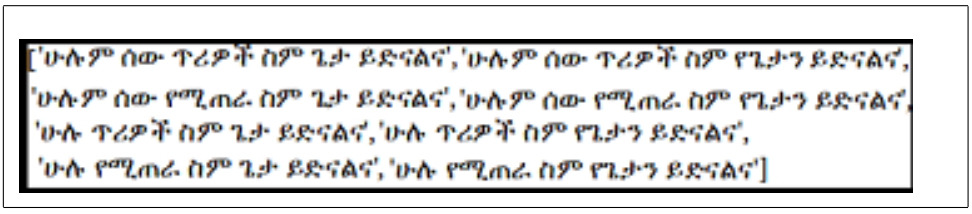
Input: *Translated_phrase_list*

Output: list of combined sentences N-grams

```
list_ngrams = []
previous_word_translations = Translated_phrase_list[0]        // the list of
translations for the first word
combination_list = []                  // holds the combined translation lists
while Translated_phrase_list do
    next_word_translations = Translated_phrase_list[next]
    combination_list=combination_list combined with next_word_translations
    next = next + 1
sentences = convert combination_list into sentences
for sentence in sentences do
    /* if number of words in a sentence is greater than four halve
       the length                                                    */
    if len(sentence) > 4 then
        max_N_gram = sentence/2 + 1
    else
        max_N_gram = len(sentence)
    list_ngrams = sentence_ngrams          // make ngrams of sentence of
    max_N-gram length and add to list
    remove duplicates in list_ngrams
return list_ngrams
```

Since the CBMT does not follow sentence order rules of the language, the words are in the order of the source language's sentence order format. Having the N-grams cover wider range, helps in getting related words in the same N-gram to be flooded.

Output 3: The translated output combined into sentence as in Algorithm 2



The combinations of the translations for Example 3.1.1 would give around 780 sentences. Output 3 shows few of the combination result of the phrases, having removed some of the possible translation for the words shown in Output 2.

Output 4: The N-grams for the translated sentences in Algorithm 2

```
[['ሁሉም ሰው ጥሪዎች ስም', 'ሁሉም ሰው የሚጠራ ስም', 'ሁሉ ጥሪዎች ስም', 'ሁሉ የሚጠራ ስም'],
['ሰው ጥሪዎች ስም ጌታ', 'ሰው ጥሪዎች ስም የጌታን', 'ሰው የሚጠራ ስም ጌታ', 'ሰው የሚጠራ ስም
የጌታን', 'ጥሪዎች ስም ጌታ', 'ጥሪዎች ስም የጌታን', 'የሚጠራ ስም ጌታ', 'የሚጠራ ስም የጌታን'],
['ጥሪዎች ስም ጌታ ይድናልና', 'ጥሪዎች ስም የጌታን ይድናልና', 'የሚጠራ ስም ጌታ ይድናልና', 'የሚጠራ ስም
የጌታን ይድናልና', 'ስም ጌታ ይድናልና', 'ስም የጌታን ይድናልና']]
```

The N-grams for the 780 outputs for Example 3.1.1 are more than 1,560. Output 4 shows the N-grams for sentences in Output 3. The flooder produces the result in Output 4 with the Book of Romans as the target corpus to be flooded.

Algorithm 3 shows the flooding process. It takes-in list of files to be flooded and list of the N-grams from Algorithm 2. In every file, each N-gram is searched and when the words in the N-gram are found, the phrase in between the starting and finishing found word of the N-gram is fetched from the line of the file where the words are found.

The Output 4 values given to Algorithm 3 will produce numerous results at line 7 of Algorithm 3. Output 5 shows some sample results out of them all. The lines found could be multiple or single. The phrase retrieved at line 9 of Algorithm 3 could have other words in between. The overlapping system favors those with the least number of not searched words found in between the searched N-grams when calculating the overlap. Not searched refers to the words not in the word and phrase list of Output 1.

Algorithm 3: Flooding Target Files

Input: *File_list* **and** *list_ngrams*
Output: list of flooded phrases from each file
Data: Files to be flooded

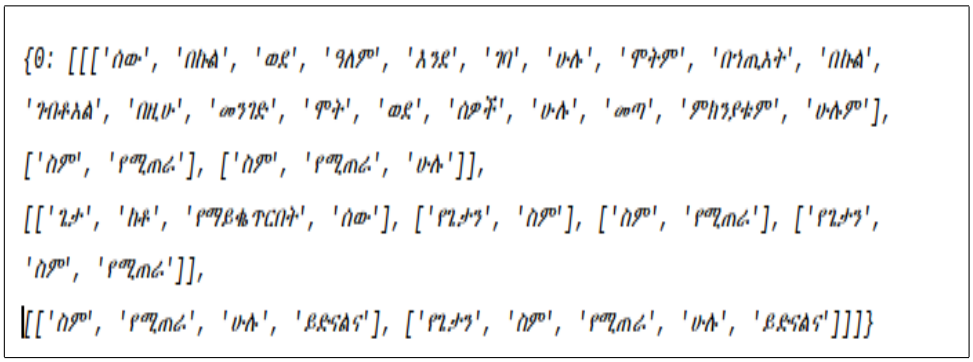
```
1  flooded_phrase_file = []  // holds list of flooded phrases from all files
2  for each file in File_list do
3      flooded_phrase = []    // holds list of flooded phrases from one file
4      for ngrams in list_ngrams do
5          frist_last = []      // holds the phrase between the first and last
           word found
6          for N_gram in ngrams do
               /* search the file for the N_gram and return the lines
                  which have the words in the N_gram                    */
7              line_list = lines_with_N_gram
8              for line in line_list do
                   /* the words in between the first and last words found
                      would make a phrase and be added to the frist_last
                      list                                                 */
9                  phrase = line[first : last]
10                 add phrase to frist_last list
11         add frist_last list to flooded_phrase list
12     add flooded_phrase list to flooded_phrase_file list along with file Id
13 return flooded_phrase_file
```

Output 5: Sample result for *line_list* at line 7 and *phrase* at line 9 of Algorithm 3

			Description
Phrase one	Initial phrase	የማጠፊ ስም የጊዜን	*Flooded phrase*
	Lines found	{288: {2: 'የማጠፊ', 1: 'ስም', 0: 'የጊዜን'}}	*At line 288 found 3 words located at index 0,1,2*
	Phrase from line	['የጊዜን', 'ስም', 'የማጠፊ']	Retrieved phrase from min to max index
Phrase two	Initial phrase	ሰው ጥሪዎች ስም ጊዜ	*Flooded phrase*
	Lines found	{108: {4: 'ሰው', 1: 'ጊዜ'}}	*At line 108 found 2 words located at index 4 and 1*
	Phrase from line	[['ጊዜ', 'ኩቡ', 'የማይቁጥርበት', 'ሰው']]	Retrieved phrase from min index 1 to max index 4 with words found in between

Output 6 shows the final output of Algorithm 3 for all the N-grams searched into a single file.

Output 6: Final output of Algorithm 3 for the N-grams searched into a single file



The first list in Output 6 is for the first list of N-grams in Output 4

3.1.3 N-gram connector

N-gram connector combines the flooded text to find the longest overlap of the translated target text. The system selects the maximum or longest overlapping phrases and merges them to form the final target sentence.

Algorithm 4 loops through the translations of each N-gram from every file. It calculates the overlap between each translation of an N-gram with the next N-gram's translation. It selects the one with the longest overlap.

