Improving Sinhala – Tamil Translation through Deep Learning Techniques

Anupama Sandamini Arukgoda

Improving Sinhala – Tamil Translation through Deep Learning Techniques

A.S. Arukgoda
Index No: 14000075
Supervisor: Dr. A.R. Weerasinghe

January 2019

Submitted in partial fulfillment of the requirements of
B.Sc in Computer Science Final Year Project (SCS4124)

UCSC

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: A. S. Arukgoda

………………………………………………

Signature of Candidate Date:

This is to certify that this dissertation is based on the work of Ms. A. S. Arukgoda under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Principle/Co- Supervisor's Name: Dr. A. R. Weerasinghe

………………………………………………

Signature of Supervisor Date:

Abstract

Neural Machine Translation (NMT) is currently the most promising approach for machine translation. Many languages have successfully achieved their state-of-the art translation accuracy with NMT. But still, due to the data-hungry nature of NMT, many of the low-resourced language pairs struggle to apply NMT and generate intelligible translations. Additionally, when the language pair is morphologically rich and also when the corpora is multi-domain, the lack of a large parallel corpus becomes a significant barrier. This is because morphologically rich languages inherently have a large vocabulary, and inducing a model for such a large vocabulary requires much more example parallel sentences to learn from. In this research, we investigated translating from and into both a morphologically rich and a low resourced language pair, Sinhala and Tamil.

To address the morphological richness, as proposed by previous work we have analyzed different sub-word segmentation techniques. We conducted a detailed analysis on these techniques with Sinhala and Tamil which also helped us learn some linguistic properties of the two languages. Furthermore, to address the scarcity of data, we employed one of the most popularly used techniques called back-translation and analyzed its applicability on the translation of Sinhala and Tamil languages in both directions. In this process we designed a new language-independent technique that performs well when the monolingual sentences are limited and could support the translation of one direction on the translation of the other direction, given two languages.

Through the course of our experiments we were able to gain an improvement of approximately 11 BLEU points for Tamil to Sinhala translation and an improvement of 7 BLEU points for Sinhala to Tamil translation over our baseline systems. Being a challenging language pair that has not been explored with NMT before with an open-domain data set, the above improvement is statistically significant and contributes towards automatic translation between these two languages.

Preface

Neural Machine Translation has not been applied for open-domain Sinhala-Tamil translation. By using the parallel corpus of 25000 sentences and monolingual corpora of Sinhala and Tamil accumulated by previous research work, we have analyzed the effect of NMT on this dataset. First, we identified the two main factors that make the translation between the two languages under consideration challenging. They were the morphological richness of the two languages and the lack of the amount of available parallel sentences. These two challenges were treated separately. We first analyzed the effect of different preprocessing techniques on Sinhala-Tamil translation, and drew conclusions based on our observations and research work reported for other languages. I carried out the analytical calculations of the observations of different experiments presented in Chapter 5, in conjunction with my supervisor.

Next we explored the applicability of two back-translation techniques proposed in the literature, and introduced a new, improved, language-independent back-translation technique that empirically showed better translation accuracy for Sinhala and Tamil. This is a new technique that contributes to the originality of the paper.

We created a Finnish-German parallel corpus to check the effect of the size of the training data on the translation accuracy to justify our efforts on finding corpus-driven techniques to improve translation accuracy rather than investing our time on increasing the parallel corpus size via manual translation. To the best of our knowledge, such an analysis has not been conducted on the languages Finnish and German before.

With constant guidance and supervision of my supervisor more conclusions were drawn about the translation of Sinhala and Tamil, which we believe, are new contributions to the body of knowledge.

Acknowledgement

Foremost, I would like to express my sincere gratitude to my supervisor, Dr. Ruvan Weerasinghe his expertise, enthusiasm, patience, motivation and encouragement. His guidance helped me in every aspect of my undergraduate research study and also to build up my career as an excellent researcher. Without his guidance and support I would not have been able to complete my work and dissertation.

I would also like to thank all the staff and colleagues at the UCSC and UCSC’s Language Technology Research Laboratory, for their support and input to make this research a success.

A special thanks goes to my parents and my family members for their patience and continuous moral support. Without them my effort would have been worth nothing. Their love, support, patience inspires me to overcome all the obstacle in life and achieve goals with success.

Table of Contents

List of Figures

List of Tables

Acronyms

BPE	Byte Pair Encoding
BAMorfessor	Boundary-Aware Morfessor
De	German
Fi	Finnish
GNMT	Google Neural Machine Translation
MT	Machine Translation
NMT	Neural Machine Translation
OOV	Out of Vocabulary
RNN	Recurrent Neural Network
SI	Sinhala
SMT	Statistical Machine Translation
TA	Tamil
UCSC	University of Colombo School of Computing

Chapter 1 - Introduction

1.1 Background to the research

Human beings naturally prefer their native language over a foreign language. In an English-dominating world, the concept of translation between languages is of utmost importance to move forward at the world's pace while preserving the native languages. The journey of Machine Translation, starting from the early 1930's up until now, is truly fascinating. Neural Machine Translation (NMT) represents a significant step-forward over a basic statistical approach, therefore is considered as the state-of-the-art for Machine Translation. While NMT has been explored in translating European Languages in large context, showing promising results, it has not been explored on Sinhala – Tamil open domain translation. Therefore, through this undergraduate research opportunity, it is our attempt to exploit this branch of Machine Translation with Deep Learning with the hypothesis that NMT can improve the quality of Sinhala – Tamil translation and thereby contribute to the effective communication between Sinhala and Tamil communities in Sri Lanka.

Sinhala and Tamil are the national languages of Sri Lanka. They are both morphologically rich and less-resourced languages. While a considerable amount of research has been carried out to translate between European languages (especially with English as a pivot language), the research on translation between two morphologically rich languages is still limited. Research reported in the literature for Sinhala-Tamil machine translation is also very limited. If we consider the properties of Sinhala and Tamil, they are both morphologically rich, they are minority languages (languages of the minority community in the world), they are low resourced (limited number of parallel corpora available) and have limited or no publicly available linguistic resources such as POS taggers and morphological analyzers. All the above properties make our task at hand more challenging.

In an early research in SMT for Sinhala and Tamil translation [1], it has been shown that due to the co-evolution of the Sinhalese and Tamils in Sri Lanka, the linguistic distance between Sinhala and Tamil is less than that between Sinhala and English, thereby making the translation between Sinhala and Tamil theoretically easier than that of Sinhala and English. In addition, the two languages Sinhala and Tamil, are head-final languages which also provide the flexibility to alter the word order.

It is our aim to exploit these commonalities between Sinhala and Tamil and design the most optimum technique to translate between Sinhala and Tamil and contribute to the improvement of the accuracy of Sinhala - Tamil translation.

The most recent research on translating between morphologically rich languages [2], has provided a promising foundation for Sinhala - Tamil machine translation, producing the best Sinhala - Tamil translator to-date based on statistical machine translation. The morphological modifications suggested in this research, which integrates morphological information as suggested in a previous research on Sinhala morphological analysis [3], has successfully increased the quality and the reliability of the Sinhala - Tamil translation.

Neural Machine Translation (NMT) is a new paradigm in Machine Translation. Within a very short period of time, it has surpassed the performance of Phrase Based Machine Translation systems (PBMT). NMT systems have stronger generalization power as they encode the source sentences as numeric vectors that represent the syntax and semantics whereas in PBMT the translation, units are encoded as strings. Also, NMT provides a more direct translation mechanism from source language to the target language, whereas PBMT consist of a collection of models (translation model, language model, reordering model etc.) which are trained separately and combined later. Therefore, from an architectural point of view, NMT is much simpler [4]. Moreover, NMT systems are capable of modeling longer dependencies thanks to the recurrent neural network (RNN) encoder decoder model originally proposed in [5].

NMT does have its drawbacks since it is less transparent and requires large data-sets. Therefore, we first analyzed the effect of data-size in NMT by translating another such morphologically-rich language pair to understand whether investing time on inventing techniques to translate optimally the available amount of parallel sentences is better than increasing the parallel corpus size via manual translation.

Our attempt is to explore Deep Neural Network models for Sinhala - Tamil translation, identify their accuracy, compare our results with the previous work and thereby propose a more general-purpose method to translate between other such agglutinative language pairs.

1.2 Research Questions

1.2.1 What is the effect of corpus size on the translation accuracy?

Sinhala and Tamil are two languages that are truly low-resourced (size of the parallel corpus is 25k) and are considerably morphologically-rich in nature. Before we provided solutions for the challenging factors in Sinhala and Tamil specifically, we needed to understand how the size of the training parallel corpus affects the translation quality. Through this research question we expected to understand how a language pair similar to Sinhala and Tamil performed with different training dataset sizes. By addressing this research question, we attempted to quantify the amount of parallel sentences (a lower-bound) required to get an acceptable translation accuracy for two languages and to understand whether investing our time on manual translation is better or worse than inventing NMT and corpus driven techniques to improve the overall translation. The answer to this question validated our efforts and methodologies adopted to address the next research question.

1.2.2 What is the accuracy of Sinhala - Tamil translation that can be achieved with NMT when compared to SMT?

As mentioned earlier, the currently available best, open-domain Sinhala-Tamil translator is based on Statistical Machine Translation. Over the years, NMT has improved the state of the art in many machine translation settings [6]. But NMT has shown to be notoriously weak for small amount of parallel data. In addition, NMT is challenged in the face of morphologically rich languages. NMT is said to have lower quality in out-of-domain translations. This is because in different domains and even within domain depending on the context, words tend to have different translations and meanings [7]. Therefore, researches have been conducted to address these challenging settings separately.

Sinhala and Tamil display many of the above challenging properties. Our parallel corpus of 25000 sentences is considerably small in the context of NMT. The morphological richness of these two languages and the fact that our parallel corpus is open-domain worsens the adverse effects imposed on low-resourced NMT. Hence by addressing this research question we will be able to analyze the translation accuracy of Sinhala – Tamil with NMT with respect to the performance of SMT on the same corpora.

1.2.3 Project Goal and Objectives

Our main goal is to design a technique using Deep Learning, which will improve the translation quality of Sinhala and Tamil translation produced by SMT on the same corpora. To achieve this, we explored the suitability of multiple techniques/Deep Learning models both using our parallel corpus as well as monolingual corpora for the two languages.

In this attempt, we further aim to,

- Find the effect of using corpus size on the quality of the translation with NMT.

This objective will be achieved by addressing our first research question through which we try to define an upper-bound on the corpus-size required for a similar language pair, to achieve considerably good translation. For this we have employed a large parallel corpus of 2 morphologically rich parallel corpus which was designed through this research work.

- Find the accuracy of the designed model and compare the results with previous work for Sinhala – Tamil machine translation.

The Sinhala-Tamil translation accuracy will be compared against the results obtained in [2], which has conducted a Tamil to Sinhala translation with SMT. We will be using the same corpora used by this research to compare the performance of SMT and NMT when using the same corpus.

- Introduce a language-independent translation technique applicable for other such low-resourced languages.

After analyzing the applicability of some selected techniques proposed in the literature, we also aimed to invent a new translation technique that would be language-independent and prove its effectiveness using Sinhala and Tamil.

1.3 Justification for the research

Sinhala and Tamil are two languages that have been proven to be challenging to translate with machine translation. Over the years, this task has been attempted, with the state-of-the-art at the time, which is Statistical Machine Translation (SMT). Currently, Neural Machine Translation is rapidly proving itself to be a strong competitor to SMT methods. But the analysis of performance of NMT for an open-domain translation between Sinhala and Tamil is yet to be explored. In addition, there is very limited research on applying NMT to translate two morphologically rich languages. Therefore, through our research we aim to address this gap in the body of knowledge.

Neural models have had recent success in machine translation with the advent of deep layered architectures, and are said to produce translations of higher accuracy than that was possible using previous techniques. However, the accuracy of these models still depend on a number of factors such as availability of linguistic resources, availability of large parallel corpora, the complexity of the languages, and the linguistic distance between the two languages under consideration among others. Therefore, by analyzing the translation quality of Sinhala - Tamil translation with Neural Machine Translation, we can compare the quality of the currently available Sinhala - Tamil translators based on SMT and identify the validity of the neural models across languages. It is also expected, that this study, will contribute to the improvement of information exchange and reduce the misunderstandings between the Sinhala and Tamil communities in Sri Lanka.

1.4 Methodology

1.4.1 Addressing the First Research Question

1.4.1.1 Corpus details

To address the first research question, that is to understand the effect of corpus size on the translation quality we chose another two such morphologically rich language pair with a large parallel corpus. The chosen language pair is Finnish and German. A parallel corpus between these two languages was not publicly available hence we created it by mapping the Finnish-English and German-English Europarl corpora in WMT18[1].

1.4.1.2 Experimental Methodology

By using one of popularly used NMT models [6], we find the BLEU score for samples of training datasets of different sizes and plot their translation accuracy against the size. This process will be conducted until a saturation of the BLEU scores is observed. Then the plotted graph will be extrapolated assuming they will continue perform at the same rate. This way, we will be able to identify the minimum corpus size required for an acceptable Sinhala - Tamil translation accuracy.

1.4.2 Addressing the Second Research Question

1.4.2.1 Corpora

Sinhala and Tamil parallel data is limited; hence they are considered as low resourced. For our experiments we use a parallel corpus of approximately 25000 sentences (referred to as 25k), which has a sentence length between 8 and 12, collected in the research [2].

There are many techniques suggested in the literature to improve NMT of low-resourced languages. Many of them include incorporating monolingual corpora in many ways. We exploited the applicability of such methods for Sinhala – Tamil translation as well. Therefore, in our experiments to improve Sinhala-Tamil translation accuracy, a 10-million-word monolingual corpus [8] and on the Tamil end, a 4.3-million-word Sri Lankan Tamil monolingual corpus [9] were used. Both these corpora are suitable for an open-domain translation as they have been collected from sentences from different domains such as newspaper articles, technical writing and creative writing.

[1] http://www.statmt.org/wmt18/translation-task.html

1.4.2.2 Baseline model

We will be evaluating the accuracy of translation on Sinhala to Tamil translation as well as Tamil to Sinhala translation.

Tamil to Sinhala Translation

To compare the Tamil to Sinhala translation quality obtained in our work, we will be using two baseline models.

01. Phrase-based SMT model where fully morpheme-like segmented units is considered as the smallest unit [2], which is the currently available best translator for Sinhala and Tamil.
02. We follow the neural machine translation architecture by [10]. A word-level neural machine translation system is implemented as an encoder-decoder network with recurrent neural networks. The model will be trained using the parallel corpus of 25k.

Sinhala to Tamil Translation

Only a single baseline model was used to evaluate this translation direction as the research work of [2] has not been conducted for Sinhala to Tamil translation direction.

01. Similar to the second baseline model for Tamil to Sinhala translation, except this model is translated from Sinhala to Tamil.

1.4.2.3 Experimental Methodology

To achieve our main goal, first we experimented with only the parallel corpora. For any language pair, the effectiveness of a MT system depends on 2 major factors. The availability and size of parallel corpus used for training, and the syntactic divergence between the two languages, i.e. morphological richness, word order differences, grammatical structure, rare words etc. Since both these factors are unfavorably affecting the translation of Sinhala and Tamil, we first focus on the morphological richness of the two languages and explore the parallel corpora by constructing word representations compositionally from smaller sub-word units, which occur more frequently than the words themselves. These representations are expected to be effective in handling rare words and are expected to increase the generalization capabilities of neural MT beyond the vocabulary observed in the training set.

Here we have considered 3 forms of sentence representations.

1. Full word-form sentences
2. Fully morphologically segmented sentences
3. Segmenting using BPE

The next phase would be to focus on the requirement of large datasets for the success of NMT. This is a major challenge in the context of Sinhala and Tamil translation as they are both low resourced languages. Therefore, we intended to explore different means of addressing this challenge provided in the literature, those would make use of the parallel corpora mentioned earlier, and attempted to incorporate monolingual corpora of Sinhala and Tamil that are much larger as suggested in [11, 12, 13]. For these experiments we used the representation that has shown the best performance in the above set of experiments.

