Enhancing Neural Machine Translation for the Sinhala-Tamil language pair with limited resources

Ashmari Pramodya[1], K T Y Mahima[1], Randil Pushpananda[1] and Ruvan Weerasinghe[1]

[1] University of Colombo School of Computing, Sri Lanka

Email: pkashmari1996@gmail.com, yasasm@scorelab.org, {rpn,arw}@ucsc.cmb.ac.lk

***Abstract*—Neural Machine Translation has emerged as a promising approach for language translation. Transformer-based deep learning architectures have also significantly enhanced translation performance across various language pairs. However, several language pairs with limited resources face challenges in adopting Neural Machine Translation because of their data requirements. This study investigates methods for expanding the parallel corpus to enhance translation quality.**

We establish a series of effective guidelines for enhancing Tamil-to-Sinhala machine translation based on cutting-edge Neural Machine Translation techniques like fine-tuning hyperparameters and data augmentation through both forward and backward translation. We validate our methods empirically using standard evaluation metrics. Based on our conducted experiments, we observed that Neural Machine Translation models trained on larger sets of back-translated data outperform other methods of synthetic data generation in Transformer-based training settings. We investigated if we could effectively use the Transformer architecture in the limited-resource context of translating Tamil to Sinhala. Our research demonstrated that Transformer models can surpass the top Statistical Machine Translation models, even in language pairs with limited resources. We achieved an improvement of 3.43 BLEU points in translation quality compared to the statistical translation models.

***Index Terms*—Neural Machine Translation, Low Resourced Languages, Back translation, Hyper-parameters, Sinhala, Tamil**

I. Introduction

Machine translation, initially proposed in 1949 by Hutchins [1], has been largely dominated by Statistical Machine Translation (SMT) models.However, Neural Machine Translation (NMT) using deep learning, as evidenced by [2]–[4], has subsequently emerged and demonstrated promising results in the field of Machine Translation. Recently, NMT tends to employ Transformer [5] 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 aim to investigate the performance of Transformer models on Tamil and Sinhala machine translation. The objective is to establish best practices for low-resource Neural Machine Translation (NMT) in these languages. To address the existing gap in research, we explore various model architectures and hyperparameter tuning methods. We specifically focus on the challenge of insufficient parallel data by expanding the corpus size and assessing the impact of data size on NMT for low-resource languages. Additionally, we study the effects of back translation and forward translation mechanisms in machine translation. To provide a comprehensive assessment, we compare the performance of our Transformer models with Statistical Machine Translation (SMT). This research contributes to the field by filling a current void, offering insights into best practices for Transformer-based models in Sinhala and Tamil 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.

II. Literature Review

A. Neural Machine Translation

NMT systems utilize advanced deep learning methods to translate text, relying on extensive datasets for machine translation. The goal of NMT is to construct and train a single, extensive Artificial Neural Network (ANN) capable of effectively translating languages [2] and it learns the mapping from a source language to its corresponding target language in a complete, comprehensive manner [6].

B. 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 [7], [8].

One of the techniques to improve NMT for low-resource languages is back translation [9]. This involves training a

Correspondence: Ashmari Pramodya (e-mail: pkashmari1996@gmail.com) Received: 01-04-2024 Revised: 15-04-2024

Accepted: 17-04-2024

Ashmari Pramodya , Yasas Mahima , Randil Pushpananda and Ruvan Weerasinghe are from University of Colombo School of Computing , Sri Lanka (pkashmari 1996 @ gmail .com , yasasm @ scorelab .org , rpn@ ucsc.cmb.ac.lk, arw@ucsc.cmb.ac.lk)

DOI: https://doi.org/10.4038/icter.v17i1.7274

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 [9]. 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. [10] solely used synthetic parallel data from both the source and target sides to create the NMT model. Further, according to the Ennrich et al. [9] 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. [11] 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. [12] suggested a method based on sentence similarity score to filter quality synthetic data utilizing bilingual word embeddings [12] and sentence similarity metrics [13]. Further, there are a few possible methods of incorporating the monolingual corpora into machine translation, including Dual learning [14] and unsupervised machine translation using monolingual corpora alone for both sides [15].

C. 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.

There are numerous ways to choose hyper-parameters, most often with manual tuning and random search or grid search [16]. Apart from that, other methods, such as Bayesian optimization [17], genetic algorithms [18], and gradient updates [19] 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.

1) Hyper-parameter Tuning for Low Resource Languages: The difference between low and high-resource NMT is more than parallel data availability. It has been shown that in bilingually low-resource scenarios, Phrase-Based Statistical Machine Translation (PBSMT) models outperform NMT models while for the high-resource the situation is reversed [**?**]. Recently, Sennrich and Zhang [20] re-visited low-resource NMT and showed that low-resource NMT is very sensitive to hyper-parameters, architectural design and other design choices. Unfortunately, their outcomes are limited to a recurrent NMT architecture. Recently in low-resource scenarios Duh et al. [21] findings show that statistical machine translation (SMT) and neural machine translation (NMT) will work similarly, but neural systems require more careful tuning to match performance which they performed there experiments on transformer architecture. Most recently, Araabi and Monz [22] study the effects of hyper-parameter settings for the Transformer architecture under various low-resource data conditions. Their experiments show that a proper combination of Transformer configurations combined with regularization techniques results in substantial improvements over a Transformer system with default settings for all low-resource data sizes. Studies [23] used fewer number of attention heads between 2 and 4, compared to the 8 heads from vanilla transformer.