Algorithm 4: Finding The Maximum Overlapping N-grams

Input: *flooded_phrase_file*

Output: list of long overlapping list of N-grams

```
max_overlap = final_max_overlap = [[], [], 0]
for each file_translation in flooded_phrase_file do
    max_translations = []
    for N_gram_translations in file_translation do
        max_N_gram_overlap_list = []; max_N_gram_overlap = 0;
        for translation in N_gram_translations do
            max_overlap_list = []; max_overlap = 0
            if translation not the last translation then
                for next_N_gram_trans in file_translation[next] do
                    ovrlp = maxm_overlap(translation, next_N_gram_trans)
                    if max_overlap < ovrlp then
                        max_ovrlap_list = [[transltion], [nxt_ngrm_trans], ovrlp]
                        max_overlap = ovrlp
                    if file_translation not the last file's translation then
                        for next_file_next_N_gram_translation in
                        next_file_translation[next] do
                            overlap =
                            maxm_overlap(translation, next_file_next_ngram_translation)
                            if max_overlap < overlap then
                                max_overlap_list =
                                [[translation], [next_file_next_ngram_translation], overlap]
                                max_overlap = overlap
            else if last translation then
                max_N_gram_overlap_list = [[translation], [translation], 0]
                max_N_gram_overlap = 0
            if max_N_gram_overlap < max_overlap then
                max_N_gram_overlap_list = max_overlap_list
                max_N_gram_overlap = max_overlap
        add max_N_gram_overlap to max_translations
    for max_translation in max_translations do
        if max_translation not in max_overlap {max_overlap add
        max_translation}
    final_max_overlap = max_overlap without the overlap number
return final_max_overlap
```

The longest overlap refers to the one which has most of its words found in the next N-gram.

Output 7: Output of Algorithm 4 for line 13 and 25

		Description
Input	*[['ጌታ', 'ሰዶ', 'የማይቈጥርበት', 'ሰው'], ['የጌታን', 'ስም'], ['ስም', 'የሚጠራ'], ['የጌታን', 'ስም', 'የሚጠራ']], [['ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና'], ['የጌታን', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና']]*	
Second phrase and next N-gram overlap calculation	*['የጌታን', 'ስም'] ['ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና'] 1*	*Only 1,'ስም' is found in both*
	['የጌታን', 'ስም'] ['የጌታን', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና'] 2	*2 words,'የጌታን' and 'ስም' are found in both*
Selected maximum overlap for second phrase	[['የጌታን', 'ስም'], ['የጌታን', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና'], 2]	The highest overlap from the above two is selected
Final selected output for the whole input	*[['የጌታን', 'ስም', 'የሚጠራ'],['የጌታን', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና'],3]*	*The highest overlap from all the input phrases is selected*

The last row of Output 7 shows the value for $maximum\_N-gram\_overlap\_list$ at line 25 of Algorithm 4. The row above it shows the output at line 13 for $maximum\_overlap\_list$ of Algorithm 4. Output 8 shows the final output of Algorithm 4.

Output 8: Final Output of Algorithm 4

```
[[['ስም', 'የሚጠራ'], ['ስም', 'የሚጠራ']],

[['የጌታን', 'ስም', 'የሚጠራ'], ['የጌታን', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና']],

[['የጌታን', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና'], ['የጌታን', 'ስም', 'የሚጠራ', 'ሁሉ',
'ይድናልና']]]
```

Output 9 shows the results in Output 8 merged to form the final output.

Output 9: Final Output

Initial list	*Merged within themselves*	*Merged with the next phrase*	*Final merged result*	*In sentence format*
[['ስም', 'የሚጠራ'],	*['ስም', 'የሚጠራ']*	*['የጌታ', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና']*	*['የጌታ', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና']*	*'የጌታን ስም የሚጠራ ሁሉ ይድናልና*
['ስም', 'የሚጠራ']]				
[['የጌታ', 'ስም', 'የሚጠራ'],	*['የጌታ', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና']*			
['የጌታ', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና']]				
[['የጌታ', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና'],	*['የጌታ', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና']*	*['የጌታ', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና']*		
['የጌታ', 'ስም', 'የሚጠራ', 'ሁሉ', 'ይድናልና']]				

Finally, the translation for the Example 3.1.1 English phrase **"everyone who calls on the name of the lord will be saved"** is finally found to be **'የጌታን ስም የሚጠራ ሁሉ ይድናልና'** having gone through the whole process of the CBMT.

Concisely, the CBMT will take the source text and use a bilingual dictionary to translate each word. These translated target words are combined to form phrases of variable length and one overlapping the other. These phrases are called N-grams and the longest N-gram match is searched for in the target corpus. Since there is no need for parallel corpora in this approach, the target corpus need not be translated to source language. The paper, which proposed the method, has not yet implemented the synonym generator proposed for rare words; this thesis implements the synonym finder using WordNet.

3.2 The NMT system

The NMT is one that tries to simulate human brain of neurons to translate amongst languages. The neurons are made to connect with each other and thought to establish connections among things they are being trained to learn about (relation between words in our case).

The NMT learns like a human child. Human children are born without a fully developed skull, as their brain has not yet fully settled on a size. This is because the neurons had not yet made a connection amongst each other to produce a fully functional brain. For these connections or synapses (as they are called) to form, they observe and pay attention to activities around them attentively. This is, briefly, what the neural network attempts to replicate.

The neurons in the cells of a neural network are thought of the relations among objects and trained to make connections among the vectors of these objects. Once they are trained, (the brain has formed) then new information is interpreted based on the connections formed during training.

In this paper RNN (recurrent neural networks) are used than the simple feed forward neural network. In the feed forward network, the neurons are being thought once without a feedback to relearn. However, in RNN, the information is fed to the system again recurrently. The output is feed back to the neuron to learn from both the fresh input and its output, which improves performance as it learns from its previous mistakes. Figure 3.2 shows their difference.

The neural cell used in this paper is the LSTM (the long short-term memory) neural cell. LSTM, introduced by Hochreiter and Schmidhuber, allows cells to forget and remember information[Hochreiter and Schmidhuber, 1997]. There is also another similar proposed methodology called GRU (Gated recurrent Unit)[Cho et al., 2014]. The uniqueness of these methods is, instead of only being aware of their current input, they have the memory of all those that preceded it. In order to do that they have a forget gate and input gate.

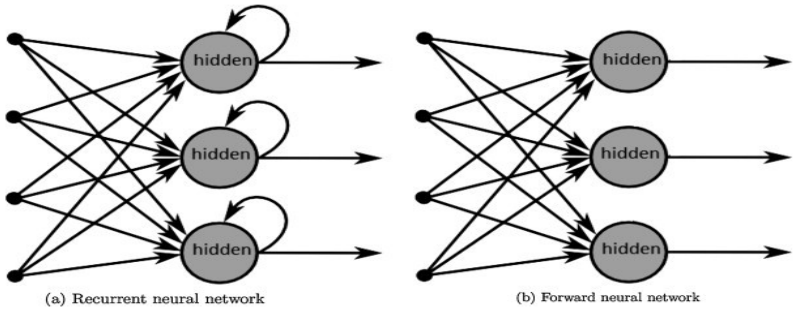
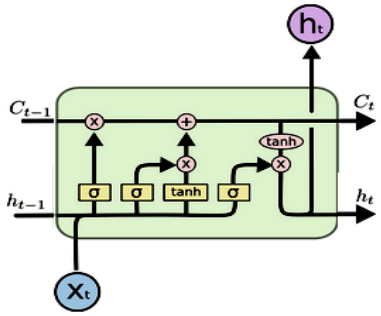


Figure 3.3: Recurrent neural network and feed forward network [De Mulder et al., 2014]



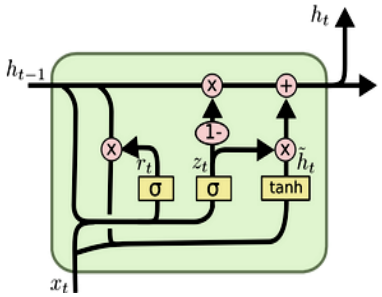
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 3.4: LSTM cell with its computation [Colah, 2015]

The cells in Figure 3.4 have the following functionality: the forget gate (f_t) is used to remove all the unwanted information from the previous inputs. In our case, the information about the word, ten or twenty steps back does not directly affect the current input, so we can forget about it. The current state (C_t) is then updated with the new input (i_t) having forgotten the unnecessary information. Finally the output (h_t) is activated using tan function after passing through the sigmoid computation.

