Pre-training and Fine-tuning Multilingual Sequence-To-Sequence Models for Domain-Specific Low-Resource Neural Machine Translation

Sarubi Thillainathan

208037K

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree Master of Science in Computer Science and Engineering

Department of Computer Science & Engineering

University of Moratuwa

Sri Lanka

April 2022

DECLARATION

I, Sarubi Thillainathan, declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the Masters thesis/Dissertation under my supervision.

Name of Supervisor: Dr. Surangika Ranathunga

Signature of the Supervisor: Date:

Name of Supervisor: Prof. Sanath Jayasena

Signature of the Supervisor: Date:

ABSTRACT

Limited parallel data is a major bottleneck for morphologically rich Low-Resource Languages (LRLs), resulting in Neural Machine Translation (NMT) systems of poor quality. Language representation learning in a self-supervised sequence-to-sequence fashion has become a new paradigm that utilizes the largely available monolingual data and alleviates the parallel data scarcity issue in NMT. The language pairs supported by the Self-supervised Multilingual Sequence-to-sequence Pre-trained (SMSP) model can be fine-tuned using this pre-trained model with a small amount of parallel data.

This study shows the viability of fine-tuning such SMSP models for an extremely low-resource domain-specific NMT setting. We choose one such pre-trained model: mBART. We are the first to implement and demonstrate the viability of non-English centric complete fine-tuning on SMSP models. To demonstrate, we select Sinhala, Tamil and English languages in extremely low-resource settings in the domain of official government documents.

This research explores the ways to extend SMSP models to adapt to new domains and improve the fine-tuning process of SMSP models to obtain a high-quality translation in an extremely low-resource setting. We propose two novel approaches: (1) Continual Pre-training of the SMSP model in a self-supervised manner with domain-specific monolingual data to incorporate new domains and (2) multistage fine-tuning of the SMSP model with in- and out-domain parallel data.

Our experiments with Sinhala (Si), Tamil (Ta) and English (En) show that directly fine-tuning (single-step) the SMSP model mBART for LRLs significantly outperforms state-of-the-art Transformer based NMT models in all language pairs in all six bilingual directions. We gain a +7.17 BLEU score on Si→En translation and a +6.74 BLEU score for the Ta→En direction. Most importantly, for non-English centric Si-Ta fine-tuning, we surpassed the state-of-the-art Transformer based NMT model by gaining a +4.11 BLEU score on Ta→Si and a +2.78 BLEU score on Si→Ta.

Moreover, our proposed approaches improved performance strongly by around a +1 BLEU score compared to the strong single-step direct mBART fine-tuning for all six directions. At last, we propose a multi-model ensemble that improved the performance in all the cases where we obtained the overall best model with a +2 BLEU score improvement.

Keywords: Neural Machine Translation, Pre-trained Language Models, Pre-training, Fine-tuning, Low-Resource languages, mBART

DEDICATION

With deepest gratitude, I dedicate this research to my Grandpa, *Late M. Saravanamuttu*. And, of course, to my forever loving family for supporting my dreams, no matter what!

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisors Dr Surangika Ranathunga and Prof. Sanath Jayasena, for the continuous support and guidance given for the success of this research. Without your insights and tremendous mentorship, I could not have achieved this level. Thank you for always stepping in to help whenever I needed regardless of your busy schedules.

I wish to convey my sincere appreciation to Prof. Gihan Dias and National Language Processing Center (NLPC) members for their valuable insights and support given for this research. I would also like to thank the entire Department of Computer Science and Engineering staff, both academic and non-academic, for their help and for providing me with the resources necessary to conduct my research. This research was supported by the University of Moratuwa AHEAD project Research Grant.

Lastly, I want to thank my family and friends who supported me though this journey.

Thank you!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
MT	Machine Translation
SMT	Statistical Machine Translation
NMT	Neural Machine Translation
MNMT	Multilingual Neural Machine Translation
RNN	Recurrent Neural Networks
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
DAE	Denoising Autoencoder
MLE	Maximum Likelihood Estimate
TL	Transfer Learning
SMSP	Self-supervised Multilingual Sequence-to-sequence Pre-trained
FT	Fine-Tuning
LRL	Low-Resource Language
LM	Language Model
M-FT	Multilingual Fine-Tuning
B-FT	Bilingual Fine-Tuning
CPT	Continual Pre-Training

Chapter 1

INTRODUCTION

1.1 Background

Machine Translation (MT) refers to the process of automatically translating one human language to another using a machine. It is one of the most challenging Natural Language Processing (NLP) tasks [1]. For decades, Statistical Machine Translation (SMT) has been widely used, which is typically a phrase-based system that translates the sequence of words or phrases from a source language to a target language [4]. In the recent past, Deep Learning techniques have been successfully used to translate one language to another. This technique is known as Neural Machine Translation (NMT). This state-of-the-art NMT is shown to outperform SMT [5]. However, NMT requires a large amount of parallel data for training to produce high-quality translations. Finding large parallel corpora is challenging when it comes to morphologically rich LRLs. Consequently, NMT tends to perform poorly in low-resource settings. Adapting the threshold proposed by Ranathunga et al. [1], we define an NMT task as low-resource and extremely low-resource when the available parallel corpus is below 0.5M and below 0.1M, respectively. However, this is not a hard threshold.

Due to data scarcity issues, transferring knowledge from already trained (known as pre-trained) Deep Learning models to low resource NMT has been widely studied [3, 6, 7]. One prominent state-of-the-art approach is extracting and sharing the learnt representations from Self-Supervised Multilingual Sequence-to-sequence Pre-trained (SMSP) models (mBART [6, 7], mT5 [8]) to a downstream NMT task which has shown significant performance gains due to the knowledge transfer from the pre-trained models [6, 7, 9]. These SMSP models are used to initialize the NMT model and then further train it with parallel data (known as Fine-tuning). Also, researchers discovered that the NMT framework could naturally incorporate multiple languages [10]. Hence, there has been a massive increase in MT systems that involve more than one language, commonly known as multilingual NMT systems (MNMT) [10].

NMT research has been conducted among Sri Lankan languages, mainly focusing on bilingual pairs such as Sinhala-Tamil [11, 12, 13, 14], Sinhala-English [15, 16] and Tamil-English [17]. Initial NMT results were not superior compared to SMT. The major issue was Long Short Term Memory (LSTM) based NMT [5] requires significant amounts of parallel data not available in our low-resource NMT setting. However, recently introduced Transformer [18] models showed superior performance even in low resource settings. Studies evidenced that the Transformer model beats LSTM based NMT and SMT in extremely low resource settings for Tamil-Sinhala [14] and Sinhala-English [15].

1.2 Research Problem

Even though the state-of-the-art NMT based models showed promising results for LRL pairs, still it was much lower than what high resource language pairs could obtain in NMT [19]. Also, MT on domain-specific settings such as government documents with extremely LRLs is still quite challenging. So far, only vanilla bilingual NMT models have been implemented for the Sinhala, Tamil and English languages, with some proposed techniques such as data augmentation [12], back-translation [20, 21], transliteration [13] and Byte Pair Encoding (BPE) [15]. Some of these improved models have surpassed SMT by a slight margin [12, 13, 15, 20, 21].

One promising approach is fine-tuning the SMSP model with parallel data has emerged as a new paradigm in low-resource Neural Machine Translation. As most of the SMSP models [6, 7, 8] support the languages considered in this study, there is an opportunity for us to use them in the context of the considered language pairs. Some studies [6, 7] considered Sinhala-English and Tamil-English pairs as part of their study along with other languages to demonstrate the viability of fine-tuning the SMSP model: mBART for low-resource bilingual and multilingualism NMT settings in the open domain. However, Si-En and Ta-En results are much lower than high resource language pairs. Also, domain-specific cases and non-English centric studies such as Si-Ta pairs have not been studied.

Even though pre-trained models have offered promising results for LRLs, simple one-step fine-tuning with parallel data yields sub-optimal results in the context of languages under-represented in the SMSP models (i.e. languages for which the SMSP model has been

pre-trained with low amounts of monolingual data) [2, 6, 7]. This problem gets aggravated when the considered translation is domain-specific. Therefore, improving the pre-training and fine-tuning processes of the SMSP models for low-resource language NMT (LRL-NMT) is vital to reap their full benefit on LRLs.

1.3 Research Scope and Objectives

This study focus to utilize SMSP model mBART for domain-specific LRLs. The objectives of this research are as follows:

- Improve the fine-tuning process of mBART SMSP models using in- and out domain data for an extremely low-resource NMT setting in Sinhala, Tamil and English languages.
- Propose non-English centric fine-tuning on mBART SMSP models.
- Utilize the monolingual data and implement Continual Pre-Training (CPT) on the mBART SMSP model for domain adaptation
- Propose ensembling techniques on fine-tuned mBART SMSP models.

1.4 Contributions

We make the following contributions to this thesis:

- Demonstrated the viability of fine-tuning SMSP models for an extremely low-resource NMT setting in Sinhala, Tamil and English languages.
- The first study demonstrated the viability of non-English centric fine-tuning on SMSP models.
- Extended the self-supervised training from the SMSP model to incorporate new domains - [Continual Pre-training for Domain Adaptation].
- Introduced Multistage Fine-tuning on SMSP models.
- Ensembled the models to get the best result for the considered languages.

1.5 Publications

- **Sarubi Thillainathan**, Surangika Ranathunga and Sanath Jeyasena, "Fine-tuning Self-Supervised Multilingual Sequence-To-Sequence Models for Extremely Low-Resource NMT" in The 7th Moratuwa Engineering Research Conference (MERCon 2021), Moratuwa, Sri Lanka, Jul 27-29, 2021.

  - **Published** and **won the best paper award** in the Big Data, Machine Learning, and Cloud Computing Track.

- **Sarubi Thillainathan**, Surangika Ranathunga and Sanath Jeyasena, "Pre-training and Fine-tuning Multilingual Sequence-To-Sequence Models for Domain-Specific Low-Resource Neural Machine Translation"

  - Submitted to Information Processing Management Journal.

Chapter 2

LITERATURE SURVEY

2.1 Overview

Under this section, initially, we discuss NMT and NMT related studies on the languages considered in this study languages used in Sri Lanka. We also provide an in-depth analysis of previous research that has contributed to the domains of Multilingual NMT (MNMT), Transfer Learning (TL) and techniques proposed to improve it. Finally, we critically analyze the available state-of-the-art pre-trained models for NMT and techniques tried out so far to fine-tune the pre-trained models.

2.2 Neural Machine Translation (NMT)

An NMT [5, 22, 23] system is implemented as a single end-to-end system that directly trains the source sentences as input and target sentences as the output. Initial NMT has used two Recurrent Neural Networks (RNN); one RNN encodes (encoder) a variable-length source sentence using a bidirectional RNN into a fixed-length vector representation, then this vector is decoded using another RNN (decoder) into a variable-length target sentence. It is typically called an RNN encoder-decoder approach [5]. However, RNN suffers from the vanishing gradient problem where it fails to capture long dependencies. Due to this, Long Short Term Memory (LSTM) [24] and then Gated Recurrent Units (GRU) [25] based RNN have been proposed.

Despite these achievements, LSTM struggles with longer sequences of text, where the fixed-length internal representation fails to decode all words in the output sequence. An attention mechanism was added to the LSTM model [23] to alleviate this issue, but it was not quite successful. Later, the Transformer model was proposed by Vaswani et al. [18] to address this issue. It is a simple network solely based on the attention mechanisms, dispensing recurrence and convolutions entirely. This model had achieved promising results by surpassing LSTMs [5, 22, 23]. It can successfully handle the longer sentences more

effectively because it is a multi-head self-attention mechanism. Since then, Transformer based architecture has become the state-of-the-art paradigm for NMT.

2.3 NMT for Sinhala, Tamil, and English Languages

Some NMT research exists for Sinhala-Tamil [11, 12, 13, 14], Sinhala-English [15, 16] and Tamil-English [17]. Early LSTM-based NMT research for the Sinhala-Tamil language pair could not beat SMT [11]. The major issue was that LSTM-based NMT [5] requires significant parallel data. However, it is unavailable in the low-resource NMT setting. Various strategies were proposed to improve NMT in the context of these three language pairs. Tennage et al. [12] applied different data augmentation techniques and showed improvement in NMT models. Back-translation was also successfully merged to above mentioned LRLs [20, 21, 26]. Also, studies explore the transliteration [13] method by converting all the language scripts to a common script format. Some of these improved models have surpassed SMT by a slight margin. However, later introduced Transformer [18] models showed superior performance even in low-resource settings. A comparative study [14] showed that the Transformer model beats LSTM based NMT and SMT in extremely low-resource settings.

Another line of research is applying different subword tokenization techniques such as Byte Pair Encoding (BPE) [13] to improve NMT for these languages. Choudhary et al. [27] used pre-trained word embeddings (BPEmb) on NMT to develop an efficient translation system that overcomes the Out Of Vocabulary problem for Tamil-English languages. Subsequently introduced Transformer model with Byte Pair Encoding (BPE) showed effectiveness in low-resource settings compared to the Transformer model without BPE [15, 16].

In studies [19, 28, 29], Sinhala-English and Tamil-English pairs reported improved results within a Multilingual NMT system. These languages benefited from knowledge transfer from the high-resource languages that are trained together. Since almost all the Multilingual NMT systems are English-centric, Sinhala-Tamil needed to explore respect to Multilingual NMT systems. Even though these models showed better translation for the languages considered in this study in a low-resource setting, still it was much lower than

what NMT could obtain for high resource language pairs.

2.4 Multilingual Neural Machine Translation (MNMT)

Early NMT (both RNN and Transformer models) mainly was a bilingual system that typically handled one translation task. Researchers later discovered that the NMT framework could naturally incorporate multiple languages; thus, there has been a massive increase in work on NMT systems that involve more than two languages [10]. MNMT gets categorized mainly into three types:

- One-to-Many (O2M): Translate from one source language to multiple target languages [30, 31, 32]; as shown in Figure 2.1-(i).
- Many-to-One (M2O): Translate multiple input source languages to one specific target language [32, 33]; as shown in Figure 2.1-(ii).
- Many-to-Many (M2M): Translate multiple input source languages to multiple target languages, in other words, multi-way NMT [19, 29, 31, 32, 34, 35]; as shown in Figure 2.1-(iii).

