Using Back-Translation to improve domain-specific English-Sinhala Neural Machine Translation

Koshiya Epaliyana

208038N

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

Department of Computer Science & Engineering

University of Moratuwa
Sri Lanka

June 2021

DECLARATION

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

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

Signature: Date:

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

Name of Supervisor: Dr. Surangika Ranathunga

Signature of the Supervisor: Date:

Name of Supervisor: Prof. Sanath Jayasena

Signature of the Supervisor: Date:

ABSTRACT

Machine Translation (MT) is the automatic conversion of text in one language to other languages. Neural Machine Translation (NMT) is the state-of-the-art MT technique w builds an end-to-end neural model that generates an output sentence in a target language given a sentence in the source language as the input.

NMT requires abundant parallel data to achieve good results. For low-resource settings such as Sinhala-English where parallel data is scarce, NMT tends to give sub-optimal results. This is severe when the translation is domain-specific. One solution for the data scarcity problem is data augmentation. To augment the parallel data for low-resource language pairs, commonly available large monolingual corpora can be used. A popular data augmentation technique is Back-Translation (BT). Over the years, there have been many techniques to improve vanilla BT. Prominent ones are Iterative BT, Filtering, Data Selection, and Tagged BT. Since these techniques have been rarely used on an inordinately low-resource language pair like Sinhala - English, we employ these techniques on this language pair for domain-specific translations in pursuance of improving the performance of Back-Translation. In particular, we move forward from previous research and show that by combining these different techniques, an even better result can be obtained. In addition to the aforementioned approaches, we also conducted an empirical evaluation of sentence embedding techniques (LASER, LaBSE, and FastText+VecMap) for the Sinhala-English language pair.

Our best model provided a +3.24 BLEU score gain over the Baseline NMT model and a +2.17 BLEU score gain over the vanilla BT model for Sinhala → English translation. Furthermore, a +1.26 BLEU score gain over the Baseline NMT model and a +2.93 BLEU score gain over the vanilla BT model were observed for the best model for English → Sinhala translation.

Keywords: Neural Machine Translation, Back-Translation, Data selection, Iterative Back-Translation, Iterative filtering , Low-resource languages, Sinhala

ACKNOWLEDGEMENTS

To start with, I would like to convey my sincere gratitude to my supervisors Dr. Surangika Ranathunga and Professor Sanath Jayasena for the tremendous support and guidance they provided me with, throughout the entire period of the research. I'm grateful for your insights, advice, and encouragement. Without your guidance and support, I could not have achieved this milestone. Your thorough knowledge of Machine Learning, Natural Language Processing, and Deep learning continuously helped me to push myself to learn more, dig deep into study material, and try out new techniques/approaches.

I wish to thank Prof. Gihan Dias for his valuable insights and guidance from the early stage of this research. I would also like to thank both the academic and non-academic staff of the Department of Computer Science and Engineering, for providing me with the resources necessary to conduct my research. This research was supported by the University of Moratuwa AHEAD project Research Grant.

Finally, I would like to give my thanks to my friends and family for all their love and support.

Thank you!

LIST OF ABBREVIATIONS

MT	Machine Translation
NLP	Natural Language Processing
NMT	Neural Machine Translation
SMT	Statistical Machine Translation
BT	Back-Translation
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
RBMT	Rule Based Machine Translation
FDA	Feature Decay Algorithm
INR	Infrequent n-gram Recovery
RCTM	Recurrent Continuous Translation Model
RNNEncdec	RNN Encoder-Decoder

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Background

Neural Machine Translation (NMT) has transpired as the factual solution for Machine Translation (MT), surpassing the Statistical Machine Translation (SMT) techniques. All supervised NMT models fall under the family of encoder-decoder architectures. The basic task of an encoder is to read and encode a sentence into a fixed-length vector whereas the task of a decoder is to output the translation of the input sentence from the encoded vector. Encoder-decoder system is jointly trained on parallel sentences so that given an input sentence, the probability of correct translations could be maximized [1].