Now this LSTM helps remember words before the current word input but forgets those that are far behind; helping it focus on what is important and forget what does not matter.

In the case of the GRU few memory units are removed, cell state and hidden states are merged, and input gate and forget gate are made into a single update gate. Figure 3.5 depicts the implementation of the GRU.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 3.5: GRU cell with its computation [Colah, 2015]

The GRU may be faster than the LSTM when it comes to smaller data given the smaller number of units[Chung et al., 2014], but it has more or less similar performance in terms of accuracy. Since the paper is focused on performance (accuracy) of the translation, the basic LSTM is chosen. Also LSTM units are more easily accessible than the GRU, which merges some of them.

These LSTM cells are the ones used for both the encoding and decoding of the information in this research. The encoder takes in the input vectors of the words to be translated and formulates a connection among them; encoding. Then this encoded data is fed to the decoder as well as the desired output. The decoder then forms the connection amongst the decoders last output and the desired output. This is the training phase where the neurons within the LSTM cells are being thought of the relation amongst the words to be translated and the final translated output.

After such connections are formed, new input is given to the system and it encodes it similar to that of the training phase. The decoder, however, does not have the desired output fed to it as before, but rather only the last output of the encoder as well as its previous word output (if it is not the first word output) is fed to it. The decoder afterwards has to predict what

the output needs to be based on the training it had been given previously. This phase is called inference or evaluation. Figure 3.6 shows an example of decoding and doing an inference. For the decoding a greedy decoder is used which selects the first fit word that has highest probability of occurrence (probability of being translated and appearing next to the one previous to it).

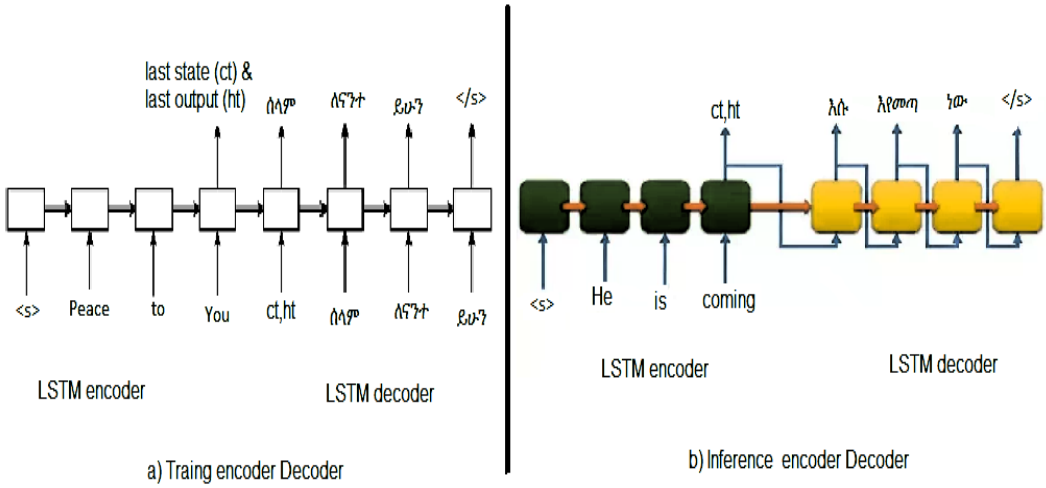


Figure 3.6: Encoder and decoder model for training and inference

The system is further improved by inserting an attention layer among the encoder layer and decoder layer. The attention helps the decoder to look back into the last outputs of the encoder layer and not just the final output of the last neuron. A simple NMT with attention is shown in Figure 3.7.

The attention layer boosts the performance of the decoder by providing further information from the input (the decoders output). Instead of having the last encoded word information, it can look back and see the other encoded words preceding it. It forms a context vector that helps it to contextualize (pay attention) to the most important word for that time step. There are mainly two widely used attention models, Luong attention[Luong et al., 2015] and Bahdanau [Bahdanau et al., 2016] attention model. Their main difference is in how they calculate the score for

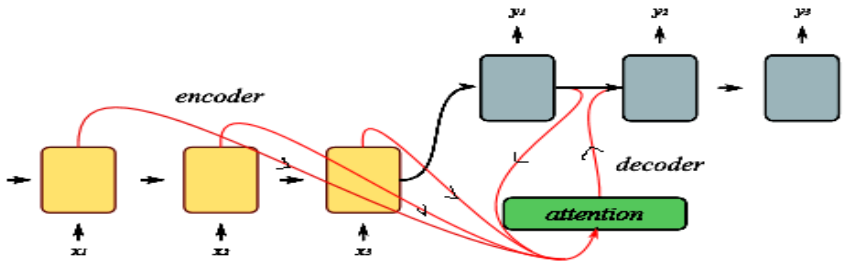


Figure 3.7: Encoder and decoder model with attention layer

each word to form the context vector. For these research the Luong attention model is used because the Bahdanau attention model was designed mainly for bidirectional RNN (recurrent neural networks) while Luong attention model is more general. Equation 3.1 through Equation 3.4 shows Luong attention model's computation.

$$\alpha_{ts} = \frac{exp(score(h_t, \overline{h}_s))}{\sum_{s'=1}^{S} exp(score(h_t, \overline{h}_{s'}))} \quad \textbf{[Attention Weights]} \tag{3.1}$$

$$c_t = \sum_{s} \alpha_{ts} \overline{h}_s \quad \textbf{[Context vector]} \tag{3.2}$$

$$a_t = f(c_t, h_t) = tanh(W_c[c_t : h_t]) \quad \textbf{[Attention Vector]} \tag{3.3}$$

$$score(h_t, \overline{h}_s) = h_t^T W \overline{h}_s \quad \textbf{[Luong's multiplicative style]} \tag{3.4}$$

The score function, calculated using Equation 3.4, is used to compare the output of the decoder (h_t) with the output of the encoder (h_s) in order to find the attention weight calculated using Equation 3.1. The attention weights ($alpha_{ts}$) are then used for the context vector(c_t) calculated by Equation 3.2. This context vector as well as the output of the decoder is then used to produce the final output of the decoder using Equation 3.3.