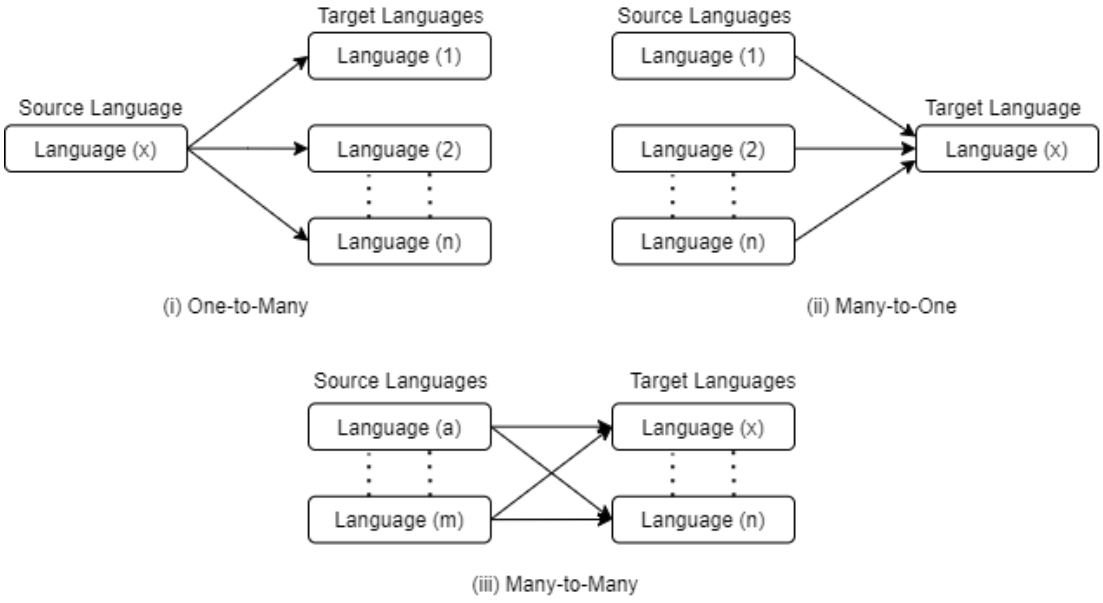

Figure 2.1: Overview of MNMT Categories [1]

Initial studies [30, 31] mainly focused on One-to-Many (as shown in Figure 2.1-(i)) systems with one common basic encoder and dedicated decoders as described in Figure 2.2-(i). The first end-to-end trial of MNMT was carried out by Dong et al. [30], which simultaneously translated sentences from one source language to multiple target languages. Since the source language is the same in all these MT tasks, the NMT model benefited by having one common encoder shared among all these MT tasks and dedicated task-specific decoders. However, that was not a superior approach that failed to include the attention mechanism. Later, the multi-way (M2M) MNMT system proposed by Firat et al. [34] extended the initial studies to have multiple sources and target languages. They used a separate encoder for each source language, a separate decoder for each target language and most importantly, only a single attention mechanism shared across all the language pairs. (Here, one encoder per source language, meaning that a single encoder is shared for translating that particular language to multiple target languages.) The only difference between the bilingual models and the proposed MNMT ones is the N number of encoders and M number of decoders as described in Figure 2.2-(iii).

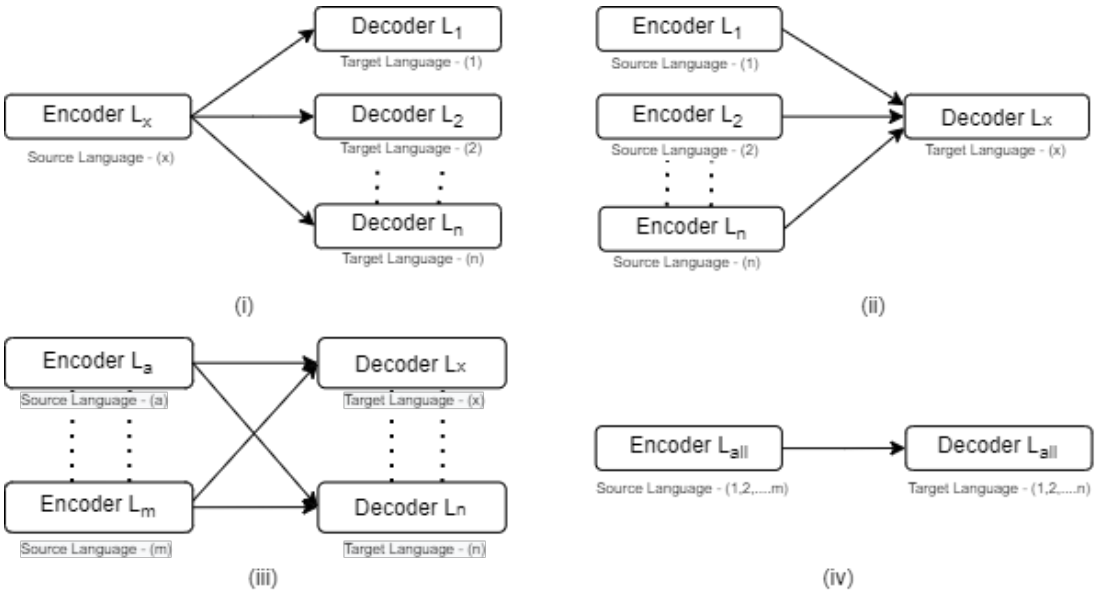

Figure 2.2: Overview of MNMT Architectures [1]

In the aforementioned MNMT systems, the number of parameters grows linearly with the number of languages, but it grows quadratically in the bilingual system [10]. However, the challenge is that the shared attention mechanism needs to bear the burden of connecting

all different language pairs, which might fail to learn the necessary representation for the best translation quality [10]. After Ha et al. [35] and Johnson et al. [32] showed that there is no need to have multiple encoder/decoder models with a large number of parameters, researchers started focusing on the universal/single MNMT model, as shown in Figure 2.2-(iv).

2.4.1 Universal Encoder and Decoder Architecture for MNMT

The universal/single encoder and decoder framework, as described in Figure 2.2-(iv) for MNMT proposed by Ha et al. [35], is inspired by the multi-source NMT (M2O) [33]. Here they enforced the language-specific coding for each input/source language (i.e. @de@Flussufer; @en@bank) and the target forcing mechanism where the beginning and at the end of every source sentence, a special symbol indicating the language they would translate into (<E> @de@darum @de@geht <E>). However, Johnson et al. [32] introduced a much simpler and more effective approach by introducing an artificial token at the beginning of the input sentence to specify the required target language without changing the standard bilingual NMT model architecture. Here they haven't specified the source language; instead, the model learnt this automatically. This approach has become a turning point for MNMT and has became very popular.

Extensive experiments conducted on MNMT [19, 29, 30, 32, 33, 34, 35] evidenced the performance improvement over the bilingual NMT system and observed strong knowledge transfer from high resource languages to LRLs. Languages that have scarce parallel corpora have benefited from data in other languages used to train together. Here, the model can learn an interlingua by learning semantic information of shared tasks in a more generalized way, resulting in a better translation quality than bilingual baseline NMT systems. MNMT models also enable zero-resource MT [32, 34, 36, 37, 38] as well.

2.4.2 Strategies to improve MNMT

There have been various efforts to improve the MNMT systems, such as Task-specific attention models [39] or an explicit neural interlingua into a multilingual encoder-decoder NMT, an attentional encoder that converts language-specific embeddings to language-

independent ones [37]. Apart from that, strategies to improve one-to-many cases in the MNMT system are typically performed low by designing two special labels and a new parameter sharing mechanism that divides each decoder layer's hidden units into shared and language-dependent ones [40]. Unified Transliteration and Subword Segmentation leverage the language similarity while exploiting parallel data from related language pairs [41]. Studies adapt trained models to work with new language pairs and continuously add new language pairs to grow the MNMT systems with dynamic vocabulary [42]. Besides that, comparison studies investigated the translation quality between dominant neural architectures Recurrent and Transformer [43]. Their studies showed that the Transformer approach delivered the best performing multilingual models, with a larger gain over corresponding bilingual models than observed with RNNs. Also, Multilingual models consistently outperformed bilingual models with respect to all considered error types such as lexical, morphological, and reordering.

2.5 Transfer Learning (TL) in NMT

Transfer Learning (TL) is a Machine Learning technique where the knowledge of an already trained ML model is applied to a different but related problem. The first study to apply TL to NMT was conducted by Zoph et al. [44]. Here they first trained an NMT model on a high-resource language pair and then used the resulting trained network (the parent model) to initialize and continue to training for the LRL pair (the child model), as shown in Figure 2.3. In other words, high-resource pair (X→Y) is used to help a low-resource pair (A→Y) where Y is usually English. So low-resource NMT model will not start with random weights instead of with the weights from the parent model. TL on NMT resulted in better performance improvement over the baseline NMT models. Zoph et al. [44], Dabre et al. [45], Nguyen and Chiang [46] showed that the parent language can make a difference in performance improvement on child NMT. So selecting a parent language from the same (or linguistically similar) language family as the child's language has a larger impact on TL [44, 46].

TL can be categorised mainly as a warm and cold start. If the parallel data was presented for the LR language while training, the parent model is called a warm start or a

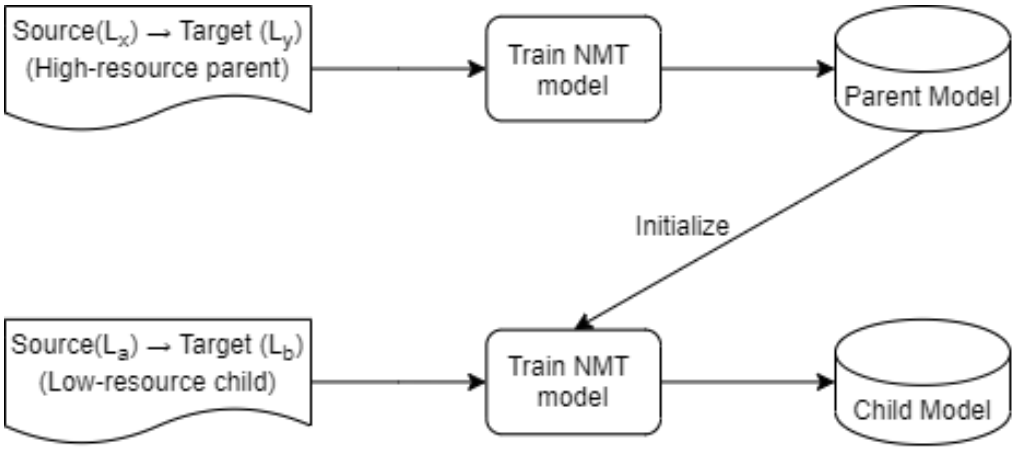

Figure 2.3: Overview of Transfer Learning

cold start [47]. Since the warm start was expected to be accurate; most of the previous work [44, 45, 46] has focused on it. In contrast, training a strong high resource model which is capable of quickly adapting to a new language that never has never been seen before is also possible as a cold start [42]. Most of the initial studies used the bilingual parent model where parent and child share the same target language [44, 45, 46] as a warm start. Later research focused on training an MNMT model that can be either O2M, M2O or M2M [47] and then fine-tuning for an LR language. Overall, these transfer learning techniques typically reduce the data requirement of the child pair and significantly improve the results on the child pair with the faster convergence over a model trained from scratch.

Several studies have been conducted to improve the TL with many dimensions, mainly on different fine-tuning techniques or TL protocols.

2.5.1 Fine-tuning techniques

Typically weights of the parent model are initialized to the low-resource NMT training, thus avoiding the need to train the low-resource NMT model from scratch. Here training starts with the parent model weights instead of random initialization. The pre-trained model parameters get fine-tuned for the selected translation task through this training. Fine-tuning can be done in many ways, as listed below.

1. No fine-tuning at all.

   Here parent model is copied entirely to the child model [46, 48, 49]. All the layers

have been frozen without any change at all.

2. Fine-tuning the embedding layer only.

   In some cases, we only want to fine-tune the embedding layers according to child vocabulary [42, 44, 46]. When the parent and child have the same target language, the parent decoder embedding can be directly used. Otherwise, the decoder embedding has to be initialized before fine-tuning the child task randomly. It completely depends on the similarity or relatedness between the parent and child language pairs. Alternatively, we can have one common shared vocabulary for parent and child languages.

3. Fine-tuning the whole parent model.

   Once the parent model is initialized, all the parameters are fine-tuned according to the child language by adjusting weights [42, 49, 50, 51]. This is a common and widely used approach where no layers are frozen from the parent model.

4. Partial fine-tuning.

   Here only the selected layers of encoders and decoders are fine-tuned [44, 48, 50, 52].

2.5.2 Transfer Learning Protocols

The basic simple transfer learning approach is to fine-tune the child pair with a high resource parent model where target language was common on both child and parent. Several studies have also explored many ways to conduct effective TL on NMT.

1. Hybrid Transfer Learning (HTL)

   Sharing lexicon embedding between parent and child languages without leveraging back translation or manually injecting noises [53].

   - First, train the High-Resource Languages as the parent model with its vocabularies.

- Then, combine the parent and child language pairs using the oversampling method to train the hybrid model initialized by the previous parent model.
- Finally, fine-tune the morphologically rich child model using a hybrid model.

2. First, train a multilingual NMT model on out-of-domain data, then fine-tune on in-domain parallel and back-translated pseudo-parallel data [25, 54].

   For example: First train a multilingual NMT model on out-of-domain Ja↔En and Ru↔En data, then fine-tune it on in-domain Ja↔En and Ru↔En data, and further fine-tune it on Ja↔Ru data. Studies show that this stage-wise fine-tuning is beneficial for high-quality translation [54].

3. First, train an NMT model on an out-of-domain parallel corpus, and then fine-tune it on a parallel corpus that mixes with in-domain and out-of-domain corpora [55]- Mixed Fine Tuning.

4. First, train on unrelated high-resource language pair, then fine-tune it on a similar intermediate language pair and then finally fine-tune it on the LRL pair [54, 56].

5. Training SMSP model using large scale monolingual data (known as pre-trained models) and then fine-tuning with the parallel data.

   This is a state-of-the-art strategy to extract and share learnt representations from pre-trained models with other downstream tasks, such as NMT. Studies show this leads to significant performance gains due to the knowledge transfer from pre-trained models [6, 7].

2.6 Pre-trained Models for NMT

Building large NMT models is a great challenge for NMT due to the lack of parallel data for LRLs. Instead of relying on inadequate parallel corpora, studies leverage the easily found large scale unlabeled monolingual data. A good universal language representation can be learned from the monolingual data by capturing linguistic characteristics, lexical meanings, and syntactic and semantic structures. In the recent past, pre-trained models such as BERT - Bidirectional Encoder Representation from Transformer [9], OpenAI GPT

- Generative Pre-training [57], BERT based pre-trained models such as RoBERTa [58] showed a powerful ability to learn universal language representations using monolingual data. Notable studies [59, 60, 61] have explored how to incorporate BERT to inject prior contextual word embedding knowledge on the encoder part and GPT for the decoder part of NMT. However, these models are not well-suited for NMT since NMT requires an encoder-decoder architecture. Later work was introduced to train complete sequence-to-sequence models with encoder-decoder architectures, which pontifically befitted for NMT [3].

