Exploring Low-resource Neural Machine Translation for Sinhala-Tamil Language Pair

Ashmari Pramodya

University of Colombo School of Computing, Sri Lanka
ash@ucsc.cmb.ac.lk

Abstract

At present, Neural Machine Translation has become a promising strategy for machine translation. Transformer-based deep learning architectures in particular show a substantial performance increase in translating between various language pairs. However, many low-resource language pairs still struggle to lend themselves to Neural Machine Translation due to their data-hungry nature. In this article, we investigate methods of expanding the parallel corpus to enhance translation quality within a model training pipeline, starting from the initial collection of parallel data to the training process of baseline models. Grounded on state-of-the-art Neural Machine Translation approaches such as hyper-parameter tuning, and data augmentation with forward and backward translation, we define a set of best practices for improving Tamil-to-Sinhala machine translation and empirically validate our methods using standard evaluation metrics. Our results demonstrate that the Neural Machine Translation models trained on larger amounts of back-translated data outperform other synthetic data generation approaches in Transformer base training settings. We further demonstrate that, even for language pairs with limited resources, Transformer models are able to tune to outperform existing state-of-the-art Statistical Machine Translation models by as much as 3.28 BLEU points in the Tamil to Sinhala translation scenarios.

1 Introduction

Since 1949, when machine translation was initially proposed (Hutchins, 1995), Statistical Machine Translation (SMT) models dominated the machine translation field for decades. However, the advent of Neural Machine Translation (NMT) using deep learning (Bahdanau et al., 2014; Cho et al., 2014; Sutskever et al., 2014) has revolutionized the field, enabling superior performance in translation tasks. Recently, NMT tends to employ Transformer (Vaswani et al., 2017) architecture which is a novel architecture grounded only on attention mechanisms. While it has shown remarkable results for high-resource languages, such as English, it struggles with low-resource languages like Sinhala and Tamil, which are morphologically rich and low-resourced languages. Despite the existence of the best open-source Sinhala-Tamil translator using SMT, NMT has not been widely experimented on in an open-domain setting. Hence, improving NMT for low-resourced languages remains an open research problem with proven success.

In this paper we aims to investigate the performance of Transformer models on Tamil and Sinhala machine translation, with the goal of establishing best practices for low-resource NMT. We explore various model architectures and hyperparameter tuning methods to develop an accurate model for these languages. To address the issue of insufficient parallel data, we expand the corpus size and evaluate the impact of data size on NMT for low-resource languages. We also examine the effects of back translation and forward translation mechanisms for machine translation. Finally, we compare the performance of our Transformer models with SMT. Our research represents a novel contribution, as there is currently an absence of exploration into the best practices for Transformer-based models in Sinhala and Tamil Neural Machine Translation (NMT) within low-resource contexts.

The rest of the paper is structured as follows: the state-of-the-art studies are critically analyzed in Section 2, Section 3 describes the methodology, and Section 4 presents the detailed experimental settings, including the utilised data sets, tools, and training protocol of MT. In Section 5 we present the experimental results. Finally, Section 6 presents the future works and concludes the paper.

Proceedings of the Student Research Workshop associated with RANLP-2023,
pages 87–97, Varna, Bulgaria, Sep 4–6, 2023.
https://doi.org/10.26615/issn.2603-2821.2023_010

2 Literature Review

2.1 Low Resourced Machine Translation

Bilingual sentence pairs are a large collection of annotated data that is essential for training a model with adequate translation quality. However, for numerous languages, we are unable to access large parallel data sets. As a result, numerous research attempts have been made to incorporate monolingual corpora into machine translation (Haddow et al., 2022; Ranathunga et al., 2023).

One of the techniques to improve NMT for low-resource languages is back translation (Sennrich et al., 2015). This involves training a target-to-source (backward) model on the parallel data available and using that model to construct synthetic translations in the monolingual sentences of the targeted language. To train the final source-to-target (forward) model, the existing authentic parallel data are combined with the newly created synthetic parallel data without differentiating between the two (Sennrich et al., 2015). The authentic parallel data provided for NMT isn't large enough to train a backward model that produces qualitative synthetic data. As a result, giving priority to the issue of the lack of parallel data, numerous methods have been proposed to improve the efficiency of the backward model.

Park et al. (2017) solely used synthetic parallel data from both the source and target sides to create the NMT model. Further, according to the Sennrich et al. (2015) the amount of monolingual data only increases the quality of translation to a certain extent, and then it begins to degrade. This phenomenon allows to impose constraints on the amount of monolingual data that can be employed in translation tasks. Moreover, as a result of low-quality synthetic data, the back-translated data may face numerous issues and long-term negative impacts on translation efficiency. Hoang et al. (2018) propose an iterative back-translation approach to address this issue and improve the performance, by using the monolingual data more than once. Additionally, Xu et al. (2019) suggested a method based on sentence similarity score to filter quality synthetic data utilizing bilingual word embeddings (Xu et al., 2019) and sentence similarity metrics (Imankulova et al., 2017). Further, there are a few possible methods of incorporating the monolingual corpora into machine translation, including Dual learning (Xia et al., 2016) and unsupervised machine translation using monolingual corpora alone for both sides (Lample et al., 2017).