D. Research in Sinhala Tamil Languages

Sinhala belongs to the Indo Aryan language family, and Tamil belongs to the Dravidian [24]. Both Sinhala and Tamil have a broad morphological vocabulary: There are 110 noun word forms and 282 verb word forms in Sinhala [24], and Tamil has about 40 nominal vocabulary forms and 240 verb forms. Syntactically, the two languages are also close. Using the SMT approach, it was able to get a better result for Sinhala Tamil translation [25].

And NMT has been explored on Sinhala Tamil in only few studies . The most recent research [26] on Improving Sinhala – Tamil Translation through Deep Learning Techniques provided the prominent foundation for Sinhala and Tamil machine translation in NMT by semi-supervised manner using bidirectional recurrent neural networks. This has been conducted for open domain context where Tennage et al. [27] also report works for the NMT using recurrent neural networks for a specific domain. And most recently [28] study use of monolingual word embedding approach for developing the translation in between Sinhala-Tamil language pair only using monolingual corpora. Our attempt is to design a suitable technique for an open-domain translation for such morphologically rich, low-resourced pair of languages by using the transformer architecture

III. Methodology

Here we focus on two main research directions for solving low resource problem: (a) exploring hyper parameters with available less data, and (b) devise methods to exploit additional opportunistic data sources.

A. 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 [29].

Exploring all possible values for so many hyper-parameters at the same time is computationally incredibly expensive. So we will vary the hyper-parameters come under vocabulary representation , architecture tuning and regularization.

Since our baseline SMT study has made use of the same corpora, we were able to make a fair comparison between SMT and NMT in the context of Sinhala and Tamil this way.

B. Corpus Extension

We will focus our attention on two main methods to extend the corpus size.

1) Corpus Extension by Adding Authentic Parallel Data: We explore several resources to collect parallel sentences such as Pre-existing parallel sentences and parallel sentences can be mined by crawling the web.

2) Corpus Extension by Using Synthetic Data: Here we will be using back-translation [9] and Forward Translation, which involves creating artificial source-side sentences by translating a monolingual set in the target language and we will be also using the synthetic data on the target side. [30]. The synthetic data will be generated through two sources namely. Transformer-base, Google translate (GNMT).

3) Back Translation: The back-translation has been used in current state-of-the-art neural machine translation systems [31], outperforming other approaches in high resource languages and improving efficiency in low-resource languages [3], [32]. The approach entails using parallel data set to build a target-to-source (backward) model, which is then used to produce synthetic translations of a large number of monolingual sentences in the target language. The available authentic parallel data is then mixed with the produced synthetic parallel data without distinguishing between the two to train a final source-to-target (forward) model [13]. The quality of the forward translation model depends on the NMT architecture used in building the models, the quality of the backward model [11], the suitability of the synthetic data generation method used [31] and the ratio of the authentic data to the synthetic data.

i) First trained a backward model (Target - Source) using our authentic source language .
ii) Second, translate the target-side monolingual (*mono_target*) data to generate source side synthetic data (*syn_source*)
iii) Then Merge all authentic and synthetic parallel (*syn_source*, *mono_target*) corpora for creating the new data-set.
iv) Finally train final forward model using newly created data-set.

4) Forward Translation: Forward translation (reverse back-translation or self-learning) was used to improve NMT [33]. Forward translation increases the efficiency of a translation model by using source-side monolingual data instead of target-side monolingual data. A source-to-target model is trained using the available authentic data. The available (usually huge) source-side monolingual sentences are then used to produce synthetic translations using this model. This data (synthetic target) is paired with the source-side data to create the synthetic parallel data-set. The resulting huge data is used to train a better source-to-target translation model. One benefit of this technique is that we have clean and original data on the source side to train a better encoder, while on the target side, we have synthetic data, which may cause the decoder to produce ungrammatical translations.

i) First, the forward translation model (Source - Target) is trained with authentic parallel corpus .
ii) Second, the monolingual source monolingual data (*mono_source*) are translated into the target by translation model (*syn_target*)
iii) The monolingual source data and its translations are combined as synthetic corpus (*mono_source*, *syn_target*) and it is merged with authentic data.
iv) Finally train final forward model using newly created data-set.

5) Synthetic Data through Google Translate: To make the most of Google Translate, we combine the back-translation technique with Google Translate to create a parallel corpus for training our translation method. This is an approach that is close to the one suggested by [34].

IV. Experimental Setup

A. Hyper-parameter Exploration

1) Data set: Our baseline training data consists of roughly 25000 sentences which have a sentence length in the range of 8 and 12 words, gathered by Pushpananda et al. [25]. The test set consist of 1000 sentences. The data-set sizes are given in Table I.