In summery, the NMT has the following flow; the sentence from source language is inserted into the encoder as a vector, the encoder cell of LSTM will form the connection between the words of the source and provides it to

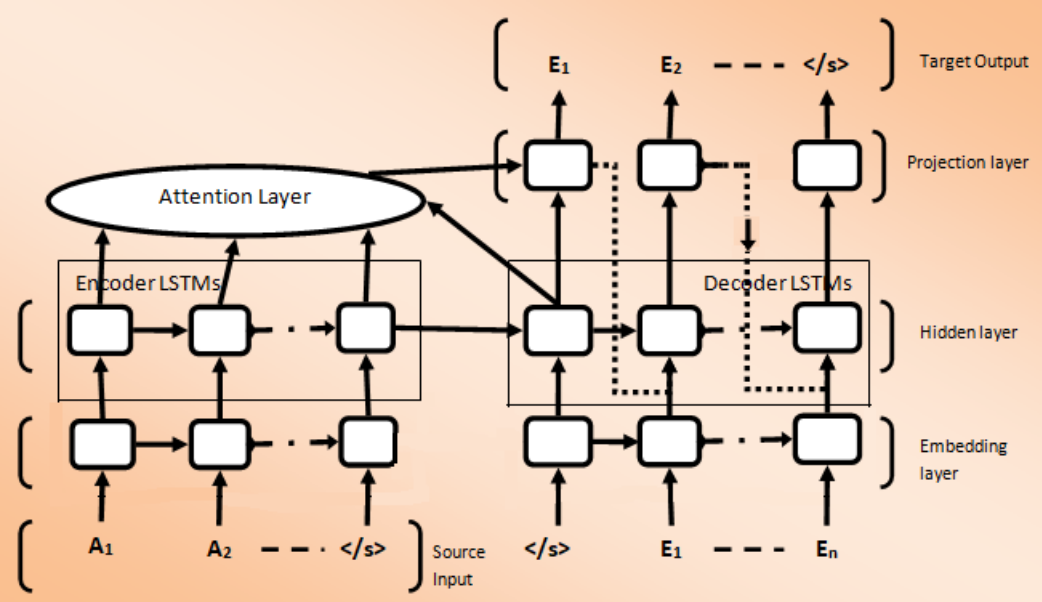


Figure 3.8: Our implementation of the NMT

the decoder and the decoder with the help of the attention layer will produce the most probable translation.

The neural network machine translator used in this research has the components discussed above with the connection shown in Figure3.8. The source input sentence in Amharic with words labeled $A_1, A_2..A_n$ along with the end of sentence tag '</s>' are given to the embedding layer which converts them to vector and hands them over to the encoder with LSTM cells to be encoded. The encoded input is then given to the attention layer. In time of training the expected output in English $(E_1, E_2..E_n)$ is given to an embedding layer and with the help of the attention layer it is decoded by the decoder with LSTM cells and provided to the projection layer. projection layer is responsible for converting the vector back to the final words in English. In the case of inference or testing however the English target words are not given as an input to the decoder.

3.3 The combination of the NMT and the CBMT

To combine the two systems the NMT model must be made to accept two inputs or two sources. This paper uses the proposed method of design with NMT accepting two source inputs by Barret [Zoph and Knight, 2016]. Their paper which has been given the title 'Multi-Source Neural Translation' [Zoph and Knight, 2016] researches the potential of using the translation of a document to different languages and using two inputs of different languages to an NMT. According to their paper, having two inputs, where one is the source sentence and the other a translation of the source to another language different from the target, helps the NMT produce a better result. Figure 3.9 shows their proposed approach.

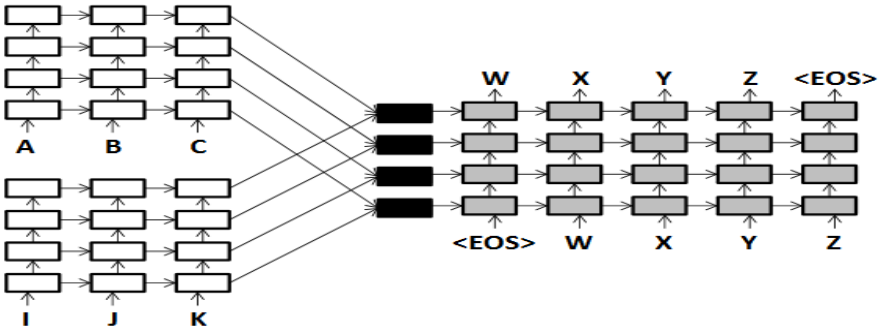


Figure 3.9: Two source input NMT proposed by Zoph and Knight [Zoph and Knight, 2016]

The source sentence and the sentence translated using the CBMT are encoded separately and are given to the attention layer. The attention layer will focus on the two inputs at the same time rather than separately. There will be a single decoder, which receives the output of the attention layer and provides the final translation. An example of our proposed system is shown in Figure 3.10.

Based on the paper [Zoph and Knight, 2016], the final outputs of the

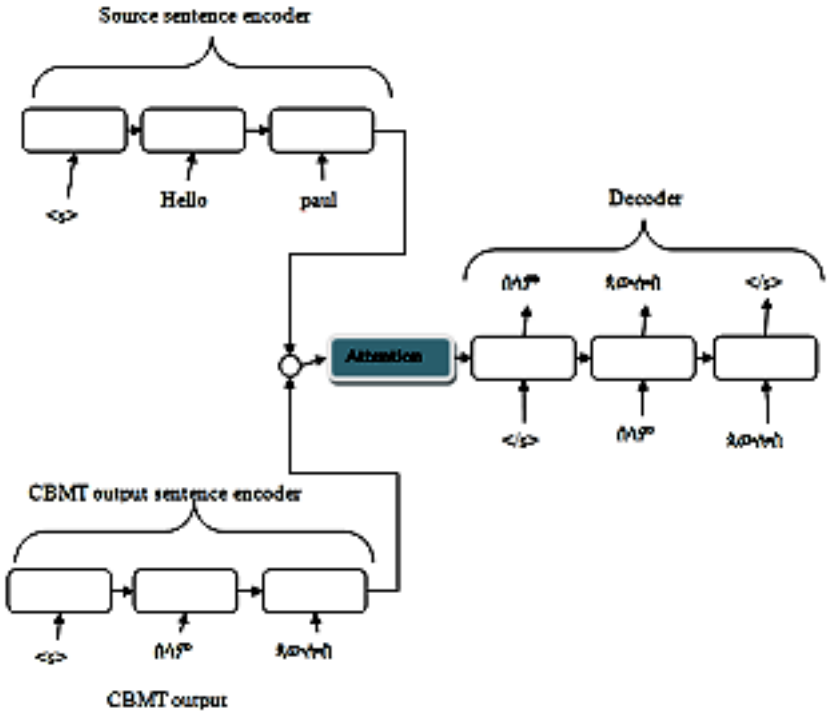


Figure 3.10: Combination of CBMT and NMT

encoders ($\overline{h}_t$) are concatenated and a linear transformation is applied to the concatenated output which is activated by *tanh* using Equation 3.5.

On the other hand, the final states (c_t) are simply added as shown by Equation 3.6. In the attention, the different context vectors are calculated separately and concatenated to produce the final output of the attention vector based on the Luong attention mechanism[Luong et al., 2015] using Equation 3.7.

$$h = tanh(W_c[h_1; h2]) \tag{3.5}$$

$$c = c_1 + c_2 \tag{3.6}$$

$$\overline{h}_t = tanh(w_c[h_t; c_t^1; c_t^2] \tag{3.7}$$

In this paper, the English source and the translated Amharic output of the CBMT will be used to train the NMT.

Chapter 4

Experiments

This chapter explains the procedures and steps taken to implement the proposed methodology. The methodology used to answer the research questions raised at the commencement of the research. To remind of the research questions proposed,

RQ1 [**Performance**] Will the combination of CBMT and NMT be complimentary and perform better than the approaches themselves?

RQ2 [**CBMT impact**] Will the error introduced by the CBMT in the combinational system of NMT and CBMT affect the performance of the system significantly?