2.2 Hyper-Parameter Exploration

Knowing which hyper-parameters to select while training a model is crucial. The parameters chosen prior to the start of training are referred to as hyper-parameters. The optimization of hyper-parameters basically referred to as finding the most optimal tuple that will minimize the predefined loss function on a given set of data.

The difference between low and high-resource NMT is not limited to having more parallel data. It has been shown that in bilingual low-resource scenarios, Phrase-Based Statistical Machine Translation (PBSMT) models outperform NMT models. However, in high-resource scenarios, NMT outperforms PBSMT models (Koehn and Knowles, 2017). Moreover, Sennrich and Zhang (2019) study on low-resource NMT, shows that it is extremely sensitive to hyper-parameters, architectural design, and other design considerations. Unfortunately, their outcomes are limited to a recurrent NMT architecture. Recently, Duh et al. (2020) show that SMT and NMT Transformers work similarly in low-resource scenarios, but the NMT systems require more careful tuning to achieve the same performance as SMT. Most recently, Araabi and Monz (2020) researched the effects of hyper-parameter settings for the Transformer architecture under various low-resource data conditions. Their experiments demonstrate that compared to a Transformer system with default settings for all low-resource data sizes, the appropriate combination of Transformer configurations and regularization algorithms yields significant improvements.

There are numerous ways to choose hyper-parameters, most often with manual tuning and random search or grid search (Bergstra and Bengio, 2012). Apart from that, other methods, such as Bayesian optimization (Bergstra et al., 2011), genetic algorithms (Chapelle et al., 2002), and gradient updates (Maclaurin et al., 2015) direct the hyperparameter selection based on the objective function. However, in order to get accurate performance, all of these approaches require the training of several networks with different hyper-parameter settings.

2.3 Research in Sinhala-Tamil Language Pair

Sinhala and Tamil are the national languages of Sri Lanka. Sinhala belongs to the Indo-Aryan lan-

guage family, while Tamil is a member of the Dravidian language family (Pushpananda et al., 2014). Both Sinhala and Tamil have a broad morphological vocabulary, Sinhala has up to 110 noun word forms and up to 282 verb word forms (Welgama et al., 2011) and Tamil has around 40 noun word forms and up to 240 verb word forms (Pushpananda et al., 2014). Moreover, syntactically, both Sinhala and Tamil are also close. As a result, the SMT method was able to produce a superior performance in Tamil to Sinhala translation (Pushpananda and Weerasinghe, 2015).

However, Only a few research studies have examined NMT in the Sinhala-Tamil language pairs. Research by Arukgoda et al. (2019) on improving Sinhala–Tamil translation through deep learning techniques provides a prominent foundation for Sinhala and Tamil machine translation in a semi-supervised manner using bidirectional recurrent neural networks. This study has been undertaken for the open-domain context whereas Tennage et al. (2017) also report studies for the NMT using recurrent neural networks for a specific domain. Moreover, recently (Nissanka et al., 2020) explored the use of a monolingual word embedding approach for developing the translations between Sinhala and Tamil language pairs just utilizing monolingual corpora. Additionally, in the context of Tamil to Sinhala machine translation, Pramodya et al. (2020) contrasted Transformers, Recurrent Neural Networks, and SMT with default parameter settings.

3 Methodology

In this article, we concentrate on two primary research directions to address the low-resource problem: (1) exploring hyper parameters with available parallel data (25k), and (2) devise methods to exploit additional opportunistic data sources.

3.1 Hyper-parameter Exploration

Transformer, like all NMT models, involves the setting of different hyper-parameters, but researchers often use the default values, even though their data conditions differ significantly from those used to evaluate the default values (Gu et al., 2018). Computing the full set of possible values for several hyper-parameters at once is computationally intensive. Hence, we will adjust the hyper-parameters that come under vocabulary representation, architecture tuning and regularization settings.

Text-To-Text Transfer Transformer (T5) : At present, with the burgeoning of Transfer Learning, Deep Learning has excelled in various complex tasks. Specifically, in NLP, we leverage transfer learning by pre-training on a task-agnostic objective and then fine-tuning it on particular downstream problems. By leveraging a unified text-to-text format and a massive training data-set (C4 : Colossal Clean Crawled Corpus), the original T5 (Raffel et al., 2019) model achieved state-of-the-art results on a variety of NLP benchmarks. Moreover, the mT5 (Xue et al., 2020) model is a multilingual variant of the original T5 model, aimed at remedying this problem. Although the mT5 model was trained on mC4 (about 26 Terabytes), a multilingual variation of the C4 data set, it closely follows the architecture and training process of T5. Because of this, it still has all the benefits of the T5 model and supports a total of 101 distinct languages.