Throughout the research, the above-mentioned data-sets will be explored on different neural models, post processing the output and evaluating their accuracies with the BLEU score, to accomplish the best possible translation accuracy for the Sinhala – Tamil pair.

1.5 Scope including delimitations

1.5.1 In-Scope

- To evaluate the effect of the corpus size in NMT using two other morphologically rich languages.
- Designing an optimum technique for Sinhala - Tamil translation with Deep Learning.
- To propose a new technique that could be adopted in the translation of other such low-resourced languages.

1.5.2 Out-Scope

- When exploring different architectures, rather than beginning with generic networks, the most popular architectures proposed for NMT [6] were explored, and then progressively fine-tuned them to reach higher BLEU scores.
- Literature has proposed many techniques to be applied under low resourced settings, ranging from supervised techniques to transfer learning techniques till unsupervised techniques. But we will be exploring a selected genre of supervised techniques only (back-translation), given the time constraints.

1.5.3 Delimitations

- The evaluation of the translation quality was measured using the BLEU score. Being an automatic evaluation metric, BLEU has its inherent drawbacks. The same translation scored with the BLEU score could be given a higher score by a linguist since BLEU does not consider synonyms, and could penalize acceptable changes in the word order etc. The ideal approach will be to evaluate the translation by human translators. But this is time consuming and expensive, therefore our sole indication of the translation accuracy will be the BLEU score.

1.6 Thesis Outline

The thesis is organized as follows.
We present a comprehensive review of the NMT approach and specifically work reported in the literature on the translation between morphologically rich languages and low-resourced languages (Chapter 2). We also cover the work conducted so far for Sinhala-Tamil translation, highlighting the research gap that has been addressed by this research in this chapter.

The research design together with the high-level architecture for addressing the first and the second research questions raised in this research are given in Chapter 3, while Chapter 4 presents the implementation aspect of the different techniques and frameworks used together with the relevant validation and testing methods followed.

We compare the results obtained by following the different approaches that we have employed in this research with a detailed analysis in Chapter 5, and summarize the research findings, conclusions drawn and future work that can be done based on the current findings in the last chapter.

1.7 Conclusion

This chapter laid the foundations for the dissertation. It introduced our general focus area and the more specific research problem and research questions and hypotheses. Then the research was justified analyzing the significance of the research, the methodology was briefly described (a comprehensive description is presented in Chapter 3) and justified, the dissertation was outlined, and the limitations were given. On these foundations, the dissertation can proceed with a detailed description of the research.

Chapter 2 - Literature Review

In this chapter, we lay out a brief yet comprehensive description of the theoretical background to understand this work in depth. The thesis analyzes different techniques to improve the translation between the morphologically rich and low resource language pair Sinhala and Tamil using NMT. We begin with an introduction to the broad context of Neural Machine Translation, the different treatments proposed in the literature for languages that are morphologically rich, followed by the treatments proposed for languages that are low resourced, identify the work conducted for Sinhala-Tamil translation over the years and end with a description of the metric used to measure the translation accuracy and its potential disadvantages.

2.1 Neural Machine Translation

Deep Neural Networks have achieved excellent performance improvements in different learning tasks. Although DNNs work well with large labeled training data on classification tasks, they were not possible to be applied on sequence-to-sequence problems. Machine translation is a sequence prediction problem. Not only both input and output are sequences, they are sequences of different lengths which made the task more challenging. But the pioneering work of [5, 10] presented an end-to-end sequence learning approach that makes minimal assumptions on the sequence length and structure, outperforming the traditional phrase-based translation systems. Unlike traditional SMT systems which consist of many sub-components such as the translation model, the Language Model etc. that should be trained separately, NMT proposes a method to train a single, large neural network that reads a sentence and directly outputs the corresponding translation. Currently, Neural Machine Translation is the state-of-the-art technique for Machine Translation for many languages.

The most popular architecture for NMT is the encoder-decoder architecture. As explained in the work of [5], a neural network (ideally a Recurrent Neural Network) performing as an encoder reads and encodes an input source-language sentence into a fixed-size vector which is also known as the context vector. The decoder, which is another neural network (ideally a Recurrent Neural Network), can be considered as a conditional recurrent Language Model which decodes the translation from the encoded vector. The encoder-decoder will be jointly trained to maximize the conditional likelihood on the bilingual training data.

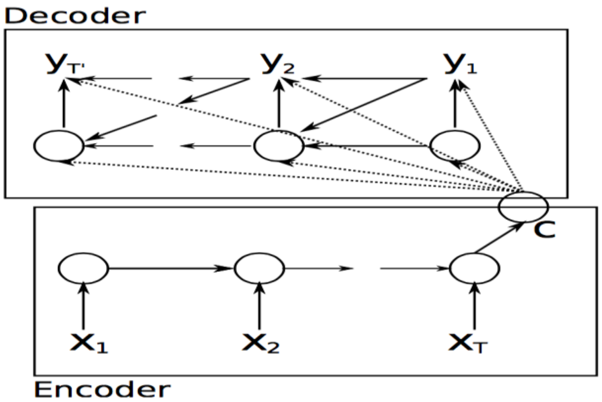


Figure 2.1: Encoder - Decoder Architecture

Though this approach was a breakthrough at the time, a potential issue in the proposed encoder-decoder architecture was, the encoding neural network compresses all the information of a given source sentence into a fixed length vector, regardless of its length. This made it difficult for the neural network to deal with long sentences. It has been empirically shown that NMT performs well on short sentences but its performance degrades rapidly as the length of the sentences increase [14].

This issue was addressed in the work of [15]. Their work introduces an extension to the encoder-decoder model which learns to align and translate jointly. Furthermore, in this work they have encoded the dependencies of a sentence from left to right as well as from right to left. That is, each time a word is translated, it checks for the set of positions in a source-sentence where the most relevant information is concentrated. Then the model predicts a target word, based on the context vectors associated with these source positions and all the previous generated target words. This approach does not attempt to squash a whole input sentence regardless of its length, into a fixed size vector. This mechanism is also known as ‘attention’. This extension significantly improved translation performance over the basic encoder-decoder approach.

Since then, much research work has been conducted using these attentional encoder decoder approach. A noticeable feature among these works is that they have been mostly explored on European languages [6]. The reason for the translation of these languages to reach such a stage of proliferation is the availability of large parallel corpora. The performance of the existing neural models were poor for under-resourced languages such as Sinhala and Tamil. Thus, translating Sinhala and Tamil is challenging.

NMT is not without its own challenges. In [7] the authors have analyzed the challenges NMT faces under six aspects. Their findings report that NMT performs poorly in open domain conditions. In the face of low amount of training data, NMT is said to produce translations of very low quality. This has also been observed in [4], which states that the SMT remains to be the best option for low-resourced settings. Rare words, length of the sentences directly affects the translation quality of NMT and the fact that NMT systems are less interpretable due to its decoding choices being buried in high dimensional matrices makes NMT less appealing.

2.2 Translating Morphologically Rich Languages

A Morphologically Rich Language (MRL) is one which grammatical relations like Subject, Predicate, Object, etc., are indicated by changes to the words instead of relative position or addition of particles. Translating between morphologically rich languages is still uncommon and is challenging. However, translating from a morphologically rich language to English and vice versa has been studied in large context. Dealing with morphologically rich languages is an open problem in language processing as the complexity in the word forms inherent to these languages makes translation complex. A common technique to address this, is to integrate morphological information. It has been explored under both SMT [17] and NMT contexts and has shown promising improvements. But this technique limits itself to be applicable for those languages that have linguistic resources, which is not a luxury available for Sinhala and Tamil.

Most of the machine translation systems are trained using a fixed vocabulary. But translation itself is an open vocabulary problem. Therefore, having to deal with out-of-vocabulary words, and rare words is unavoidable. If the translating languages are low-resourced (size of the parallel corpora is small), this problem is worsened because of the increased size of the vocabulary. Hence, translation mechanisms that go below the word level have been explored.

2.2.1 Byte-Pair-Encoding (BPE)

A simple yet effective technique was proposed by [18] to represent the rare words as a sequence of sub-words. In this work, the compression algorithm Byte-Pair-Encoding (BPE) has been tuned to merge the most frequent pair of characters iteratively.

The training is done using two vocabularies: training vocabulary and symbol vocabulary. As the first step, all the words are segmented into characters and the characters are added to the symbol vocabulary. This step is done recursively merging the most frequent symbol bigram to the vocabulary, and in each step all its occurrences are replaced by a new symbol (merged symbol bigram). This is repeated for a number of times which is the only parameter that should be defined by the user.

Selecting this hyper-parameter (no. of merge operations to be used), depends on both the language and the size of the corpus. It needs to be decided on a trial and error basis. A very low value for this parameter would lead to a character-level segmentation, where as a very high value would lead to word-level representation. Regardless of the simplicity of this technique, BPE has become the state-of-the-art preprocessing technique for NMT.

With the same intuition, researches have been conducted suggesting a character-level representation [25] of the sentences. But both these approaches tend to generate longer input sequences, thus exacerbates the handling of long-term dependencies. Also, BPE exploits only statistical information. Therefore, approaches that capture the semantics or morphological information were more desirable and suitable for some languages.

2.2.2 Morfessor

Morfessor [19], is an algorithm that works in an unsupervised manner to extract morpheme-like segments from a raw, un-annotated corpus without using any linguistic knowledge. This was mainly developed for languages which are complex and have concatenative morphology such as Finnish and Turkish. It aims to generate the most probable segmentation of words to their prefix, suffix and stem by relying on the Minimum Description Length. This approach has been heavily explored in both SMT and NMT research work and has shown promising performance improvements for many languages. Since there are no publicly available linguistic resources for Sinhala and Tamil, and also since this technique had shown promising performance improvements for Sinhala-Tamil machine translation [2, 3] we have employed Morfessor algorithm in our research. We have also empirically compared the effect of morpheme-like units and BPE on the translation quality in our setting.

2.3 Translating Low-resourced languages

Similar to many other deep learning tasks, the success of NMT is strongly dependent on the availability of large parallel corpora. Since this is a luxury many of the languages (specially minority languages) do not have, many techniques have been proposed over the years to address this. Let us analyze such approaches that have become popular over the years.

One of the first researches reported to incorporate monolingual corpora is [11]. The intuition is that even though it is quite difficult to obtain parallel corpora for two languages, it is much easier to obtain large monolingual corpora. This research proposes generating synthetic sentences by back-translating sentence in the monolingual corpora and thereby making the overall parallel corpus size larger. This technique has been applied for back-translation of both source-side monolingual corpora [11] and target-side monolingual corpora [12]. While this paved way to improve the translation quality of low-resourced languages it has also been shown empirically that such models tend to "forget" source-side information if trained on much more monolingual data than parallel data, imposing a constraint on the amount of monolingual data that can be used.

One of the main reasons for the popularity of the back-translation technique was it required no changes to be done to the network architecture. Therefore, many techniques have been introduced to improve the quality of the back-translator, since it is another imperfect MT system. [13], proposes a filtering technique which chooses the back-translated synthetic sentences with the highest quality. This improves the final translation quality leading to higher BLEU scores.

Another research addressing this issue proposes data augmentation [20]. Inspired by the many data-augmentation techniques adopted in computer vision research work, the authors generate synthetic sentences to give more context to rare words. This is done by replacing common words with rare words for contexts they can be applied, and consequently replace its corresponding word in the other language by the rare word's translation. This technique has been proven to perform better than the earlier mentioned back translation techniques.

The most common solution for the difficulty of NMT to learn representation of the words (or the smallest token) is to segment words into sub-words as stated above. In [21], the authors have conducted multiple researches to analyze the effect of word embeddings on the translation quality. In [22] they propose a method to train the word embeddings with monolingual data and have presented three methods through which these embeddings can be can be mixed with the parallel word embeddings to provide a better representation to rare words and there by improve the translation quality. This research [22] by the same authors as [21] show that this method of leveraging external embeddings enable a virtually infinite source vocabulary which exclusively improve the translation accuracy in low resourced scenarios even though it does not show encouraging performance in high-resourced settings.

The related work reviewed so far still require a strong cross-lingual signal. Researches that completely remove the need of parallel data has also been proposed. In [23], [24] unsupervised techniques that rely on nothing but monolingual corpora have been introduced, providing hope for languages that have almost no parallel corpora. [24], provides one of the first working approaches for fully working unsupervised NMT. When analyzed in depth, the core idea of this paper is a well laid-out combination of other techniques suggested to improve NMT in low-resource settings. The core ideas in this method are,

- Train the language pair in both directions in tandem.
- Lock the embedding table to bilingual embedding induced from monolingual data.
- Share the encoder between the two languages.
- Alternate between denoising auto-encoder steps and back translation steps.

This paper is one of the solid breakthroughs in NMT and the results show BLEU scores of 10-11 for English - French translation without using parallel corpora at all. In addition, the authors have also shown how these results could be improved further by introducing small parallel corpora converting this unsupervised technique to a semi-supervised technique.

Another popular treatment for low-resource NMT is transfer learning. Transfer learning is the method where a model developed for a particular task is reused as the starting point for model on a second task. This concept has been adopted for NMT with different approaches. The idea is to train a parent model with one pair for languages and transfer its parameters, including the source word embeddings to model where the second languages pair is translated. This concept of "sharing" enables the system to transfer the translation knowledge from one language pair to the other. These approaches are yet to be explored on the Sinhala and Tamil language pair.

2.4 Sinhala - Tamil Machine Translation

The first attempt on Sinhala - Tamil translation has been reported in the year 2003 [1]. This research reports Statistical Machine Translation carried out on the 3 most common languages in Sri Lanka namely Sinhala, Tamil and English. The results of this research show that Sinhala – Tamil translation performs better than Sinhala – English translation. This observation has been justified by concluding that the linguistic distance between Sinhala and Tamil is lower than that of English - Sinhala pair, owing to the co-evolution of Sinhalese and Tamils in Sri Lanka. Other than scarcity of data, morphology richness is the main factor that needs to be considered for many language pairs to develop a successful MT system. In an earlier research done to investigate how Sinhala – Tamil

SMT performance varies with the amount of parallel training data used, the error analysis has shown that the morphological richness leads to poor BLEU scores. According to literature, integrating morphology information to MT is one of the solutions for morphologically rich language pairs. However, usable linguistic resources like morphological analyzers and part-of-speech taggers are publicly unavailable for Sinhala and Tamil, making machine translation for this languages pair challenging.

The best currently available Sinhala – Tamil translator has been produced through the most recent research for morphologically rich languages [2], based on statistical machine translation. The authors have integrated an unsupervised morphological modification approach, suggested in a previous research on Sinhala morphological analysis [3] to overcome the issues related to morphological richness. This has resulted in dramatic improvements in the translation quality and the reliability of the Sinhala - Tamil translation.

On a case-study that compares SMT and NMT, it has been shown that NMT generates outputs that have lower post-edit effort with respect to NMT and delivers state of the art results especially for language pairs involving rich morphology prediction. They also show that NMT has an edge especially on lexically rich texts [26]. Therefore, it is our hypothesis that we can improve Sinhala and Tamil translation accuracy with deep neural models.

2.5 Neural Machine Translation for Sinhala and Tamil

So far, NMT has been explored on Sinhala and Tamil only in one study [26]. In this research the authors have analyzed ways of improving NMT using word phrases when the parallel corpus size is considerably small. This research is conducted in the domain of official government documents thereby investigating the effects of word phrases in domain specific NMT. Our attempt is to design a suitable technique for an open-domain translation for such morphologically rich, low-resourced pair of languages.

2.6 Bilingual Evaluation Understudy (BLEU)

This is one of the most popular translation evaluation technique introduced by the work of [16]. It can be defined by the following equation.

$$BLEU = BP * exp \sum_{i=1}^{n} \lambda_i \log(precision_i)$$

$$BP = \min\,(1, \frac{translated\ sentence\ length}{reference\ sentence\ length})$$

Precision is calculated by dividing the number of correctly translated words by the number of words in the translation output. Here BP is called the brevity-penalty. Usually there is no penalty for dropping words in precision-based metrics and it is addressed by the BLEU with a brevity-penalty. If the output length is too short compared to the reference translation, then the brevity-penalty reduces the score of the of the translated sentence. Rather than calculating the precision using correct number of words, [10] has proposed to consider the correctly translated bigrams, trigrams etc. so that the sequence of the word in the translations are also taken into consideration. Since the length of the highest correlation between human judgment normally equals to 4. Through the same research BLEU-4 was introduced by limiting the order of n to 4.

$$BLEU - 4 = BP * \prod_{i=1}^{4} \lambda_i \log(precision_i)$$

BLEU metric has the possibility of using multiple references. But it has not constrained how n-gram matches can be drawn from multiple references. Also it disregards synonyms and the order the matching n-grams occur are some of the reasons as to why this metric may not always correlate with human judgment.