The first research question helps answer whether the combination of CBMT and NMT would produce a more contextual and accurate translation than the outputs of the individual methods. It will focus on the translation of the language pairs of English-Amharic given the same size of input data for both the individual methods and their combination.

The second research question is raised to answer whether the combinational system is tolerant to the errors introduced by CBMT or whether the

errors introduced by the CBMT significantly lower its performance. It will evaluate the performance of the combinational system compared to an ideal system given original translation in place of CBMT output.

4.1 Procedure

This section lists the procedures taken in order to answer the raised research questions. They were not strictly followed in the order they have been written here. Some steps were completed in parallel with each other since they do not overlap. However, the list will provide a clear and simple means to understand and follow the experiment, if need be in order to replicate the research. The procedures taken were as follows:

1. Develop the CBMT system based on the initial paper[Miller et al., 2006] to answer the first research question1
2. Prepare a parallel text document in English and Amharic from the New Testament NIV Bible
    - Build a bilingual dictionary to work with the CBMT system
3. Develop an NMT system to be used as a reference to answer research question one1
4. Develop the combination system, the neural network that accepts two inputs, one from CBMT and the other from the source English text
5. Evaluate the performance of each system (the CBMT, the NMT and their combination) to answer the first research question1
6. Compare the obtained results with each other and form a conclusion
7. Evaluate the performance of an ideal combinational system where the inputs are the original Amharic text rather than the CBMT output and the source English text to answer the second research question2

8. Compare the obtained results from procedure seven with results obtained in procedure five and form a conclusion

4.2 Tools used

The main programming language used for the system is python and its libraries, such as NLTK (natural language toolkit). Tensorflow library was used to build the NMT part of the system.

Python : is an interpreted high-level programming language for general-purpose programming created by Guido van Rossum and first released in 1991. Its high-level built in data structures, combined with dynamic typing and dynamic binding; makes it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Both the CBMT and the NMT in this research were built with this language.

Tensorflow : is an open-source software library for data flow programming across a range of tasks. The Google Brain team developed it and released it under the Apache 2.0 open source license in 2015. It is a symbolic math library, and is used for machine learning applications such as neural networks. The NMT in this research uses the Tensorflow libraries throughout.

NLTK : natural language toolkit; it is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text-processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. It also includes graphical demonstrations and sample data sets accompanied by a cookbook. It also has a book that explains the principles behind the underlying language processing tasks that NLTK supports.

4.3 Corpus used

The NMT aspect of the research demands for a well-organized parallel corpus. For the language pairs of this research, English - Amharic, there are few sources of such parallel corpus readily available. Reasonably parallel translation sources between the pair are the Bible and the FDRE constitution. This research uses the Holy Bible.

The research uses the New International Version Bible of all other versions because both the Amharic and English versions are translated from the same Dead Sea scrolls. This makes it more accurately parallel than other versions of the Bible translation.

The whole New Testament of the Bible is used as a corpus providing a total of 8603 phrases and sentences for the research. Two books of the Bible, Paul's letter to the Romans (Romans) and The Gospel according to Mark (Mark), were used as a test data.

Mark is said to be the first Gospel written[Clay, 2018] hence it is selected as a representative of the Gospels, which takes 47% of the New Testaments word count. Mark has a total of 713 phrases and sentences. The book of Romans, a Pauline epistle, is taken as a representative of all the other epistles, which takes 33% of the total word count in the New Testament. Romans is the longest of all the letters and can stand as a showcase for Paul's writing which is 23% of the New Testament[Apologika, 2014]. The book of Romans has a total of 471 phrases and sentences. The combination of Romans and Mark provides a total of 1184 phrases and sentences for testing.

In order to conduct evaluation of the built translation systems, these two books were removed from the New Testament corpus and taken as a standalone text on their own. Testing was also conducted while they were still in the main corpus to evaluate the impact of their presence for the CBMT.

Google translate has been used as the main agent of translation for the manually built bilingual dictionary used by the CBMT. 77% of the total 6,793 vocabulary words have been translated using Google. In addition to Google translate; we have done a manual translation for 1,426 vocabulary words in the book of Romans and other 150 vocabulary words using the Bible.

Manual translation here refers to the translation of each word using every entry it has in the Bible by a human. Figure 4.1 shows the outcome of such translation for the word *Acknowledge*. Manual translation helps to address the variations in Amharic translated words caused by gender (female or male), plural form and the different persons (first, second and third persons). English words do also have different Amharic translations based on their context as shown in Figure 4.1. *Acknowledge* has been translated into four main stem words አወቀ, አሰበ,ተቀበለ and መሰከረ .

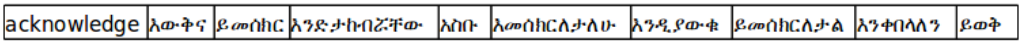


Figure 4.1: Manual translation result of the word acknowledge

Not only single words but also phrases have been translated as well. Figure 4.2 shows English phrases translated to words or phrases in Amharic.

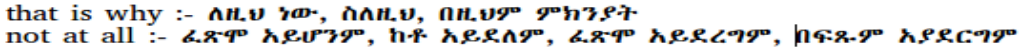


Figure 4.2: Manual translation result of phrases

Dataset

We have fed the same dataset to all systems with minor variations. In the CBMT, we have used the book of Mark and the book of Romans as a test set. The flooded texts for the book of Romans were the book of Romans itself and Paulian epistles without Romans. The flooded texts for the book

of Mark were the book of Mark itself and the gospels without Mark. The books have been flooded to themselves in order to evaluate the performance of the CBMT when the searched text is found in the flooded text and also to see the impact of the bilingual dictionary on the CBMT.

The combinational system has two different test sets and different models. The first test set has the output of the CBMT and the source sentence as an input for the NMT. The second test set gives the original target text and the source sentence as an input to the NMT models and we have called it the ideal approach. This was done so to see the impact of the CBMT output and the errors introduced by the CBMT on the combinational system.

In the basic NMT and combinational system, similar dataset as the CBMT is used. We have used Paulian epistle without Romans to train a basic NMT model and the combinational model. Then we have tested the model using the book of Romans as the test or holdout set. We have used 10-fold cross validation[Dietterich, 1998] to train and test the basic NMT model and the combinational model with Romans. In a similar manner, we have used 10-fold cross validation[Dietterich, 1998] to train the basic NMT model and the combinational model with the book of Mark. We have also used holdout validation[Raschka, 2018] with a random 80% training and 20% test data split alongside the 10-fold cross validation for both Mark and Romans to obtain a more general representation of the results.

4.4 Experiment Setup

We evaluate the proposed approach for the language pair English-Amharic using the same training parameters for both the basic NMT and the combinational system. The encoder-decoder setup has 1024 LSTM cells or hidden units with 1024 word embedding and the data has been trained for 100 epochs.

Epoch is the measure of time the whole data passing though the system. The data is divided into mini batches and iterated trough the system one batch at a time. When all the batches are iterated and the whole data has gone through the system, that is one epoch.

All systems (the CBMT, the NMT and the combination system) were run and tested on Google colab's 12GB RAM online resource. Google colab is a cloud based data science work space provided by Google for free.

4.5 Evaluation method

In this research, the method of evaluation chosen is the BLEU score. It answers whether the machine translated output is close to a reference which is translated by a human.

BLEU or Bilingual Evaluation Understudy when expanded, was proposed to produce a method for the automatic evaluation of machine translation, which is faster, and language-independent[Kishore et al., 2002]. The quality is considered the correspondence between a machine's output and that of a human[Kishore et al., 2002]. BLEU score is defined in the range between 0 and 1 (or in percentage between 0 and 100); where 1 is a perfect match with the reference and 0 is for no word matched.