3.2 Exploring Additional Data with Different Domains

We explored several resources to collect parallel sentences such as crawling the web to mine parallel sentences that already exist and to mine completely novel parallel sentences.

3.3 Exploring Synthetic Data

Here we applied Back Translation (Sennrich et al., 2015), which involve creating artificial source-side sentences by translating a monolingual set in the target language. Further, we employed synthetic data on the target side (Zhang and Zong, 2016). Specifically, the synthetic data was generated through two sources namely 1) Using Transformer-base, and 2) Google translate (GNMT).

Back Translation: Back-translation is a popular method in state-of-the-art machine translation tasks (Edunov et al., 2018). It has shown superior performance compared to other translation approaches in high-resource language settings, and it has also been proven to be effective in low-resource language situations (Cho et al., 2014). This technique involves building a backward model from parallel data, which is used to generate synthetic translations of monolingual sentences in the target language. The produced synthetic parallel data is mixed with the real parallel data to train a final source-to-target (forward) model.

Forward Translation: Forward translation (Zhang and Zong, 2016) improves NMT efficiency by us-

ing source-side monolingual data to generate synthetic translations and create a synthetic parallel data set. This leads to a better source-to-target translation model trained with a massive amount of data, while also allowing the target side to learn from synthetic data for improved grammatical correctness

Synthetic Data through Google translate: In order to maximize the use of Google Translate, the back-translation method and Google Translate are used to generate a parallel corpus for training our translation model. This is an approach that is close to the one suggested by (Pham and Nguyen, 2019). However, we had to do some post-processing to the synthetic data as it contains English words in between the words.

4 Experimental Setup

4.1 Hyper-parameter Exploration

Dataset : Our baseline training data comprises around 25000 phrases with a length between 8 and 12 words which was used in the SMT collected by Pushpananda and Weerasinghe (2015). They have used two approaches to collect these parallel data. The first approach was identifying Sinhala-Tamil parallel documents such as magazines, books, articles and the second approach was translating Sinhala sentences to Tamil with the help of professional translators. The corpus statistics for the parallel corpus is given in Table 7 in appendix. We investigated the hyper-parameters for the Tamil to Sinhala translation direction. We were able to fairly compare SMT and NMT in the setting of Tamil and Sinhala because our baseline SMT study (Pushpananda and Weerasinghe, 2015) employed the same corpora (25k).

BPE Effect: In order to improve the translation of rare words, word segmentation approaches such as Byte-Pair-Encoding (BPE) (Sennrich et al., 2016) have become standard practice in NMT.

For Tamil to Sinhala translation, we used BPE merge values of 1k, 2k, 5k, and 10k. The BPE model was trained on the complete training corpus, enabling us to assess the influence of various levels of BPE segmentation on translation performance Here, we selected the merge values by training the models on default parameter settings, and we used smaller numbers of merging operations since we were dealing with smaller training data conditions.

Architecture and Regularization: All the research experiments were carried on openNMT toolkit (Klein et al., 2017). For our experiments, we reduced the number of attention heads in encoder and decoder layers as well as the model dimension. Moreover, we sampled the size of the feed-forward network and also the batch size. Further, we conducted experiments with different values for the learning rate (1,2) and warm-up steps (8000,4000) using the learning rate scheduler, as implemented within OpenNMT-py (Klein et al., 2017). Regularization is used to increase generalization ability and minimize over-fitting in neural networks. Thus, in our experiments, we employed Dropout, one of most effective regularization strategy introduced by (Srivastava et al., 2014). Following (Sennrich and Zhang, 2019), we investigated the impact of regularization by applying dropouts to Transformer. Moreover, we also experimented with larger label-smoothing factors. The selected hyper-parameters for our experiments, and their values are presented in Table 1. Notably, these hyper-parameters and their values depend on preliminary experiments and previous findings (Sennrich and Zhang, 2019; Fonseka et al., 2020; Duh et al., 2020) that identify the hyper-parameters that have the greatest impact on translation efficiency.

Hyper-parameter	Values
Number of Layers in encoder/decoder	5,6
Attention Heads	2, 4
Embedding dimension	256, 512
Feed Forward dimension	1024, 2048
Drop Out	0.2, 0.3, 0.4
Label smoothing	0.1, 0.2
Batch size	2048, 4096
Warm-up steps	8000, 4000
Learning rate(define by OpenNMT (Klein et al., 2017))	1, 2

Table 1: Hyper-parameters considered during the tuning of Transformer with 25k parallel data.

Furthermore, we fine tuned the mT5 model with our data and used it for evaluation of Tamil to Sinhala Translation tasks. To train the mT5 model, we used the Simple Transformers library[1] (based on the Huggingface Transformers library) and the training and testing data will be the same as earlier experiments. For the experiments, we used a training/evaluation batch size of 20 and a maximum sequence length (max seq length) of 96. In our initial experiments, the model worked with relatively long text due to the maximum sequence length of 96. However, we ran out of GPU memory (CUDA

[1] https://github.com/ThilinaRajapakse/simpleTransformers

memory error), and then we decided to reduce the batch size to 4 instead of reducing the maximum sequence length.