2.7 Summary

Improving the quality of the translation with the available data is one of the hardest problems in MT. Currently, there is a huge demand for integrating linguistic information and increase the corpus size using different supervised and unsupervised NMT techniques for MT. In this chapter, we have mainly discussed the theory and the work reported in the literature starting with basic NMT, the different treatments to address the data-sparsity problem arose by morphological richness and the lack of parallel corpora. These treatments include integrating linguistic information, generating synthetic parallel sentences using monolingual corpora, data augmentation, including word-embeddings and using unsupervised techniques which require no parallel corpora. Finally we have discussed our translation quality metric, its advantages and disadvantages. From this literature review, we see that there is still the need to analyze the performance of NMT on morphologically rich language pairs for open-domain translation. In our research we aim to address this gap in the body of knowledge using Sinhala and Tamil. Such a technique can be easily adopted by other such morphologically rich, low resourced language pairs and thereby verify the applicability of NMT across languages.

Chapter 3 - Research Design

In this chapter, a discussion on the overall design to address each research question is presented in two stages. In the first stage, the experimental design to address the first research question, together with an introduction to the corpus we used is provided. As the second stage, an introduction to the Sinhala-Tamil parallel and monolingual corpora, the different treatments/preprocessing techniques we have used to increase the accuracy and the intuition behind them are discussed. An explanation on the unit of evaluation and some limitation it imposes has also been presented in detail.

3.1 Research Design for the First Research Question

The corpus details and the high-level architecture to address our first research question, which is “What is the effect of corpus size on the translation accuracy?” are given below.

3.1.1 Corpora

To identify the effect of corpus size, we have built a quantitative research design. Since Sinhala and Tamil are low-resourced, yet morphologically rich, we selected two such morphologically rich language pair with ample parallel sentences. The languages that were chosen were Finnish and German. Finnish is a member of the Finnic language family and is typologically between fusional and agglutinative languages. It modifies and inflects nouns, adjectives, pronouns, numerals and verbs, depending on their roles in the sentence leading to a very productive morphology in which a stem can give rise to several thousand words. German is also considered as a morphologically rich language and the translation of these two languages with English has shown to be difficult in the literature [6].

The direct translation between Finnish and German has not been reported in the literature. Therefore a parallel corpus between these two languages was not publicly available. We extracted the Finnish-English and German-English corpora from the Europarl corpora in WMT18, selected the common English sentences in the two corpora, and mapped the

Finnish sentences to the corresponding German sentences and thereby created a parallel corpus of approximately 1 Million sentences.

3.1.2 Experimental Setup

Since the plan was to investigate the variation of the translation performance in the Finnish-German (Fi-De) language pair with the size of the parallel training data, the experiment was conducted iteratively while doubling the amount of training data in each iteration. First we randomly selected 25,000 sentences, and in the next iteration it was doubled up-to 50,000 we conducted the experiment for 6 iterations (until the training corpus size was 800,000). Similar to our Sinhala-Tamil experiments, the validation dataset consisted of 1,000 sentences and the testing dataset had 2,500 sentences.

We chose one of the most popularly used 2-layer Bi-directional Recurrent Neural Network (BRNN) with LSTM units, with Attention mechanism. Selected word embeddings size was 500. The vocabulary size was limited to 20,000. This experimental setup is shown in Figure 3.1. The BLEU score for each training corpus-size value was plotted and analyzed to draw conclusions from the observations.

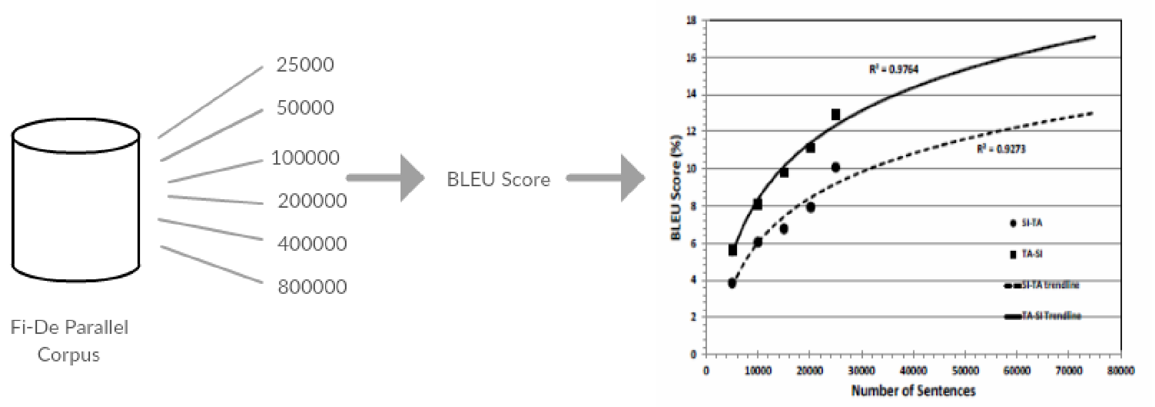


Figure 3.1: High-level Architecture for the First Research Question

3.2 Research Design for the Second Research Question

The corpus details and high-level architecture to address our first research question, which is “What is the accuracy of Sinhala - Tamil translation that can be achieved with NMT when compared to SMT?” is given below.

3.2.1 Corpora

Sinhala and Tamil parallel data is limited; hence they are considered as low resourced. For our experiments we use a parallel corpus of approximately 25000 sentences which has a sentence length between 8 and 12, collected in the research [2].

There are many techniques suggested in the literature to improve NMT of low-resourced languages. Many of them include incorporating monolingual corpora in many ways. We will be exploiting the applicability of such methods for Sinhala – Tamil translation as well. Therefore, in our experiments to improve Sinhala-Tamil translation accuracy, we will be using a 10-million-word monolingual corpus [8] and on the Tamil end, we will be using a 4.3-million-word Sri Lankan Tamil monolingual corpus [9]. Both these corpora are suitable for an open-domain translation as they have been collected from sentences from different domains such as newspaper articles, technical writing and creative writing.

Table 3.1: Characteristics of the Parallel Dataset

Corpus Statistics	**Sinhala**	**Tamil**
Number of sentence pairs	26,187	
Total Number of Words (T)	262,082	227,486
Vocabulary Size	38,203	54,543
V/T %	14.58	23.98

3.2.2 Network Hyper-parameters

After referring the literature, we chose the hyper-parameters that were popularly used. A 2-layer Bi-directional Recurrent Neural Network (BRNN) with LSTM units, with

Attention mechanism was one of the architectures most popularly used. We used the default values suggested by the OpenNMT framework as the optimizer, which was Stochastic Gradient Descent with a learning rate of 1.0.

3.2.3 High-Level Architecture

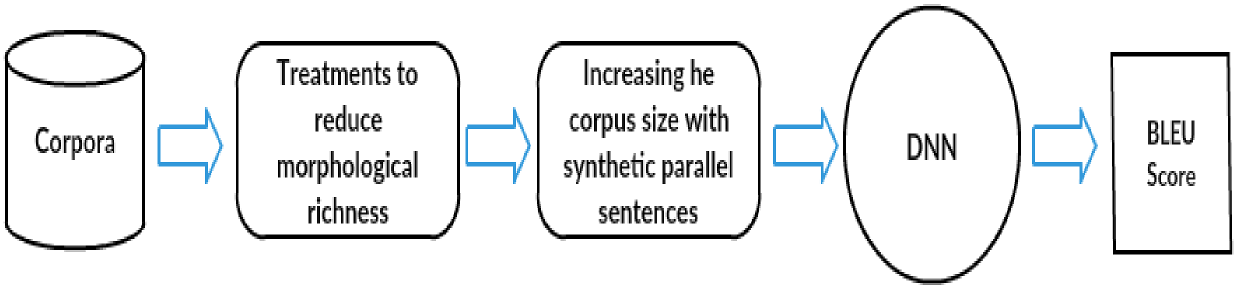

Corpora
Treatments to reduce morphological richness
Increasing he corpus size with synthetic parallel sentences
DNN
BLEU Score

Figure 3.2: High-Level Architecture for the Second Research Question

As we have identified earlier, there are two main properties inherent to Sinhala and Tamil that makes their translation demanding. They are,

i. Sinhala and Tamil are both morphologically rich languages.

ii. The amount of parallel sentences available for this language pair is limited.

These two factors should be treated separately to get good translation accuracies. One method to address the morphological richness of the two languages would be to use linguistic information using linguistic resources such as morphological analyzers, POS taggers etc. But Sinhala and Tamil do not have publicly available linguistic resources. Therefore we make use of sub-word segmentation approaches.

Sub-word segmentation is a technique of text preprocessing that segments words into smaller tokens (sub-words). Such approaches can be commonly divided into unsupervised segmentation and linguistically-driven segmentation. The goal of text segmentation in SMT is to reduce the number of unseen words i.e. words that do not occur in the training data.

We first obtained three different representations of our original corpora. The first one being the original full word-form corpus and the second being the corpora segmented into morpheme-like units. For this segmentation, we used the tool Morfessor 2.0 which provides a morpheme segmentation algorithm that works in an unsupervised manner and extracts morpheme-like segments from the words in an un-annotated/raw corpus. In the absence of some morphs in the dictionary learnt from the un-annotated corpus, Morfessor does not produce character-level segmentation, leading to the OOV problem. But we changed the algorithm in such a way that, such OOV words are segmented into characters.

The third form of representation was obtained by preprocessing the full-word corpora using the algorithm Byte-Pair-Encoding (BPE). This algorithm requires the tuning of the number of merge operations, which is a parameter required by the algorithm. This parameter solely depends on the language and the dataset, therefore we attempted to identify an optimal value for our datasets empirically.

Some example sentences preprocessed with each technique is given below.

1. Full word-form sentences

   SI : නිදසුනක් | කිවහොත් | ශ්රී | ලංකාවට | අදාළ | ප්රතිගාමී | බලවේග | ක්රියාත්මක | වන | ආකාර | හතරක් | පවතී | .

   TA: உதாரணமாக | கூறுவதாயின்| இலங்கைக்கு | பொருத்தமான | முற்போக்கு | சக்திகள் | செயற்படும் | நான்கு | முறைகள் | இருக்கின்றன | .

2. Fully morphologically segmented sentences

   SI : නි | දස | ුන | ක් | කිව | හොත් | ශ්රී | ලංකාව | ට | අ | දා | ළ | ප්රති | ගාමී | බල | වේග | ක්රියාත්මක | වන | ආකාර | හතර | ක් | ප | වතී | .

   TA: உ | தார | ண | மாக | கூறு | வ | தா | யின் | இ | ல | ங்கை | க்கு | பொருத்த | மான | மு ற் | போக்கு | சக்தி | கள் | செயற்பட | ு | ம் | நா | ன் | கு | மு | றை | கள் | இரு | க் | கின்ற | ன | .

3. Segmenting using BPE

   SI : නි@@ | ද@@ | සු@@ | නක් | කි@@ | ව@@ | හොත් | ශ්රී | ලංකාවට | අදාළ | ප්රති@@ | ගා@@ | ම@@ ◌ී | බලවේ@@ | ග | ක්රියාත්මක | වන | ආ@@ | කාර | හ@@ | තර@@ | ක් | පවතී .

   TA : உ@@ | தாரண@@ | மாக | கூற@@ | ◌ுவ@@ | தாய@@ | ◌ின் | இலங்க@@ | ◌ைக்கு | பொ@@ | ருத்த@@ | மான | மு@@ | ற்ப@@ | ◌ோ@@ | க்கு | சக்த@@ | ◌ிகள் | செயற்ப@@ | டும் | நா@@ | ன்க@@ | ◌ு | முற@@ | ◌ைகள் | இருக்க@@ | ◌ின்ற@@ | ன | .

First, we considered only our 25000 parallel corpora. The parallel corpora will be applied on the most popular NMT architectures, fine-tuning the model's hyper-parameters to get the best translation accuracy (i.e. the highest BLEU score). From these experiments the form of representation and the appropriate architecture which showed the maximum BLEU score were identified empirically. There onwards, the experiments were conducted using the selected representation (either full word-form, BPE or the morpheme-like units) and the selected architecture.

3.2.4 Attempts to improve translation accuracy employing machine learning techniques

3.2.4.1 Using the encoder introduced by Google (GNMT)

For the experiments so far, we adopted a BRNN encoder. In the research work presented by Google, they have introduced a new encoder where only the first layer is a single bidirectional layer and the other layers are unidirectional RNN layers. The bidirectional states in this layer are concatenated and residual connections are fed to the next layers which are uni-directional. This is called the GNMT encoder (Figure 3.3). The intention behind this introduction was to increase the speed of the model. The Google Neural Machine Translator is said to be a 8-layer encoder decoder model, therefore having bidirectional RNN with many parameters than a forward RNN on every layer would decrease the training speed.

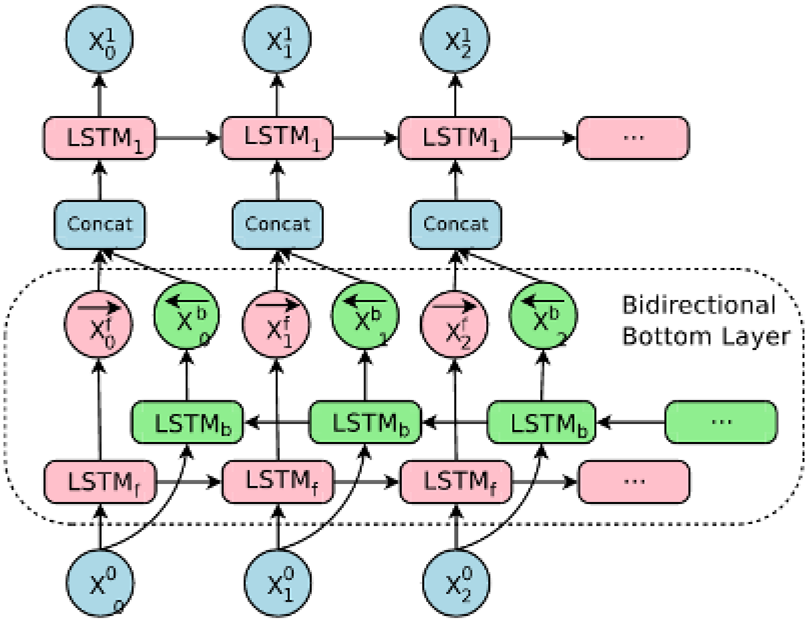


Figure 3.3: GNMT Encoder

We applied our dataset preprocessed using the best representation on the same 2-layer encoder decoder model, but using a GNMT encoder rather than a BRNN encoder. The main reason for us to adopt this encoder was to understand the effect of the number of parameters in the model, on translation in low-resourced settings.

3.2.4.2 Checkpoint Smoothing

We went a step further to improve the BLEU scores, and that is by using an ensemble technique. Ensemble methods are learning algorithms that combine multiple individual methods to create a learning algorithm that is better than any of the individual parts. Checkpoint smoothing is one such ensemble technique which can be using in a single training process [30]. The idea is, rather than using the model generated from the final epoch, we average the parameters of the models from multiple epochs and translate using the averaged models. This increases the generalization power of the models, resulting in better translations.