The BART [3] model is the first complete self-supervised sequence-to-sequence (encoder-decoder) model trained on large scale (English) monolingual data. Later BART was extended to incorporate more than one language, resulting in mBART. mBART is a Self-supervised Multilingual Sequence-to-sequence Pre-trained (SMSP) model. It is pre-trained on a large-scale unlabeled monolingual corpus of 25 different languages, result mBART25 [6]. An extended work of the mBART model has incorporated up to 50 languages; mBART50 [7]. Another well-known sequence-to-sequence model is T5 [62], a Text-to-Text Transfer Transformer trained over a large English monolingual corpus similar to BART [3]. mT5 [8] is the multilingual version of T5, which includes 101 languages. Subsequently, introduced mT6 [63] was additionally trained with parallel data on mT5.

2.7 Self-supervised Multilingual Sequence-to-sequence Pre-training

Most state-of-the-art sequence-to-sequence models [61] have been pre-trained as self-supervised learning tasks using monolingual data. Self-supervised learning is the same as supervised learning. Here training data labels are generated automatically instead of relying on manually annotated data [61]. In training, monolingual data is noised and fed as a source and original monolingual data as the target. At the end of the training, the model can take a partially noised input and predict the denoised words to recover the original sentence. Here we briefly describe the BART pre-trained model and discuss the mBART model we used in this study.

Input	A B C . D E .
Token Masking	A _ C . _ E .
Token Deletion	A . C . E .
Text Infilling	A _ . D _ E .
Sentence Permutation	D E . A B C .
Document Rotation	C . D E . A B

Table 2.1: Sample input and its transformations output after applying different noising functions [3]

2.7.1 BART

BART [3] is the first method for pre-training a complete sequence-to-sequence denoising auto-encoder. BART is trained on a large scale (English) monolingual data by,

1. First corrupting text with an arbitrary noising function.

2. Then, learn a model to reconstruct the original text.

BART mainly applies techniques such as Token Masking, Token Deletion, Text Infilling, Sentence Permutation, and Document Rotation to noising the text [3]. The sample input and corresponding arbitrary noising function outputs are shown in Table 2.1. As we can see, these arbitrary noising transformations even allow changing the sentence length.

BART is implemented as a standard sequence-to-sequence Transformer architecture [18] by having a bidirectional auto-encoder and a left-to-right autoregressive decoder. BART comprises two pre-trained models, a bidirectional encoder as BERT [9] and a left-to-right decoder as GPT [57]. The base BART model consists of 6 layers of encoder and decoder, and the large model contains 12 layers of encoder and decoder. Pre-training optimizes the negative log-likelihood of the original document.

2.7.2 mBART

The mBART model is a multilingual BART, extending the pre-training into multiple monolingual languages. Same as BART, the mBART model follows 12 layers of encoder and 12 layers of a decoder. Additionally, it includes a layer-normalization layer on top of both the encoder and decoder.

There are two mBART models one is mBART25 [6], which supports 25 languages and mBART50 [7], which supports 50 languages. The monolingual corpus has been extracted from the Common Crawl (CC) for pre-training. Like BART, all the monolingual data is corrupted with a noising function then trained on a single denoising autoencoder that maps corrupted sentences to the original sentences, as shown on the left side of Fig. 2.4. This model is pre-trained using two types of noise function: random span masking and order permutation [6]. mBART is more memory efficient than mT5, which is comparatively large and supports 101 languages. Also, mBART has shown relatively better results than mT5 for translation [64]. Thus in this study, we choose the mBART50 [7] model, which supports both Sinhala and Tamil, as mBART25 doesn't support Tamil.

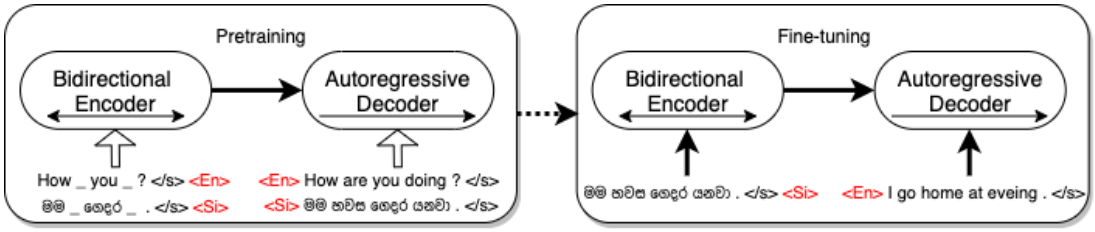

Figure 2.4: Overview of Multilingual Denoising Autoencoder Pre-training (left) and fine-tuning on NMT (right) [2]. A special token "language id" is added to both the encoder and decoder.

2.8 Fine-tuning Multilingual Self-Supervised Pre-trained Models for Low-resource NMT

As discussed in Sections 2.5 and 2.6, transferring and sharing the knowledge from high-resource languages to LRLs has been widely studied. In MNMT models, LRLs obtain knowledge by joint training with high-resource languages. On the other hand, different transfer learning approaches are also widely known to share the knowledge from already trained NMT models. Among these, one of the state-of-the-art transfer learning methods is fine-tuning the multilingual self-supervised pre-trained models (trained monolingual data) for low-resource MT. An overview of pre-training and fine-tuning is described in Fig. 2.5.

Even though the aforementioned pre-trained models [3, 6, 7] were trained as an encoder-decoder architecture, these models themselves cannot be directly used as an NMT model. Instead, these pre-trained models can initialize NMT training, thus avoiding the need to

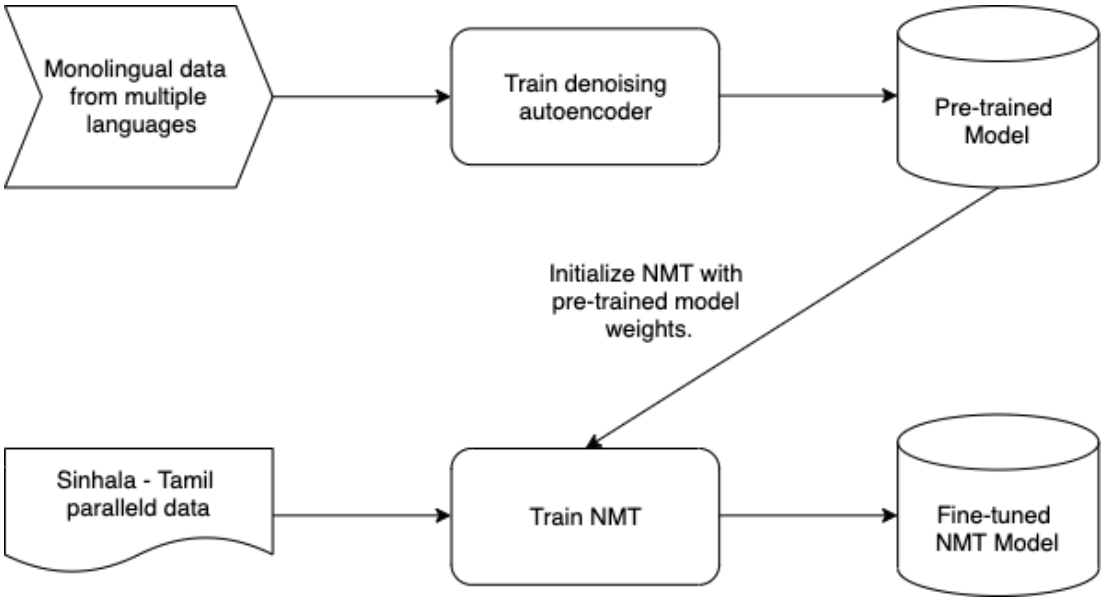

Figure 2.5: Overview of Pre-training and fine-tuning.

train the low-resource NMT model from scratch. This process is referred to as *fine-tuning*- try to fine-tune the pre-trained model parameters to incorporate a new language or domain.

Currently available multilingual sequence-to-sequence pre-trained models are mBART, mT5, and mT6. These models can fully initialize both the encoder and decoder of an NMT model with the corresponding source and target languages. In particular, the mBART model has shown promising results [6, 7] on supervised and unsupervised MT at both the sentence and document levels. Significant gains have been observed in LRL pairs such as English-Vietnamese/Turkish [6, 7].

mBART related previous studies [6, 7] have considered Sinhala-English and Tamil-English pairs in demonstrating the viability of fine-tuning the mBART model for low-resource bilingual and multilingual NMT settings [6, 7]. Their fine-tuning experiments are categorized into three groups upon the size of the parallel data, low-resource ($<$1M), medium resource ($>$1M and $<$10M), and high resource ($>$10M). The English-Sinhala/Tamil language pairs fall into the low-resource category. They used FLoRes [65] data with 565K Open Subtitles and GNOME/KDE/Ubuntu sentences obtained from the OPUS repository[1] for English-Sinhala, and WMT'20[2] data for English-Tamil pair with 609k sentences from various sources such as Wiki Titles v2, WikiMatrix, Indian Prime Minister's news updates

[1] `http://opus.nlpl.eu/`

[2] `http://www.statmt.org/wmt20/translation-task.html`

and many more. Results show improvements in low-resource pairs compared to NMT models trained from scratch in English→Sinhala/Tamil and Sinhala/Tamil→English directions.

However, all the experiments conducted by the above research have been English centric, i.e. the tested language pair included English either on the source or target side. In other words, they have only tested Bilingual Fine-Tuning (B-FT) with Sinhala-English and Tamil-English, but not Sinhala-Tamil. Even though Tang et al. [7] experimented with fine-tuning multiple language pairs simultaneously - which is called Multilingual Fine-Tuning (M-FT), i.e. (M2O (Any→English), O2M (English→Any), and M2M (Any↔Any) with English as a pivot language), they also did not test non-English centric fine-tuning. Some studies [66, 67] considered non-English centric fine-tuning where only one language in the pair is included in the mBART model. Madaan et al. [66] fine-tuned the mBART25 [6] model for Hindi-Marathi and Spanish-Portuguese, where the mBART25 model supported only Hindi and Spanish.

2.9 Continual learning on Self-Supervised Pre-trained Models (Extending the pre-trained models)

Continual learning to incorporate new languages or domains has been successfully studied in MNMT MT models. Lakew et al. [42] adapted trained MNMT models to work with new language pairs and continuously added new language pairs to grow the MNMT systems with dynamic vocabulary. Some studies pre-trained the NMT model on a large open-domain corpus and freeze all the pre-trained model parameters [68]. Then they injected a set of domain-specific adapter layers for every target domain, and these injected adapters were fine-tuned to maximize performance on the corresponding domains.

Similarly, when training large self-supervised pre-trained models using a large corpus of monolingual data, it is impossible to cover all the languages available in the world and different domains. So far maximum of around 101 languages was considered in the large pre-trained model [8]. Studies have addressed these limitations on pre-trained models by extending the self-supervised pre-trained models to either support new language or new domains [7, 69, 70, 71].

Tang et al. [7], Liu et al. [69], Chen et al. [70], Susanto et al. [71] extended the pre-trained models to incorporate new languages. Their main goal is to add new languages to the selected pre-trained models. Tang et al. [7] have extended the initial mBART-25 model to support more languages. They added randomly initialized vector space as embedding layers for the newly added languages. They combined monolingual data of the original and newly added languages and then continued pre-training in a self-supervised manner to extend the mBART model.

Liu et al. [69] adapted mBART to unseen languages by mixing target language monolingual data with similar source language text that is supported by mBART and then continued to pre-trained the mBART model. A very recent study submitted for WNMT20 shared news translation task by Chen et al. [70] focused on Tamil↔English and Inuktitut↔English low resource setting pairs. They continued to pre-train the mBART25 [6] model across 13 languages on all monolingual data provided by WMT20 to support unseen languages such as Tamil and Inuktitut. Along with this, they explored multilingual/bilingual fine-tuning, data augmentation, and reranking [70].

2.10 Summary

Two solutions applicable in the context of LRL-NMT are Transfer learning (TL) (as discussed in Section: 2.5) and Multilingual Neural Machine Translation (MNMT) (as discussed in Section: 2.4). Several studies evidenced that MNMT has a strong positive impact on LRLs [19, 29, 32], but still, MNMT struggles in the context of extremely LRL. Also, building such large MNMT models is quite challenging and costly for those working with limited computational resources. On the other hand, TL [44] is widely studied in the context of LRL. Several studies have explored many ways to conduct effective transfer learning on NMT [25, 53, 54, 55, 56].

One of the state-of-the-art transfer learning methods is fine-tuning an SMSP model with parallel data for a downstream NMT task, as discussed in Section: 2.8. Studies show this leads to significant performance gains over the Transformer model trained from scratch due to the knowledge transfer from pre-trained models [6, 7]. However, extremely LRL domain-specific NMT cases are not carefully studied. Also, the viability of fine-tuning

these SMSP models on non-English centric language pairs has not been studied, despite there being a need. So there is an urge to empirically analyze the robustness of SMSP models on extremely LRL NMT and non-English centric cases as this LRL NMT tends to perform poorly compared to High-resource pairs.

Another major concern is that languages under-represented in the SMSP models (i.e. languages for which the SMSP model has been pre-trained with a low amount of monolingual data) [2, 6, 7] obtained sub-optimal results while fine-tuning. Even though pre-trained models have offered promising results for LRLs, simple one-step fine-tuning with parallel data is not always a better solution. All the studies we are aware of [7, 69, 70, 71] focused on extending the SMSP models to adapt to new languages. To the best of our knowledge, there is no direct work on continual pre-training to adapt SMSP models to new domains in the context of NMT. Therefore, improving the pre-training and fine-tuning processes of the SMSP models for LRL-NMT is left to explore.

Chapter 3

METHODOLOGY

3.0.1 Overview

First, we explain the two direct fine-tuning strategies considered in this study. As shown in Figure 3.1, we are directly fine-tuning (one-step fine-tuning) the SMSP model using in-domain parallel data using Bilingual Fine-Tuning (B-FT) and Multilingual Fine-Tuning (M-FT). In contrast to the previous research that focused only on English-centric fine-tuning [6, 7], our study shows that non-English centric fine-tuning is also viable using self-supervised multilingual sequence-to-sequence pre-trained models. Different to some research that assumed only one language is in mBART while fine-tuning [66, 67], our studies focus on fine-tuning both languages in a pair of languages. Thus we are the first to conduct entire end-to-end fine-tuning with the latest mBART50 model [7].