Early encoders and decoder models were constructed mainly by using recurrent neural network (RNN) based methods [2, 3] and convolution neural network (CNN) based methods [4, 5]. "Long short-term memory networks (LSTMs) are a type of a Recurrent Neural Network" (RNN) which are known for their capability to learn long-term dependencies. A drawback of early encoder-decoder-based approaches like RCTMs (Recurrent Continuous Translation Models) [1], and RNNEncdec (RNN Encoder-Decoder) [2] is that "the network needs to compress all the necessary information of the source sentence into a fixed-length vector." To diminish this issue, Bahdanau et al. [6] proposed an enhancement to the encoder-decoder model which is a cable of learning to align and translate at the same time. In this technique, the decoder generates each word based on the context vectors associated with the relevant words in the source sentence and the words the decoder has previously generated. This dynamic calculation of the context vector has been obtained by an attention mechanism [6].

One of the main drawbacks of recurrent models is their sequential nature which causes issues like preventing parallelization between training examples which be-comes worse with longer sequence lengths since memory constraints limit batching across examples. These problems can be alleviated by Transformer models [7] which are more parallelizable. Transformers steer clear of recurrence and use only attention mechanisms to model dependencies by replacing the recurrent layers with multi-headed self-attention (Recurrent layers are most frequently used in encoder-decoder architectures). Hence, the encoders and decoders in Transformers were built using self-attention network (SAN) based methods (Attention mechanism helps to pass information between the encoder-decoder pair effectively in both directions. In addition to the aforementioned property, self-attention helps the encoder encode the sequence much more efficiently).

1.2 Research Problem

For an NMT model to perform well, a large parallel corpus is needed. For high-resource language pairs such as German-English and French-English, finding parallel corpora is not difficult. However, for low-resource language pairs (low-resource language is a language that lacks a unique writing system, lacks (or has a limited) a presence on the World Wide Web, lacks linguistic expertise specific to that language, and/or lacks electronic resources such as parallel and monolingual corpora, vocabulary lists, etc [8]), finding parallel corpora between the two languages is a challenge. Building parallel corpora for Machine Translation(MT) in low-resource languages is both time-consuming and expensive since professional translators are limited as there are very limited bilingual speakers of these languages. For MT in restricted domains such as official government documents, the problem is even more severe. In other words, if the translation task is domain-specific, the challenge is even harder. Sinhala is a morphologically rich but low-resource language and does not have the luxury of large parallel datasets [9]. However, compared to the limited amounts of parallel data, the Sinhala-English language pair has large monolingual corpora for an open domain such as Wikipedia and news.

To alleviate the difficulties in finding large parallel corpora for NMT, data augmentation can be used. In data augmentation, largely available monolingual corpora are used to enlarge the number of parallel sentences available for training. One such data augmentation method is Back-Translation (BT) [10].

Back-Translation is the process of translating a monolingual corpus in the target language by an already existing MT system, in the reverse translation direction, into the source language. Then the obtained synthetic source language sentences along with their respective target language sentences are used to construct a synthetic parallel corpus, which is then added to the already existing parallel corpus to form an augmented parallel data set. Back-Translation is a language and architecture-independent data augmentation technique, which can be used in both NMT and SMT.

The translation models trained with additional synthetic parallel data tend to contain novel words, which implies that using additional synthetic source sentences and monolingual target sentences can improve the word-level fluency of MT systems [10]. Basic Back-Translation has issues with respect to the quality of the synthetic parallel corpus generated. Low-quality synthetic data could degrade the performance of NMT systems acutely [11]. Hence, methods to improve the quality of back-translated data have been introduced. We have identified 4 main approaches to improve the performance of BT: iterative BT [12], filtering [13], data selection [14, 15] and Tagged BT [16, 17].