BLEU uses a modified form of precision to compare a candidate translation against multiple reference translations[Kishore et al., 2002]. The metric modifies simple precision since machine translation systems have been known to generate more words than are in a reference text.

$$p_n = \frac{\sum_{C \in \{candidates\}} \sum_{N-gram \in C} Count_{clip}(N-gram)}{\sum_{C' \in \{candidates\}} \sum_{N-gram' \in C'} Count_{clip}(N-gram')} \quad \textbf{[Precision score]} \tag{4.1}$$

In Equation 4.1, the precision score for the entire document is calculated.

It first computes the N-gram matches sentence by sentence. Then, It adds the clipped N-gram counts for all the candidate sentences and divide by the number of candidate N-grams in the test corpus to compute a modified precision score, p_n, for the entire test corpus. $Count_{clip}$ is calculated as $Count_{clip} = min(Count, MaxRefCount)$. The maximum number of times a word occurs in any single reference translation($MaxRefCount$) is counted and it clips (take the minimum of the two) the total count of each candidate word($Count$) by its maximum reference count($MaxRefCount$), adds these clipped counts up, and divides by the total (unclipped) number of candidate words $Count_{clip}(N-gram')$ [Kishore et al., 2002].

Then, brevity penalty, BP, is computed to avoid penalizing candidate translations, which are longer than the references translation twice.

$$BP = \begin{cases} 1 & \text{if } c > \text{r} \\ e^{(1-r/c)} & \text{if } c \leq \text{r} \end{cases} \quad \textbf{[Brevity penalty]} \tag{4.2}$$

As shown in Equation 4.2, c is the length of the candidate translation and r is the effective reference corpus length. The brevity penalty is 1.0 when the candidate's length is the same as any reference translation's length. In order not to punish short sentences more than long ones, BP is calculated over the entire corpus using decaying exponential in r/c.

Then, the BLEU score is calculated by first computing the geometric average of the modified N-gram precisions, p_n , using N-grams up to length N and positive weights w_n summing to one. Where N is usually taken to be four and $w_n = 1/N$. Then multiplying the result by the exponential brevity penalty factor. Equation 4.3 shows its implementation.

$$BLEU = BP.exp(\sum_{n=1}^{N} w_n \log p_n) \quad \textbf{[BLEU score]} \tag{4.3}$$

Few translations will attain a score of 1 unless they are identical to a reference translation. For this reason, even a human translator will not necessarily score 1[Kishore et al., 2002].

Chapter 5

Results and Discussions

This chapter provides the results obtained along with a brief explanation of the factors. The results are depicted as per the order of the experiments and the final discussion summarizes all results and explains their relation to the research questions.

In order to answer the first research question about **Performance**, the results from CBMT, NMT and the combinational system are first presented in Section 5.1, Section 5.2 and Section 5.3.1 respectively. These results from these sections are compared and the first research question which asks *'Will the combination of CBMT and NMT be complimentary and perform better than the approaches themselves?'* is answered in Section 5.4.1.

In order to answer the second research question about **CBMT impact**, the results from the combinational system and the ideal combinational system are presented in Section 5.3.1 and Section 5.3.2 respectively. These results from these two sections are compared and the second research question which asks *'Will the error introduced by the CBMT in the combinational system of NMT and CBMT affect the performance of the system significantly?'* is answered in Section 5.4.2.

5.1 CBMT Result

The system was tested using a custom-made dictionary using Google translate and manual translation. The vocabulary of the dictionary was generated from the English version of the NIV Bible.

Per the discussion in Section 4.1; at the end of step two the CBMT was ready for testing. Table 5.1 depicts the CBMT test results obtained, using BLEU score evaluation method.

In Tabel 5.1 the column 'Flooded Data' refers to the target corpus given to the CBMT system to flood it with translated phrases.

Table 5.1: Results for the CBMT

	Flooded Data	Test data	BLEU score
1	Paulian epistle without Romans	Romans	27.91
2	Romans	Romans	70.46
3	Mark	Mark	21.98
4	Gospels without Mark	Mark	21.43

We have implemented manual translation for the book of Romans on about 80% of its total vocabulary. Hence it has a better performance yield than the book of Mark, whose translation was solely dependent on Google translate. This was so both when they were flooded to the text that contained them [when Romans was flooded to Romans and Mark Flooded to Mark] (by 48 %) and when they were flooded to the text without them [Romans to Paulian epistles not containing Romans and Mark to the Gospels without Mark] (by 6%) Table 5.1.

However, the translation of Romans did not produce a 100% as would be expected when it was part of the document. This was mainly because the system selects the overlapping N-gram based on the number of words matched, two consecutive phrases that may have a high overlap but which

are not the correct ones may be selected. Figure 5.1 shows the occurrence of such scenario where the obtained result was the one with the highest overlap but was the wrong match.

source	paul an apostle of christ jesus by the command of god our savior
Translated N-gram	[['ጳውሎስ ሐዋርያው ክርስቶስ ኢየሱስ ትእዛዝ', 'ጳውሎስ ሐዋርያ ክርስቶስ ኢየሱስ ትእዛዝ'], ['ሐዋርያው ክርስቶስ ኢየሱስ ትእዛዝ አምላክ', 'ሐዋርያ ክርስቶስ ኢየሱስ ትእዛዝ አምላክ'], ['ክርስቶስ ኢየሱስ ትእዛዝ አምላክ የኛ'], ['ኢየሱስ ትእዛዝ አምላክ የኛ አዳኝ']]
Flooded document result	[[['ኢየሱስ', 'ክርስቶስ'], ['ክርስቶስ', 'ኢየሱስ'], ['ጳውሎስ', 'የክርስቶስ', 'ኢየሱስ', 'ባሪያ', 'ሐዋርያ']], [['ክርስቶስ', 'ጸጋ', 'ከሁላችሁ', 'ጋር', 'ይሁን', 'አሜን', 'በዘላለማዊ', 'አምላክ', 'ትእዛዝ']], [['ክርስቶስ', 'ጸጋ', 'ከሁላችሁ', 'ጋር', 'ይሁን', 'አሜን', 'በዘላለማዊ', 'አምላክ', 'ትእዛዝ']], [['አምላክ', 'ትእዛዝ']]]
Highest overlapping N-gram	['ክርስቶስ', 'ጸጋ', 'ከሁላችሁ', 'ጋር', 'ይሁን', 'አሜን', 'በዘላለማዊ', 'አምላክ', 'ትእዛዝ']
correct match	['ጳውሎስ', 'የክርስቶስ', 'ኢየሱስ', 'ባሪያ', 'ሐዋርያ']

Figure 5.1: Overlap calculation error

When the book of Romans is removed from the New Testament and tested with Romans the result obtained is less accurate than that found when Romans was part of the flooded text by 42% as show in Table 5.1. In the same manner, when the book of Mark was removed from the flooded text and tested, its BLEU score was less than that found when it was part of the flooded text by 0.54% as in Table 5.1. This shows that the dependency of CBMT is largely on the bilingual dictionary than the target corpus.

5.2 NMT Result

The third step mentioned in Section 4.1 was completed to enable the testing of the NMT system. The NMT test results obtained using BLEU score

evaluation method are depicted in Table 5.2.

The validation methods used are 10-fold and Holdout as discussed in Sub-Section 4.3. Figure 5.2 shows the distribution of the 10 results obtained for 10-fold cross validation using box-plots while rows 2 and 3 in Table 5.2 show the average of those results.