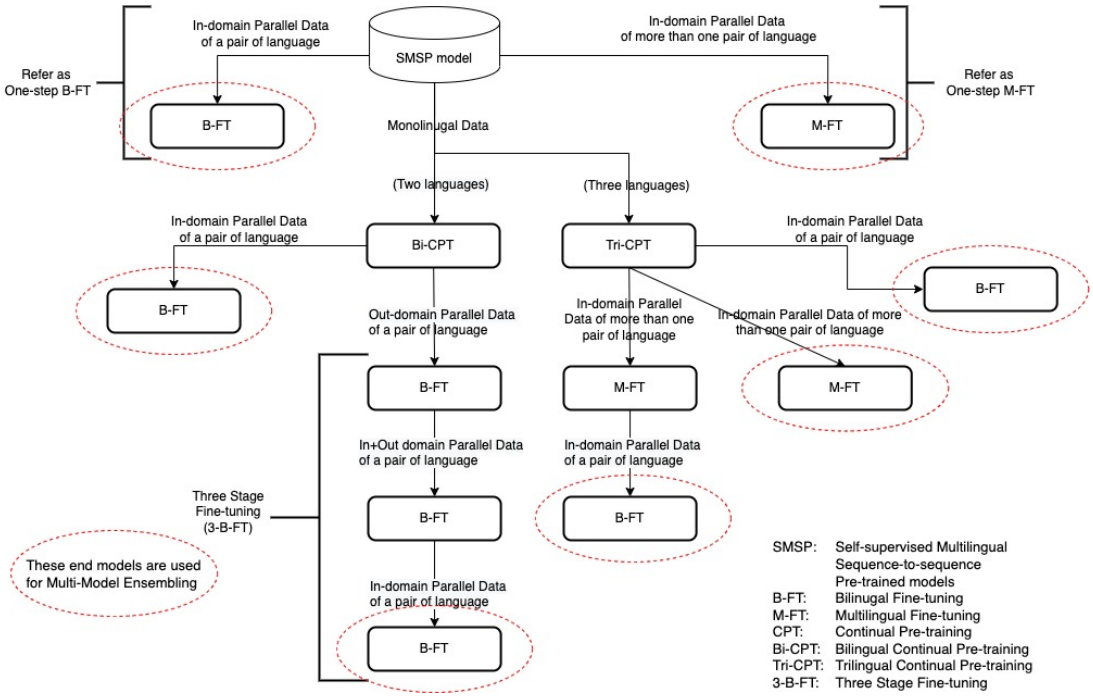

Figure 3.1: Overview of Methodology

However, it has been already shown that these techniques yield sub-optimal results for languages under-represented in the SMSP model [2, 6, 7]. Fine-tuning these SMSP models

using a small amount of in-domain parallel data has not been sufficient to learn domain-specific representation over the largely trained open-domain representation. Hence, in this study, we explore different strategies to improve the pre-training and fine-tuning of the SMSP models to achieve a better result on domain-specific LRL-NMT. We propose

- Continual Pre-Training (CPT) the SMSP model for domain adaptation: Here, we pre-train the SMSP model with additional monolingual data in order to alleviate the impact of the under-representation of LRLs in the SMSP model. We experiment with Bilingual Continual Pre-training (Bi-CPT) or Trilingual Continual Pre-training (Tri-CPT). We do not go beyond three languages, as our dataset has only three languages.

- Multistage fine-tuning of the continuously pre-trained SMSP model with parallel data for the NMT task. We fine-tune the SMSP model more than once by utilizing out-domain and in-domain parallel data. We mainly introduce two multistage fine-tuning cases: Three-stage Bilingual fine-tuning and Bilingual fine-tuning on a multilingually fine-tuned model. In both cases, the final fine-tuning occurs with an in-domain parallel corpus.

- Ensembling different fine-tuned models. Since we have explored different fine-tuning strategies on the selected base SMSP model, we had multiple similar fine-tuned SMSP models. We propose an ensemble of some of these models.

Fig. 3.1 summarises how our techniques are implemented/related.

3.0.2 Bilingual Fine-tuning using Self-Supervised Multilingual Sequence-to-sequence Pre-trained Models

Our main goal is to fine-tune the SMSP mBART model in an extremely low-resource domain-specific NMT setting, where the number of parallel sentences is less than 100k. We select Sinhala-Tamil-English languages to demonstrate the extremely low-resource scenario, where we train six bilingual models via pairwise combinations.

We call this one-step B-FT our strong B-FT for this study. The overview of this fine-tuning process is described in Fig. 2.5. We take the SMSP model to initialize the weights

of both the encoder and decoder of the downstream NMT task. Once the weights are initialized, we feed parallel data and further fine-tune the encoder and decoder of the downstream NMT task. Here training starts with the SMSP model weights instead of random initialization.

3.0.3 Multilingual Fine-Tuning using Self-Supervised Multilingual Sequence-to-sequence Pre-trained Models

As our second baseline, we use multilingual fine-tuning (M-FT) of the SMSP model as demonstrated in [7]. Up to now, almost all the studies have focused on English-centric MNMT systems. As discussed in Section 2.4, MNMT systems are mainly categorized into three groups where the English language is either on the target/source side or used as a pivot language to connect in the M2M setting. Even the viability of M-FT demonstrated by Tang et al. [7] only considered English-centric MNMT systems. In contrast, we fine-tune multilingual NMT systems in a non-English centric manner in three ways:

1. Many-to-One (M2O): $Many \rightarrow LanguageX$
2. One-to-Many (O2M): $LanguageX \rightarrow Many$
3. Many-to-Many (M2M): $Many \leftrightarrow Many$, with Language X as a pivot language.

3.1 Continual Pre-training for Domain Adaptation

As mentioned above, SMSP models are trained on large-scale open-domain monolingual data. Directly fine-tuning these SMSP models with small amounts of in-domain parallel data is not the best choice for using these SMSP models in domain-specific NMT. This is because small domain-specific parallel corpora are not enough to teach domain-specificity to a model that already contains a much larger representation in the open domain. Therefore, continual pre-training of these models with domain-specific monolingual data would alleviate the low representations that LRLs have in the SMSP models for the considered domain. Thus, we introduce continual denoising pre-training to incorporate a new domain on the SMSP model.

Some studies carried out continual pre-training of encoder-only pre-trained models (BERT [9] and RoBERTa [58]) with a large amount of in-domain data and later fine-tuned them for tasks such as text classification [72, 73, 74, 75, 76]. To the best of our knowledge, there is no direct work on continual pre-training to adapt SMSP models to new domains in the context of NMT. All the studies we are aware of [7, 69, 70, 71] focused on extending the SMSP models to adapt to new languages.

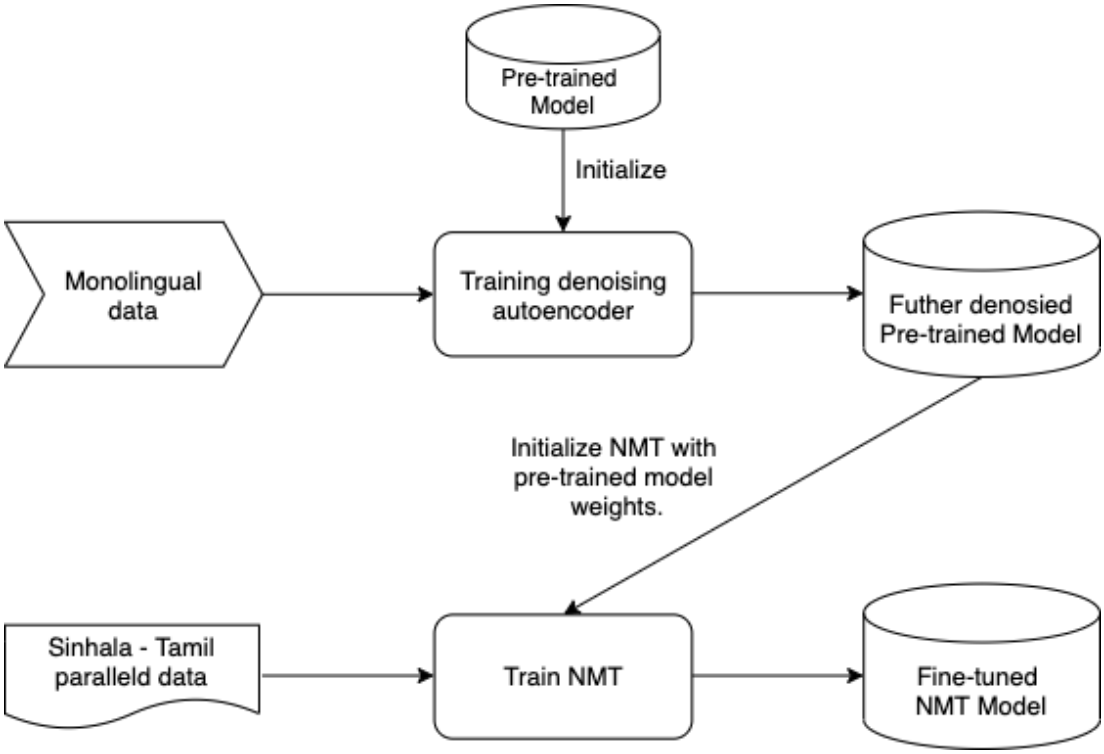

Figure 3.2: Overview of Continual Pre-training for Domain Adaptation.

In the SMSP model pre-training, monolingual data is noised and fed as the source, and original monolingual data as the target, which refers to self-supervised learning as mentioned in Section: 2.7. We follow the same denoising objectives as the selected SMSP model.

Our continual pre-training data covers K languages: $D = D_1, ..., D_K$, where each D_i is monolingual data in a language i. Equation: 3.1 is the noising function g that corrupts the text. We train the model to predict the original text X given $g(X)$. More formally, we aim to maximize L_θ:

$$L_\theta = \sum_{D_i \in D} \sum_{X \in D_i} logP(X|_g(X); \theta) \tag{3.1}$$

- where X is an instance in language i
- the distribution P is defined by the Seq2Seq model.

We conduct CPT using different types of monolingual data as below:

Case A: CPT with in-domain monolingual data.

Case B: CPT with out-domain monolingual data, which is larger in quantity.

Case C: CPT with mixed domain (in-domain + out-domain) monolingual data.

Case D: Multistage CPT on large out-domain data and then with in-domain data. Here we first pre-train with large out-domain monolingual data (case B) and then pre-train with in-domain monolingual data (case A).

First, we take monolingual data from all the languages we considered in the translation system. E.g. when we want to translate between the $X \rightarrow En$ language pair, we take monolingual data of language X and English (bilingual cases). If we want to consider more than one language pair, e.g., translating between $X \rightarrow En$ and $Y \rightarrow En$, we take monolingual data from X,Y and English (trilingual cases). Then we conduct CPT using the selected denoising objective of the SMSP model.

3.2 Multistage Fine-tuning

Different transfer learning protocols have been studied to adapt trained NMT models (both RNNs and simple Transformers) to new domains [25, 53, 54, 55, 56]. However, the effectiveness of these transfer learning protocols has not been studied in the context of SMSPs. Only single-step fine-tuning has been conducted so far. This single step of fine-tuning may not be the best way to adapt to a new domain. Thus, we introduce multistage fine-tuning on SMSP models in this research, as shown in Figure 3.1. Here we are introducing multistage fine-tuning where we try to fine-tune the mBART model consecutively two or more times as shown in Figure 3.3.

We categorize multistage fine-tuning as follows:

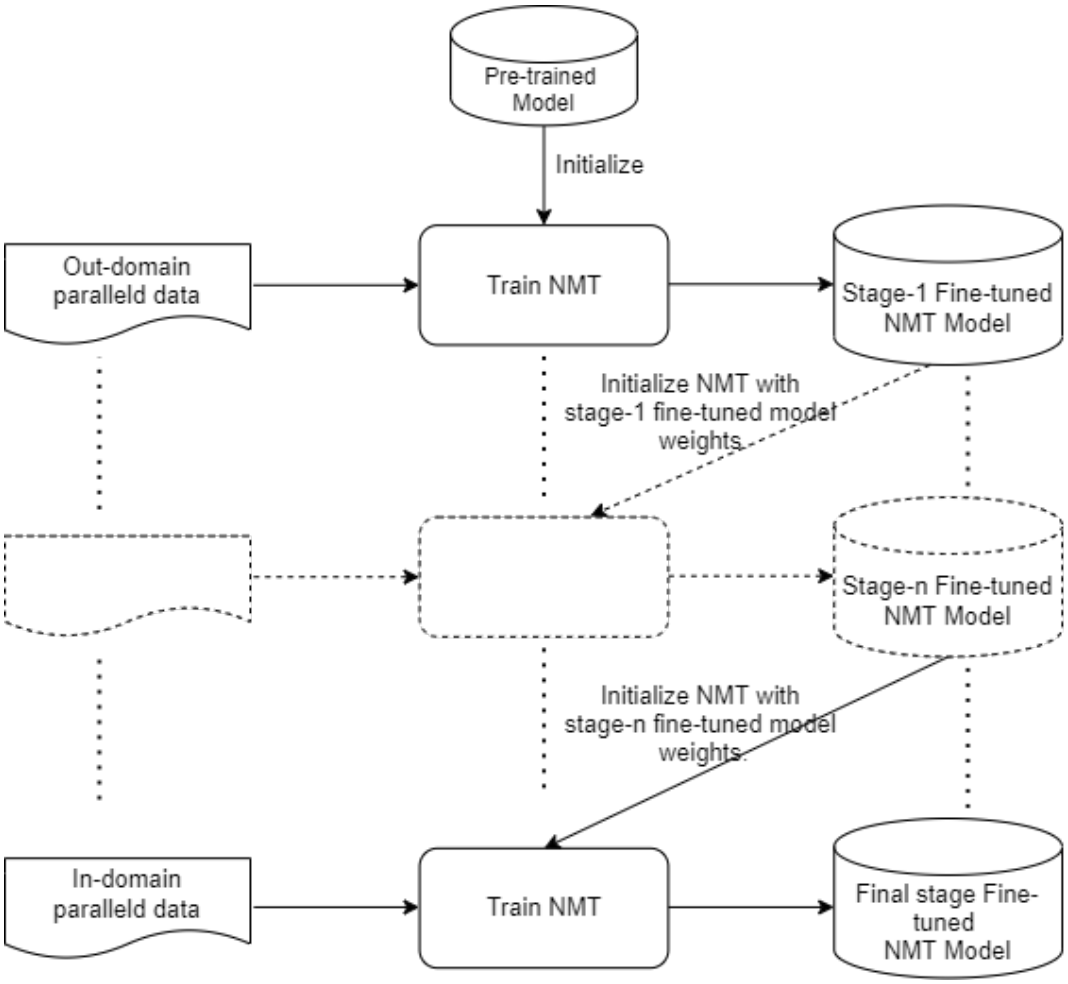

Figure 3.3: Overview of Multistage Fine-tuning.