Table I: Parallel corpus statistics

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

2) BPE effect: In order to improve the translation of rare words, word segmentation approaches such as Byte-Pair-Encoding (BPE) [35] have become standard practice in NMT. To evaluate the effect of different degrees of BPE segmentation on performance, we consider merge operations ranging from 1k to 10k, training BPE on the full training corpus For that, we used values like 1000, 2000, 5000, and 10000 for Tamil to Sinhala translation as the number of merge-operations. Smaller numbers for merge operations were used because of the small training data condition [36].

3) Architectural Tuning: A current observation in neural networks, and in particular in Transformer architectures, is that increasing the number of model parameters improves performance [37]. However, those results were mainly obtained for scenarios with a significant amount of training data, and it is uncertain if they are specifically relevant to low-resource scenarios. While [38] show that using fewer Transformer layers improves the quality of low-resource NMT, For our random search experiments, we sample the number of attention heads from [1 to 4] and the model dimension from [256,512]. We experimented with various transformer encoders and decoders layers from [2 to 5]. We also sample the size of the feed-forward network (FFN), varying our samples over [1024, 2048].

4) Regularization:: Regularization is used to prevent neural networks from over-fitting and increase generalization capacity. Dropout is an effective regularization strategy introduced by [39]. It is only applied during the training process. Following [20], we analyze the impact of regularization by applying dropouts to Transformer . We also experiment with larger label-smoothing factors. We used label smoothing in the ranges of 0.1 and 0.2, as well as dropout values of 0.2, 0.3 and 0.4.

5) Text-To-Text Transfer Transformer (T5): With the burgeoning of Transfer Learning, Deep Learning has achieved many wonders. More specifically, in NLP, with the rise of the Transformer [5], various approaches for 'Language Modeling' have arisen wherein we leverage transfer learning by pre-training the model for a very generic task and then fine-tuning it on specific downstream problems. By leveraging an unified text-to-text format and a massive training data-set (C4 (Colossal Clean Crawled Corpus)), the original T5 [40] (Text-To-Text Transfer Transformer) model achieved state-of-the-art results on a variety of NLP benchmarks. The mT5 [41] model is a multilingual variant of the original T5 model, aimed at remedying this problem. mT5 closely follows the architecture and the training procedure of T5 but is trained on mC4 (26 Terabytes), a multilingual variant of the C4 data-set. It retains all the advantages of T5, but it also supports a total of 101 different languages.

So we will fine tuned the mT5 model with our data and used it for evaluation of Tamil to Sinhala Translation task.

B. Corpus Extension by Authentic Parallel Data

1) Data set: We explore different types of resources

- Found Bitext: Pre-existing parallel sentences may be found via various sources such as Opus[1] JW300[2]
- Minded Bitext: Parallel sentences can be mined by crawling the web, for example via Paracrawl[3]. The challenge with using this crawled data is that it can be more noisy. 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[4].

  Bible Scraping: Studies [42] have used Bible as a corpus for natural language processing and also for NMT for low resource languages. Here[5] you can find a multilingual parallel corpus created from translations of the Bible. Unfortunately for Sinhala and Tamil it is not available. So we scraped the online bible found in Wordproject Bibles Index[6] which uses KJV version of english and other languages .

  Text Extraction from offline sources like Textbooks (provided by educational publications), and online newspapers.

2) Models: Before combining the gathered parallel datasets, we conducted additional cleaning and removed duplicates. Following this, we trained the datasets independently. Initially, training was done separately, and we assessed their performance. However, the BLEU scores did not exhibit significant improvement across the entire Bible corpus. This may be attributed to the Bible alignment being based on verses rather than sentences. Additionally, the presence of lengthy sentences requiring splitting posed a challenge due to irregular punctuation usage. To address this, we narrowed down the corpus by selecting sentences with lengths between 1 and 20. The test set utilized comprises 10% of the training dataset, ensuring exclusivity. BLEU scores are presented for both Test Sets A and B. Notably, Test Set B excludes Bible sentences from the evaluation set, containing only News Crawl sentences, aligning with the test set used for hyperparameter tuning experiments. In summary, the two test sets employed in our experiments can be outlined as follows:

i) Test Set 1 : 10% of training data
ii) Test set 2 : 1000 sentences (Test set used for baseline systems)

Table II: The parallel corpora available.

Corpus	**Sentence pairs**
Bible	31k
Gnome[7]	0.9k
Ubuntu[8]	5k
Open Subtitles[9]	8k
JW300[10]	4M
TextBooks	0.8k
Sinhala Tamil aligned [11]	0.9k
Translation data related to the COVID-19 [12]	0.3k

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

C. Corpus extension by adding synthetic data

1) Monolingual Corpus: For our experiments we use 10M word Sinhala monolingual corpus and 400,000 word Tamil monolingual corpus [25]. Both these corpora are suitable for an open-domain translation as they have been collected from sentences from different domains such as newspaper articles, technical writing and creative writing. Monolingual corpus statistics are given in Table III.