4.2 Exploring Additional Data

Previous studies have been conducted (Arukgoda et al., 2019; Nissanka et al., 2020) using only 25k parallel data. In this, we aimed to determine the impact of corpus size on the quality of Tamil to Sinhala translation by expanding the parallel dataset with additional bi-text data. We investigated various resources to collect parallel sentences, including those that already exist and those that can be mined by crawling web pages(refer to the Appendix for more details). Sinhala and Tamil sentence tokenization was done using indicNLP library [2]. Further, models were trained using the default parameters of the Transformer-based architecture and the BPE merge operation value was set to 5k. In order to fairly assess the translation, we employed the BLEU score metric (Bi-Lingual Assessment Understudy) (Papineni et al., 2002).

Before merging the collected parallel datasets, we performed additional cleaning and deduplication. Only after this step, we trained the datasets independently. Initially, we trained the datasets separately and evaluated their performance. However, the BLEU scores did not show any significant improvement in the entire corpus of the bible. This could be attributed to the fact that the bible was aligned by verse, rather than by sentences.

Further, it contains sentences that are too long and need to be split, which is challenging due to irregular usage of splitting punctuations. Therefore, we reduced the corpus by taking the sentences which have sentence length between 1 and 20. The used test set consists of 10% of the training data set which are mutually exclusive. We show the BLEU scores on both Test sets. When compared to Test set A, Test set B does not contain any bible sentences in the evaluation set. Basically, it only contains News Crawl as same as the test set used in section 5.1. The two test sets used in our experiments can be summarized as follows:

i. Test A: 10% of the training data (Eval contains domain-specific data).
ii. Test B: 1000 news sentences (This test set is used for evaluating the baseline systems presented in Table 3).

[2] https://github.com/anoopkunchukuttan/indic_nlp_library

4.3 Exploring Synthetic Data

Monolingual Corpus: For our experiments, we used a Sinhala monolingual corpus with 10M words and a Tamil monolingual corpus with 400,000 words (Pushpananda and Weerasinghe, 2015). Both of these corpora are ideal for an open-domain translation as they have been collected from sentences from different domains such as newspaper articles, technical writing and creative writing.

We conducted experiments to evaluate synthetic data in both source and target sides for the machine translation. In addition, we also examined, how the models perform when training data is augmented with synthetic data which was generated using various MT approaches. In particular, we investigated generated synthetic data not only by Transformer base methods (our NMT model) but also by Google neural Machine translate (GNMT) model.

In addition to backward translation where monolingual corpora were used in the target language, we also also looked at forward translation where monolingual corpora were used in the source language.

Inspired by Poncelas et al. (2019), we continued to create NMT models with increasing sizes of data to assess the effects of synthetic data. Here we used the same default training settings for Transformer-base architecture in order to evaluate how much synthetic data is required to switch back to the commonly used Transformer configurations. Moreover, by employing a random search of hyper parameters we tuned the models at different data-sets only in Tamil $\rightarrow$ Sinhala translation direction, and those configurations were subsequently utilized to train Sinhala $\rightarrow$ Tamil translation direction models. We demonstrate those results in T-tuned columns in Table 6 and Table 7.

5 Results

This section presents the results obtained from the experiments conducted on the Tamil and Sinhala language pairs.

5.1 Hyper Parameter Exploration

We used Transformer-base and SMT (Pushpananda and Weerasinghe, 2015) as our baselines. It took approximately 10-12hrs time to train our models for 15k train steps. For different selected subsets, we obtained significant improvements over Transformer-base. The best results obtained so far from tuning are presented in the Table 2. The re-

	A	B	C	D	E
Layers	5	5	5	5	5
Embedding dimension	512	512	512	512	512
Heads	4	2	2	4	2
Feed-forward dimension	2048	2048	2048	2048	2048
Dropout	0.4	0.3	0.4	0.4	0.3
Label smoothing	0.2	0.2	0.2	0.2	0.2
Batch-size	2048	2048	2048	4096	2048
Warm-up steps	8000	4000	8000	8000	4000
Learning rate (define by OpenNMT)	2	1	2	2	1
BLEU	**16.39**	16.13	16.11	15.83	15.60

Table 2: The best hyper-parameter configurations obtained

sults demonstrate that, reducing the Transformer attention heads, and the number of layers, along with increasing the rate of different regularization techniques are highly effective (+5 BLEU) for the 25k data-set.