The aforementioned four approaches individually improve the quality of BT models, mainly for low-resource language pairs. However, how they would work for Sinhala-English language pairs and how they would perform when they are combined are not yet known.

1.3 Research Objectives

The objectives of this research are as follows:

1. Implement a Back Translation algorithm for Sinhala-English, by improving and combining the following existing techniques to further improve Back Translation;

   (a) iterative BT

   (b) filtering

   (c) data selection

   (d) tagged BT

2. Utilize the constructed synthetic data to improve Sinhala-English NMT over the baseline models.

3. Empirical evaluation of different sentence embedding techniques for filtering sentences in Sinhala-English (which are translations of each other).

1.4 Contributions

We made the following contributions to this thesis:

1. Used Back-Translation to improve the translation performance of domain-specific Sinhala-English NMT models.

2. Empirically showed that by combining Data selection, Filtering, and Iterative Back-Translation, we could achieve better results than the existing techniques for further improving BT.

3. Achieving better results than the existing techniques lead to setting a new baseline for Sinhala-English domain-specific machine translation.

4. Experimental evaluation of the following sentence embedding techniques for filtering sentences in Sinhala-English (which are translations of each other):

   (a) FastText combined with VecMap

   (b) LASER

   (c) LaBSE

1.5 Publications

- **Koshiya Epaliyana**, Surangika Ranathunga, Sanath Jayasena ”Improving Back-Translation with Iterative Filtering and Data Selection for Sinhala-English NMT” 2021 Moratuwa Engineering Research Conference (MERCon) ***(Accepted)***.

1.6 Organization

The remainder of this thesis is organized as follows. Chapter 2 consists of the previous work done on Back-Translation and different approaches to improving Back-Translation. Chapter 3 presents the methodology we followed to build our Machine Translation models. Chapter 4 describes the experimental setup, data, and experimental details. Chapter 5 discusses the results of our NMT models obtained using training on augmented parallel data. We conclude this document with a discussion of the results obtained and our observations.

Chapter 2

LITERATURE SURVEY

In this section, we discuss previous research on Back-Translation as well as the techniques proposed by various researchers on improving Back-Translation. We discuss Iterative BT, Filtering, and Data selection approaches such as Transductive Data selection, selecting difficult to predict words, and selecting data closer to the target domain. In addition to the aforementioned approaches, we also discuss methods like Incrementally filtered BT, Tagged BT, Iterative Tagged BT, Sampling, using both source-side and target-side monolingual data, Noised BT, and using a pivot language. All these techniques have contributed to improving Back-Translation. Hence, we discuss and critically analyze these techniques to identify the best ones amongst them.

2.1 Basic Back-Translation

Back-Translation is the process of translating a monolingual corpus in the target language by an already existing MT system, in the reverse translation direction, into the source language. Back-Translation was first introduced by Sennrich et al. [10]. They have applied BT between a high-resource language pair as well as a low-resource language pair. The gains from BT were higher for the high-resource language pair than for the low-resource language pair. However, Poncelas et al. [18] have empirically shown that BT is likely to give better results for low-resource scenarios. They have also shown that when the size of the training corpus increases, the performance also increases not only for the model trained only on authentic parallel data but for the model trained only on synthetic parallel data and the model trained on both authentic and synthetic (hybrid) data. Figure 2.1 is a depiction of the Back-Translation process proposed by Sennrich et al. [10].

Even though Back-Translation improves over basic NMT models, BT itself has limitations and drawbacks. As observed by Poncelas et al. [18], the coverage