Table III: Corpus of the Monolingual Data-set

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

2) Models: Here we translate the monolingual Sinhala (*mono_sin*) data to generate corresponding synthetic back-translated Tamil (*syn_ta*) data, building pseudo parallel sentence pair {*mono_sin, syn_ta*}; and translate the monolingual Tamil data (mono_ta) to generate synthetic Sinhala (*syn-sin*), building pseudo parallel sentence pair {*mono_ta, syn_sin*}. Motivated by [43], we continue to create NMT models with increasing sizes of synthetic parallel data to evaluate the effects of back-translated data, using the same training settings with baseline models. We keep adding synthetic data (*syn_sin or syn_ta*) on one side, and corresponding monolingual data on other side to the new pseudo-parallel corpus each time. Under such setting, syn_ta will be used as back-translated data for Tamil → Sinhala translation, and as forward-translated data for opposite direction (Sinhala → Tamil), and so do the syn_sin data, it will be used as back-translated data for Sinhala → Tamil translation, and as forward-translated data for opposite direction,

V. Results and Evaluation

A. Hyper-parameter Exploration

1) BPE effect: To evaluate the effect of different degrees of BPE segmentation on performance, we consider merge operations ranging from 1k to 10k. Reducing BPE merge operations from 10k to 5k improves performance (+2 BLEU).

2) Architectural and Regularization effect: We use Transformer-base and Transformer-big and SMT as our baselines, with the hyper-parameters and optimizer settings described in [5]. We use the OpenNMT-py toolkit [44] for our experiments and multiperl script as evaluation metric.

The hyper-parameters we selected , techniques used were discussed thoroughly in section IV and their values are presented in Table IV. Our random selection of hyper-parameters and their values are dependent on preliminary experiments and previous findings [20], [21], [45] that show which hyper-parameters have the greatest effect on translation efficiency. . For different randomly selected subsets, we obtain significant improvements over Transformer-base.

Table IV: Hyper-parameters considered during the tuning of Transformer

Hyper-parameter	**Values**
Number of Layers in encoder/decoder	2, 3, 4, 5
Attention Heads	1, 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 sets	4000, 8000

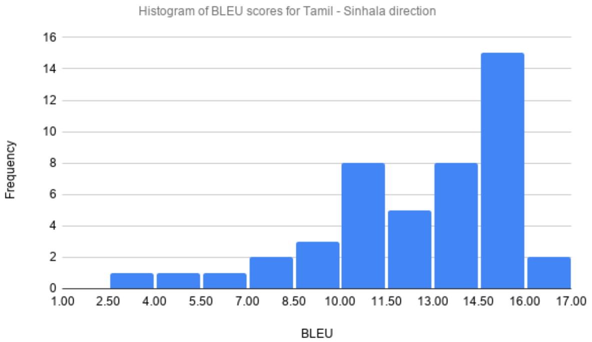


Figure 1: For various NMT models with different hyper-parameters, histogram of test-set BLEU scores.

The best results obtained so far from tuning is shown in Table IV. The models were trained in 15k training steps. To train a model it took 10hrs to 15hrs time. And resulted in 52 models for Tamil to Sinhala translation direction. In Figure 1, we show the distribution of BLEU scores for 52 NMT models with various hyper-parameter settings. Because of the limited time, after getting the best configuration for Tamil to Sinhala direction, we trained a model in Sinhala to Tamil translation direction which resulted in an increase of 2.5 BLEU point over the baseline system. We found that with careful hyper parameter tuning we can outperform SMT results with 3.28 BLEU point for Tamil to Sinhala translation direction using the same training data set as shown in Table VI.

Table V: The best hyper-parameter configurations obtained via random search

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

Model	BLEU Tamil - Sinhala
SMT	13.11
Transformer-base	11.49
Transformer-tuned	16.39
mT5_TA-SI	11.56

Table VII: Comparison of SMT and NMT translated outputs

Test	அதன் பின் அது ஆகாயத்தில் பறக்கத் தொடங்கியது இஸ்ரேலினை நோக்கியே
Reference	අනතුරුව ඒවා ආපසු අහසට නැගුණේ ඊශ්‍රායලය බලා පිටත් වීමටය .
SMT	ඉන්පසුව එය අහස පියාසර කරන්නට පටන්ගත්තේ ඊ ශ්‍ර යල □
Transformer base	ඉන් පසුව එය අහසින් කුරුල්ලන් ආරම්භ වූයේ ඊශ්‍රායලය දෙස ය .
Transfomer tuned	අනතුරුව මේ අහසේ පියාසර කරන්නට පටන් ගත්තේ ඊශ්‍රායලය වෙත ය .
mT5_TA-SI	අනතුරුව එය අහසට නැඹුරු වූයේ ඊශ් රායලය දෙස ය.

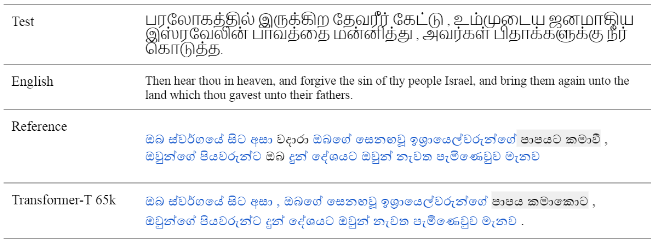
3) Human Evaluation: We used the Human ranking strategy at the sentence level to assess the performance of the models outlined in Table 6. These models were trained with an equivalent volume (25k) of parallel data. For the Human evaluation, we enlisted the participation of 10 final-year undergraduates from the translation studies department at the University of Kelaniya, Sri Lanka. Ten sentences randomly selected from the test set were assigned to each participant, who was then tasked with ranking the translated outputs. We asked participants to rank the sentences from best to worst in order of quality. We did not allow any ties. We followed the guidelines in [46].