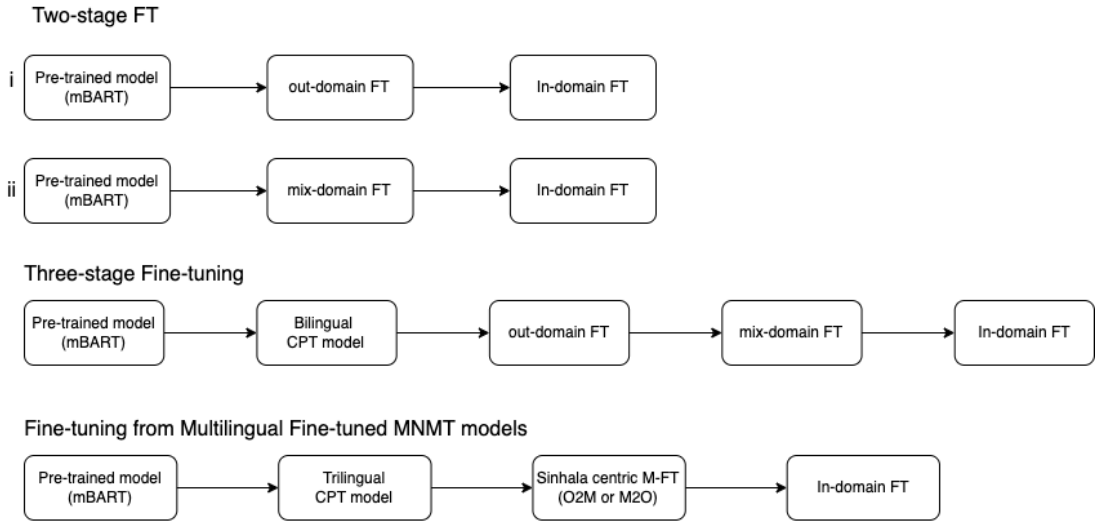

Figure 3.4: Different ways of Multistage Fine-tuning.

3.2.1 Two-stage FT

Here we conducted fine-tuning two consecutive times.

Case i: First fine-tune with Out-domain parallel data and then fine-tune with in-domain parallel data.

Case ii: Fine-tune with mixed domain parallel data, then fine-tune with in-domain parallel data. Here we up-sampled the in-domain data size to match the large out domain data and created the mixed-domain parallel data.

3.2.2 Multistage Fine-tuning Combine with Continual Pre-Training

We use the further denoised in-domain CPT SMSP model (as described in Section 3.1) as a base model to initialize the multistage fine-tuning process. We implement two different multistage fine-tuning strategies as follows:

1. Three-stage Fine-tuning [CPT+3-B-FT]: We borrowed the idea of mix-fine-tuning from Chu et al. [55], where they first train a Transformer-based NMT model on an out-of-domain parallel corpus and then fine-tune it on a parallel corpus that is a mix of the in-domain and out-of-domain corpora. In our study, we propose the following:

   (a) Initialize multistage training with the CPT model.

   (b) B-FT with out-domain parallel data.

   (c) B-FT with mixed domain parallel data where we up-sampled the in-domain data size to match with the larger out-domain corpus and created the mixed domain parallel data.

   (d) Finally, B-FT with in-domain parallel data.

2. Fine-tuning from Multilingual Fine-tuned MNMT models [CPT+M-FT+B-FT]:

   (a) Initialize training with the CPT model.

   (b) Conduct non-English-centric M-FT such as O2M M-FT or M2O M-FT as described in Section 3.0.3.

(c) Finally, B-FT with in-domain parallel data.

3.2.3 Ensemble of Fine-tuned Self-Supervised Multilingual Sequence-to-sequence Pre-trained Models

The ensemble is a Machine Learning technique that combines several base models to produce one optimized predictive model [77]. In NMT ensembling, the input sentence is translated using multiple models, and then the output from each model is averaged. As we propose multiple techniques to improve the fine-tuning process of SPSM models, we have multiple similar models that use identical target vocabularies and the same decoding (in other words, all have the same base SMSP models). Hence we carry out mainly two ensembling techniques:

1. Checkpoint Ensemble

   This is the bare minimum way to conduct the ensemble from a single training process [78]. Precious studies applied checkpoint ensembling on RNN based NMT system by combining the last N checkpoints of a single training [79, 80, 81]. In this study, we apply this checkpoints ensembling on SMSP models. We combine 3 different saved checkpoints from a particular single training. In other words, when we are fine-tuning the SMSP model, we save the last ten checkpoints. We select 3 checkpoints that yield the best results in the validation test set among all the saved checkpoints. Then we apply the checkpoints ensembling using these selected 3 checkpoints. Likewise, we experiment with combining 2 checkpoints too. Finally, we report the best ensemble result among both of the above scenarios. Note that we can combine any number of models we want.

2. Multi-model Ensemble

   Besides our baselines one-step B-FT and M-FT, we explored different techniques to improve the fine-tuning process of the SMSP model. Due to that, we have multiple similar SMSP models that have been fine-tuned with in-domain parallel data at the final stage. We combine a maximum of 3 such SMSP fine-tuned models (due to our computational resource limitations) for the ensemble. As shown in Fig. 3.1, we

explore different fine-tuning approaches to improve from baseline B-FT such as Bi-CPT continued with B-FT, Bi-CPT continued with Three-stage B-FT. Among these different approaches, we pick the top 3 best models.

Chapter 4

IMPLEMENTATION

4.1 Experimental setup

4.1.1 Architecture

For our experiments, we select the mBART50 which is referred to as the mBART [7] SMSP model. It supports all the considered languages while the mBART25 [6] model does not support Tamil. In particular, mBART has shown promising results for supervised and unsupervised NMT [6, 7]. mBART is memory efficient and has shown relatively better results than mT5 [64]. We follow the same mBART model architecture - the standard sequence-to-sequence Transformer [18], with 12 layers of encoder-decoder with the model dimension of 1024 on 16 heads. For training, we use the FairSeq[1] tool.

4.1.2 Dataset

For our experiments, we pick two languages that are underrepresented in the mBART model: Sinhala (Si) and Tamil (Ta), along with English (En). Our domain-specific parallel datasets are from Sri Lankan official government documents [15, 82], consisting of annual reports, crawled contents from government institutional websites, committee reports, procurement documents and Acts. According to the statistics of our dataset given in Table 4.1, our dataset is smaller than 100k. Thus this creates an extremely low-resource domain-specific translation task.

We also gather publicly available out-domain parallel datasets from OPUS[2] and WMT MT tasks[3]. For Sinhala-English, we use the FLoRes V1[4] [65] training dataset, obtained from OPUS[5]. For Tamil-English, we use WMT20[6] MT news tasks' parallel datasets. Un-

[1] https://github.com/pytorch/fairseq
[2] https://opus.nlpl.eu/
[3] https://www.statmt.org/wmt20/translation-task.html
[4] https://github.com/facebookresearch/flores/tree/main/floresv1
[5] https://opus.nlpl.eu/
[6] https://www.statmt.org/wmt20/translation-task.html

fortunately, we do not have out-domain parallel sentences for the Sinhala-Tamil pair. The statistics of these out-domain datasets are given in Table 4.2.

We use domain-specific monolingual data obtained from Epaliyana et al. [26] for monolingual data. We have adequately enough out-domain monolingual data. We pick out-domain monolingual data from the online news site NewsFirst[7] [83] and FLoRes[8] [65]. The statics are given in Table 4.3.

4.1.3 Preprocessing

The government corpus has been cleaned and verified manually with the help of professional translators, as mentioned by Fonseka et al. [15]. Sentences containing only dates, special characters and numbers have been removed. Cleaning scripts of the Moses[9] [4] tool were used to remove misaligned sentences. English sentences were tokenized using Moses toolkit[10] [4], while Sinhala and Tamil used an internal tokenizer [84]. We use the SentencePiece[11] model learned over monolingual Common Crawl (CC) data in the mBART [7] model, containing 250,000 sub-word tokens.

Dataset	**No. of Sentence**
Sinhala-Tamil	66,348
Tamil-English	66,348
Sinhala-English	74,468
Validation Set (for all pairs)	1,623
Test Set (for all pairs)	1,603

Table 4.1: Statistics of the parallel dataset of official government documents

Language pair	**Dataset**	**No. of Sentence**
Sinhala-Tamil	FLoRes	646,781
Tamil-English	WMT20 News	305,671

Table 4.2: Statistics of the out-domain parallel corpus

[7] https://www.newsfirst.lk/
[8] https://github.com/facebookresearch/flores/tree/main/floresv1
[9] http://www.statmt.org/moses/
[10] http://www.statmt.org/moses/
[11] https://github.com/google/sentencepiece

4.2 Addressing the Zero Width Joiner (ZWJ) issue

As mentioned earlier in Section 4.1.3, we use the SentencePiece[12] tokenizer. However, empty tokens where Unicode appears as space in output vocabulary have been removed while training the SentencePiece model. Due to that, the Zero Width Joiner character (200d) has been replaced by whitespace. Hence language scripts that require Zero Width Joiner get altered and result in a wrong output. This issue is there for languages like Sinhala, Kannada and Malayalam[13]. Ideally, we should not replace the Zero Width Joiner (200D) with whitespace since it indicates joining two chars without zero width (no whitespace). Also, the zero-width joiner must be present to decode the segmentation of decoder outputs to raw text successfully. These special characters should be kept as they are while learning the SentencePiece model. We resolved this issue and the fix was successfully merged with SentencePiece[14] to eliminate this ZWJ issue.

However, we cannot eliminate this issue when using the already trained (pre-trained) models as the models were trained without ZWJ over the large scale monolingual data. Hence we propose a post-processing solution to add the ZWJ character in the possible occurring places for Sinhala languages[15]. Here we mainly cover frequently occurring *"yansaya"* and *"rakāransaya"* of the Sinhala script. Further details of the post-processing logic are provided in Appendix. A.0.1.

Language	Dataset	Domain	No. of Sentences
Sinhala	Government data	in-domain	44,115
English	Government data	in-domain	42,773
Tamil	Government data	in-domain	24,220
Sinhala	News-first	out-domain	650,000
English	News-first	out-domain	627,301
Sinhala	FLoRes	out-domain	646,781
English	FLoRes	out-domain	646,781

Table 4.3: Monolingual Data

[12] https://github.com/google/sentencepiece
[13] https://en.wikipedia.org/wiki/Zero-width_joiner
[14] https://github.com/google/sentencepiece/pull/630
[15] https://en.wikipedia.org/wiki/Sinhala_script#Consonant_conjuncts

4.3 Baselines

We considered the following baselines.

1. Phrase-based Statistical Machine Translation (SMT):

   We used the SMT model proposed by Fernando et al. [82], which has been trained using the Moses toolkit[16] [4] with the default features and parameters of a 5-gram language model.

2. LSTM based NMT:

   We used a 2-layer bidirectional LSTM as the encoder and a 2-layer LSTM as the decoder described by Sennrich and Zhang [85] for low-resource NMT. We trained this LSTM based NMT 4 consecutive times with early stopping criteria (with the patience of 5 valid steps) and saved all the checkpoints. The model gave the highest BLEU score upon validation dataset identified as the best model among these saved checkpoints. During the inference phase, we used a beam search of 5.

3. LSTM based NMT Ensemble:

   From all the saved checkpoints in the aforementioned LSTM based NMT, we selected the top 4 models based on the results of the validation set for the ensemble.

4. **Transformer Baseline:**

   We consider this as our main state-of-the-art baseline. Here, we adapted the Transformer model with BPE proposed by [15]. We tuned the hyperparameters with 5 layers of encoder-decoder with 2 attention heads. We continued our training up to 300 epochs and saved checkpoints. The model with the highest BLEU score on the validation dataset was identified as the best model among these checkpoints.

4.4 Fine-tuning SMSP Model

For all directions, we train with 0.3 dropout, 0.2 label smoothing, 2500 warm-up steps, and 3e-5 maximum learning rate as described by Liu et al. [6]. We use a maximum of up

[16] `http://www.statmt.org/moses/`

to 100k training updates for B-FT cases.

We extend the fine-tuning by adding multiple language pairs together, resulting in M-FT. We applied the same setting as Section 4.4 and used a maximum of 300k training updates. Additionally, we applied a temperature sampling rate of 1.5. Since our main goal is to conduct M-FT in a non-English centric manner, we chose Sinhala centric M-FT. It is the most requested use case in Sri Lanka since the government documents are mostly produced in Sinhala and should be translated to Tamil and English. This, of course, has been identified as a common requirement in general for MT [86].

4.5 Continual Pre-training for Domain Adaptation

Data selection for the different continual pre-training strategies proposed in Section:3.1 is described below:

Case A: Multilingual Pre-training on in-domain monolingual data.

We use available in-domain government monolingual data (see Table 4.3). However, the monolingual data is not large.

In the second attempt, we utilize parallel data (as shown in Table 4.1) as monolingual data for each selected language pair and the available small monolingual government data to create a combined dataset. First, we use only the selected two languages for pre-training (Bi-CPT). Then we move to CPT with three languages for M-FT, which we refer to as Tri-CPT.

Case B: Multilingual Pre-training on large out-domain monolingual data.

We use the large available out-domain monolingual data mainly obtained from news data crawled from Sri Lankan news websites [83] (see Table 4.3).

Case C: Multilingual Pre-training on mixed domain monolingual data.

For this case, we took the Flores [65] out-domain monolingual data and mixed it with our in-domain government monolingual dataset mentioned in Case A.

Case D: multistage Multilingual Pre-training on large out-domain data and then with in-domain data.

Here we first pre-train with large out-domain monolingual data (case B) and then pre-train with in-domain government monolingual data (case A).

Noise function

We use the nosing techniques used by the mBART model [6]: Random Span masking and Order Permutation [6]. First, we remove spans of text and replace them with a mask token. We mask 0.3% of words in each instance (with random masking 0.1) by randomly sampling a span length according to a Poisson distribution ($\lambda = 3.5$). We also permute the order of sentences within each instance. The decoder input is the original text with one position offset. A language id symbol <LID> is used as the initial token to predict the sentence.

4.6 Multistage Fine-tuning

We use the in-domain CPT model as a base model to initialize the multistage fine-tuning. We have two different multistage fine-tuning techniques: Three-stage Fine-tuning and Multilingual Fine-tuning followed by Bilingual Fine-tuning as described in Section 3.2. For $En-Ta$ and $En-Si$ pairs, we tested with Three-stage Fine-tuning techniques. Since we do not have an out-domain dataset for $Si-Ta$, we do not conduct the Three-stage fine-tuning experiment for that pair.

4.7 Evaluation setup

The final model is selected based on the validation likelihood. For decoding, we use beam search with beam size 5 as used by [6]. The final results are calculated against the true-target tokenized data and reported in BLEU [87].

For ensembling cases described in Section: 3.2.3, we combine a max of 3 models and evaluate under the same decoding and BLEU scores calculation as mentioned above.

Chapter 5

RESULTS AND DISCUSSION

5.1 Bilingual Fine-tuning using Multilingual Denoising Pre-trained Models

Comparison Between Full Precision Fine-Tuning and Mixed-Precision Fine-Tuning

When training a neural network, usually we use Full Precision, a 32-bit floating-point (FP32) arithmetic calculation by default. Else, we can use Mixed-Precision training, which combines single-precision (FP32) with half-precision (FP16) format. Mixed-Precision training has additional performance benefits on NVIDIA GPUs. It requires shorter training time, lower memory requirements, enabling larger batch sizes, larger models, or larger inputs[1].