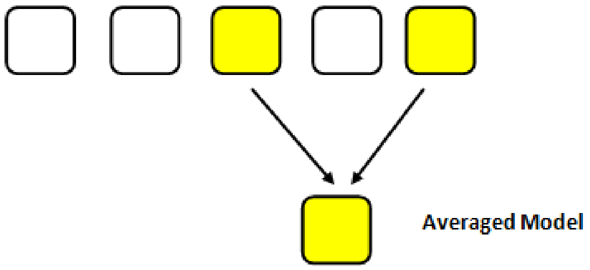


Figure 3.4: Checkpoint Smoothing

Out of the many techniques proposed in the literature to improve the translation of low-resourced languages (discussed in Chapter 2), we have limited our experimentation to supervised techniques using back-translation, as it is a widely-popular technique that makes the maximum use of both parallel and monolingual sentences. We have experimented the applicability of two such techniques on the context of Sinhala and Tamil, and went a step further to introduce a new technique which generated better BLEU scores for Sinhala and Tamil, but could also be used for any language pair.

The two selected techniques are,

01. Back-translating target-side monolingual sentences using our own model and increasing the parallel corpus size by incorporating the back-translated, synthetic sentences. (Normal Back-Translation).
02. Improving the quality of the back-translated, synthetic sentences by using the filtering mechanism proposed in [13]. (Filtered Back-Translation).

After analyzing the applicability of these two techniques, we then introduced a technique which improves the translation quality of both translation directions of the two languages. This new technique is called “Incrementally Filtered Back-Translation”. These three techniques are explained in detail below.

3.2.4 Back-Translation Techniques

Recently, researchers have shown that back-translating monolingual data can be used to create synthetic parallel corpora which in return can be used in combination with authentic parallel data to train a high quality NMT system. Once this technique was introduced by the work of [11, 12], it was soon adopted and explored on many languages. The two main reasons for the popularity of this concept are, this technique does not require the architecture of the network to be changed. Also, it exploits the fact that although the amount of parallel sentences between two languages is limited, it is quite easy to find a large number of monolingual sentences separately for the two languages under consideration.

3.2.4.1 Corpus Details

Even though we have 10-Million word Sinhala monolingual corpora and a 4.3-million-word Sri Lankan Tamil monolingual corpus, we extracted only the sentences that have a length of 8-12 words since our parallel sentences had the same range of tokens per sentence. Tables 3.2 and 3.3 show the characteristics of both the original monolingual corpora and the amount of sentences we extracted of Sinhala and Tamil, respectively.

Table 3.2: Corpus details of the original Monolingual Corpora

Corpus Statistics	**Sinhala**	**Tamil**
Number of sentences	1,067,173	407,578
Total Number of Words (T)	13,158,152	4,178,440
Vocabulary Size	933,153	301,251

Table 3.3: Corpus details of the Monolingual sentences selected for Back-Translation

Corpus Statistics	**Sinhala**	**Tamil**
Number of sentences	180,793	40,453
Total Number of Words (T)	1,577,921	352,813
Vocabulary Size	154,782	65,228

The setup we have created by creating a corpus of monolingual sentences with a length of 8-12, shows that even the monolingual sentences we have for Sinhala and Tamil are low. Therefore we required a technique that makes the maximum use of both available parallel sentences and monolingual sentences.

3.2.4.2 Normal Back-Translation

Using target side monolingual data to improve NMT performance for general under resourced languages was proposed [12]. According to this research, using synthetic source side sentences generated from back translation has increased the quality of translation by a significant amount. Therefore we used the target-language monolingual sentences to create synthetic parallel sentences.

To translate from Tamil to Sinhala, we first took 22k (we took multiples of the authentic parallel dataset size which was 22k) target-side (Sinhala) monolingual sentences and back translated them using the best model trained from Sinhala to Tamil using only authentic parallel sentences. Then by combining the synthetic source sentences and the target-side monolingual sentences, we created a synthetic parallel corpus which was then merged with the authentic corpus. The above steps were repeated by increasing the amount of monolingual sentences as multiples of 22k (22k, 44k, 66k) until we ran out of monolingual sentences to add or the increase in the BLEU score was less that 0.4 (this was considered as the convergence condition). This is depicted in Figure 3.6 and is also presented as an algorithm in Chapter 4.

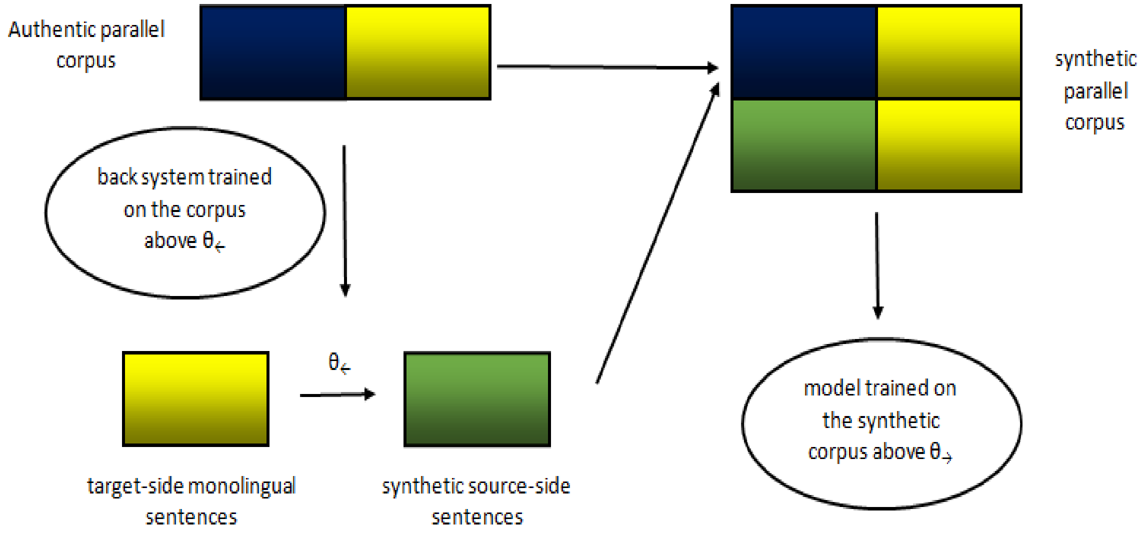

Authentic parallel corpus
back system trained on the corpus above $\theta_{\leftarrow}$
target-side monolingual sentences
$\theta_{\leftarrow}$
synthetic source-side sentences
synthetic parallel corpus
model trained on the synthetic corpus above $\theta_{\rightarrow}$

Figure 3.5: Normal Back-Translation Technique

3.2.4.3 Filtered Back-Translation

As a corpus-based paradigm, the translation quality strongly depends on the quality and the quantity of the training data provided. This required the model that we use to back-translate the target-side monolingual sentences to be of high-quality. This prompted us to check the applicability of the back-translating technique proposed in [13], which has the added step of filtering the best synthetic corpus to be merged with the full parallel corpus in each step. This algorithm is depicted in the Figure 3.7 and is also stated in Chapter 4.

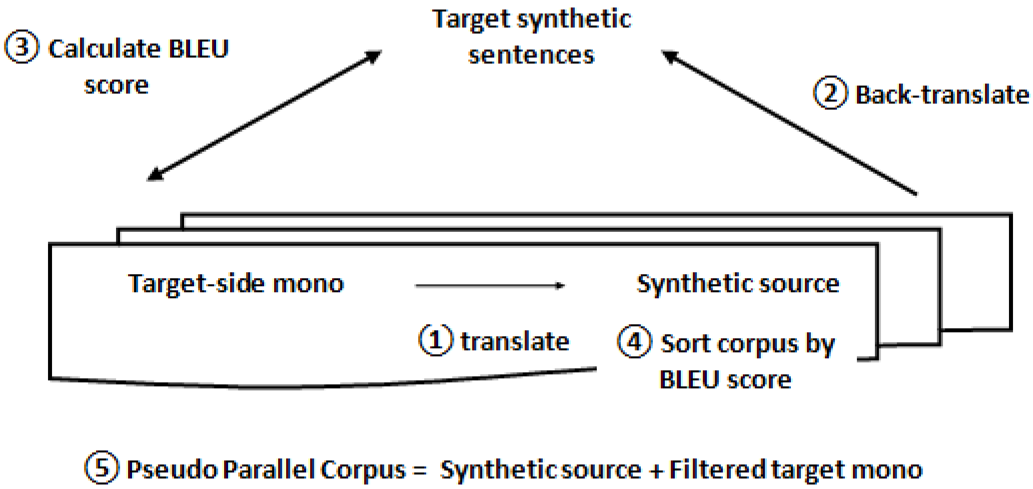

③ Calculate BLEU score
Target synthetic sentences
② Back-translate
Target-side mono
Synthetic source
① translate
④ Sort corpus by BLEU score
⑤ Pseudo Parallel Corpus = Synthetic source + Filtered target mono

Figure 3.6: Filtered Back-Translation Technique

3.2.4.4 Incrementally Filtered Back-Translation

Both Normal Back-translation and Filtered Back-Translation techniques showed promising improvement in the BLEU score for Tamil to Sinhala translation. But for Sinhala to Tamil translation, such an improvement could not be observed. We noticed that the previous back-translation techniques are focused on translating in one direction at a time. This encouraged us to design a technique which would make the maximum use of the limited monolingual sentences we had and also make the two translation directions benefit from each other.

We start off with two models created using authentic parallel sentences trained from language-1 to language-2 (model-1) and vice versa (model-2). Then we select some amount of monolingual sentences from language-2 and create a pseudo parallel corpus using the two models following the filtered back translation technique. Using this parallel corpus a model should be trained in the direction of language-1 to language-2. This was considered as model-3.

As the next step we selected some amount of monolingual sentences from language-1 and created a pseudo parallel corpus using the two models model-3 and model-2, following the filtered back translation technique. This ensures that at each step the back-translation is done using the best model created for the two translation directions. With this technique we witnessed an improvement in the translation quality in both directions. This algorithm is depicted in the figure and is also stated in Chapter 4.

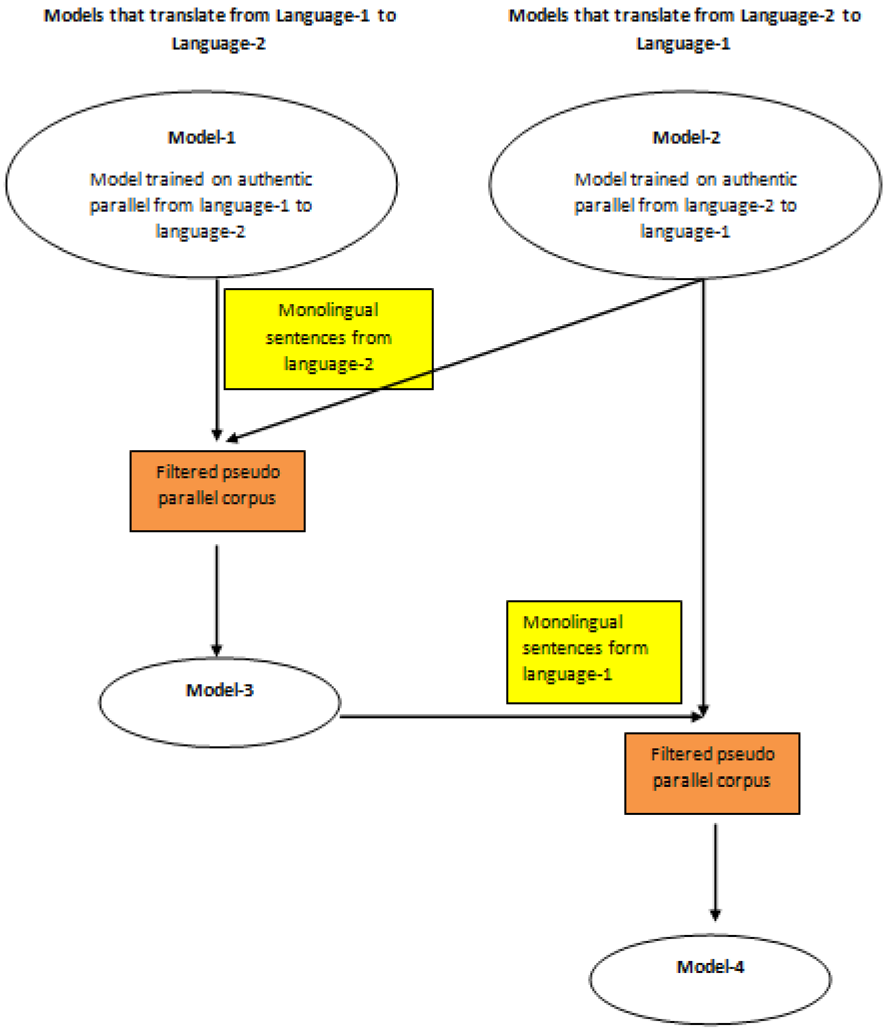

Models that translate from Language-1 to Language-2
Models that translate from Language-2 to Language-1
Model-1
Model trained on authentic parallel from language-1 to language-2
Model-2
Model trained on authentic parallel from language-2 to language-1
Monolingual sentences from language-2
Filtered pseudo parallel corpus
Monolingual sentences form language-1
Model-3
Filtered pseudo parallel corpus
Model-4

Figure 3.7: Incrementally Filtered Back-Translation Technique

In order to get an unbiased result, we performed 3-fold cross-validation technique for each experiment.

3.3 Limitations

Due to the time constraints we will be analyzing only the supervised machine learning techniques. Even though the literature has proposed different transfer learning, unsupervised and semi-supervised techniques, they will be left to be explored as future work.

3.4 Evaluation Plan

As mentioned is Chapter 1, the quality of the different translation approaches we explore will be measured using the automatic evaluation technique BLEU (Bilingual Evaluation Understudy). We evaluated the results obtained for the following models.

- Baseline model
- Models with full word-form
- Models designed with different sentence representations
    - Fully morpheme-like segmentation model
    - BAMorfessor model
    - Byte-Pair-Encoding model

An error analysis was also done to compare the effect of our treatments at different stages, against the full word-form baseline model as follows.

01. Count the number of total words (TotW) and unique words (UniW) in each training (Tr) and testing (Te) datasets.
02. Count the number of out-of-vocabulary (OOV) words in the test dataset (as a percentage of test dataset).
03. Count the number of new words that are not present in the target-side training or reference dataset.

3.5 Summary

In this chapter, we have discussed the high-level architecture and the overall design to address the two research questions we are focusing on.

As the initial step, the methodology to find the effect of the amount of training sentences was presented. This discussion also contains how we designed our research process to address the challenges of Sinhala–Tamil translation step by step. Different word-segmentation techniques and back-translation techniques employed during our research work were discussed in detail in this chapter. Finally, the results we observed by following these techniques how these techniques could be improved further in future work are discussed in Chapter 5 and Chapter 6.

Chapter 4 - Implementation

In this chapter, we describe the implementation aspect of the methods we have followed to generate efficient translations and deal with the inherent challenges of Sinhala and Tamil mentioned in Chapter 1. We begin explaining the implementation details behind the new preprocessing technique we are introducing named BAMorfessor. Next, the 3 different algorithms belonging to the back-translation genre explaining how they were adopted in our research work are presented. These algorithms were explained graphically in Chapter 3. Furthermore, this chapter provides details on the research tools used throughout the research process.

4.1 Preprocessing Techniques

4.1.1 Introducing a new word representation form (BAMorfessor)

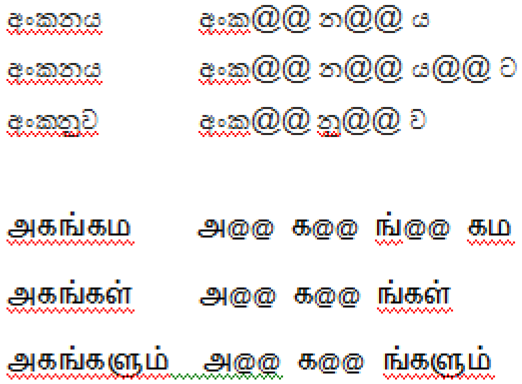
As mentioned in section 5, we observed comparatively better translations with the morpheme-like segmented units than the benchmark full-word form representation. And we also observed that this was not reflected in the BLEU scores because of our post-processing technique. We were inspired by the way the BPE approach, kept track of the boundary of each word, which made the post-processing much easier. Through this observation we wanted to combine the best of both worlds, therefore we came up with a representation of words where the segmentation was done using Morfessor but kept track of the boundaries of the word with another symbol (we used the '@@' sign). This representation will be referred to as the BAMorfessor (Boundary Aware Morfessor) technique

SI : නි@@ | දස@@ | ුන@@ | ක් | කිව@@ | හොත් | ශ්රී | ලංකාව@@ | ට | අ@@ | දා@@ | ළ | ප්රති@@ | ගාමී | බල@@ | වේග | ක්රියාත්මක | වන | ආකාර | හතර@@ | ක් | ප@@ | වතී | .