Table VIII: 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.

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

According to the rankings in Table VIII, Transformer-Tuned received the highest percentage of top rankings (43%). Additionally, mT5_TA-SI secured the second-highest ranking, with 32%. In contrast, Transformer-Base and SMT were ranked the lowest. Furthermore, when assessing Transformer-Tuned, mT5_TA-SI, and Transformer-Base using BLEU scores, the results align with the human evaluation rankings. However, it's worth noting that the BLEU evaluation scores differ when evaluating all four models. Specifically, the BLEU metric considers synonyms and paraphrases only if they are present in the set of multiple reference translations. Moreover, Neural Machine Translation (NMT) systems inherently capture word similarity, often leading to the inclusion of synonyms in translation outputs. But Sinhala and Tamil language pair does not have the luxury of having multiple references due to been low resourced. This can be reason for having different results for Ranking and BLEU metrics calculations.

B. Increasing the corpus size using authentic parallel data



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 සෙනහවූ ඉශ්‍රායෙල්වරුන්ගේ පාපයට කමාවී , ඔවුන්ගේ පියවරුන්ට ඔබ දුන් දේශයට ඔවුන් නැවත පැමිණෙවුව මැනව
Transformer-T 65k	ඔබ ස්වර්ගයේ සිට අසා , ඔබගේ සෙනහවූ 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)

Table IX illustrates the performance impact of incorporating the Bitext, Paracrawl, and Bible datasets into our initial training set. We observed a significant enhancement in translation for both Tamil to Sinhala and Sinhala to Tamil directions. For instance, on Test Set A, BLEU scores improved by 4.12 points (from 11.49 to 15.61) for Tamil to Sinhala, and by 6.12 points (from 4.98 to 11.10) for Sinhala to Tamil. This positive trend persisted in Test Set B, particularly after adding the biblical corpus. However, the second row of Test Set B revealed a notable performance drop for parallel training data that differed from the evaluation domain. To address this, there may be a need to create a validation set that better matches the evaluation domain or employ domain adaptation techniques for diverse domains to enhance performance and robustness. Our findings suggest that incorporating additional data types is a promising research avenue, especially for Neural Machine Translations (NMTs) dealing with limited resources such as Tamil and Sinhala languages. Figure 1 showcases an example sentence translated from the Bible corpus, demonstrating accurate conveyance of the intended meaning. It's noteworthy that the writing style of Bible verses poses a unique challenge for machine translation, differing from typical news article sentences.

C. Corpus extension using Synthetic parallel data

This section presents the results of the experiments conducted in adding pseudo parallel data to our baseline NMT systems using monolingual corpora described in Table III. Here we evaluate synthetic data in both source and target sides for the machine translation. In this study, we examine about how models work when training data is augmented with synthetic data which was generated using various MT approaches. In particular, we investigate back-translated data generated not only by Transformer-base (our NMT model) but also by Google Neural Machine translate (GNMT) model and combinations of both. Other than backward translation where monolingual corpora were used in the target language, we also investigated on forward translation where monolingual corpora were used in the source language.

Table IX: The effect of additional resource types for NMT. Observe that adding Bible, Text extracted and found-bitext to baseline tends to improve performance for NMT, with NMT gaining significant benefits.

	Data size	Tamil- Sinhala		Sinhala - Tamil	
		Test 1	Test2	Test 1	Test 2
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 X: Results of corpus extension by using synthetic data generated by Transformer-base model

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

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**

From the results shown in Table X and Table XI, adding synthetic data on both sides can improve the performance in the translation direction from Tamil to Sinhala, and all BLEU scores are higher when compared to that built only with authentic data. Surprisingly, in opposite translation direction (Sinhala to Tamil) synthetic data has a significant negative impact on results when the data size is increased. With the addition of synthetic data, BLEU scores increases but when the synthetic data size is 75k BLEU scores have been declining, but they are not lower than the baseline. This can be observed in both approaches we used to generate synthetic data.

However, the quality improvements vary depending on the method used to generate the Synthetic data. We can observe from Table XI, Models built with synthetic data generated by GNMT perform better than those built with data generated by Transformer-base. When comparing models with an equal amount of Transformer base or GNMT-created data, we find that the latter outperforms the former by around two BLEU points. However, Synthetic data have opposite effects on the two translation paths, as seen in the two tables above. More specifically, when translating Tamil to Sinhala, the monolingual synthetic data from both sides have positive effects. Another observation is that models trained with GNMT-created data outperform the best tuned baseline model (16.39). Based on the empirical results obtained in Table 10 the back translation with GNMT is the best performing approach among all other approaches evaluated in this work, forward translation under performs back translation in both approaches we used to create synthetic data.