Hence we fine-tune each pair of directions two times under the same hyper-parameters, one with the default setting Full Precision and the second one with Mixed-precision training. Latter one typically tends to reduce the training time. As shown in Table 5.1, there is not much BLEU score difference between Full Precision and Mixed-precision. However, Mixed-precision training has reduced the training time. Considering our resource limitation for training, we choose Mixed-precision training going forward.

Models	**Si→Ta**	**Ta→Si**	**Si→En**	**En→Si**	**Ta→En**	**En→Ta**
Bilingual full precision FT	29.82	37.87	37.84	35.89	33.63	25.96
Bilingual mixed precision FT	29.75	37.57	37.72	36.11	33.36	25.8

Table 5.1: Comparison between full precision training and mixed precision Fine-Tuning. Results are reported in BLEU score.

Baseline Results and Discussion

The performance results of our bilingual fine-tuned models against the baselines described in Section 4.3 are given in Table 5.2 and a few examples of translated sentences are provided in Appendix. A.0.2. We observe quantifiable performance gains for the B-FT in all 6 cases. We significantly improved the BLEU score while translating Sinhala/Tamil sentences to English, gaining a +7.17 BLEU score on Si→En translation and +6.74 BLEU

[1] `https://pytorch.org/blog/accelerating-training-on-nvidia-gpus-with-pytorch-automatic-mix`

for the Ta→En direction over the state-of-the-art Transformer Baseline. Liu et al. [6] observed that when English is on the target, the NMT model obtained greater improvement than when English is on the source. In these extremely low-resource settings, we also observe the same behaviour as we found in the study [6] for low and medium resources. We strongly believe it is due to the English language is benefiting from its large-scale monolingual data used during pre-training, compared to other languages. To train the mBART model, 55,608M English tokens have been used, while only 243M, 595M tokens have been used for Sinhala and Tamil, respectively [6, 7]. However, we argue that, in order to examine the full potential of these pre-trained models, their performance in the non-English centric translation should also be considered since many languages are under-represented in these pre-trained models.

Models	**Si→Ta**	**Ta→Si**	**Si→En**	**En→Si**	**Ta→En**	**En→Ta**
SMT	22.27	26.91	22.62	20.27	17.96	14.16
LSTM	17.39	22.3	21.66	20.05	15.62	13.35
LSTM Ensemble	19.95	24.5	25.39	22.63	19.93	15.98
Transformer Baseline	26.97	33.46	30.55	29.47	26.62	21.75
Bilingual FT (B-FT)	**29.75**	**37.57**	**37.72**	**36.11**	**33.36**	**25.8**
Improvement	(+2.78)	(+4.11)	(+7.17)	(+6.64)	(+6.74)	(+4.05)

Table 5.2: Comparison with SMT, LSTM, Transformer Architectures against our Bilingual Fine-tuning models for Sinhala (Si), Tamil (Ta) and English (En) - Results are reported in BLEU score.

Most importantly, for the first time, we experimented with non-English centric B-FT where both languages in a pair of languages are included in the mBART model for Sinhala-Tamil. We managed to obtain a +2.78 improvement on Si→Ta and +4.11 on Ta→Si directions over Transformer Baseline. We get the lowest improvement when Tamil is on the target side, such as Si→Ta +2.78 and En→Ta +4.05. Even though the pre-trained model used 595M Tamil tokens, which is higher than 243M tokens for Sinhala, Sinhala tends to perform well when being the target compared to Tamil.

We can see that LSTM models are less effective in extremely low-resource settings. By ensembling the top 4 models, the LSTM model surpassed SMT by a slight margin except for Si→Ta and Ta→Si. We can evidence that LSTM models require a large corpus to obtain a reasonable amount of accuracy. The transformer model is much stronger compared to

SMT and LSTM models. However, our proposed B-FT using a multilingual denoising pre-trained model outperforms the Transformer model in all the cases. This demonstrates the usefulness of pre-trained models for low-resource NMT, where it is difficult to find large scale parallel data to train NMT from scratch.

5.2 Multilingual Fine-tuning using Multilingual Denoising Pre-trained Models

Our Sinhala centric multilingual FT results against the baseline B-FT results are provided in Table 5.3. As we can see, we could not obtain a reasonable improvement over the multilingual FT. We strongly believe it is because of inadequate parallel data. We used extremely LRL pairs only; unrelated languages like Tamil and English are on the target/-source side. Nevertheless, we observe a slight improvement in O2M M-FT and M2O M-FT compared to the M2M M-FT case like on English centric MNMT [7, 19, 29, 32]. We also evidence that up-sampling plays a major role in these MNMT systems as stated by Arivazhagan et al. [19], Aharoni et al. [29], Johnson et al. [32]. In our experiments, Si-Ta acts as a low-resource pair compared to the Si-En data set as we have around 66k, and 74k sentences, respectively. When we up-sample Si-Ta to match the size of Si-En, we were able to maximize the performance and beat the bilingual baseline such as Si→Ta (+0.55), while observing performance reduction on the Si-En side such as Si→En (-0.13).

Models	**Si→Ta**	**Ta→Si**	**Si→En**	**En→Si**
B-FT	29.75	37.57	37.72	36.11
O2M M-FT (Si→Ta,Si→En)	30.3 (+0.55)	N/A	37.59 (-0.13)	N/A
M2O M-FT (Ta→Si,En→Si)	N/A	37.62 (+0.05)	N/A	36.03 (-0.08)
M2M M-FT (Si↔Ta,En↔Si)	28.44 (-1.31)	36.38 (-1.19)	36.05 (-1.67)	34.93 (-1.18)

Table 5.3: Results of Bilingual Fine-tuning models and Multilingual Sinhala Centric Fine-tuning models - Results are reported in BLEU score.

5.3 Continual Pre-training for Domain Adaptation

Extensive evaluation of Si↔En pair results is reported in Table 5.4. Our experiments show the effectiveness of continual pre-training to adapt to new domains. Even though we don't have adequate in-domain data, our experiments validate that having little in-domain data would have more impact than out-domain or mixed domain data. We pick Bilingual

Continual Pre-training on in-domain monolingual data for going forward. When we look at Table 5.5, we can evidence that continual pre-training helps extremely LRLs to improve further.

	Models	**Si→En**	**En→Si**
	B-FT	37.72	36.11
Case A	Bi-CPT in-domain mono data + B-FT	38.21 (+0.49)	36.38 (+0.27)
	Bi-CPT combined parallel and in-domain data + B-FT	**38.51** (+0.79)	**36.64** (+0.53)
Case B	Bi-CPT out-domain mono data + B-FT	38.09 (+0.37)	36.21 (+0.1)
Case C	Mixed in-domain and out-domain Bi-CPT + B-FT	38.3 (+0.58)	36.33(+0.22)
Case D	Multistage Bi-CPT + B-FT	38.38 (+0.66)	36 (-0.11)

Table 5.4: Fine-tuning Results from Continual Pre-trained models against our strong baseline Bilingual Fine-tuned models for Si↔En pairs - Results are reported in BLEU score.

We even conducted the Trilingual Continual Pre-training on in-domain monolingual data. We utilize the parallel data (as in Table 4.1) available for Si-En (74k sentences) as monolingual data for Sinhala and English and along with the Tamil side sentences from the parallel Si-Ta (66k sentences) data set. After denoising, we conducted B-FT for each pair of directions. Results are given in Table 5.5, which is lower than Bilingual Continual Pre-training on in-domain monolingual data. We believe it is because of the language relatedness. However, for En→Ta, trilingual denoising gives better results; we believe it is because Tamil benefited from the upsampling in the Trilingual denoising.

Models	**Si→Ta**	**Ta→Si**	**Si→En**	**En→Si**	**Ta→En**	**En→Ta**
B-FT	29.75	37.57	37.72	36.11	33.36	25.8
Bi-CPT+B-FT	**30.72** (+0.97)	**38.08** (+0.51)	**38.51** (+0.79)	**36.64** (+0.53)	**34.29** (+0.93)	26.3 (+0.5)
Tri-CPT+B-FT	30.34 (+0.59)	37.93 (+0.36)	37.52 (-0.2)	36.56 (0.45)	33.61 (+0.25)	**26.62** (+0.82)

Table 5.5: Fine-tuning Results of Bilingual and Trilinugal Continual Pre-training on in-domain monolingual data for all the six directions - Results are reported in BLEU score.

5.4 Multistage Fine-tuning

5.4.1 Two-stage FT

We analyzed the Si-En pairs, and the results are given in Table 5.6. Even though we did not get a reasonable improvement on two-stage FT, we observed the positive impact on these different two-stage multi-stage FT. Compared to out-domain fine-tuning, we found

that mixed fine-tuning plays a major role. As we need to analyze carefully, we are moving to three-stage fine-tuning where we combine the case mentioned above (case (i) and case (ii)). Also, we combine Continual Pre-Training and Multistage Fine-tuning, which we discuss in the next section.

	Models	**Si→En**	**En→Si**
	B-FT	37.72	36.11
Case i	O/D, I/D FT	38.03 (+0.31)	36.65 (+0.54)
Case ii	Mix/D, I/D FT	38.41 (+0.69)	36.76 (+0.65)

Table 5.6: Fine-tuning Results from Continual Pre-trained models against the our strong baseline Bilingual Fine-tuned models for Si↔En pairs - Results are reported in BLEU score.

5.4.2 Multistage fine-tuning Combined with Continual Pre-Training

As shown in Table 5.7, combining our proposed Continual Pre-Training and Multistage Fine-tuning approaches improves the performance over baseline B-FT. Our results prove that utilizing the available in-domain monolingual and out-domain parallel data enhances fine-tuning performance in the extremely low-resource domain-specific settings. We obtained quantifiable improvements over the Three-stage fine-tuning, where we obtained +1.46 improvement in the Ta→En direction and +1.17 in the Si→En direction. However, when Sinhala/Tamil are on the target side, our improvements are low, up to +0.8 improvement. As observed in our initial bilingual FT experiments results (Section: 5.1), here also we can witness that, even though we fine-tune with large out-domain data, we still require a reasonable amount of in-domain data to learn good language representation on a particular domain for morphologically rich languages than English.

Multilingual FT helps Si↔Ta directions positively over Si↔En directions. O2M M-FT and M2O M-FT show a positive side and slightly improved with Tri-CPT - O2M M-FT and M2O M-FT. As we can see, we obtained the highest of +0.75 BLEU improvement on Si→Ta and +0.47 BLEU improvement on the Ta→Si side on multilingual FT. In contrast, in the Si↔En direction, we lagged by -0.24 BLEU score on En→Si and less improvement on Si→En by +0.24 BLEU score. Hence we choose to fine-tune further from Multilingual Fine-tuned MNMT models for Si↔Ta directions. We observe that fine-tuning further from

Models	Si→Ta	Ta→Si	Si→En	En→Si	Ta→En	En→Ta
B-FT (Baseline)	29.75	37.57	37.72	36.11	33.36	25.8
Bi-CPT+B-FT	30.72 (+0.97)	38.08 (+0.51)	38.51 (+0.79)	36.64 (+0.53)	34.29 (+0.93)	26.3 (+0.5)
Bi-CPT+3-B-FT	-	-	**38.89** (+1.17)	**36.91** (+0.8)	**34.82** (+1.46)	26.45 (+0.65)
O2M-M-FT	30.3 (+0.55)	-	37.59 (-0.13)	-	-	-
M2O-M-FT	-	37.62 (+0.05)	-	36.03 (-0.08)	-	-
M2M-M-FT	28.44 (-1.31)	36.38 (-1.19)	36.05 (-1.67)	34.93 (-1.18)	-	-
Tri-CPT+B-FT	30.34 (+0.59)	37.93 (+0.36)	37.52 (-0.2)	36.56 (0.45)	33.61 (+0.25)	**26.62** (+0.82)
Tri-CPT+O2M-M-FT	30.5 (+0.75)	-	37.93 (+0.21)	-	-	-
Tri-CPT+M2O-M-FT	-	38.04 (+0.47)	-	35.87 (-0.24)	-	-
Tri-CPT+M-FT+B-FT	**31.16** (+1.41)	**38.9** (+1.33)	-	-	-	-

Table 5.7: Multistage fine-tuning against the our strong baseline Bilingual Fine-tuned models - Results are reported in BLEU score.

Multilingual Fine-tuned MNMT models have performed better and improved by +1.41 BLEU score on Si→Ta and +1.33 on Ta→Si.

5.5 Ensemble of Fine-tuned Self-supervised Multilingual Sequence-to-sequence Pre-trained Models

All the possible combinations of ensembling results are given in Table 5.9. As we can see, the multi-model ensemble overall improved the performance in all the cases whereas checkpoints ensemble methods performed comparatively less. However, the result distribution of the multi-model ensemble case is almost similar for each pair of directions. Among them, we picked the highest score combination for each pair of directions. We observe that even the baseline B-FT stands as a strong model. Out of the six translation directions, baseline B-FT was not identified as one of the best models only for the Ta-En direction. Finally, we obtained a maximum +2.56 BLEU point improvement on Ta→En and a minimum of +1.84 BLEU score improvement in En→Si directions.

Rank	Si→Ta	Ta→Si	Si→En	En→Si	Ta→En	En→Ta
1	Tri-CPT+M-FT+B-FT	Tri-CPT+M-FT+B-FT	Bi-CPT+3-B-FT	Bi-CPT+3-B-FT	Bi-CPT+3-B-FT	Tri-CPT+B-FT
2	Bi-CPT+B-FT	Bi-CPT+B-FT	Bi-CPT+B-FT	Bi-CPT+B-FT	Bi-CPT+B-FT	Bi-CPT+3-B-FT
3	Tri-CPT+O2M-M-FT	Tri-CPT+M2O-M-FT	Tri-CPT+O2M-M-FT	Tri-CPT+B-FT	Tri-CPT+B-FT	Bi-CPT+B-FT

Table 5.8: Top 4 improved models from baseline B-FT

Ensemble	Models	Si→Ta	Ta→Si	Si→En	En→Si	Ta→En	En→Ta
No Ensemble	B-FT (Baseline)	29.75	37.57	37.72	36.11	33.36	25.8
	Rank-1	31.16 (+1.41)	38.9 (+1.33)	38.89 (+1.17)	36.91 (+0.8)	34.82 (+1.46)	26.62 (+0.82)
Checkpoint Ensemble	B-FT (Baseline)	30.07	37.86	37.77	36.21	33.4	25.82
	Rank-1	31.26	38.8	39.58	36.85	35.4	26.55
Multi-Model Ensemble	B-FT & Rank-1	31.43	39.45	39.39	37.6	34.98	26.69
	B-FT & Rank-2	31.1	38.57	38.76	37.1	34.58	27.31
	B-FT & Rank-3	31.3	38.84	38.4	37.52	34.53	26.81
	B-FT & Rank-1,2	**31.85 (+2.1)**	39.29	**39.92 (+2.2)**	37.71	35.62	**27.77 (+1.97)**
	B-FT & Rank-1,3	31.64	**39.73 (+2.16)**	39.61	**37.95 (+1.84)**	35.53	27.16
	B-FT & Rank-2,3	31.52	39.08	39.27	37.58	34.93	27.33
	Rank-1,2	31.83	39.49	39.61	37.73	35.83	27.54
	Rank-1,3	31.41	38.58	39.79	37.61	**35.92 (+2.56)**	26.86
	Rank-1,2,3	31.73	39.47	39.72	37.76	35.86	27.41

Table 5.9: Ensemble Results for all the six directions. Results are reported in BLEU score.