Table 5.2: Results for the NMT

	Training input Data	Test data	Validation	BLEU score
1	Paulian epistle without Romans	Romans	Holdout	10.24
2	Romans	Romans	10-fold	11.95
3	Mark	Mark	10-fold	12.42
5	Romans	Romans	Holdout	7.28
6	Mark	Mark	Holdout	10.12

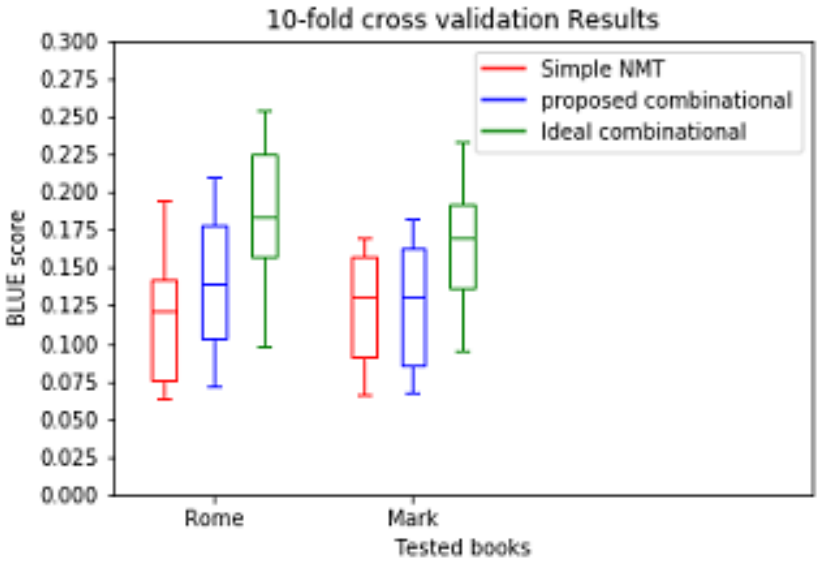


Figure 5.2: Box plot of 10-fold NMT results

The test result obtained from Mark was better than that from Romans by an average of 1.62% (2.84% while using holdout validation and 0.45% while

using 10-fold validation as shown in Table 5.2). Although small difference, it attributes to Marks' writing having similar words unlike the diverse word selection in Romans[Clay, 2018].

5.3 CBMT and NMT Combination Result

There are two test cases for this section, the proposed test case and the ideal test case.

In the first case, we have given the NMT the source sentence and the output of the CBMT as an input per the proposed methodology. Table 5.3 shows the results obtained from such a setup.

In the second case, we have given the NMT the English source sentence and the original Amharic as an input creating an ideal system. The test results are depicted in Table 5.4 for the ideal combinational system.

The validation methods used are 10-fold cross validation and Holdout as discussed in Sub-Section 4.3. Figure 5.2 shows the distribution of the 10 results obtained for 10-fold cross validation using box-plots while rows 2 and 3 in Table 5.3 and Table 5.4 show the average of those results.

5.3.1 Test results using CBMT Output

On the completion of the fourth step mentioned in Section 4.1 the proposed system, combination of NMT and CBMT, was fit for testing. In this section the output of the CBMT is given to the Multi-input NMT as presented in the proposed methodology. Table 5.3 shows the result.

Table 5.3: Results for the combination of NMT and CBMT

	Training input Data	Test data	Validation	BLEU score
1	Paulian epistle without Romans	CBMT output Romans	Holdout	11.55
2	Romans	CBMT output Romans	10-fold	14.07
3	Mark	CBMT output Mark	10-fold	12.36
5	Romans	CBMT output Romans	Holdout	13.84
6	Mark	CBMT output Mark	Holdout	12.73

5.3.2 Test results using the original Amharic Text

Step seven mentioned in Section 4.1 was completed to evaluate the ideal case for the combinational system. Table 5.4 shows the results

Table 5.4: Ideal case Results for the combination of NMT and CBMT

	Training input Data	Test data	Validation	BLEU score
1	Paulian epistle without Romans	original Amharic Romans	Holdout	11.63
2	Romans	original Amharic Romans	10-fold	18.46
3	Mark	original Amharic Mark	10-fold	17.41
5	Romans	original Amharic Romans	Holdout	25.74
6	Mark	original Amharic Mark	Holdout	25.52

In the first case, when the CBMT output is used as an input to the NMT, the Book of Romans performed better than the book of Mark by 1.71% when using 10-fold validation and 1.11% when using holdout validation as shown in Table 5.3. The CBMT output of Romans is better than that by the book of Mark and its impact has propagated to the combinational system. In the ideal case scenario the results are more or less the same. The result for the book of Romans was better than the book of Mark by 1.05% while using 10-fold cross validation and by only 0.22% while using holdout validation.

5.4 Discussion of all Results

The total results obtained from the systems have been collected and shown in Table 5.5 for comparison. CBMT results where they were flooded to the text containing them is not included as it may be biased and represent an ideal system.

Table 5.5: Summary of all results

System	Dataset	Validation	BLEU score	average BLUE
CBMT	book of Romans	-	27.91	**27.91**
	book of Mark	-	21.43	21.43
NMT	book of Romans	Holdout	7.28	9.61
		10-fold	11.95	
	book of Mark	Holdout	10.12	11.27
		10-fold	12.42	
Combinational system with CBMT output	book of Romans	Holdout	13.84	13.95
		10-fold	14.07	
	book of Mark	Holdout	12.73	12.54
		10-fold	12.36	
Combinational system with original text (ideal)	book of Romans	Holdout	25.74	22.1
		10-fold	18.46	
	book of Mark	Holdout	**25.52**	**21.46**
		10-fold	17.41	

The CBMT without being provided the target flooded data has performed better on average by 14.23 BLEU points over the NMT (18.3% for Romans and 10.16% for Mark).

The ideal combinational system, which takes the original target Amharic text as the second input, has performed better on average with 11.34 BLEU score gain over the NMT as shown in Table 5.5 (12.49% for Romans and 10.19% for Mark). The ideal system, however, did not outperform the CBMT on average but produced results in the same range (2.39% worse for Romans

and 0.03% better for Mark).

The combinational system with CBMT output given as the second input for the NMT, achieves on average 2.805 BLEU score point over the simple NMT as shown in Table 5.5 (4.34% for Romans and 1.27% for Mark).

5.4.1 Performance discussion

The first research question about **Performance** asks *'Will the combination of CBMT and NMT be complimentary and perform better than the approaches themselves?'*. With available data the proposed system does perform better than the NMT but does not perform better than the CBMT.

The ideal combinational system has performed better on average with 8.54 BLEU score gain over the combinational system with CBMT output given as the second input for the NMT as shown in Table 5.5 (8.15% for Romans and 8.92% for Mark).

5.4.2 CBMT impact discussion

The second research question about **CBMT impact** which asks *'Will the error introduced by the CBMT in the combinational system of NMT and CBMT affect the performance of the system significantly?'* is answered yes. The performance of the combinational system of NMT and CBMT is significantly affected by the error introduced by the CBMT.

5.5 Threats to Validity

5.5.1 Internal Threat to validity

The selected datasets for testing are the book of Mark and the book of Romans, which are proper samples for the New Testament Bible when seen as a whole as described in Section 4.3. However, the test data fails in representing the books written by other authors, though minor. The book of Acts and the Book of Revelation are the most significant of those left out; as Acts takes 13% and revelation takes 7% of the New Testament[Apologika, 2014].