We further trained NMT models using a parallel corpus composed of a combination of the datasets mentioned in Table IX and synthetic data generated through the two approaches outlined earlier. The final results revealed that adding synthetic data, particularly on the source side, significantly improves performance in the translation from Tamil to Sinhala direction. The inclusion of synthetic data generated through the GNMT approach demonstrated better outcomes compared to the synthetic data generated on both sides.

Table XII shows the improvement of BLEU score with the addition of different data to the baseline (25k) parallel corpus in Tamil to Sinhala translation direction. So our Final model Trained with 65k authentic data + 100k GNMT_syn_ta achieved BLEU score of 20.86 where we were able to outperformed the SMT by 7.75 BLEU points.

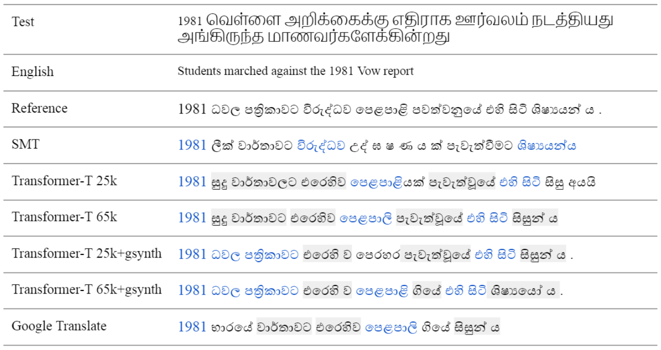
Table XII: Results for Transformer-base and Transformer Tuned for various data sizes in Tamil to Sinhala

System	Data type	Data size	Tamil - Sinhala	
			Base	Tuned
Transformer 25k	Model trained with 25k baseline data	25k	11.49	16.39
Transformer 65k	Model trained with 65k authentic data	65k	14.04	16.06
Transformer 25k+synth	Model Trained with 25k baseline data + 100k GNMT_syn_ta	125k	18.44	19.01
Transformer 65k+synth	Model Trained with 65k authentic data + 100k GNMT_syn_ta	165k	18.59	20.86

D. Analysis on Translated Sentences

To analyze predicted translations by the best performing models, we have considered the best and worst translation examples from our test set. Fig 3 and Fig 4 shows two of the best and and Fig 5 shows worst performance translation example sentences from test set compared to the predicted translation. For best performance translation in sentences, predicted translation is perfectly fluent and adequate and similar to the gold data.

However, for some cases named entity has also been mistranslated, In case of worst performance translation example, the predicted example sentence is completely inadequate from all the models as this is a sentence related to Sinhala literature. But most of the time translated output sentences are very adequate because they give semantically correct meaning with the use of attention mechanism. We observed that NMT outperforms SMT in all most all the cases with our new model.

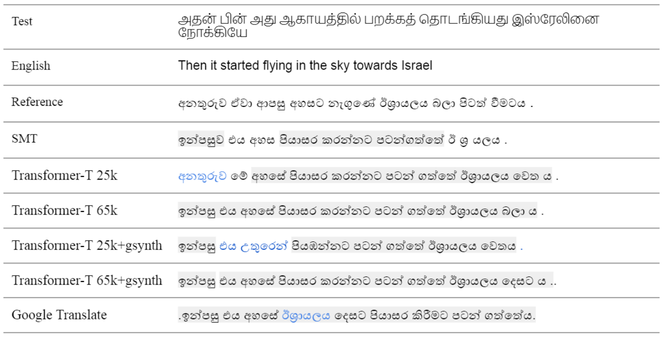


Test	1981 வெள்ளை அறிக்கைக்கு எதிராக ஊர்வலம் நடத்தியது அங்கிருந்த மாணவர்களேக்கின்றது
English	Students marched against the 1981 Vow report
Reference	1981 ධවල පත්‍රිකාවට විරුද්ධව පෙළපාළි පවත්වනුයේ එහි සිටි ශිෂ්‍යයන් ය .
SMT	1981 ලීක් වාර්තාවට විරුද්ධව උද් ස ෂ ණ ය ක් පැවැත්වීමට ශිෂ්‍යයන්ය
Transformer-T 25k	1981 සුදු වාර්තාවලට එරෙහිව පෙළපාළියක් පැවැත්වූයේ එහි සිටි සිසු අයයි
Transformer-T 65k	1981 සුදු වාර්තාවට එරෙහිව පෙළපාලි පැවැත්වූයේ එහි සිටි සිසුන් ය
Transformer-T 25k+gsynth	1981 ධවල පත්‍රිකාවට එරෙහි ව පෙරහර පැවැත්වූයේ එහි සිටි සිසුන් ය .
Transformer-T 65k+gsynth	1981 ධවල පත්‍රිකාවට එරෙහි ව පෙළපාළි ගියේ එහි සිටි ශිෂ්‍යයෝ ය .
Google Translate	1981 හාරයේ වාර්තාවට එරෙහිව පෙළපාලි ගියේ සිසුන් ය

Fig. 3: Translation example with with comparing SMT with NMT systems with different data sizes (gray color highlighted words give semantically correct meaning)Transformer 65k+gsynth is the best performing model. Refer Table XII