TA: உ@@ | தார@@ | ண@@ | மாக | கூறு@@ | வ@@ | தா@@ | யின் | இ@@ | ல@@ | ங்கை@@ | க்கு | பொருத்த@@ | மான | முற்@@ | போக்கு | சக்தி@@ | கள் | செயற்பட@@ | ூ | ம் | நா@@ | ன்@@ | கு | மு@@ | றை@@ | கள் | இரு@@ | க்@@ | கின்ற@@ | ன |.

The steps followed to prepare this representation were,

01. Segment words in the monolingual corpora using Morfessor.
02. Create a mapping between the words and their segmentations (Figure 4.1).
03. Insert the '@@' sign to the segmented tokens, except for the last token in each word.



අංකනය අංක@@ න@@ ය
අංකනය අංක@@ න@@ ය@@ ට
අංකනුව අංක@@ නු@@ ව

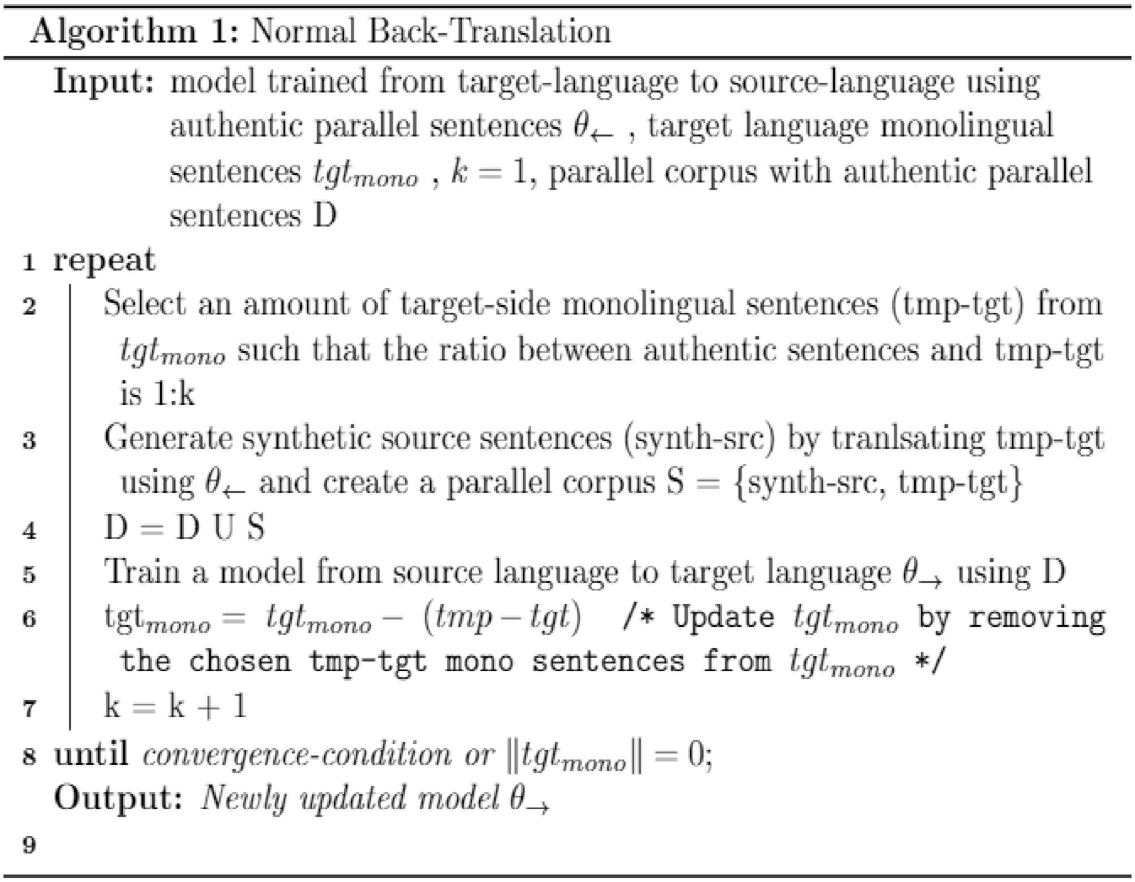
அகங்கம அ@@ க@@ ங்@@ கம
அகங்கள் அ@@ க@@ ங்கள்
அகங்களும் அ@@ க@@ ங்களும்

Figure 4.1: BAMorfessor Mapper

Through this representation post processing was made easier as it required only a regular expression to concatenate morpheme-like units with the special character '@@' with the next morpheme-like unit.

4.3 Back Translation Algorithms implemented

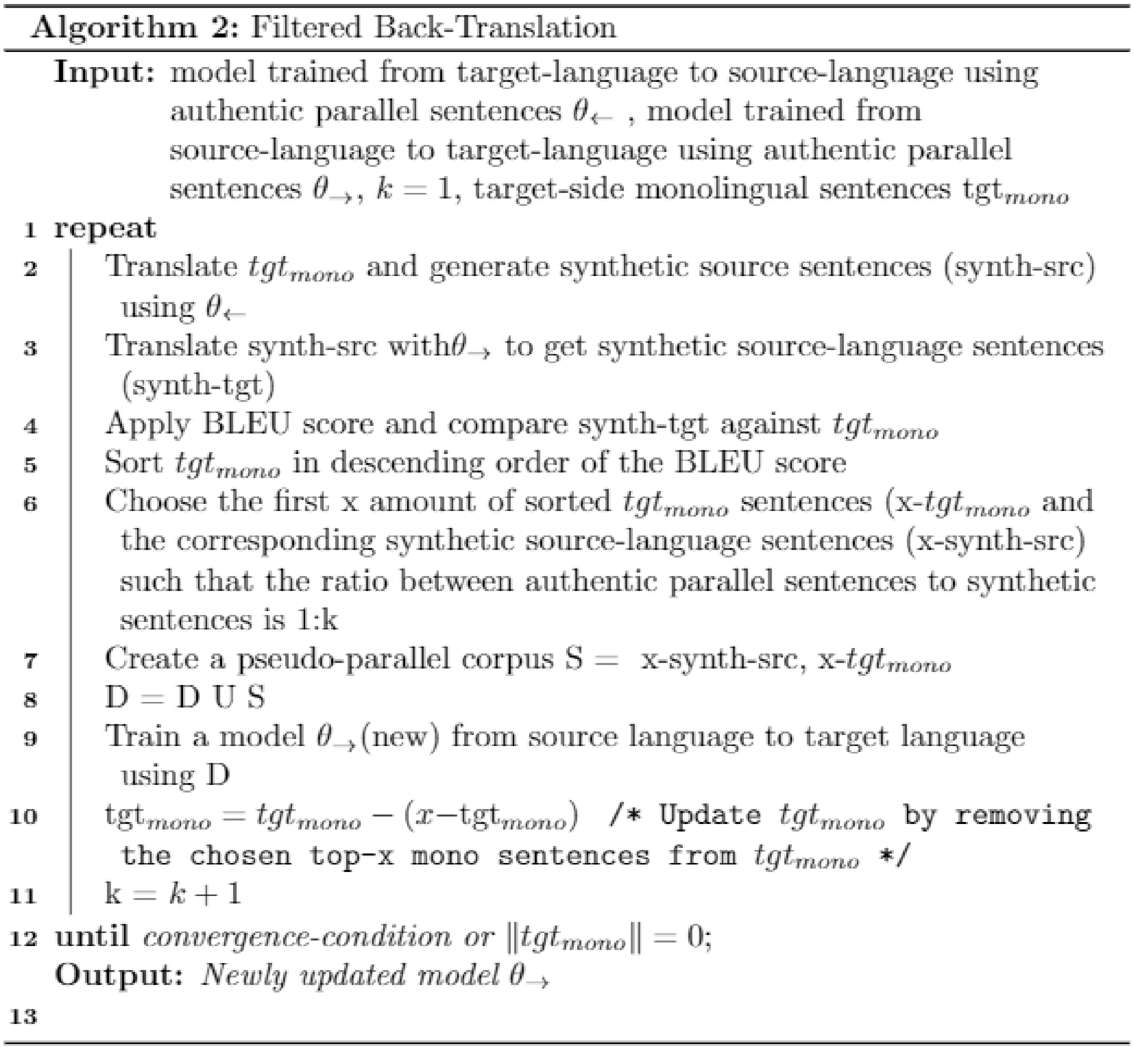
4.3.1 Normal Back-Translation



```
Algorithm 1: Normal Back-Translation
  Input: model trained from target-language to source-language using
         authentic parallel sentences $\theta_{\leftarrow}$ , target language monolingual
         sentences $tgt_{mono}$ , $k = 1$, parallel corpus with authentic parallel
         sentences D
repeat
    Select an amount of target-side monolingual sentences (tmp-tgt) from
     $tgt_{mono}$ such that the ratio between authentic sentences and tmp-tgt
     is 1:k
    Generate synthetic source sentences (synth-src) by tranlsating tmp-tgt
     using $\theta_{\leftarrow}$ and create a parallel corpus S = {synth-src, tmp-tgt}
    D = D U S
    Train a model from source language to target language $\theta_{\rightarrow}$ using D
    tgt$_{mono}$ = $tgt_{mono} - (tmp - tgt)$   /* Update $tgt_{mono}$ by removing
     the chosen tmp-tgt mono sentences from $tgt_{mono}$ */
    k = k + 1
until convergence-condition or $\|tgt_{mono}\| = 0$;
  Output: Newly updated model $\theta_{\rightarrow}$
```

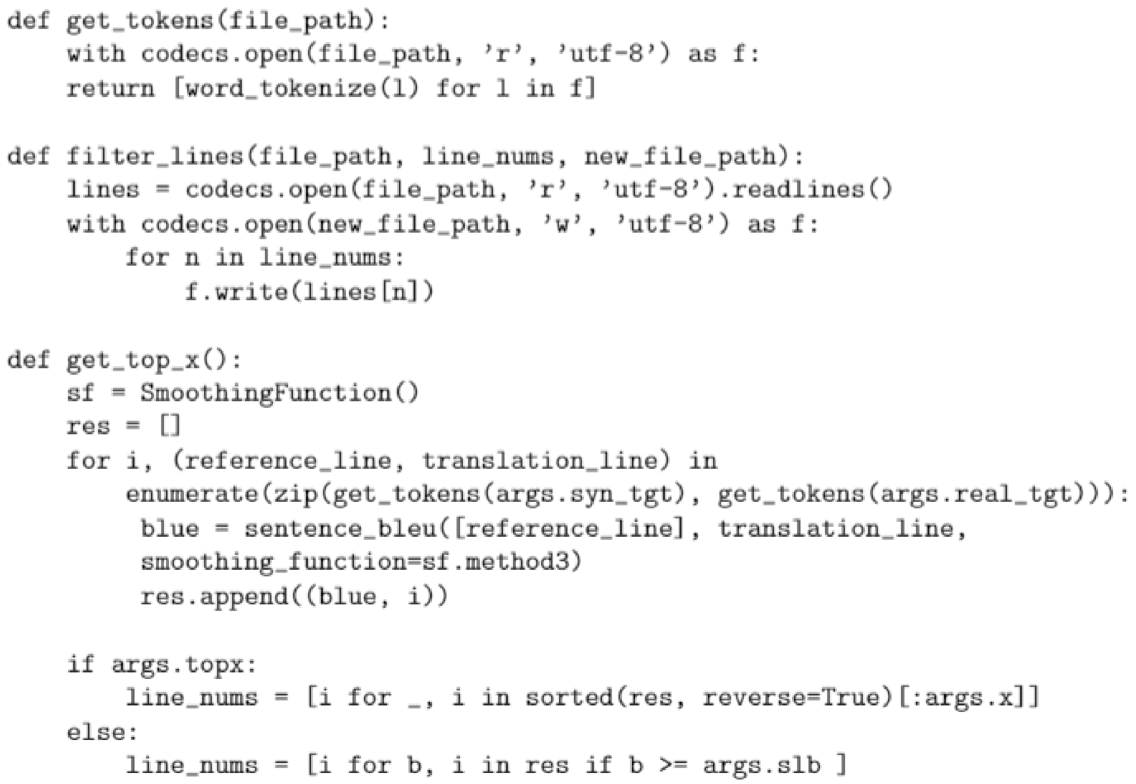
In these techniques the convergence condition is: if the BLEU score between two iterations are not greater than 0.4, exit the algorithm.

4.3.2 Filtered Back-Translation



```
Algorithm 2: Filtered Back-Translation
Input: model trained from target-language to source-language using
       authentic parallel sentences $\theta_{\leftarrow}$ , model trained from
       source-language to target-language using authentic parallel
       sentences $\theta_{\rightarrow}$, $k = 1$, target-side monolingual sentences tgt$_{mono}$
repeat
    Translate $tgt_{mono}$ and generate synthetic source sentences (synth-src)
     using $\theta_{\leftarrow}$
    Translate synth-src with$\theta_{\rightarrow}$ to get synthetic source-language sentences
     (synth-tgt)
    Apply BLEU score and compare synth-tgt against $tgt_{mono}$
    Sort $tgt_{mono}$ in descending order of the BLEU score
    Choose the first x amount of sorted $tgt_{mono}$ sentences (x-$tgt_{mono}$ and
     the corresponding synthetic source-language sentences (x-synth-src)
     such that the ratio between authentic parallel sentences to synthetic
     sentences is 1:k
    Create a pseudo-parallel corpus S =  x-synth-src, x-$tgt_{mono}$
    D = D U S
    Train a model $\theta_{\rightarrow}$(new) from source language to target language
     using D
    tgt$_{mono}$ = $tgt_{mono}$ − ($x$−tgt$_{mono}$)  /* Update $tgt_{mono}$ by removing
     the chosen top-x mono sentences from $tgt_{mono}$ */
    k = $k + 1$
until convergence-condition or $\|tgt_{mono}\| = 0$;
Output: Newly updated model $\theta_{\rightarrow}$
```

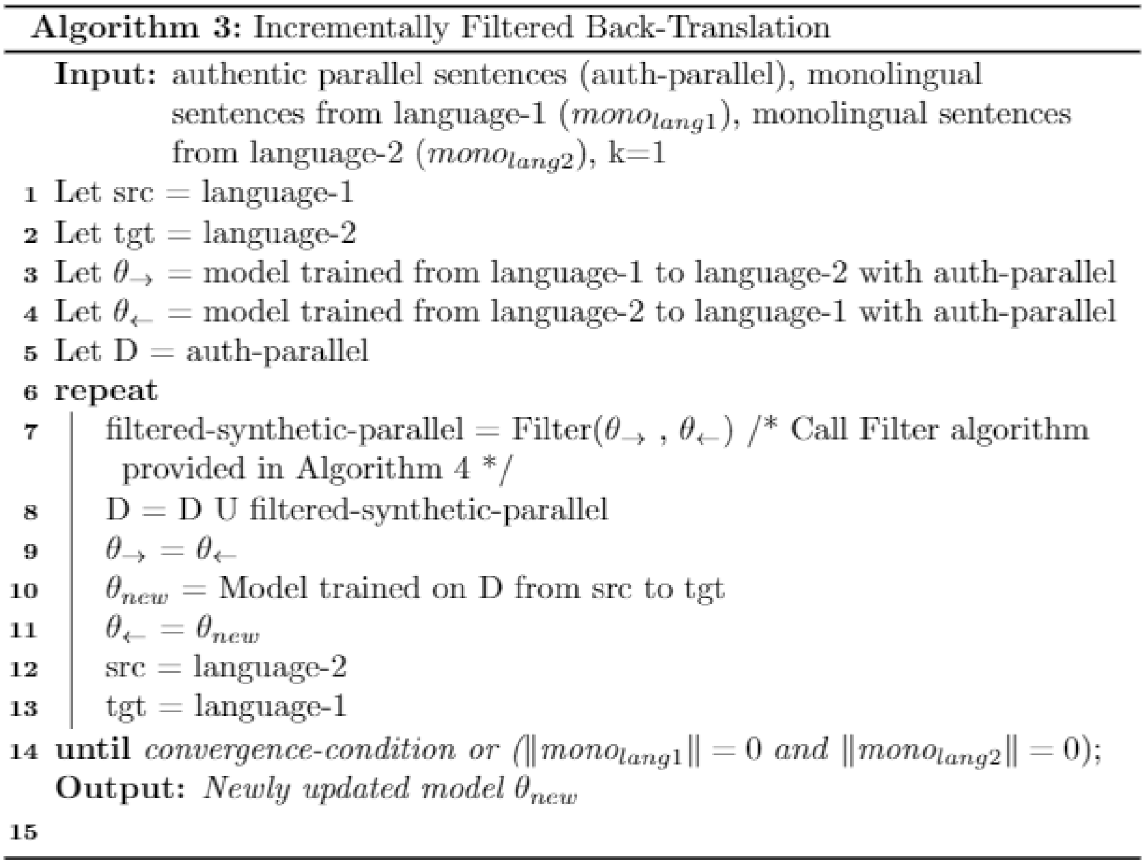
The python code to filter the top-x taget monolingual sentences, is provided in Figure 4.2. The get_top_x method finds the BLEU scores of the monolingual sentences and the synthetic source sentences and sorts them in the descending order. Depending on the argument provided for x (topx), the first x sentences are returned.