5.6 Manual Analysis of the Translated Output

We manually analyze the translated sentences for each direction. A few examples of translated sentences are provided in Appendix. A.0.2. One common observation is that our models fail to handle the name of the places and roads. Those named entities are completely domain-specific, and with limited data, we could not cover those names. We can apply data augmentation techniques to include those specific named entities in future work.

Another major observation is that our models perform less when Tamil is on the target side. It is mainly because the Tamil language has free word ordering [88] and is more inflectional than Sinhala [89]. We observed free word order, joint words and similar syntactic constructions while analyzing the Tamil outputs. Output sentences used similar or synonyms compared to the reference (Ref) sentences. When the complexity of a language increases, the amount of training data required to learn language-specific information also increases. We can conclude that when both languages have similar amounts of monolingual data for pre-training, fine-tuning results depend on the complexity of the target language.

Chapter 6

CONCLUSION AND FUTURE WORK

For LRLs, transferring the knowledge from the already trained (pre-trained models) SMSP by fine-tuning remains one promising approach. One such SMSP model is mBART. According to previous studies [6, 7], fine-tuning the mBART models has shown promising results for English centric fine-tuning; we took this line of research even further and showed the viability of fine-tuning SMSP models non-English centric extremely low-resource domain-specific settings. Even though some languages are under-represented in the pre-trained model, we showed that the pre-trained model is robust enough to obtain significant improvements for non-English centric MT.

Apart from that, we extended the SMSP model further by pre-training it with different combinations of monolingual data. We introduced multistage fine-tuning to adapt an SMSP model to new domains in an extremely low-resource setting involving non-English-centric language pairs. We explored different fine-tuning strategies by utilizing out- and in-domain parallel data. These techniques showed quantifiable improvements in the context of Sinhala and Tamil, which are under-represented in the selected SMSP Model (mBART).

From our experiments, we can conclude that the translation accuracy heavily depends on the amount of monolingual data and the domain of the data used to pre-train the SPSM model. We also need to note that when monolingual data is roughly equal, the fine-tuned result depends on the target language complexity. Apart from that, our proposed approaches proved that even though the pre-trained model has been trained over large-scale monolingual data, extending these models by adding little in-domain monolingual data helps SMSP models improve even further. Also, using out-domain data for multistage fine-tuning tends to improve the English side rather than Sinhala/Tamil on target.

In future work, we plan to experiment more with extremely LRL non-English-centric MNMT cases. We found that M-FT cases struggle to provide an improvement over B-FT cases. In the next phase, we will be focusing on improving the pre-training and fine-tuning process of M-FT cases, which would result in a single MNMT model that can handle all

the translation directions at once.

[1] S. Ranathunga, E.-S. A. Lee, M. P. Skenduli, R. Shekhar, M. Alam, and R. Kaur, "Neural machine translation for low-resource languages: A survey," 2021.

[2] S. Thillainathan, S. Ranathunga, and S. Jayasena, "Fine-tuning self-supervised multilingual sequence-to-sequence models for extremely low-resource nmt," in *2021 Moratuwa Engineering Research Conference (MERCon)*. IEEE, 2021, pp. 432–437.

[3] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[4] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, 2007, pp. 177–180.

[5] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *arXiv preprint arXiv:1409.3215*, 2014.

[6] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, "Multilingual denoising pre-training for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, 2020.

[7] Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu, and A. Fan, "Multilingual translation with extensible multilingual pretraining and finetuning," *arXiv preprint arXiv:2008.00401*, 2020.

[8] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mt5: A massively multilingual pre-trained text-to-text transformer," *arXiv preprint arXiv:2010.11934*, 2020.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] R. Dabre, C. Chu, and A. Kunchukuttan, "A survey of multilingual neural machine translation," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–38, 2020.

[11] P. Tennage, P. Sandaruwan, M. Thilakarathne, A. Herath, S. Ranathunga, S. Jayasena, and G. Dias, "Neural machine translation for sinhala and tamil languages," in *2017 International Conference on Asian Language Processing (IALP)*. IEEE, 2017, pp. 189–192.

[12] P. Tennage, P. Sandaruwan, M. Thilakarathne, A. Herath, and S. Ranathunga, "Handling rare word problem using synthetic training data for sinhala and tamil neural machine translation," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[13] P. Tennage, A. Herath, M. Thilakarathne, P. Sandaruwan, and S. Ranathunga, "Transliteration and byte pair encoding to improve tamil to sinhala neural machine translation," in *2018 Moratuwa Engineering Research Conference (MERCon)*. IEEE, 2018, pp. 390–395.

[14] A. Pramodya, R. Pushpananda, and R. Weerasinghe, "A comparison of transformer, recurrent neural networks and smt in tamil to sinhala mt," in *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*. IEEE, 2020, pp. 155–160.

[15] T. Fonseka, R. Naranpanawa, R. Perera, and U. Thayasivam, "English to sinhala neural machine translation," in *2020 International Conference on Asian Language Processing (IALP)*. IEEE, 2020, pp. 305–309.

[16] R. Naranpanawa, R. Perera, T. Fonseka, and U. Thayasivam, "Analyzing subword techniques to improve english to sinhala neural machine translation," *International Journal of Asian Language Processing*, vol. 30, no. 04, p. 2050017, 2020.

[17] B. Janarthanasarma and T. Uthayasanker, "A survey on neural machine translation for english-tamil language pair."

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[19] N. Arivazhagan, A. Bapna, O. Firat, D. Lepikhin, M. Johnson, M. Krikun, M. X. Chen, Y. Cao, G. Foster, C. Cherry *et al.*, "Massively multilingual neural machine translation in the wild: Findings and challenges," *arXiv preprint arXiv:1907.05019*, 2019.

[20] A. Arukgoda, A. Weerasinghe, and R. Pushpananda, "Improving sinhala-tamil translation through deep learning techniques." in *NL4AI@ AI* IA*, 2019.

[21] L. Nissanka, B. Pushpananda, and A. Weerasinghe, "Exploring neural machine translation for sinhala-tamil languages pair," in *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*. IEEE, 2020, pp. 202–207.

[22] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1700–1709.

[23] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[26] K. Epaliyana, S. Ranathunga, and S. Jayasena, "Improving back-translation with iterative filtering and data selection for sinhala-english nmt," in *2021 Moratuwa Engineering Research Conference (MERCon)*. IEEE, 2021, pp. 438–443.

[27] H. Choudhary, A. K. Pathak, R. R. Saha, and P. Kumaraguru, "Neural machine translation for english-tamil," in *Proceedings of the third conference on machine translation: shared task papers*, 2018, pp. 770–775.

[28] T. Banerjee, A. Kunchukuttan, and P. Bhattacharyya, "Multilingual indian language translation system at wat 2018: Many-to-one phrase-based smt," in *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation: 5th Workshop on Asian Translation: 5th Workshop on Asian Translation*, 2018.

[29] R. Aharoni, M. Johnson, and O. Firat, "Massively multilingual neural machine translation," *arXiv preprint arXiv:1903.00089*, 2019.

[30] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, "Multi-task learning for multiple language translation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 1723–1732.

[31] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," *arXiv preprint arXiv:1511.06114*, 2015.

[32] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017.

[33] B. Zoph and K. Knight, "Multi-source neural translation," *arXiv preprint arXiv:1601.00710*, 2016.

[34] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," *arXiv preprint arXiv:1601.01073*, 2016.

[35] T.-L. Ha, J. Niehues, and A. Waibel, "Toward multilingual neural machine translation with universal encoder and decoder," *arXiv preprint arXiv:1611.04798*, 2016.

[36] O. Firat, B. Sankaran, Y. Al-Onaizan, F. T. Y. Vural, and K. Cho, "Zero-resource translation with multi-lingual neural machine translation," *arXiv preprint arXiv:1606.04164*, 2016.

[37] Y. Lu, P. Keung, F. Ladhak, V. Bhardwaj, S. Zhang, and J. Sun, "A neural interlingua for multilingual machine translation," *arXiv preprint arXiv:1804.08198*, 2018.

[38] S. M. Lakew, M. Federico, M. Negri, and M. Turchi, "Multilingual neural machine translation for zero-resource languages," *arXiv preprint arXiv:1909.07342*, 2019.

[39] G. Blackwood, M. Ballesteros, and T. Ward, "Multilingual neural machine translation with task-specific attention," *arXiv preprint arXiv:1806.03280*, 2018.

[40] Y. Wang, J. Zhang, F. Zhai, J. Xu, and C. Zong, "Three strategies to improve one-to-many multilingual translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2955–2960.

[41] V. Goyal, S. Kumar, and D. M. Sharma, "Efficient neural machine translation for low-resource languages via exploiting related languages," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 2020, pp. 162–168.

[42] S. M. Lakew, A. Erofeeva, M. Negri, M. Federico, and M. Turchi, "Transfer learning in multilingual neural machine translation with dynamic vocabulary," *arXiv preprint arXiv:1811.01137*, 2018.

[43] S. M. Lakew, M. Cettolo, and M. Federico, "A comparison of transformer and recurrent neural networks on multilingual neural machine translation," *arXiv preprint arXiv:1806.06957*, 2018.

[44] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," *arXiv preprint arXiv:1604.02201*, 2016.

[45] R. Dabre, T. Nakagawa, and H. Kazawa, "An empirical study of language relatedness for transfer learning in neural machine translation," in *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*, 2017, pp. 282–286.

[46] T. Q. Nguyen and D. Chiang, "Transfer learning across low-resource, related languages for neural machine translation," *arXiv preprint arXiv:1708.09803*, 2017.

[47] G. Neubig and J. Hu, "Rapid adaptation of neural machine translation to new languages," *arXiv preprint arXiv:1808.04189*, 2018.

[48] A. F. Aji, N. Bogoychev, K. Heafield, and R. Sennrich, "In neural machine translation, what does transfer learning transfer?" in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7701–7710.

[49] B. Ji, Z. Zhang, X. Duan, M. Zhang, B. Chen, and W. Luo, "Cross-lingual pre-training based transfer for zero-shot neural machine translation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 115–122.

[50] T. Kocmi and O. Bojar, "Efficiently reusing old models across languages via transfer learning," *arXiv preprint arXiv:1909.10955*, 2019.

[51] M. Maimaiti, Y. Liu, H. Luan, and M. Sun, "Multi-round transfer learning for low-resource nmt using multiple high-resource languages," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 18, no. 4, pp. 1–26, 2019.

[52] Y. Kim, Y. Gao, and H. Ney, "Effective cross-lingual transfer of neural machine translation models without shared vocabularies," *arXiv preprint arXiv:1905.05475*, 2019.

[53] M. Maimaiti, Y. Liu, H. Luan, and M. Sun, "Enriching the transfer learning with pre-trained lexicon embedding for low-resource neural machine translation," *Tsinghua Science and Technology*, p. 1, 2020.

[54] A. Imankulova, R. Dabre, A. Fujita, and K. Imamura, "Exploiting out-of-domain parallel data through multilingual transfer learning for low-resource neural machine translation," *arXiv preprint arXiv:1907.03060*, 2019.

[55] C. Chu, R. Dabre, and S. Kurohashi, "An empirical comparison of domain adaptation methods for neural machine translation," in *Proceedings of the 55th Annual Meeting*

of the Association for Computational Linguistics (Volume 2: Short Papers), 2017, pp. 385–391.

[56] G. Luo, Y. Yang, Y. Yuan, Z. Chen, and A. Ainiwaer, "Hierarchical transfer learning architecture for low-resource neural machine translation," *IEEE Access*, vol. 7, pp. 154 157–154 166, 2019.

[57] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[58] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[59] S. Clinchant, K. W. Jung, and V. Nikoulina, "On the use of BERT for neural machine translation," in *Proceedings of the 3rd Workshop on Neural Generation and Translation*, 2019, pp. 108–117.

[60] J. Yang, M. Wang, H. Zhou, C. Zhao, W. Zhang, Y. Yu, and L. Li, "Towards making the most of bert in neural machine translation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 9378–9385.

[61] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Science China Technological Sciences*, pp. 1–26, 2020.

[62] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[63] Z. Chi, L. Dong, S. Ma, S. H. X.-L. Mao, H. Huang, and F. Wei, "mt6: Multilingual pretrained text-to-text transformer with translation pairs," *arXiv preprint arXiv:2104.08692*, 2021.

[64] E.-S. A. Lee, S. Thillainathan, S. Nayak, S. Ranathunga, D. I. Adelani, R. Su, and

A. D. McCarthy, “Pre-trained multilingual sequence-to-sequence models: A hope for low-resource language translation?” *arXiv preprint arXiv:2203.08850*, 2022.

[65] F. Guzmán, P.-J. Chen, M. Ott, J. Pino, G. Lample, P. Koehn, V. Chaudhary, and M. Ranzato, “The flores evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english,” *arXiv preprint arXiv:1902.01382*, 2019.

[66] L. Madaan, S. Sharma, and P. Singla, “Transfer learning for related languages: Submissions to the wmt20 similar language translation task,” in *Proceedings of the Fifth Conference on Machine Translation*, 2020, pp. 402–408.

[67] S. Cahyawijaya, G. I. Winata, B. Wilie, K. Vincentio, X. Li, A. Kuncoro, S. Ruder, Z. Y. Lim, S. Bahar, M. L. Khodra *et al.*, “Indonlg: Benchmark and resources for evaluating indonesian natural language generation,” *arXiv preprint arXiv:2104.08200*, 2021.

[68] A. Bapna, N. Arivazhagan, and O. Firat, “Simple, scalable adaptation for neural machine translation,” *arXiv preprint arXiv:1909.08478*, 2019.

[69] Z. Liu, G. I. Winata, and P. Fung, “Continual mixed-language pre-training for extremely low-resource neural machine translation,” *arXiv preprint arXiv:2105.03953*, 2021.

[70] P.-J. Chen, A. Lee, C. Wang, N. Goyal, A. Fan, M. Williamson, and J. Gu, “Facebook ai’s wmt20 news translation task submission,” *arXiv preprint arXiv:2011.08298*, 2020.

[71] R. H. Susanto, D. Wang, S. Yadav, M. Jain, and O. Htun, “Rakuten’s participation in wat 2021: Examining the effectiveness of pre-trained models for multilingual and multimodal machine translation,” in *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, 2021, pp. 96–105.

[72] I. Beltagy, K. Lo, and A. Cohan, “Scibert: A pretrained language model for scientific text,” *arXiv preprint arXiv:1903.10676*, 2019.

[73] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott, "Publicly available clinical bert embeddings," *arXiv preprint arXiv:1904.03323*, 2019.