Test	இந்த வரலாற்று கதையுடன் வெளிப்படும் மேலும் முக்கியமான பாடமொன்று இருக்கின்றது
English	There is one more important lesson that emerges with this historical story
Reference	මේ ඉතිහාස කතාවත් සමඟ මතුවන තවත් වැදගත් පාඩමක් තිබෙ
SMT	මෙම ඓතිහාසික කතා නද රය කින් මතුවන තවත් වැදගත් පාඩමක් තිබෙ.
Transformer-B	මේ ඓතිහාසික කතාවලට බලපාන තවත් වැදගත් පාඩමක් තිබෙ.
Transformer-T	මේ ඓතිහාසික කතාවෙන් හෙළි වන තවත් වැදගත් පාඩමක් තිබෙ
mT5_TA-SI	මේ ඉතිහාසය තුළින් පැහැදිලි වන තවත් වැදගත් පාඩමක් තිබෙ.

Figure 1: Tamil Sinhala translation example with SMT and NMT systems trained on 25k parallel data on Transformer-B(base),Transformer-T(tuned)(gray color highlighted words give semantically correct meaning)

As Sennrich and Zhang (2019) report reducing the batch size is effective. We also demonstrated that setting the batch size to 2048 performs better than when the batch size is 4096. We were able to outperform SMT results by 3.28 BLEU points with our optimized Transformer. The BLEU scores obtained by our optimized Transformer model were compared with the baseline models presented in Table 3. We also compared the translation quality with that of Google Translate on the same test data set. Examples of translations between SMT and NMT systems can be seen in Figure 1. In summary, our Transformer-T model provided more semantically accurate translations compared to the other models.

5.1.1 Human Evaluation

We utilized the Human ranking of translation scores at the sentence level strategy to compare the performance of the models listed in the Table 3. The models were trained using an equivalent volume (25k) of parallel data. For the human evaluation, ten final-year undergraduates from the translation

Model	**BLEU** **Tamil - Sinhala**
SMT (Pushpananda and Weerasinghe, 2015)	13.11
Transformer-Base	11.49
Transformer-Tuned	16.39
mT5_TA-SI	11.56

Table 3: Comparison of BLEU score against baseline models for Tamil to Sinhala

studies department at the University of Kelaniya, Sri Lanka, participated. Ten sentences from the test set were randomly selected and distributed among the participants, who were subsequently tasked with ranking the translated output. Participants were provided with reference sentences and translated sentences from the four different systems. Furthermore, we instructed them to rank the sentences based on quality, arranging them from best to worst. Throughout this process, we ensured that no ties were permitted and that we adhered to the established guidelines in (Narayan et al., 2017). Table 4 shows the findings of our human evaluation study. We compared our tuned Transformer, mT5_TA-SI, Transformer base, with SMT to determine how much each system is rated as the best, second best, and so on. According to the partici-

Model	1st	2nd	3rd	4th
SMT	0.11	0.25	0.37	0.27
Transformer-Base	0.14	0.22	0.34	0.30
Transformer-Tuned	0.43	0.35	0.10	0.12
mT5_TA-SI	0.32	0.18	0.19	0.31

Table 4: Ranking of various systems. Rank 1st is best and rank 4th, worst. Numbers show the percentage of times a system gets ranked at a certain position.

pants' rankings presented in Table 4, Transformer-Tuned got the highest ranking percentage (43%). Moreover, the mT5_TA-SI was the second-highest-ranked approach by 32%. However, Transformer-Base and SMT, are the least ranked methods respectively. Moreover, evaluating the Transformer-Tuned, mT5_TA-SI and Transformer-Base using the BLEU score also gives the same results. However, BLEU evaluation scores are different when evaluating all four models. Notably, synonyms and paraphrases are only taken into account by the BLEU metric only if they are in the set of multiple reference translations. Further, NMT systems capture the similarity of words which may results in having synonyms in translation outputs. However,

due to the limited resources, Sinhala and Tamil language pairs do not have the luxury of having multiple references. Hence, we are able to assume that this would be the reason why the Ranking and BLEU metrics calculations produced different results.

5.2 Exploring additional Data

	Data size	Tamil- Sinhala		Sinhala - Tamil	
		Test A	Test B	Test A	Test B
baseline	25k	11.49	11.49	4.98	4.98
+ Bible	45k	13.48	9.38	7.82	4.28
+ Bible + found bitext	55k	14.22	13.25	9.89	6.87
+ Bible + found bitext + Text extracted	65k	15.61	14.48	11.10	6.99

Table 5: The effect of additional resource types for NMT. We observed that adding Bible, Text extracted and found-bitext to baseline tends to improve the performance for NMT, with NMT gaining significant benefits. Models are trained with Transformer-base configurations.

5.2.1 Analysis

The impact of different data types on the model's quality is systematically assessed in the following section.

Effectiveness of additional data

Table 5 shows the performance impact of found Bitext, Paracrawl, and the Bible datasets when combined with our initial training set. We observed noticeable improvement of translation for both directions. For example, on Test Set A, the BLEU points were improved from 11.49 to 15.61 by 4.12 points for Tamil to Sinhala direction and for Sinhala to Tamil direction there was a 6.12 BLEU points improvement from 4.98 to 11.10. The trend is observed in the Test set B as well but only after adding biblical corpus. As depicted in the second row of 5, we observed a significant drop in performance for the parallel training data that differ from the evaluation domain in Test set B. However, in order to improve performance and robustness, we might need to create a validation set that is better matched or use a domain adaptation technique for different domains. We conclude that employing additional data types is a promising research direction, particularly for NMTs with limited resources like Tamil and Sinhala languages.