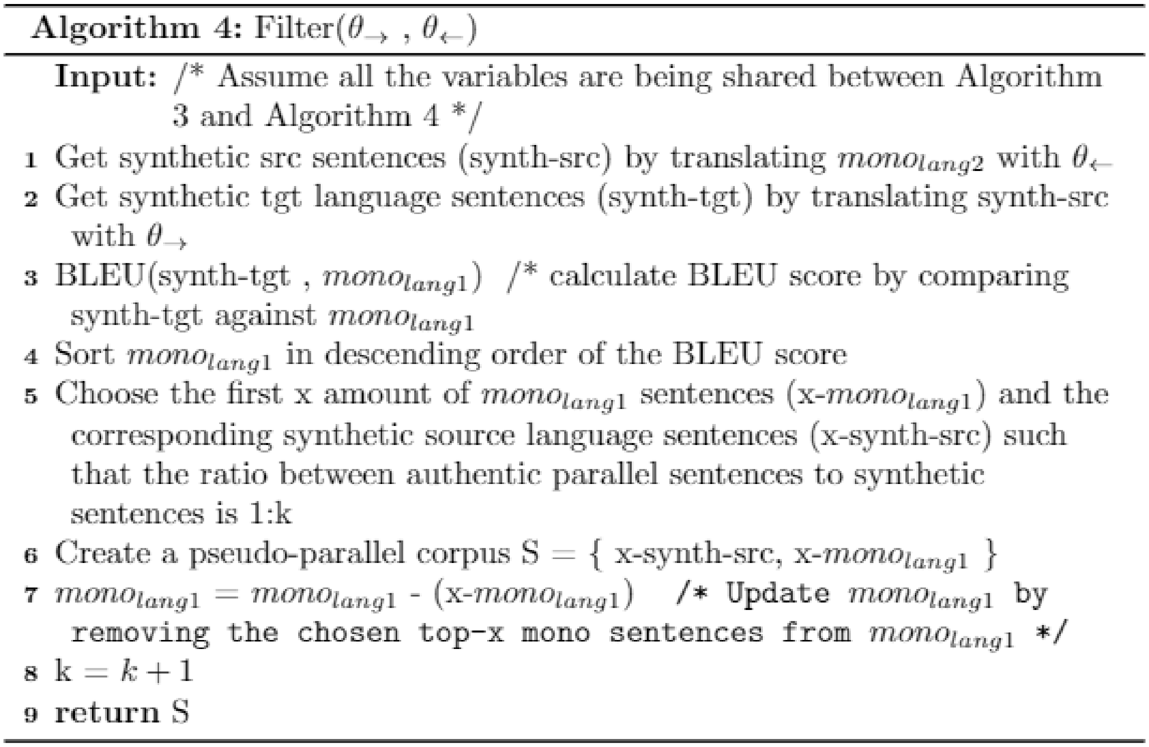
```
def get_tokens(file_path):
    with codecs.open(file_path, 'r', 'utf-8') as f:
    return [word_tokenize(l) for l in f]

def filter_lines(file_path, line_nums, new_file_path):
    lines = codecs.open(file_path, 'r', 'utf-8').readlines()
    with codecs.open(new_file_path, 'w', 'utf-8') as f:
        for n in line_nums:
            f.write(lines[n])

def get_top_x():
    sf = SmoothingFunction()
    res = []
    for i, (reference_line, translation_line) in
        enumerate(zip(get_tokens(args.syn_tgt), get_tokens(args.real_tgt))):
         blue = sentence_bleu([reference_line], translation_line,
         smoothing_function=sf.method3)
         res.append((blue, i))

    if args.topx:
        line_nums = [i for _, i in sorted(res, reverse=True)[:args.x]]
    else:
        line_nums = [i for b, i in res if b >= args.slb ]
```

Figure 4.2: Python code to implement BLEU filtering

4.3.3 Incrementally Filtered Back-Translation



```
Algorithm 3: Incrementally Filtered Back-Translation
Input: authentic parallel sentences (auth-parallel), monolingual
       sentences from language-1 ($mono_{lang1}$), monolingual sentences
       from language-2 ($mono_{lang2}$), k=1
Let src = language-1
Let tgt = language-2
Let $\theta_{\rightarrow}$ = model trained from language-1 to language-2 with auth-parallel
Let $\theta_{\leftarrow}$ = model trained from language-2 to language-1 with auth-parallel
Let D = auth-parallel
repeat
    filtered-synthetic-parallel = Filter($\theta_{\rightarrow}$ , $\theta_{\leftarrow}$) /* Call Filter algorithm
      provided in Algorithm 4 */
    D = D U filtered-synthetic-parallel
    $\theta_{\rightarrow} = \theta_{\leftarrow}$
    $\theta_{new}$ = Model trained on D from src to tgt
    $\theta_{\leftarrow} = \theta_{new}$
    src = language-2
    tgt = language-1
until convergence-condition or ($\|mono_{lang1}\| = 0$ and $\|mono_{lang2}\| = 0$);
Output: Newly updated model $\theta_{new}$
```



```
Algorithm 4: Filter($\theta_{\rightarrow}$ , $\theta_{\leftarrow}$)
Input: /* Assume all the variables are being shared between Algorithm
       3 and Algorithm 4 */
Get synthetic src sentences (synth-src) by translating $mono_{lang2}$ with $\theta_{\leftarrow}$
Get synthetic tgt language sentences (synth-tgt) by translating synth-src
  with $\theta_{\rightarrow}$
BLEU(synth-tgt , $mono_{lang1}$)  /* calculate BLEU score by comparing
  synth-tgt against $mono_{lang1}$
Sort $mono_{lang1}$ in descending order of the BLEU score
Choose the first x amount of $mono_{lang1}$ sentences (x-$mono_{lang1}$) and the
  corresponding synthetic source language sentences (x-synth-src) such
  that the ratio between authentic parallel sentences to synthetic
  sentences is 1:k
Create a pseudo-parallel corpus S = { x-synth-src, x-$mono_{lang1}$ }
$mono_{lang1}$ = $mono_{lang1}$ - (x-$mono_{lang1}$)   /* Update $mono_{lang1}$ by
  removing the chosen top-x mono sentences from $mono_{lang1}$ */
k = $k + 1$
return S
```

4.4 Validating and testing the models

Throughout the experiments conducted using only the 25000 parallel corpora the models were validated using a small extracted parallel dataset of 1000 sentences.

OpenNMT framework evaluates the training set and the validation set accuracy and the perplexity as follows. To test the models, we selected a test-set of 2500 sentences (which is 10% of the training-set size). The test-set consisted of sentences which were mutually exclusive from the training-data.

- $\text{Accuracy} = \frac{\text{No.of correct words}}{\text{No.of words}}$

- $\text{Perplexity} = e^{\text{negative loss likelihood of the true target data} \div \text{ no.of target words}}$

The training was done until the perplexity of the validation set stops decreasing.

4.5 Research Tools used

- The NMT experiments we chose OpenNMT [29], which is an open-source framework. It is a strong sequence-to-sequence implementation in Torch with many model configurations.
- Morpheme-like representation was generated based on Morfessor Categories-MAP, using Morfessor 2.0 [19].
- BPE representation was obtained by the method suggested in [18]. The implementation of their work is also publicly available.
- Beautiful-Soup web-crawler (a python implementation) was used to collect data.
- To speed-up the training process GeForce GC 1080 Ti GPU was used with a GPU memory of 16 GB.
- Python modules sci-py, matplotlib, statsmodel to generate graphs and charts in the evaluation.

4.6 Summary

Throughout this chapter we discussed the technical aspect of the newly introduced preprocessing technique, machine learning techniques and also the three flavors of back-translation we have employed to improve translation accuracy. Their effect on the translations are discussed and analyzed in Chapter 5. We have also described the tools, frameworks, GPU specifications that enabled the implementation of our experiments.

Chapter 5 - Results and Analysis

In this chapter we present the translation accuracy obtained with each technique discussed in the methodology of the research design and their implementation discussed in Chapter 4. We also provide our justification and conclusions for the observed results. The reader can see how each technique we employed progressively improved the translation accuracy measured using the BLEU score.

5.1 Results obtained for the First Research Question

The Finnish-German corpus that was prepared by us was divided into 6 samples as mentioned in Chapter 3. The Figure 5.1 below shows a scatter plot of the BLEU scores obtained in each sample.

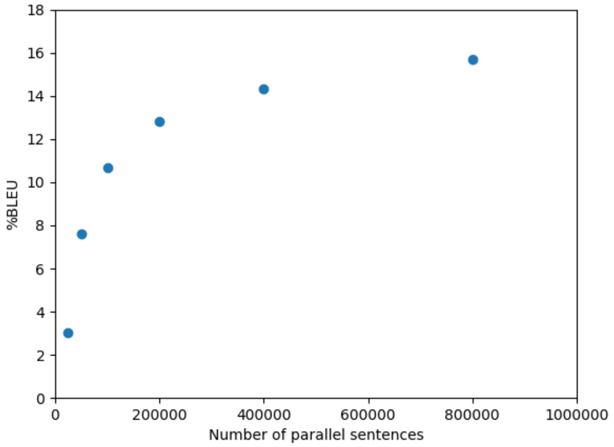


Figure 5.1: BLEU Score (%) values of Fi-De for different data sizes from 25k to 800k

When observing the scatter plot very carefully, we could see that there is a decrease in the increasing rate of the BLEU scores as the dataset size increases. We could observe that even when the dataset size is 800,000 (an amount of parallel sentences Sinhala and Tamil could only hope of achieving), the BLEU score that was reported for Fi-De was 15.70.

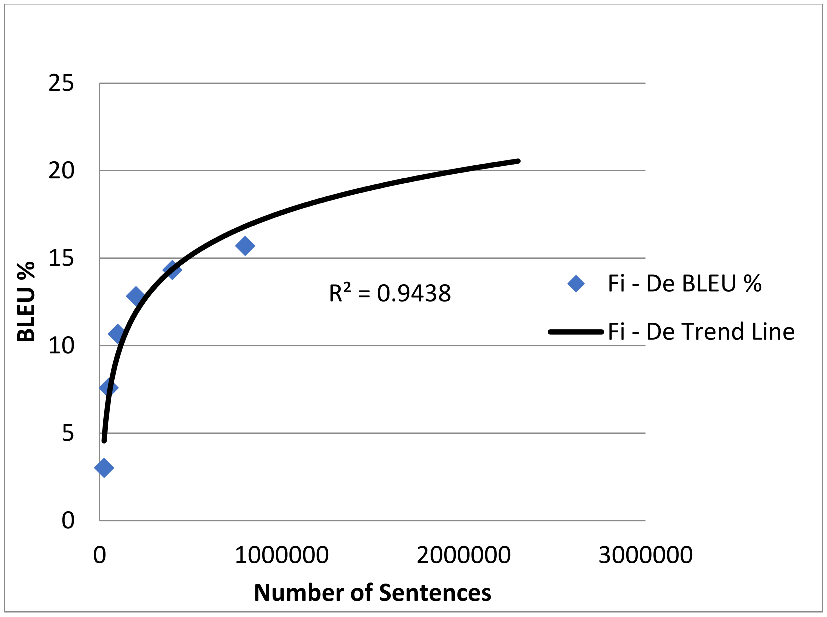


Figure 5.2: Best-fit curve for Average BLEU Score VS Number of Parallel Sentences

Next, we fitted a curve to the scatter plot in an attempt to extrapolate the possible performance we could expect from the Fi-De language to identify how much parallel sentences we would require to reach at least a BLEU score of 20 (The average BLEU score the translation tasks performed in WMT16 is 21.3, as stated in [6]). According to the R-squared values of the fitted curves approaching 1, we could expect only to achieve the BLEU score around 20 (Figure 5.2) when Fi-De has approximately an amount of 2,000,000 parallel sentences as a lower bound. The conclusion we drew by the addressing this research question is as follows.

- Collecting more parallel sentences that has been translated manually by linguists is an expensive and a time consuming task. To improve the corpus size of Sinhala and Tamil, even crowd-sourcing is a limited option as the number of people who are both fluent in Sinhala and Tamil are low. The above experiment encouraged us to research for techniques that would make the maximum use of the available corpora, rather than investing our time and money on collecting correctly, linguistically translated parallel sentences. Addressing this research question justified the investment of our efforts on the second research question.

5.2 Results obtained for the Second Research Question

5.2.1 SMT Baseline Model

In order to compare our results we initiate two baseline models where the first model is the work of [2], which is a SMT with morpheme-like units. We used the exact same corpora as given in [2] to see the performance of NMT in comparison to SMT under same conditions.

The researchers in [2] have conducted their experiments only in the direction of Tamil to Sinhala. The results they obtained are provided in Table 5.1. Since in our research work we are interested in the translation in both Sinhala to Tamil and Tamil to Sinhala directions, we use a second baseline model by training a network with the architecture provided in section 5.2.2 on our 25k full-word form parallel corpora.

Table 5.1: BLEU scores using SMT, corpus preprocessed by Morfessor

	Tamil - Sinhala
BLEU	13.11

5.2.2 Full-word form Representation

The BLEU scores obtained for both directions with the full-word form are shown in Table 5.2. Needless to say these values were discouraging.

Table 5.2: BLEU scores on translating the Full-word form corpus with NMT

Translation Direction	**Sinhala - Tamil**	**Tamil - Sinhala**
BLEU	2.47	5.41

The translations did not resemble the semantics of the reference sentences. Only a few words from each sentence were correctly translated but it did not add any value to the underlying meaning of the sentence. Such semantically correct translation outputs are given in Table 5.3.

Table 5.3: Example translations from Full-word form NMT

Reference Sentence	Translated Sentence
සමහරු පළාත් සභාවට ශක්තිමත් නායකත්වයක් සඳහා ඉදිරිපත් වී සිටිති .	සමහරු පළාත් සභා යෝජනා සකස් කිරීම සඳහා යෝජනා කළා .
ලෝකයේ කිසිදු රාජ්යයකට හුදෙකලාව පැවැතීමට නො හැකි ය .	ලොව කිසිදු ආණ්ඩුවකට කාලයක් පාලනය කළ නො හැකි ය .

The results of the error analysis (following the processes mentioned in Section 3.4) for both Sinhala to Tamil and Tamil to Sinhala are shown in the Table 5.4.