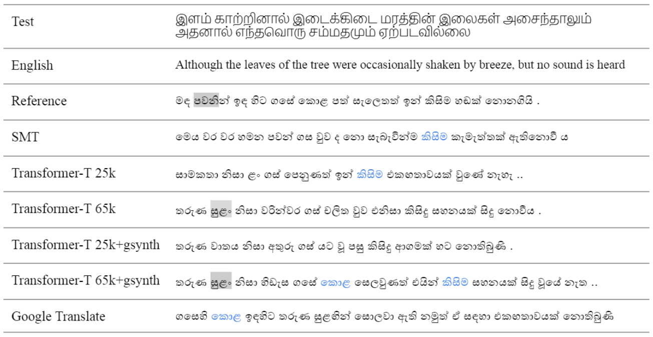
VI. Conclusion

This paper offers a comprehensive study on the low-resource language pair of Sinhala and Tamil using the transformer architecture. Our investigation concentrates on two primary directions: firstly, enhancing translation quality by exploring optimal hyperparameters for the existing baseline dataset (25k), and secondly, improving translation quality by augmenting the dataset size. This research has proved for low-resource data sizes, a proper combination of Transformer configurations together with regularization techniques results in significant improvements over a Transformer system with default settings. And also this research proved the fact suggested by Duh et al. [21], in low-resource scenarios, statistical machine translation (SMT) and neural machine translation (NMT) both can work similarly, but neural systems need more careful tuning to fit performance. Using target synthetic data increases source-to-target translation over using just parallel corpus, but the gains are smaller than adding source side synthetic data (back translated). As the amount



Test	அதன் பின் அது ஆகாயத்தில் பறக்கத் தொடங்கியது இஸ்ரேலினை நோக்கியே
English	Then it started flying in the sky towards Israel
Reference	අනතුරුව ඒවා ආපසු අහසට නැගුණේ ඊශ්‍රායලය බලා පිටත් වීමටය .
SMT	ඉන්පසුව එය අහස පියාසර කරන්නට පටන්ගත්තේ ඊ ශ්‍ර යලය .
Transformer-T 25k	අනතුරුව මේ අහසේ පියාසර කරන්නට පටන් ගත්තේ ඊශ්‍රායලය වෙත ය .
Transformer-T 65k	ඉන්පසු එය අහසේ පියාසර කරන්නට පටන් ගත්තේ ඊශ්‍රායලය බලා ය .
Transformer-T 25k+gsynth	ඉන්පසු එය උතුරෙන් පියඹන්නට පටන් ගත්තේ ඊශ්‍රායලය වෙතය .
Transformer-T 65k+gsynth	ඉන්පසු එය අහසේ පියාසර කරන්නට පටන් ගත්තේ ඊශ්‍රායලය දෙසට ය ..
Google Translate	.ඉන්පසු එය අහසේ ඊශ්‍රායලය දෙසට පියාසර කිරීමට පටන් ගත්තේය.

Fig. 4: Translation example with with comparing SMT with NMT systems with different data sizes (gray color highlighted words give semantically correct meaning)Transformer 65k+gsynth is the best performing model. Refer Table XII



Test	இளம் காற்றினால் இடைக்கிடை மரத்தின் இலைகள் அசைந்தாலும் அதனால் எந்தவொரு சம்மதமும் ஏற்படவில்லை
English	Although the leaves of the tree were occasionally shaken by breeze, but no sound is heard
Reference	මඳ පවනින් ඉඳ හිට ගසේ කොළ පත් සැලෙතත් ඉන් කිසිම හඬක් නොනගියි .
SMT	මෙය වර වර හමන පවන් ගස වුව ද නො සැබැවින්ම කිසිම කැමැත්තක් ඇතිනොවී ය
Transformer-T 25k	සාමකතා නිසා ළං ගස් පෙනුණත් ඉන් කිසිම එකඟතාවයක් වුණේ නැහැ ..
Transformer-T 65k	තරුණ සුළං නිසා වරින්වර ගස් චලිත වුව එනිසා කිසිදු සහනයක් සිදු නොවීය .
Transformer-T 25k+gsynth	තරුණ වාතය නිසා අතුරු ගස් යට වූ පසු කිසිදු ආගමක් හට නොතිබුණි .
Transformer-T 65k+gsynth	තරුණ සුළං නිසා හිඩැස ගසේ කොළ සෙලවුණත් එයින් කිසිම සහනයක් සිදු වූයේ නැත ..
Google Translate	ගසෙහි කොළ ඉඳහිට තරුණ සුළඟින් සොලවා ඇති නමුත් ඒ සඳහා එකඟතාවයක් නොතිබුණි

Fig. 5: Translation example with with comparing SMT with NMT systems with different data sizes (gray color highlighted words give semantically correct meaning) Transformer 65k+gsynth is the best performing model. Refer Table XII

of data increases, we discover that efficiency does not always improve. Furthermore, synthetic data actual effect can vary depending on languages, data sizes, and translation directions. We also discovered that having more synthetic data does not always increase translation accuracy in Sinhala to Tamil direction.