Figure 2 shows an example sentence from a Bible corpus that has been translated. The resulting translation accurately conveys the intended meaning of the target sentence. It's worth noting that the writing style of Bible verses differs from that of typical news articles sentences, which presents a unique challenge for machine translation.

Test	பரலோகத்தில் இருக்கிற தேவரீர் கேட்டு , உம்முடைய ஜனமாகிய இஸ்ரவேலின் பாவத்தை மன்னித்து , அவர்கள் பிதாக்களுக்கு நீர் கொடுத்த.
English	Then hear thou in heaven, and forgive the sin of thy people Israel, and bring them again unto the land which thou gavest unto their fathers.
Reference	ඔබ ස්වර්ගයේ සිට අසා වදාරා ඔබගේ සෙනඟවූ ඉශ්‍රායෙල්වරුන්ගේ පාපය කමාවී , ඔවුන්ගේ පියවරුන්ට ඔබ දුන් දේශයට ඔවුන් නැවත පැමිණෙවුව මැනව
Transformer-T 65k	ඔබ ස්වර්ගයේ සිට අසා , ඔබගේ සෙනඟවූ ඉශ්‍රායෙල්වරුන්ගේ පාපය කමාකොට , ඔවුන්ගේ පියවරුන්ට දුන් දේශයට ඔවුන් නැවත පැමිණෙවුව මැනව .

Figure 2: Sample Translation example of Bible verse From Transformer 65k (gray color highlighted words give semantically correct meaning)

Effect of various synthetic data

The experimental results for the analysis of diverse synthetic data are shown in Table 6 and Table 7. We can see from the findings in Table 6 and Table 7 that adding synthetic data on both sides can enhance the performance in the translation from Tamil → Sinhala direction. Moreover, all BLEU scores are also higher when compared to the networks built only using authentic data. Surprisingly, in opposite translation direction (Sinhala → Tamil) synthetic data has a significant negative impact on results when the data size is increased. Although BLEU scores are dropping when the synthetic data amount is 75k in Tamil → Sinhala direction, they are still higher than the baseline and this phenomenon can be observed in both synthetic data generation approaches we used to generate synthetic data.

Particularly, based on the outcomes shown in Table 7, models developed with synthetic data produced by GNMT outperform those developed with data produced by Transformer-Base (*NMT_syn_ta*). Here NMT models are trained with default configurations of Transformer base architecture in both translation directions When comparing models trained on Transformer base architecture, with an equal amount of GNMT (*GNMT_syn_ta*, *GNMT_syn_si*) or Transformer-Base (*NMT_syn_sin*, *NMT_syn_ta*) synthetic data, we find that the GNMT one outperforms the Transformer-Base by around 2.0 BLEU points. However, the difference is only 0.01 BLEU points when the GNMT and Transformer base models' (100k) hyper-parameters are tuned.

As depict in the Table 6 and Table 7, synthetic data have opposite effects on the two translation directions. Specifically, when translating Tamil to Sinhala direction, the monolingual synthetic

data from both sides have positive effects. Moreover, the back translation with both approaches, Transformer-base (*NMT_syn*) and Google Translate (*GNMT_syn*) have nearly same performance when the NMT models are tuned with correct hyper parameters. Moreover, forward translation under performs back translation in both synthetic data generation methods we used.

Direction	Tamil → Sinhala		Sinhala → Tamil	
	T-base	T-tuned	T-base	T-tuned
baseline	11.46	16.39	4.89	8.08
+ Synthetic Tamil data *(NMT_syn_ta)*				
25k	12.32	14.42	5.97	7.52
50k	13.03	14.12	**7.12**	**8.10**
75k	15.12	17.74	6.54	6.95
100k	**16.46**	**19.00**	6.65	6.96
+ Synthetic Sinhala data *(NMT_syn_sin)*				
25k	11.74	14.03	6.05	6.81
50k	13.64	13.54	**6.34**	7.14
75k	13.69	14.13	6.12	**8.26**
100k	**13.71**	**14.42**	5.39	6.56

Table 6: Results of corpus extension by using synthetic data generated by Transformer-base model

How much synthetic data do we need for Sinhala-Tamil language pair models to reach reasonable translation quality ?