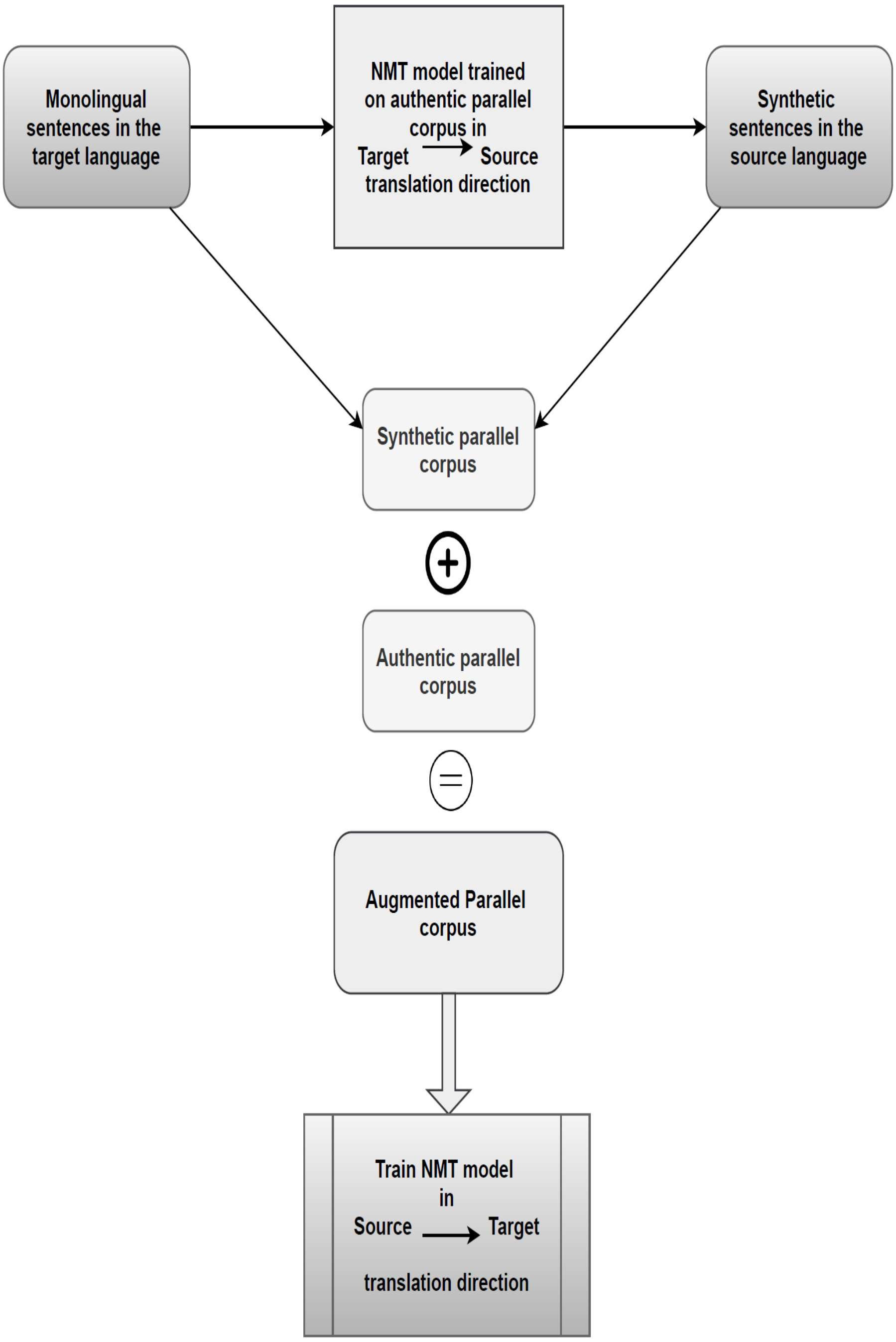

Monolingual sentences in the target language
NMT model trained on authentic parallel corpus in
Target
Source
translation direction
Synthetic sentences in the source language
Synthetic parallel corpus
Authentic parallel corpus
Augmented Parallel corpus
Train NMT model in
Source
Target
translation direction

Figure 2.1: Back-Translation process

("the percentage of tokens (words, numbers, and other characters) of the test set that are covered by the vocabularies which are used to build the NMT models") has decreased with the size of the synthetic parallel data. The reason for this is, that when more synthetic data are added, the more its vocabulary starts to dominate. The vocabulary of synthetic data is restricted compared to authentic data. In addition to the restricted vocabulary, synthetic sentences also introduce translation errors of the NMT system that has been used to translate the monolingual corpus. These affect the performance of the NMT system adversely causing the performance to deteriorate based on the nature of the synthetic corpus. Hence, various research has proposed different techniques to further improve Back-Translation by mitigating its sub-optimal nature. Out of all these techniques, four take dominance; Iterative BT, Filtering, Data selection and Tagged BT.