Developing machine translation models for low-resource languages with restricted online representation poses challenges. However, our initial findings indicate that even a modest amount of parallel data (a few hundred thousand example translations) can yield substantial improvements when employing contemporary neural architectures. Hence, we emphasize the importance of persistently pushing the boundaries in discovering and curating exploitable parallel text for low-resource languages. Additionally, in future, the enhancement of NMT systems for low-resource scenarios could benefit from the exploration of transfer-learning approaches as a key component of the improvement strategy.

References

[1] W. J. Hutchins, "Machine translation: A brief history," in *Concise history of the language sciences*. Elsevier, 1995, pp. 431–445.

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

[3] 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.

[4] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[5] 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.

[6] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[7] B. Haddow, R. Bawden, A. V. M. Barone, J. Helcl, and A. Birch, "Survey of low-resource machine translation," *Computational Linguistics*, vol. 48, no. 3, pp. 673–732, 2022.

[8] S. Ranathunga, E.-S. A. Lee, M. Prifti Skenduli, R. Shekhar, M. Alam, and R. Kaur, "Neural machine translation for low-resource languages: A survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–37, 2023.

[9] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[10] J. Park, J. Song, and S. Yoon, "Building a neural machine translation system using only synthetic parallel data," *arXiv preprint arXiv:1704.00253*, 2017.

[11] V. C. D. Hoang, P. Koehn, G. Haffari, and T. Cohn, "Iterative back-translation for neural machine translation," in *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, 2018, pp. 18–24.

[12] G. Xu, Y. Ko, and J. Seo, "Improving neural machine translation by filtering synthetic parallel data," *Entropy*, vol. 21, no. 12, p. 1213, 2019.

[13] A. Imankulova, T. Sato, and M. Komachi, "Improving low-resource neural machine translation with filtered pseudo-parallel corpus," in *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, 2017, pp. 70–78.

[14] Y. Xia, D. He, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma, "Dual learning for machine translation," *arXiv preprint arXiv:1611.00179*, 2016.

[15] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," *arXiv preprint arXiv:1711.00043*, 2017.

[16] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.

[17] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *25th annual conference on neural information processing systems (NIPS 2011)*, vol. 24. Neural Information Processing Systems Foundation, 2011.

[18] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine learning*, vol. 46, no. 1, pp. 131–159, 2002.

[19] D. Maclaurin, D. Duvenaud, and R. Adams, "Gradient-based hyperparameter optimization through reversible learning," in *International conference on machine learning*. PMLR, 2015, pp. 2113–2122.

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

[21] K. Duh, P. McNamee, M. Post, and B. Thompson, "Benchmarking neural and statistical machine translation on low-resource african languages," in *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020, pp. 2667–2675.

[22] A. Araabi and C. Monz, "Optimizing transformer for low-resource neural machine translation," *arXiv preprint arXiv:2011.02266*, 2020.

[23] P.-J. Chen, J. Shen, M. Le, V. Chaudhary, A. El-Kishky, G. Wenzek, M. Ott, and M. Ranzato, "Facebook ai's wat19 myanmar-english translation task submission," *arXiv preprint arXiv:1910.06848*, 2019.

[24] R. Pushpananda, R. Weerasinghe, and M. Niranjan, "Sinhala-tamil machine translation: Towards better translation quality," in *Proceedings of the Australasian Language Technology Association Workshop 2014*, 2014, pp. 129–133.

[25] R. Pushpananda and R. Weerasinghe, "Statistical machine translation from and into morphologically rich and low resourced languages," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2015, pp. 545–556.

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

[27] 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.

[28] 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.

[29] J. Gu, Y. Wang, Y. Chen, K. Cho, and V. O. Li, "Meta-learning for low-resource neural machine translation," *arXiv preprint arXiv:1808.08437*, 2018.

[30] H. Li and H. Huang, "Evaluating low-resource machine translation between chinese and vietnamese with back-translation," *arXiv preprint arXiv:2003.02197*, 2020.

[31] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," *arXiv preprint arXiv:1808.09381*, 2018.

[32] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[33] J. Zhang and C. Zong, "Exploiting source-side monolingual data in neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1535–1545.

[34] N. L. Pham and V. V. Nguyen, "Adapting neural machine translation for english-vietnamese using google translate system for back-translation," 2019.

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

[36] M. Denkowski and G. Neubig, "Stronger baselines for trustable results in neural machine translation," *arXiv preprint arXiv:1706.09733*, 2017.

[37] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, and L. S. Chao, "Learning deep transformer models for machine translation," *arXiv preprint arXiv:1906.01787*, 2019.

[38] E. Van Biljon, A. Pretorius, and J. Kreutzer, "On optimal transformer depth for low-resource language translation," *arXiv preprint arXiv:2004.04418*, 2020.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[40] 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.

[41] 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," 2020.

[42] 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.

[43] A. Poncelas, M. Popovic, D. Shterionov, G. M. d. B. Wenniger, and A. Way, "Combining smt and nmt back-translated data for efficient nmt," *arXiv preprint arXiv:1909.03750*, 2019.

[44] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," *arXiv preprint arXiv:1701.02810*, 2017.

[45] 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.

[46] S. Narayan, N. Papasarantopoulos, S. B. Cohen, and M. Lapata, "Neural extractive summarization with side information," *arXiv preprint arXiv:1704.04530*, 2017.