[74] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: adapt language models to domains and tasks," *arXiv preprint arXiv:2004.10964*, 2020.

[75] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[76] R. Zhang, R. G. Reddy, M. A. Sultan, V. Castelli, A. Ferritto, R. Florian, E. S. Kayi, S. Roukos, A. Sil, and T. Ward, "Multi-stage pre-training for low-resource domain adaptation," *arXiv preprint arXiv:2010.05904*, 2020.

[77] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[78] H. Chen, S. Lundberg, and S.-I. Lee, "Checkpoint ensembles: Ensemble methods from a single training process," *arXiv preprint arXiv:1710.03282*, 2017.

[79] R. Sennrich, B. Haddow, and A. Birch, "Edinburgh neural machine translation systems for wmt 16," *arXiv preprint arXiv:1606.02891*, 2016.

[80] R. Sennrich, A. Birch, A. Currey, U. Germann, B. Haddow, K. Heafield, A. V. M. Barone, and P. Williams, "The university of edinburgh's neural mt systems for wmt17," *arXiv preprint arXiv:1708.00726*, 2017.

[81] K. Imamura and E. Sumita, "Ensemble and reranking: Using multiple models in the nict-2 neural machine translation system at wat2017," in *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, 2017, pp. 127–134.

[82] A. Fernando, S. Ranathunga, and G. Dias, "Data augmentation and terminology integration for domain-specific sinhala-english-tamil statistical machine translation," *arXiv preprint arXiv:2011.02821*, 2020.

[83] M. Rajitha, L. Piyarathna, M. Nayanajith, and S. Surangika, "Sinhala and english document alignment using statistical machine translation," in *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*. IEEE, 2020, pp. 29–34.

[84] F. Farhath, S. Ranathunga, S. Jayasena, and G. Dias, "Integration of bilingual lists for domain-specific statistical machine translation for sinhala-tamil," in *2018 Moratuwa Engineering Research Conference (MERCon)*. IEEE, 2018, pp. 538–543.

[85] R. Sennrich and B. Zhang, "Revisiting low-resource neural machine translation: A case study," *arXiv preprint arXiv:1905.11901*, 2019.

[86] A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary *et al.*, "Beyond english-centric multilingual machine translation," *Journal of Machine Learning Research*, vol. 22, no. 107, pp. 1–48, 2021.

[87] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[88] R. Futrell, K. Mahowald, and E. Gibson, "Quantifying word order freedom in dependency corpora," in *Proceedings of the third international conference on dependency linguistics (Depling 2015)*, 2015, pp. 91–100.

[89] M. Anand Kumar, V. Dhanalakshmi, K. Soman, and S. Rajendran, "A sequence labeling approach to morphological analyzer for tamil language," *IJCSE) International Journal on Computer Science and Engineering*, vol. 2, no. 06, pp. 1944–195, 2010.

Appendix A

Appendix

A.0.1 Addressing the Zero Width Joiner (ZWJ) issue

ZWJ post-processing fix is mainly required for $En \rightarrow Si$ and $Ta \rightarrow Si$ directions. i.e. when Sinhala is on the target side. While translating from xx to Sinhala, to avoid Zero-Width-joiner issue (ZWJ), we added the following codes in the file of Fairseq[1].

We added two lines of code just before the return statement of the method:

```
def post_process(sentence: str, symbol: str):
```

code block to be added:

```
sentence = sentence.replace("\u0DCA\u0dbb", "\u0DCA\u200D\u0dbb")
sentence = sentence.replace("\u0DCA\u0dba", "\u0DCA\u200D\u0dba")
```

A.0.2 Output Translated Sentences

[1] fairseq/fairseq/data/data_utils.py

	Sentence 1	Sentence 2	Sentence 3
Input (Si)	ආසියානු සංවර්ධන බැංකු ආධාර යටතේ මාර්ග ජාල වැඩිදියුණු කිරීමේ ව්‍යාපෘතිය / ADB (Asian Development Bank Funded Road Network Improvement Project / Asian Development Bank)	ආයතනික සහ ප්‍රතිපත්තිමය සහය : ඉදිකිරීම් උපදේශන සේවය (institutional and Policy Support : construction Supervision Consultant)	විමර්ශනයට ලක් කරන ලද ආයතන සම්බන්ධයෙන් පහත සඳහන් කිරීක්ෂණයන් සිදු කර ඇත (the following general observations are made in relation to the institutions examined .)
Ref (Ta)	ஆசிய அபிவிருத்தி வங்கி உதவியின் கீழ் வீதி வலையமைப்புகளை மேம்படுத்தும் செயற்றிட்டம் / ADB	நிறுவன ரீதியான மற்றும் கொள்கை ரீதியான உதவி : நிர்மாண ஆலோசனை சேவை	பரிசீலனை மேற்கொள்ளப்பட்ட நிறுவனங்கள் தொடர்பாகப் பின்வரும் அவதானிப்புக்கள் மேற்கொள்ளப்பட்டன .
Transformer Baseline	ஆசிய அபிவிருத்தி வங்கியின் நிதி உதவியின் கீழ் வீதி வலையமைப்புச் செயற்றிட்டம் / ADB	நிறுவன ரீதியான மற்றும் கொள்கை ரீதியான உதவி : நிர்மாண ஆலோசனை சேவை	விசாரணை செய்யப்பட்ட நிறுவனங்கள் தொடர்பாக பின்வரும் அவதானிப்புக்கள் மேற்கொள்ளப்பட்டன .
B-FT	ஆசிய அபிவிருத்தி வங்கி உதவியின் கீழ் வீதி வலையமைப்பு மேம்பாட்டுச் செயற்றிட்டம் / ADB	நிறுவன ரீதியான மற்றும் கொள்கை ரீதியான உதவி : நிர்மாண ஆலோசனைச் சேவை	விசாரணைக்குட்படுத்தப்பட்ட நிறுவனங்கள் தொடர்பாக பின்வரும் அவதானிப்புக்கள் மேற்கொள்ளப்பட்டன .
Input (Ta)	பின்னதூவலில் இருந்து மாத்தறை வரை காலித் துறைமுகத்துக்கு நுழைவுப் பாதை : 05 % (Pinnaduwa to Matara ; GPAR : 05 %)	கிழக்கு மாகாணத்தில் வறிய மக்களுக்கான உட்கட்டமைப்பு வசதிகளை மேம்படுத்தல் : JICA , உள்ளூர் நிதியம் (Pro-Poor Eastern Infrastructure Development Project : Japan International Coorporation Agency & Government of Sri Lanka)	ஆசிய அபிவிருத்தி வங்கியும் இலங்கை அரசும் (Asian Development Bank and Government of Sri Lanka)
Ref (Si)	පින්නදුව සිට මාතර දක්වා සහ ගාල වරාය දිවියුම් : 05 %	නැගෙනහිර පළාතේ සුපිරි සුප්‍රජා ජනයා සඳහා වූ යටිතල පහසුකම් වැඩිදියුණු කිරීමේ ව්‍යාපෘතිය : JICA සහ දේශීය අරමුදල්	ආසියානු සංවර්ධන බැංකුව සහ ශ්‍රී ලංකා රජය
Transformer Baseline	පින්නදුව සිට මාතර දක්වා ගාල වරාය දිවියුම් : 05 %	නැගෙනහිර පළාතේ සුපිරි සුප්‍රජා ජනයා සඳහා වූ යටිතල පහසුකම් වැඩිදියුණු කිරීම : JICA සහ දේශීය අරමුදල්	ආසියානු සංවර්ධන බැංකුව සහ ශ්‍රී ලංකා රජය
B-FT	පින්නදුව සිට මාතර දක්වා සහ ගාල වරාය දිවියුම් : 05 %	නැගෙනහිර පළාතේ සුපිරි සුප්‍රජා ජනයා සඳහා වූ යටිතල පහසුකම් වැඩිදියුණු කිරීමේ ව්‍යාපෘතිය : JICA සහ දේශීය අරමුදල්	ආසියානු සංවර්ධන බැංකුව සහ ශ්‍රී ලංකා රජය
Input (Si)	සම්මෝචිත වර්ෂය තුළ අයවැයෙහි හා සැබෑ ආදායම් හා වියදම් අතර ප්‍රමාණාත්මක විචලනයන් නිරීක්ෂණය වූ හෙයින් අයවැය ලේඛනය ඵලදායී පාලන කාරකයක් ලෙස භාවිත කර නොතිබුණු බව කිරීක්ෂණය විය .	අන්තර්වාර වාර්තාවට අනුව පරීක්ෂා කළ ආයතන 32 ද ඇතුලත්ය .	2014 වර්ෂයේ අයවැය ඇස්තමේන්තුවෙන් තෝරාගත් අමාත්‍යාංශවල වැය ශීර්ෂයන් පිළිබඳ සාකච්ඡා කිරීම සඳහා වන පාර්ලිමේන්තු විශේෂ කාරක සභාව ,
Ref (En)	significant variances were observed between the budgeted and the actual income and expenditure for the year under review , thus indicating that the budget had not been made use of as an effective instrument of control .	includes 32 Institutions examined as per Interim Report .	Select Committee of Parliament to discuss the Heads of Expenditure of the Ministries selected from the Budget Estimates of the year 2014 ,
Transformer Baseline	since the budget had been observed between the budgeted and the actual income and expenditure , thus indicated that the budget had not been made use of as an effective instrument of the budget had not been made .	32 institutions checked according to the Interpretation report .	Select Committee to discuss the Heads of Expenditure of Ministries selected from the Budget Estimate 2014 ,
B-FT	Since considerable variances were observed between the budgeted and the actual income and expenditure during the year under review , it was observed that the budget had not been made use of as an effective instrument of control .	the total number of Institutions examined according to the Interim Report was 32 .	Select Committee of Parliament to discuss the Heads of Expenditure of Ministries selected from the Budget Estimates of 2014 ,
Input (En)	Pro-Poor Eastern Infrastructure Development Project : Japan International Coorporation Agency & Government of Sri Lanka	Enhance technological knowledge need to conduct experiments on lesser known and underutilized plants .	Housing Project Operating Expenses
Ref (Si)	නැගෙනහිර පළාතේ දුසී සුප්‍රජා ජනයා සඳහා වූ යටිතල පහසුකම් වැඩිදියුණු කිරීමේ ව්‍යාපෘතිය : JICA සහ දේශීය අරමුදල්	දැනුවත් බවින් අඩු සහ භාවිතය අඩු ශාක ගවේෂණය කිරීමට අවශ්‍ය තාක්ෂණික දැනුම දියුණු කිරීම .	නිවාස ව්‍යාපෘතිය මෙහෙයුම් වියදම්
Transformer Baseline	නැගෙනහිර පළාතේ දුසී සුප්‍රජා ජනයා සඳහා වූ යටිතල පහසුකම් සංවර්ධන ව්‍යාපෘතිය : ජපන් ජාත්‍යන්තර සහයෝගිතා නියෝජිතායතනය සහ ශ්‍රී ලංකා රජය	දැනුවත් බවින් අඩු සහ භාවිතය අඩු ශාක ගවේෂණය කිරීමට අවශ්‍ය තාක්ෂණික දැනුම ඉහළ නැංවීම් .	නිවාස ව්‍යාපෘතිය මෙහෙයුම් වියදම්
B-FT	නැගෙනහිර පළාතේ දුසී සුප්‍රජා ජනයා සඳහා වූ යටිතල පහසුකම් වැඩිදියුණු කිරීමේ ව්‍යාපෘතිය : ජපන් ජාත්‍යන්තර සහයෝගිතා නියෝජිතායතනය සහ ශ්‍රී ලංකා රජය	දැනුවත් බවින් අඩු සහ භාවිතය අඩු ශාක ගවේෂණය කිරීමට අවශ්‍ය තාක්ෂණික දැනුම දියුණු කිරීම .	නිවාස ව්‍යාපෘතිය මෙහෙයුම් වියදම්
Input (Ta)	மீளாய்வாண்டில் பாதிடப்பட்ட மற்றும் உள்ளவாறான வருமானம் மற்றும் செலவினத்திற்கிடையேயும் முரண்கள் அவதானிக்கப்பட்டமையால் பாதீடு சிறந்த முகாமைத்துவக் கட்டுப்பாட்டுக் கருவியாக பயன்படுத்தப்பட்டிருக்காமை அவதானிக்கப்பட்டது .	இடைக்கால அறிக்கையின்படி பரிசீலிக்கப்பட்ட 32 நிறுவனங்களை உள்ளடக்குகின்றது .	1. ஒரு வருட காலப்பகுதிக்கும் மேலாக பிரதம நிறைவேற்று அதிகாரியின பதவி வெற்றிடமாயுள்ளது .
Ref (En)	significant variances were observed between the budgeted and the actual income and expenditure for the year under review , thus indicating that the budget had not been made use of as an effective instrument of control .	includes 32 Institutions examined as per Interim Report .	1 . the post of Chief Executive Officer had been vacant over a period of one year .
Transformer Baseline	It was observed that the budgeted and the income and expenditure as at the date of the year under review had not been utilized as better management control instruments .	As per the Interim Report , 32 institutions examined as per the Interim Report .	1 . The post of Executive Officer has been vacant for a period of one year .
B-FT	It was observed that the conflicts were observed between the internal income and expenditure during the year under review and that the budget had not been made use of as an effective instrument of management control .	This includes 32 institutions examined as per the interim report .	1 . the post of Chief Executive Officer had been vacant over a period of one year .
Input (En)	spiritual Leadership Training for Hindu Youths	issuance of Budget guidelines on the preparation of the Budget Estimates .	7.National Secretariat for Non-Governmental Organizations
Ref (Ta)	இந்து இளைஞர்களுக்கான ஆன்மீக தலைமைத்துவப் பயிற்சி	வரவு செலவுத்திட்ட மதிப்பீடுகளைத் தயாரித்தல் தொடர்பான வரவு செலவுத்திட்ட வழிகாட்டல்களை வழங்குதல்	7. அரச சார்பற்ற நிறுவனங்களுக்கான தேசிய செயலகம்
Transformer Baseline	இந்து இளைஞர் யுவதிகளுக்கான ஆன்மீகப் பயிற்சி	வரவு செலவுத் திட்ட மதிப்பீடுகளைத் தயாரிப்பது தொடர்பான வரவு செலவுத்திட்ட வழிகாட்டல்களை வழங்குதல் .	7. அரச சார்பற்ற நிறுவனங்களுக்கான நடவடிக்கை செயலகம்
B-FT	இந்து இளைஞர்களுக்கான ஆன்மீக தலைமைத்துவப் பயிற்சி	வரவு செலவுத்திட்ட மதிப்பீடுகளைத் தயாரித்தல் தொடர்பான வரவு செலவுத்திட்ட வழிகாட்டல்களை வழங்குதல்	7. அரச சார்பற்ற நிறுவனங்களுக்கான தேசிய செயலகம்

Figure A.1: Output Translated Sentences