Direction	Tamil → Sinhala		Sinhala → Tamil	
	T-base	T-tuned	T-base	T-tuned
Baseline	11.46	16.39	4.89	8.08
+ google Synthetic Tamil data *(GNMT_syn_ta)*				
25k	13.25	14.85	5.30	7.86
50k	15.84	18.05	**6.29**	8.49
75k	17.32	18.26	6.09	6.25
100k	**18.44**	**19.01**	5.89	6.84
+ google Synthetic Sinhala data *(GNMT_syn_sin)*				
25k	14.89	17.42	7.14	7.28
50k	15.75	17.89	**8.08**	**8.80**
75k	16.26	18.42	7.20	8.12
100k	**17.78**	**18.69**	7.39	**8.26**

Table 7: Results of corpus extension by using synthetic data generated by Google translate

We conduct a separate set of experiments to study the impact of the amount of synthetic data, for both the source and target sides. From these experiments, we discovered that having more synthetic data does not always increase translation accuracy in Sinhala to Tamil direction. We empirically demonstrate how crucial it is to choose a high-quality NMT system for generating synthetic parallel corpus. For Tamil to Sinhala translation direction, when the ratio between authentic to synthetic parallel sentences were increased, translation performance has continuously improved. However, unlike the Tamil to Sinhala translation, we were unable to obtain a proper translation performance in the opposite direction.

When using higher morphologically rich language as the source language, the NMT architecture encoder performed well. As a result when the source side is more morphologically rich than the target side, the encoder encodes more detail about the sentence, resulting in better decoding by the decoder. The encoded sentences lack sufficient information for the decoder to deduce a successful translation when the source language is less morphologically complex than the target language. We argue that this would be the reason why translations from Tamil to Sinhala are more accurate than translations from Sinhala to Tamil. Additionally, the impact of synthetic data can vary depending on various factors such as the languages involved, data size, and translation direction. In comparison to using only a parallel corpus, incorporating synthetic target data leads to an improvement in source-to-target translation performance. However, the improvement is relatively smaller compared to using source side synthetic data (back translated).

6 Conclusion

We performed an empirical comparison of SMT and NMT in low-resource settings: Tamil-to-Sinhala. Benchmarking common models and establishing best practises are our objectives. This study has demonstrated that, for low-resource data sizes, a proper combination of Transformer configurations together with regularization techniques (Araabi and Monz, 2020) and also with proper vocabulary selection, results in significant improvements when compared with the Transformer system with default settings. Moreover, this research proved the fact suggested by (Duh et al., 2020), in low-resource scenarios, both statistical machine translation (SMT) and neural machine translation (NMT) can work similarly, but in order to get better performance, the neural systems require more careful tuning.

Developing machine translation models for low-resource languages with limited online presence can be challenging, but our preliminary results sug-

gest that even a small amount of parallel data (a few hundred thousand example translations) can make a significant difference when using current neural architectures. Therefore, we believe it is essential to continue pushing the boundaries of finding and curating exploitable parallel text for low-resource languages. In future, NMT system can be improved for low-resource scenarios by experimenting with transfer-learning approaches.

7 Acknowledgement

I would like to express my gratitude to my research supervisors Dr. Randil Pushpananda and Dr. Ruvan Weerasinghe (University of Colombo School of Computing) for their encouragement, support, and valuable feedback. Additionally, I would also like to thank K.T Yasas Mahima for the invaluable assistance provided in structuring the paper.

References

Ali Araabi and Christof Monz. 2020. Optimizing transformer for low-resource neural machine translation. *arXiv preprint arXiv:2011.02266*.

Anupama Arukgoda, A. Weerasinghe, and Randil Pushpananda. 2019. Improving sinhala-tamil translation through deep learning techniques. In *NL4AI@AI*IA*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *25th annual conference on neural information processing systems (NIPS 2011)*, volume 24. Neural Information Processing Systems Foundation.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).

Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. 2002. Choosing multiple parameters for support vector machines. *Machine learning*, 46(1):131–159.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Kevin Duh, Paul McNamee, Matt Post, and Brian Thompson. 2020. Benchmarking neural and statistical machine translation on low-resource african languages. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2667–2675.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Thilakshi Fonseka, Rashmini Naranpanawa, Ravinga Perera, and Uthayasanker Thayasivam. 2020. English to sinhala neural machine translation. In *2020 International Conference on Asian Language Processing (IALP)*, pages 305–309. IEEE.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana. Association for Computational Linguistics.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The flores evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *arXiv preprint arXiv:1902.01382*.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. 2022. Survey of low-resource machine translation. *Computational Linguistics*, 48(3):673–732.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.

W John Hutchins. 1995. Machine translation: A brief history. In *Concise history of the language sciences*, pages 431–445. Elsevier.

Aizhan Imankulova, Takayuki Sato, and Mamoru Komachi. 2017. Improving low-resource neural machine translation with filtered pseudo-parallel corpus. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 70–78.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. 2015. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR.

Shashi Narayan, Nikos Papasarantopoulos, Shay B Cohen, and Mirella Lapata. 2017. Neural extractive summarization with side information. *arXiv preprint arXiv:1704.04530*.

LNASH Nissanka, BHR Pushpananda, and AR Weerasinghe. 2020. Exploring neural machine translation for sinhala-tamil languages pair. In *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 202–207. IEEE.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Jaehong Park, Jongyoon Song, and Sungroh Yoon. 2017. Building a neural machine translation system using only synthetic parallel data. *arXiv preprint arXiv:1704.00253*.