2.2 Iterative Back-Translation

Iterative BT has been used as an approach to further improve Back-Translation [12, 19, 20]. Figure. 2.2 is a depiction of Iterative BT. It is the process of training the two Back-Translation systems (target → source and source → target) multiple times. After obtaining the Back-translated NMT model in the source → target direction, that model is used to back-translate the monolingual source sentences in the source → target translation direction and vice versa. This process is carried out iteratively until no improvement is observed in the performance of the NMT models.

Hoang et al. [12] have shown that more iterations cause much better translation quality in Back-Translated models. The synthetic parallel corpus generated through those models also were of higher quality. Hence, it is evident that, with the number of iterations, translation performance has also improved. However, Cotterell et al. [21] have shown that out of out-of-domain and in-domain data, the gains have been observed earlier(at iteration-2) for in-domain data since the original parallel data, monolingual data and test data originate from the same

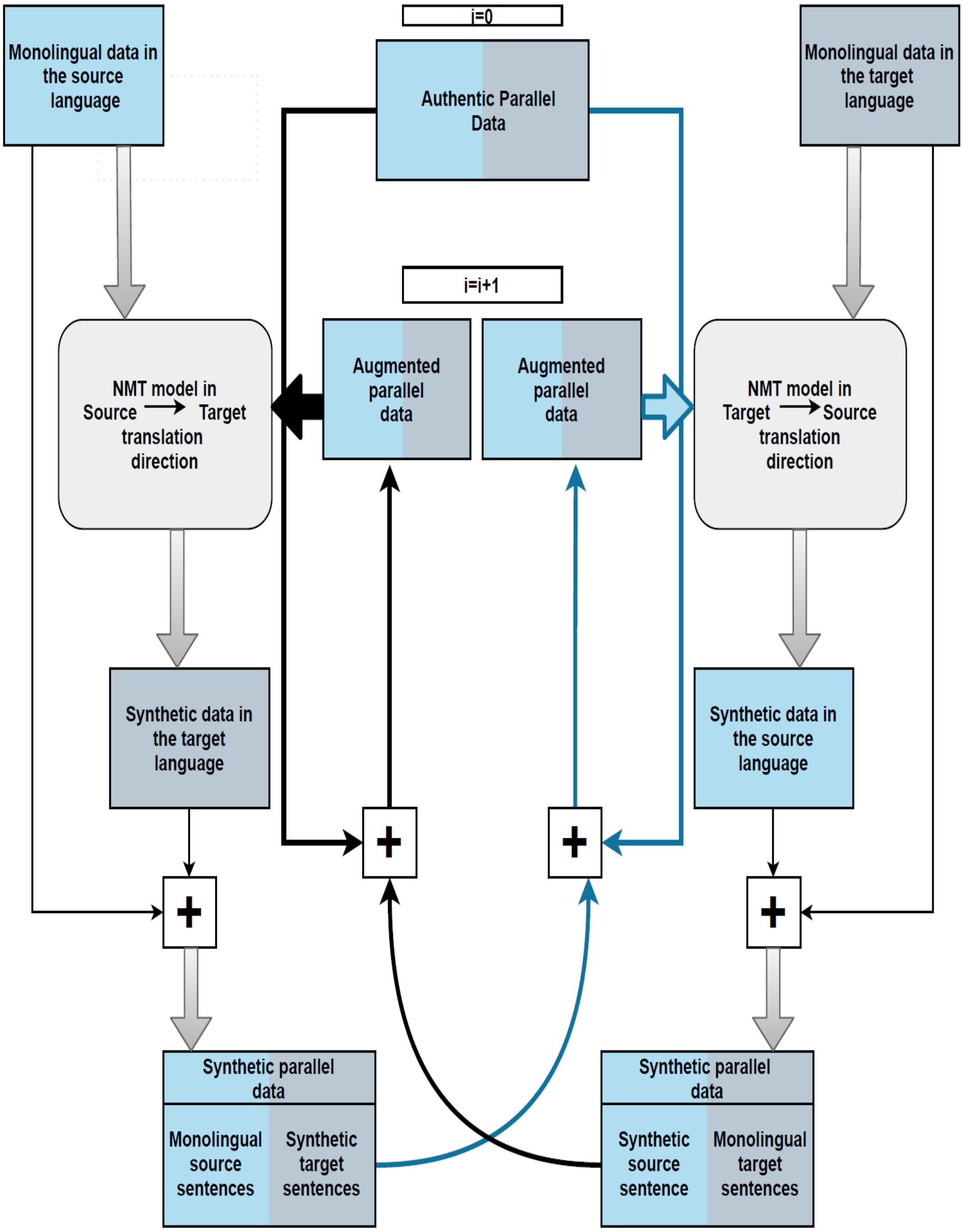

Monolingual data in the source language
i=0
Authentic Parallel Data
Monolingual data in the target language
i=i+1
NMT model in Source ⟶ Target translation direction
Augmented parallel data
Augmented parallel data
NMT model in Target ⟶ Source translation direction
Synthetic data in the target language
Synthetic data in the source language
+
+
+
+
Synthetic parallel data
Monolingual source sentences
Synthetic target sentences
Synthetic parallel data
Synthetic source sentence
Monolingual target sentences

Figure 2.2: Iterative Back-Translation process

domain. In addition to the number of iterations, as observed by both Hoang et al. [12] and Cotterell et al. [21], for in-domain data, performance gains from Iterative BT have been higher for high-resource language pairs than that for low-resource language pairs. Cotterell et al. [21] have further experimented with out-of-domain data for the high-resource language pair. Out of all the in-domain and out-of-domain experiments, the gains have been the highest in the out-of-domain approach (at iteration-3) due to Back-Translation enabling adaptation to the new domain.