Table 5.4: Error Analysis for full-word form NMT

Description	Sinhala - Tamil		Tamil - Sinhala	
	UniW	TotW	UniW	TotW
Training Source Dataset (Average)	35,635	252,139	50,111	221,194
Testing Source Dataset (Average)	9,043	27,743	10,566	24,316
OOV%	7.96	0.22	14.95	0.35

According to Table 5.4, we can see that the total number of words in training and testing datasets are higher in the Sinhala language than in the Tamil language while the unique number of words are higher in the Tamil dataset than that of Sinhala dataset. If we consider the testing dataset, approximately 8% of the unique words in the Sinhala testing dataset are Out-of-Vocabulary (OOV) words while the same value is 15% in the Tamil testing dataset which is significantly higher. When analyzing the BLEU scores, Sinhala to Tamil translation performance is lower than the Tamil to Sinhala translation. However, when considering the OOV rate of the Tamil to Sinhala and Sinhala to Tamil translation directions, Tamil to Sinhala translation has a higher OOV rate.

From these observations made through the above error analysis we can draw the following conclusions.

- As experimentally shown in [27], NMT systems consistently produce more fluent translations to the point that they completely sacrifice adequacy.
- We observe that the ratio (vocabulary size : total number of words) of Tamil is greater than Sinhala. This could mean that even though a detailed linguistic analysis has not been reported to compare the morphological richness of Sinhala and Tamil, within the context of our corpora, Tamil is morphologically richer than Sinhala.

5.2.3 Preprocessed with Morfessor

The work of [2], shows that morphologically rich languages perform better with word segmentations and in their work, they have employed the unsupervised technique Morfessor which segment the words into morpheme like units. We used the same representation and we observed that some of the words were properly segmented into affixes and stems as well as some segmentations were not properly segmented.

We preprocessed our full-word form corpus with Morfessor and translated using the same neural network architecture. Once translated, the segmented units required to be post-processed to form the full word form.

Post-Processing Technique: First we created a mapping between the morpheme-like units and their original form. Then, we sorted the original form of words in the descending order according to their length and replaced scattered morpheme-like units in the sentences with the help of the mapping scheme to get a readable output.

Table 5.5: BLEU scores from Morfessor preprocessing technique

Sentence representation	**Example Sentence**	**Tamil - Sinhala**	**Sinhala - Tamil**
Morpheme-like units	**SI :** අවුල් වි ය වුල ් ඇති වීමේ තර ් ජන ය නිරත ුරු ව ප වතී . **TA :** உ | தார | ண | மாக | கூறு | வ | தா | யின் | இ | ல | ங்கை | க்கு | பொருத்த | மான | மு ற் | போக்கு | சக்தி | கள் | செயற்பட | ு | ம் | நா | ன் | கு | மு | றை | கள் | இரு | க் | கின்ற | ன | .	4.6	4.06

As can be seen in Table 5.5, the dataset preprocessed with Morfessor does not show an improvement in the translation accuracy when compared with the full-word form. But the researchers could see that translated sentences of this representation were more meaningful than the translation output from the full-word form dataset. We identified, that this improvement in the translation accuracy was not reflected due to the post processing technique we used to concatenate the morpheme-like units after the translations. During post processing, the translations have got corrupted. This required us to look into a solution to address this, which led to the development of BAMorfessor representation (described in Chapter 4). The results by using BAMorfessor representation is given in Table 5.6 below.

Table 5.6: BLEU scores obtained with BAMorfessor representation

Sentence representation	**Example Sentence**	**Tamil-Sinhala**	**Sinhala -Tamil**
BAMorfessor	**SI :** අවුල් වි@@ ය@@ වුල@@ ් ඇති වීමේ තර@@ ්@@ ජන@@ ය නිරත@@ ුරු ව ප@@ වතී **TA :** உ@@ | தார@@ | ண@@ | மாக | கூறு@@ | வ@@ | தா@@ | யின் | இ@@ | ல@@ | ங்கை@@ | க்கு | பொருத்த@@ | மான | முற்@@ | போக்கு | சக்தி@@ | கள் | செயற்பட@@ | ு | ம் | நா@@ | ன்@@ | கு | மு@@ | றை@@ | கள் | இரு@@ | க்@@ | கின்ற@@ | ன |.	9.17	6.06

The newly proposed BAMorfessor technique showed a promising improvement, also making the post-processing much easier (we simply needed to join sub-words having '@@' signs with the next sub-word). An error analysis was done for the corpus with BAMorfessor representation to further justify the improvement in the BLEU scores.

Table 5.7: Error Analysis on the BAMorfessor representation

Description	Sinhala - Tamil		Tamil - Sinhala	
	UniW	TotW	UniW	TotW
Training Source Dataset (Average)	6,668	488,902	4,374	606,336
Testing Source Dataset (Average)	4,001	53,901	2,519	66,880
OOV%	0.22	0	0.21	0

When comparing the values of the two Table 5.4 and Table 5.7**,** we can clearly see that the number of tokens have increased with BAMorfessor presentation. This increases the average length of a sentence more than the full-word form. At the same time we see that the OOV percentage has decreased drastically in both translation directions. For Sinhala-Tamil direction the unique OOV percentage with the full-word form was 7.96% and with the BAMorfessor representation it has decreased to a value of 0.22%. For Tamil to Sinhala translation direction the unique OOV percentage was 14.95% and with the BAMorfessor representation it has decreased to a value of 0%.

The improvement of the BLEU scores with BAMorfessor representation can be reasoned out as below. With this new representation, the vocabulary size of the Sinhala corpus has been reduced to approximately 20% of the vocabulary size of the full-word form and as for Tamil, it has been reduced to 8% of the vocabulary size of its full-word form. It performed better than the morpheme-like representation as the post-processing did not affect the translations as it did for the morpheme-like representation.

These observations lead us to the following conclusions.

- When Morfessor is applied on Sinhala and Tamil, some of the units that are segmented into are exact morphemes while some are segmented into only morpheme-like units.
- This segmentation helps the model translate unseen words as even the unseen words are being segmented into units that are most likely to have been occurred in the training set. This means segmentation into morpheme-like units have reduced the data-sparsity and entropy of the dataset, which has directly influenced in the improvement of the overall translation quality.

5.3.4 Preprocessed with BPE

When preprocessing our corpora with BPE, choosing an appropriate value for its hyper-parameter (number of merge operations) was challenging. It is a value that depends both on the language and the corpus. A value is usually selected on a trial and error basis. A higher value for this parameter would lead the tokens to be almost words where as a lower value would leave the tokens at a character level.

The number of merge operations affects the average length of a sentence. As shown in [16], when the sentence size increases, more difficult it becomes for NMT to learn. Since there has been no research work reported on applying BPE encoding on a corpus such as ours, we considered a range of values for the number of merge operations and analyzed how it affects the final BLEU score.

We started our range of values from 500 and plotted the corresponding BLEU score while increasing this value by 500 until the merge operations created almost words. We were expecting at least an “elbow” pattern to choose an appropriate value from, by plotting the BLEU score against the number of merge operations. But as can be seen in the two graphs depicted in Figure 5.3 and 5.4 there is no proper statistical relationship between the independent and the dependent variable and a proper value for the hyper-parameter is highly dependent on the corpus size. Therefore, we chose the number of merge operations with the highest BLEU score within the considered window-size.

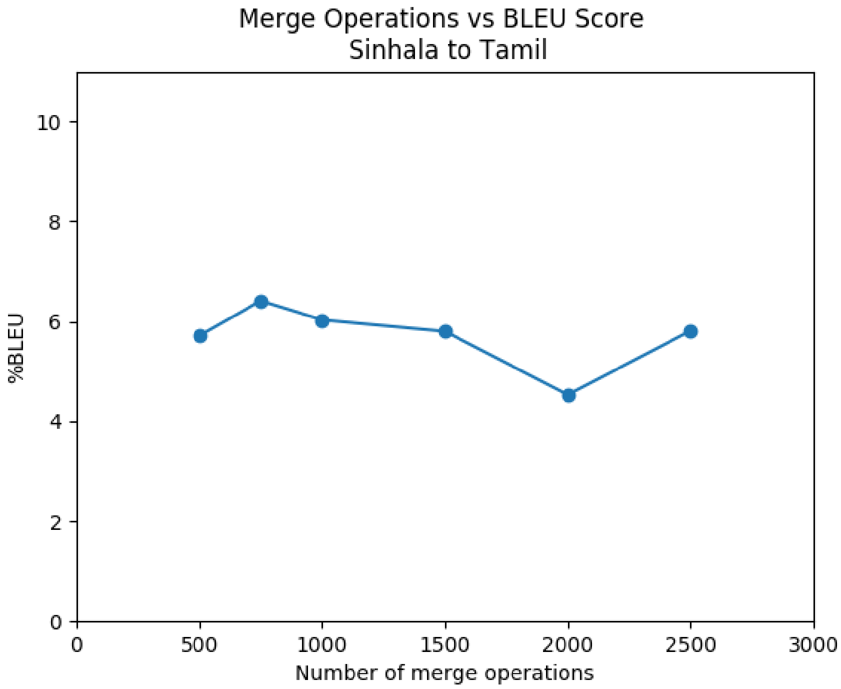


Figure 5.3: Merge Operations vs. BLEU score for Sinhala to Tamil Translation

In the Sinhala to Tamil translation, we observed that the highest average BLEU score is given at 1000 merge operations. But the difference between the BLEU score at 500 and 1000 was significant (5.71 and 6.03) which prompted us to believe that a value in between 500 and 1000 could produce a better BLEU score. As expected we observed the highest peak for Sinhala to Tamil translation when the number of merge operations was 750, and for Tamil to Sinhala, when it was 1000. These values were used for the experiments conducted thereafter. The BLEU scores for the chosen number of merge operations is shown in Table 5.8**.**

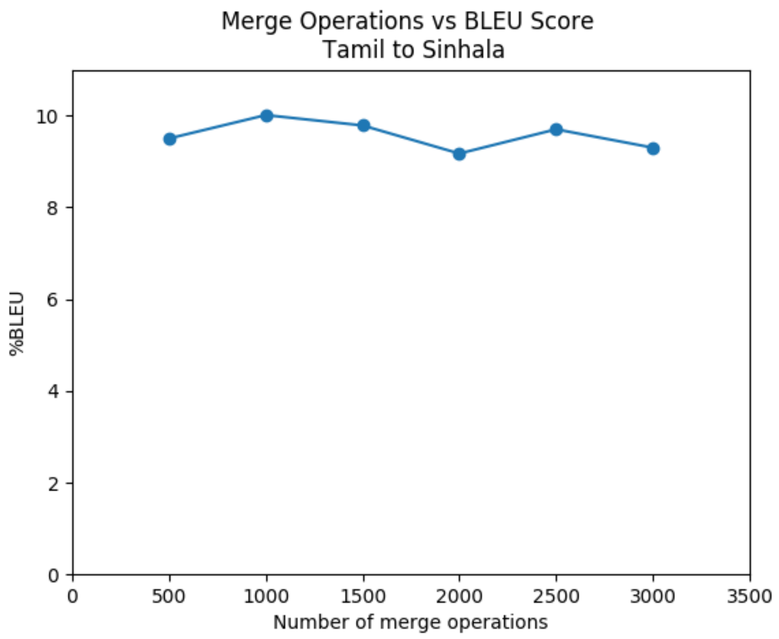


Figure 5.4: Merge Operations vs. BLEU Score from Tamil to Sinhala Translation

The BLEU score when the corpus is preprocessed with BPE is higher than when it is preprocessed with Morfessor. Through a manual evaluation of the tokens produced in each case, we observe that Morfessor generates better morphemes than BPE, and translation in both directions seem to have benefitted from the BPE segmentation than with Morfessor segmentation. The Table 5.9 shows some examples where Morfessor has produced better morphemes than BPE.

Table 5.8: BLEU Scores reported for the chosen number of merge operations

	Sinhala – Tamil (#merge-operations = 750)	**Tamil –Sinhala** (#merge-operations = 1000)
BLEU	6.41	10.01

Table 5.9: Few examples where Morfessor has produced better morphemes than BPE

Original Word	Stem + Affixes	BAMorfessor Segmentation	BPE Segmentation
ස්ත්‍රියට	ස්ත්‍රි + ය + ට	ස්ත්‍රි@@ ය@@ ට	ස්ත්‍ර@@ ියට
උරුමයට	උරුම + ය + ට	උරුම@@ ය@@ ට	උ@@ රු@@ ම@@ යට
ත්‍රස්තවාදය	ත්‍රස්ත + වාද + ය	ත්‍රස්ත@@ වාද@@ ය	ත්‍රස්තවාදය

To investigate further on the improvement of the translation quality with BPE, we conducted an error analysis as was done for the previous forms of representation (Table5.10).

Table 5.10: Error Analysis for the parallel corpus represented with BPE

Description	Sinhala - Tamil		Tamil - Sinhala	
	UniW	TotW	UniW	TotW
Training Source Dataset (Average)	939	582,204	1,161	563,035
Testing Source Dataset (Average)	906	64,082	1,121	62,300
OOV%	0	0	0	0

The vocabulary size has been brought down further with BPE representation. As can be seen in Table 5.10 the OOV percentage, both unique and total words is 0%. This shows that BPE enables NMT, an open-vocabulary translation. It has further decreased the entropy and simultaneously had increased coverage, leading to better BLEU scores.

Additionally we observed that sub-word segmentation creates new words in the target-language, that are not available in the training nor the reference sentences. We compared the number of such newly created words and their quality, generated from the three forms of representations so far.

Table 5.11: Number of new words created with each form of representation

Preprocessing Technique	**Sinhala - Tamil**	**Tamil - Sinhala**
Full-word form	0	0
BAMorfessor	469	993
BPE	1525	809

From the values in Table 5.11, we observe that new words are generated due to segmentation. Preprocessing with BPE tends to generate a higher number of new words and a manual evaluation of the created words show us that the new words generated with BAMorfessor representation are more meaningful than the new words generated with BPE. To reason out this observation, previously we saw that Morfessor segments words into better morphemes than BPE. Therefore, when concatenating the translated sub-units during post-processing, there is a better chance for stems to be combined with appropriate affixes when preprocessed with Morfessor to produce more intelligible complete words. But since the segmentation with BPE is not linguistically correct as much as with Morfessor, there is a high chance for incorrect affixes to be combined with the stems during post-processing resulting in unintelligible words.

From the above observations, we can draw the following conclusions.

- Tuning the number of merge operations is a difficult task as it is solely data-driven.

- BPE technique performed better than BAMorfessor representation. Similar observations were made in [28] on the Bengali-Hindi pair of languages. They empirically show in their work that for linguistically close languages, BPE performs better than when using Morfessor. Their intuition is that when two languages are linguistically close (with respect to morphology and word order), even though BPE does not produce good morphemes this does not affect the closely related languages because of their syntactic similarities. The same intuition can be applied to the Sinhala-Tamil language pair to justify the above observations.
- The above conclusions help us derive another conclusion. That is, since linguistically similar languages like Sinhala and Tamil benefit from the fact that the sub-units are not segmented into proper morphemes, it frees us from the need of a morphological analyzer for NMT translation tasks. Therefore we could focus are future efforts on improving the quality of the newly generated words in BPE by incorporating a Language Model.
- Sub-word segmentation tends to generate new words the model has not seen during the training stage. When the segmentation produces better morphemes, more sensible new words are generated.

After experimenting on different preprocessing techniques, next we employed two machine learning techniques to improve the accuracies.

5.3.5 Using the GNMT Encoder

As mentioned in Chapter 3, we next changed the network architecture by changing the encoder type. In the earlier experiments we were using a Bi-Directional LSTM Recurrent Encoder. Next we changed the model to use a GNMT encoder.

Table 5.12: BLEU scores when GNMT encoder is used

	Tamil - Sinhala	**Sinhala - Tamil**
BLEU	10.57	6.94

We analyzed the reasoning behind the improvement of the BLEU score with this architecture (improved BLEU score given in Table 5.12). We noticed that the number of parameters used in the model with a BRNN encoder (22,683,128) was almost as twice as the number of parameters of that with a GNMT encoder (11,104,741). When the data-set is small, we cannot afford to fit models with a high degree of freedom (too many parameters). This leads to a simpler model ergo the improvement in the translation quality.

Next we used checkpoint smoothing on the models generated from the previous step, i.e. by using smoothing the models generated when the GNMT encoder was used. Intuition behind using checkpoint smoothing is described in Chapter 3. By using this technique, we could improve the BLEU scores from the previous step as shown in Table 5.13.