Epistles by other authors such as Apostle Peter, Apostle John and James and his Brother Jude are insignificant when compared to the Epistles by Apostle Paul in terms of word count. Romans being the largest of all Paul's Epistles, it can be said that the test data more or less represents the Epistles well. Therefore, except Acts and Revelation the Test data set does fully represent the total data set, the New Testament.

5.5.2 External Threat to validity

The test dataset is solely from The New Testament and the research did not consider any other external dataset. The CBMT vocabulary does not support words other than those in the New Testament. Not all the words in the English dictionary are in the current dictionary of the system. Although, the synonym finder helps with such shortcomings, it does not fully address the problem. Especially words specific to a certain field are going to produce erroneous translation.

The failure of the system's dictionary makes it improbable for others to replicate the result for any other data set outside the New Testament. In order to properly reproduce the results, a new set of vocabulary and a well-built bilingual dictionary for the new set of vocabulary is required.

Chapter 6

Conclusion

The research set out to find a system that complements each other when translating a document from English to Amharic. It proposed the CBMT and the NMT to complement each other in terms of parallel data size, accuracy, context with translation and coherence.

From the results found we have the answer to the research questions; which were,

RQ1 [**Performance**] Will the combination of CBMT and NMT be complimentary and perform better than the approaches themselves?

RQ2 [**CBMT impact**] Will the error introduced by the CBMT in the combinational system of NMT and CBMT affect the performance of the system significantly?

The CBMT system performed better than the basic NMT and the combinational system given the same size of data. However, the ideal combination of the CBMT and NMT has a BLEU score in the same range as that of the CBMT while outperforming the simple NMT by 11.34 BLEU points. This entails that with smaller increase in parallel corpus the ideal system will outperform both individual systems.

The output from the CBMT has a great impact on the performance of the combinational systems as seen by the performance of the proposed combinational system compared to the ideal system. The proposed combinational system still has outperformed the NMT by 2.805 BLEU score points in spite of the errors introduced by the CBMT.

Therefore, a well-built CBMT with a well-built bilingual dictionary that produces a close to ideal output makes a fluent combinational system that outperforms a simple NMT and a basic CBMT system.

According to the paper by [Zoph and Knight, 2016] CBMT with good bilingual dictionary performs better than SMT and EBMT. NMT of Google outperforms all PBMT approaches. Hence combination of Google's NMT and a CBMT with strong bilingual dictionary implies a system better for our language pair. Because the data other machine translation researches used for the language pair was not found, a direct comparison can not be made. But the claims on the papers this research is based on ([Zoph and Knight, 2016] and [Wu et al., 2016]) do give us the ground to make such assertions. In conclusion, the study suggests the combinational system for translation of English to Amharic language.

6.1 Future works

The performance of the CBMT could be improved by changing the format of the dictionary. Instead of taking the whole word as a translation; using the stem words may make the system more inclusive for documents not included in the training. Building a dictionary with every English word translated to its possible Amharic stem word and also converting the flooded text into assemblage of stem words is the next step for this research.

Bibliography

[Lan, 2018] (2018). *Innovation and Expansion in Translation Process Research*. John Benjamins e-Platform.

[Apologika, 2014] Apologika (2014). who wrote most of the new testament.

[Arora et al., 2013] Arora, K., Arora, S., and Roy, M. (2013). Speech to speech translation: a communication boon. *CSI Transactions on ICT*.

[Bahdanau et al., 2016] Bahdanau, D., Cho, K., and Bengio, Y. (2016). Neural machine translation by jointly learning to align and translate. *cs.CL*.

[Besacier et al., 2000] Besacier, L., Melese, M., and Meshesha, M. (2000). Amharic speech recognition for speech translation in tourism domain.

[Brown et al., 1990] Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., , and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics Volume 16 Number 2*.

[Cho et al., 2014] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *cs.CL*.

[Chung et al., 2014] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *cs.CL*.

[Clay, 2018] Clay, C. (2018). Comparing the gospels: Matthew, mark, luke, and john.

[Colah, 2015] Colah (2015). Understanding lstm networks.

[De Mulder et al., 2014] De Mulder, w., Bethard, S., and Moens, M.-F. (2014). A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech and Language*.

[Dietterich, 1998] Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*.

[Gangadharaiah, 2011] Gangadharaiah, R. (2011). Coping with data sparsity in example based machine translation.

[Gasser, 2012] Gasser, M. (2012). Toward a rule-based system for english -amharic translation.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. 9:1735–80.

[Irfan, 2017] Irfan, M. (2017). Machine translation.

[John Hutchins, 1995] John Hutchins, W. (1995). Machine translation: A brief history.

[Kishore et al., 2002] Kishore, P., Salim, R., Todd, W., and Wei-Jing, Z. (2002). Blue: a method for automatic evaluation of machine translation. *Association for Computational Linguistics*, pages 311–318.

[Labaka et al., 2014] Labaka, G., España-Bonet, C., i Villodre, L. M., and Sarasola, K. (2014). A hybrid machine translation architecture guided by syntax. *machine translation*.

[Luong et al., 2015] Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine tr anslation. *cs.CL*.

[Miller et al., 2006] Miller, D., Carbonell, J., Klein, S., Steinbaum, M., and Grassiany, T. (2006). Context-based machine translation. *The Association for Machine Translation of the Americas (AMTA-2006).*

[Niehues et al., 2016] Niehues, T. H. J., Cho, E., and AlexWaibel (2016). Pre-translation for neural machine translation.

[Oladosu et al., 2016] Oladosu, J., Esan, A., Adeyanju, I., Adegoke, B., Olaniyan, O., and Omodunbi, B. (2016). Approaches to machine translation: A review. *FUOYE Journal of Engineering and Technology*, 1:120–126.

[Popovic, 2017] Popovic, M. (2017). Comparing language related issues for nmt and pbmt between german and english. *The Prague Bulletin of Mathematical Linguistics No. 108, 2017, pp. 209–22.*

[Precup-Stiegelbauer, 2012] Precup-Stiegelbauer, L.-R. (2012). Automatic translations versus human translations in nowadays world. *Akdeniz Language Studies Conference 2012.*

[Raschka, 2018] Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning.

[Soergel, 1998] Soergel, D. (1998). Wordnet. an electronic lexical database.

[Tadesse and Mekuria, 2000] Tadesse, A. and Mekuria, Y. (2000). English to amharic machine translation using smt. *Master's thesis, Addis Ababa University.*

[Taye et al., 2015] Taye, G., Gebreegziabher, M., Besacier, L., and Teferi, D. (2015). Phoneme based english-amharic statistical machine translation. *AFRICON 2015*, pages 1–5.

[Teshome, 2000] Teshome, E. (2000). Bidirectional english-amharic machine translation: An experiment using constrained corpus. *Master's thesis, Addis Ababa University.*

[Wu et al., 2016] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G. S., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

[Yulianti et al., 2011] Yulianti, E., Budi, I., Hidayanto, A. N., Manurung, H. M., and Adriani, M. (2011). Developing indonesian-english hybrid machine translation system. *2011 International Conference on Advanced Computer Science and Information Systems*.

[Zewgneh, 2017] Zewgneh, S. (2017). English-amharic document translation using hybrid approach. *Master's thesis, Addis Ababa University*.

[Zhang et al., 2017] Zhang, J., Zhou, L., Hu, W., and Zong, C. (2017). Neural system combination for machine translation. In *ACL*.

[Zoph and Knight, 2016] Zoph, B. and Knight, K. (2016). Multi-source neural translation. *cs.CL*.