Nghia Luan Pham and Van Vinh Nguyen. 2019. Adapting neural machine translation for english-vietnamese using google translate system for back-translation.

Alberto Poncelas, Maja Popovic, Dimitar Shterionov, Gideon Maillette de Buy Wenniger, and Andy Way. 2019. Combining smt and nmt back-translated data for efficient nmt. *arXiv preprint arXiv:1909.03750*.

Ashmari Pramodya, Randil Pushpananda, and Ruvan Weerasinghe. 2020. 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)*, pages 155–160. IEEE.

Randil Pushpananda and Ruvan Weerasinghe. 2015. Statistical machine translation from and into morphologically rich and low resourced languages. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 545–556. Springer.

Randil Pushpananda, Ruvan Weerasinghe, and Mahesan Niranjan. 2014. Sinhala-tamil machine translation: Towards better translation quality. In *Proceedings of the Australasian Language Technology Association Workshop 2014*, pages 129–133.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. 2023. Neural machine translation for low-resource languages: A survey. *ACM Computing Surveys*, 55(11):1–37.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. *arXiv preprint arXiv:1905.11901*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Pasindu Tennage, Prabath Sandaruwan, Malith Thilakarathne, Achini Herath, Surangika Ranathunga, Sanath Jayasena, and Gihan Dias. 2017. Neural machine translation for sinhala and tamil languages. In *2017 International Conference on Asian Language Processing (IALP)*, pages 189–192. IEEE.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Viraj Welgama, Dulip Lakmal Herath, Chamila Liyanage, Namal Udalamatta, Ruvan Weerasinghe, and Tissa Jayawardana. 2011. Towards a sinhala wordnet. In *Proceedings of the Conference on Human Language Technology for Development*.

Yingce Xia, Di He, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. *arXiv preprint arXiv:1611.00179*.

Guanghao Xu, Youngjoong Ko, and Jungyun Seo. 2019. Improving neural machine translation by filtering synthetic parallel data. *Entropy*, 21(12):1213.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mT5: A massively multilingual pre-trained text-to-text transformer.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.

A Appendix

A.1 Corpus Statistics

A.1.1 Parallel Corpus Statistics

The corpus statistics for the parallel corpus we used in section 4.1 is given in Table 8.

Corpus Statistics	Sinhala	Tamil
Sentence Pairs	26,187	26,187
Vocabulary Size (V)	38,203	54,543
Total number of words (T)	262,082	227,486
V/T %	14.58	23.98

Table 8: Parallel corpus statistics

The various types of resources we investigated for gathering online parallel data, as detailed in Section 4.2, are outlined below:

- **Found Bitext**: Pre-existing parallel sentences could found via various sources such as Opus [3], JW300 [4].
- **Mined Bitext**: Parallel sentences can be mined by crawling the web, for example via Paracrawl [5]. We exploit the fact that various websites exist in multiple languages and devise methods to discover and extract these parallel sentences. We basically focus into Government websites [6] and, online newspapers.
- **Bible**: Studies (Guzmán et al., 2019) have used Bible as a corpus for natural language processing and also for NMT for low resource languages. A parallel corpus of the bible in 100 languages (Tiedemann, 2012) is available online[7]. However, for Sinhala and Tamil it is not available. So we scraped the online bible found in Wordproject Bibles Index[8] which uses the KJV version of English and other languages.
- Text Extraction from sources like Textbooks (provided by educational publications).

[3] http://opus.nlpl.eu/
[4] http://opus.nlpl.eu/JW300.php
[5] https://paracrawl.eu/
[6] https://www.mohe.gov.lk
[7] https://github.com/christos-c/bible-corpus/
[8] https://www.wordproject.org/bibles/index.htm
[9] https://github.com/nlpc-uom/Sinhala-Tamil-Aligned-Parallel-Corpus
[10] https://tico-19.github.io

Corpus	Sentence pairs
Bible	31k
GNOME	8.4k
Ubuntu	4.9k
Open Subtitles	8k
JW300	4M
TextBooks	1.2k
Sinhala Tamil aligned [9]	0.9k
Translation data related to the COVID-19 [10]	0.3k

Table 9: The parallel corpora available online.

A.1.2 Monolingual Corpus Statistics

Corpus Statistics	Sinhala	Tamil
Number of sentence pairs	1,067,173	407,578
Total words	13,158,152	4,178,440
Vocabulary size	933,153	301,251

Table 10: Monolingual Corpus Statistics

A.2 Transformer-base settings

	Transformer-base default parameters
Layers	6
Embedding dimension	512
Heads	8
Feed-forward dimension	2048
Dropout	0.3
Label smoothing	0.1
Batch-size	4096
Warm-up steps	8000
Learning rate	2
BLEU	11.49

Table 11: Default hyper parameters used in Transformer-base