Table 5.13: BLEU scores after using checkpoint smoothing

	Tamil - Sinhala	**Sinhala - Tamil**
BLEU Score	11.76	7.51

The improvement in the BLEU score is significant. The ensemble of models from multiple epochs used to create the averaged model has increased the generalization power of the models, resulting in better translations.

5.3.5 Applying Back-Translation

The previous series of experiments were conducted only using the parallel sentences. In the following experiments we attempted make use of our monolingual corpora to increase the net parallel corpus size by using back-translating techniques. We analyzed the applicability of two such techniques on the context of Sinhala and Tamil and then introduced a new technique which improved the translation accuracy further.

5.3.5.1 Normal Back-Translation

We applied the algorithm for Normal back-translation presentenced in Chapter 4 and the translation accuracy for the two directions are provided below.

As can be seen from the Table 5.14, Tamil to Sinhala translation direction has benefitted from the naïve back-translation approach. At each step we added 22k amount of target-side monolingual sentences back translated from the baseline model created using authentic parallel sentences (models with the BLEU scores given in Table 5.13). This validates conventional wisdom in Deep Learning which states “more data is better data”.

Table 5.14: BLEU Scores from Normal Back-Translation

Ratio between authentic parallel sentences : synthetic parallel sentences	**Tamil - Sinhala**	**Sinhala - Tamil**
1 : 1	12.16	7.34
1 : 2	14.17	7.37
1 : 3	15.35	

We could have continued the translation for Tamil to Sinhala direction by adding more synthetic parallel sentences. But we stopped when the ratio between authentic : synthetic parallel sentences were 1 : 3 because for Sinhala to Tamil direction we could only conduct experiments until the same ratio is 1 : 2 due to the lack of target-side (Tamil) monolingual sentences (corpus details are provided in Table 3.3).

But this improvement in the translation quality could not be witnessed in Sinhala to Tamil translation. As witnessed in all the experiments conducted so far, translating Sinhala to Tamil is more difficult than translating from Tamil to Sinhala. Our conclusion from this observation is the morphological richness of Tamil than that of Sinhala in the context of our corpora. Given that the synthetic data is generated via an imperfect back-translation system Sinhala to Tamil failed to improve the BLEU scores.

This prompted the need to improve the quality of the pseudo parallel sentences generated, which led us to explore the filtered back-translation technique.

5.3.5.2 Filtered Back-Translation

We applied the algorithm for Filtered back-translation presentenced in Chapter 4 and the translation accuracy for the two directions are provided below.

The improvement of the quality of the pseudo parallel sentences achieved by suing the filtering algorithm provided in Chapter 4 has increased the translation quality more than it did in the naïve back-translation technique. This improvement is reflected from the BLEU scores presented in Table 5.15 for Tamil to Sinhala translation.

Table 5.15: BLEU Scores from Filtered Back-Translation: BLEU Scores from Filtered Back-Translation

Ratio between authentic parallel sentences : synthetic parallel sentences	**Tamil - Sinhala**	**Sinhala - Tamil**
1 : 1	14.04	7.23
1 : 2	14.75	7.58
1 : 3	15.93	

Again, Sinhala to Tamil translation direction has failed to gain any improvement in the translation quality even using the filtering technique. The BLEU scores reported for this direction almost similar to the BLEU scores from the naïve back-translation approach.

When translating between two languages, one translation direction usually performs better than the other. This difference is more prominent when the linguistic distance between the two languages are high. This prompted us to design a technique which would benefit the translation direction that performs poorly, from the translation direction that performs better. Since the performance of the model will degrade if the synthetic data is overly dominant in the training set, i.e. the benefit of using high-quality authentic parallel data maybe out-weighed by the synthetic back-translated data, we wanted a technique that will make the maximum use of minimum amount of monolingual sentences. We designed the Incrementally Filtered Back-Translation techniques to cater to those two requirements.

5.3.5.3 Incrementally Filtered Back-Translation

We applied the algorithm for Normal back-translation presentenced in chapter 4 and the translation accuracy for the two directions are provided below.

As expected, this new technique was able to create better translation accuracy for both translation directions (shown in Table 5.16). Sinhala to Tamil direction had increased its BLEU score by approximately 2 points, which was a significant improvement.

Table 5.16: BLEU Scores from Incrementally Filtered Back-Translation

Ratio between authentic parallel sentences : synthetic parallel sentences	Tamil - Sinhala	Sinhala - Tamil
1 : 1	14.04	
1 : 2		9.41
1 : 3	15.39	
1 : 4		9.71
1 : 5	16.02	

The importance in the technique is that, the improvement in the BLEU scores are seen at the earlier stages. That is, it makes the maximum use of even the limited amount of monolingual sentences. Furthermore, it is a technique that can be applied to any language pair regardless of being high-resourced or low-resourced.

5.4 Overall Observation and Discussion

An observation we made throughout the experiments were that the translation of Tamil to Sinhala performed better than the translation for Sinhala to Tamil. If we consider the characteristics of the Sinhala and Tamil parallel datasets in Table 3.1, we can clearly see that V to T ratio of the Tamil dataset is almost two times larger than the Sinhala dataset. This is an indication that Tamil is morphologically richer than Sinhala within our corpora. Also it can be observed from the Table 32.1, Sinhala has more number of total words than Tamil. Since in the parallel corpus, the sentences of the two languages have the exact meaning of each other, it can be stated that Sinhala requires more words than Tamil to be used to convey the same meaning. When a language is morphologically richer, the inflectional morphemes add more information about time, count, singularity/Plurality etc. Therefore a morphologically richer language requires only less number of words to convey a message than a relatively les morphologically rich language. Therefore we conclude that within the context of our corpora, Tamil behaves morphologically richer than Sinhala.

NMT is an end-to-end translation. In an encoder-decoder architecture, the encoder encodes a source sentence in an almost language independent representation which will later be decoded on the decoder-side. When the sure-side is morphologically richer than the target-side, the encoder tends to encode more information about the sentence, leading to a better decoding by the decoder. When the source-language is less morphologically rich than the target-side, the encoded sentences does not contain much information for the decoder to deduce a good translation. This justifies why Tamil to Sinhala translation direction produces better translations than for Sinhala to Tamil translations.

Through the course of experiments, we have improved the NMT benchmark by a BLEU score of 11 for Tamil to Sinhala direction, and 7 for Sinhala to Tamil translation direction. This showed us that to improve the translation between two languages, identifying the challenging properties unique to the two languages under consideration and treating them, could take us a long way to reach acceptable translation accuracies.

5.5 Summary

The results from each technique we used were presented in this chapter. Together with the results, an analysis of the BLEU scores, comparing different experiments was also discussed. The potential conclusions that can be drawn from these results are presented in Chapter 6.

Chapter 6 - Conclusions

6.1 Introduction

This dissertation is on developing an NMT system for improving the translation between the morphologically rich and low resource language pairs Sinhala and Tamil. This chapter provides an overall picture on the conclusions drawn from the overall research work conducted by us.

6.2 Conclusions about research problem and research questions

As empirically shown in [7], the amount of parallel sentences in the training set has a positive effect on the translation quality. This was validated as a proof of concept by us, by translating a previously unexplored, highly morphologically rich pair of languages, with different training set sizes. In agreement with the research work of both [2, 7], we observed that the increase in dataset size not only increased the translation quality but after a point, the rate of change of the BLEU score decreases (Figure 5.2). Therefore addressing our first research question "What is the effect of the corpus size on the translation accuracy?", justified the time and effort invested on introducing new techniques that improve the translation quality with even the little amount of data that is available, rather than manually translating sentences (via human translators), to increase the corpus size.

"What is the accuracy of Sinhala-Tamil translation that can be achieved with NMT when compared with SMT?" was the second research question we addressed. This was done by first identifying the challenging properties of Sinhala and Tamil which are their morphological richness and the unavailability of a large number of parallel sentences. These two factors were treated with different techniques. First we explored different preprocessing techniques to reduce the data sparsity, OOV problems that are inherent to morphologically rich languages. There we see that using the 22k parallel corpora NMT did not perform as well as when the same corpus is translated with SMT. This is contradictory to the findings reported by [4] which states that NMT performs better than SMT for the same corpus size. The main reason for this is, NMT requires more parameters to be trained than SMT, hence when the dataset is small, the randomness/degree of freedom within the neural network is too high resulting less intelligible translations. The fact that our focus is on an open domain translation also increases the randomness as there is high probability that the word sense of the same word in two contexts is dissimilar (high word sense ambiguity).

By using word segmentation with Morfessor into morpheme-like units, we could see an improvement in the translation quality. When preprocessed with BPE, the BLEU score could be increased further. When analyzing the sub-word units generated by each of these techniques, we see that Morfessor has segmented each word into almost good morphemes, hence morpheme-like units. With BPE, majority of the words had not been segmented into morphologically sensible units. A similar observation was made in [28], where it is mentioned that for Bengali-Hindi as they are syntactically similar languages, translation using BPE performs better. Sinhala and Tamil too are syntactically similar languages. Because of this, even though the sub-word units are morphologically incorrect, Sinhala and Tamil translation is not affected. This leads to a more general conclusion that for syntactically similar languages, even if a morphological parser which could segment the words into exact morphemes, were available, BPE would perform better. Such an analysis has not been done for Sinhala and Tamil prior to our research work.

Through the course of experiments conducted with naïve back-translation and filtered back-translation proposed in [12, 13] we show their applicability on Tamil to Sinhala translation. The observations conformed to the common wisdom of “more data is better data” in the context of Deep Learning. But the expected improvement in the translation quality through these techniques were not witnessed in the translations conducted from Sinhala to Tamil which questions their applicability across languages.

One of our observations from our research work and previous work is that given two languages, translation in one direction performs better than the other. This distinction is more prominent when one language is morphologically richer than the other. This prompted us to design an algorithm that benefits from this fact and improve the quality of both translation directions. This algorithm, also known as ‘Incrementally Filtered Back-Translation’, manages to help the translations reach high accuracies with minimum amount of monolingual sentences. This is an original contribution by us to the body of knowledge.

Throughout the experiments we noticed that Tamil to Sinhala performs better than Sinhala to Tamil. As discussed in Chapter 5, this is due to the high level of morphological richness shown by Tamil in comparison with Sinhala. While this conclusion was drawn depending on the corpus statistics, we shall not generalize this statement to say that Tamil language is morphologically richer than Sinhala language as it is something that needs to be analyzed linguistically in depth. But our conclusion is that, within the context of our corpora, while both languages are morphologically rich, Sinhala is morphologically poorer with respect to Tamil.

6.3 Limitations

A considerable limitation that we came across was deciding on a suitable merge operation value when using BPE. Although we used the same value for both languages for this parameter, we believe that the translation accuracy could be increased if we could fine tune this value for the two languages separately and this can be explored in future research work.

6.3 Implications for further research

Our work has paved the way for languages that are both morphologically rich and low resourced, to improve their translation accuracy. As we have only focused on supervised techniques currently, we intend to explore the effect of transfer learning and unsupervised techniques on the same preprocessing techniques. Furthermore we believe that the translation accuracy could be further increased if we could fine tune this value for the two languages separately and find the optimum parameter value for this corpus.

The applicability of Incrementally Filtered Back-translation technique can be explored on other languages to establish its validity across languages. Another detailed research that our research open doors to is to analyze how well the improvement in the translation quality is reflected by the BLEU score.

References

[1] R. Weerasinghe, "A Statistical Translation Approach to Sinhala-Tamil Language Translation," *5th International Information Technology Conference*, pp. pp. 136 - 141, 2003.

[2] R. Pushpananda, R. Weerasinghe and M. Niranjan, "Statistical Machine Translation from and into Morphologically Rich and Low Resourced Languages," *Computational Linguistics and Intelligent Text Processing*, pp. pp. 545-556, 2015.

[3] V. Welgama, R. Weerasinghe and M. Niranjan, "Evaluating a machine learning approach to Sinhala morphological analysis," 2009.

[4] L. Bentivogli, A. Bisazza, M. Cettolo and M. Federico, "Neural versus Phrase-Based Machine Translation Quality : a Case Study," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin Texas, USA, 2016.

[5] I. Sutskever, O. Vinyals and Q. Le, "Sequence to Sequence Learning with Neural Networks," *Advances in neural information processing systems*, pp. pp. 3104-3112, 2014.

[6] R. Sennrich, B. Haddow and A. Birch, "Edinburgh Neural Machine Translation Systems for WMT16," in *In Proceedings of WMT*, 2016b.

[7] P. Koehn and R. Knowles, "Six challenges for neural machine translation," in *In Proceedings of First Workshop on Neural Machine Translation. Association for Computational Linguistics*, 2017.

[8] R. Weerasinghe, D. Herath, V. Welgama, N. Medagoda, A. Wasala, and E. Jayalatharachchi, "UCSC Sinhala Corpus - PAN Localization Project-Phase I," 2007.

[9] R. Weerasinghe, R. Pushpananda, and N. Udalamatta, "Sri Lankan Tamil Corpus," Technical report, University of Colombo School of Computing and funded by ICT Agency, Sri Lanka, 2013.

[10] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *In Proceedings of the Empiricial Methods in Natural Language Processing*, 2014.

[11] J. Zhang and C. Zong, "Exploiting source-side monolingual data in neural machine translation," in *In proceedings of EMNLP*, 2016.

[12] R. Senrich, B. Haddow and A. Birch, "Improving Neural Machine Translation

Models with Monolingual Data," *In Proceedings of the Annual Meeting of the Association of Computational Linguistics, ACL,* pp. pp.567-573, 2016.

[13] A. Imankulova, T. Sato, and M. Komachi. , "Improving Low-Resource Neural Machine Translation with Filtered Pseudo-Parallel Corpus," *In Proceedings of the 4th Workshop on Asian Translation,* p. pp. 70–78, 2017.

[14] C. Kyunghyun, B. Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014.

[15] D. Bahdanau, C. Kyunghyun and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," 2014.

[16] P. Kishore, S. Roukos, T. Ward, and W. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," *In Proceedings of the 40th annual meeting on association for computational linguistics,* pp. pp.311-318, 2002.

[17] P. Koehn, F. J. Och, and D. Marcus, "Statistical Phrase-Based Translation," *In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Association for Computational Linguistics,* vol. Volume 1, p. pp. 48–54, 2003.

[18] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," 2015.

[19] P. Smit, S. Virpioja, S. Gronroos, and M. Kurimo, "Morfessor 2.0: Toolkit for statistical morphological segmentation," *In Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics,* pp. pp. 21-24, 2014.

[20] M. Fadaee, A. Bisazza, and C. Monz, "Data Augmentation for Low-Resource Neural Machine Translation," in *In Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL*, 2017.

[21] M. A. D. Gangi and M. Federico, "Can monolingual embeddings improve neural machine translation?," in *In Proceedings of of CLiC-it*, 2017.

[22] D. Gangi and M. Federico, "Monolingual Embeddings for Low Resourced Neural Machine Translation," 2017.

[23] P. Ramachandran, P. J. Liu and Q. V. Le, "Unsupervised pretraining for sequence to sequence learning," 2016.

[24] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," in *In International Conference on Learning Representations*, 2018.

[25] J. Chung, K. Cho, and Y. Bengio., "A character-level decoder without explicit segmentation for neural machine translation," in *Association for Computational Linguistics (ACL)*, 2016.

[26] P. Tennage, P. Sandaruwan, M. Thilakarathne, A. Herath, S. Ranathunga, G. Dias, and Jayasena, "Neural Machine Translation for Sinhala and Tamil Languages," 2017.

[27] A. Toral and V. M. Sanchez-Cartagena, "A multifaceted evaluation of neural versus phrasebased machine translation for 9 language directions," *In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics,* vol. Volume 1, p. pp. 1063–1073, 2017.

[28] T. Banerjee and P. Bhattacharya, "Meaningless yet Meaningful: Morphology Grounded Subword-level NMT," in *In Proceedings of the Second Workshop on Subword/Character Level Model*, New Orleans, Louisiana, June 6, 2018.

[29] G. Klein, Y. Kim, Y. Deng, J. Senellart and A. M. Rush, "OpenNMT: Open-source toolkit for neural machine translation," 2017.

[30] H. Chen, S. Lundberg and S. Lee, "Checkpoint Ensembles: Ensemble Methods from a Single Training Process," 2017.