Artetxe et al. [19] have experimentally shown that Iterative BT tends to converge to similar outcomes/results regardless of the initial MT model used. To elaborate more, despite the initial system used; RBMT (Rule-Based Machine Translation), Supervised NMT, Supervised SMT, and Unsupervised SMT, the final system obtained after Iterative BT is more or less the same.

Furthermore, Abdulmumin et al. [20] have introduced batch Back-Translation, which is an enhancement to the standard Iterative BT where only batches of monolingual data have been used. They claim that using the entire monolingual corpus at once degrades the forward model by introducing a lot of noise. Furthermore, they explain that "the noise (bad signals) from the preceding iteration supposedly overwhelms the good signals of the subsequent iterations." Hence, the impact of the noise has been reduced by reducing the number of sentences to be back-translated at each iteration [20].

2.3 Data Selection

Data selection refers to selecting the most appropriate monolingual data to be translated to create synthetic data. Appropriate monolingual data refers to sentences that are closer to the domain of interest than other sentences in the large monolingual corpus. In addition to that, data selection also refers to selecting target monolingual data using prediction loss i.e. sentences with words that have been identified as hard to predict words.

2.3.1 Transductive Data Selection algorithms

One of the possible Data Selection methods is Transductive learning where the data is selected based on the test set which consists of texts to be translated. One of the main features of Transductive learning is that one has already come across both training and testing datasets before training the model. Hence, Ponselas et al. [14, 15] have introduced two Transductive data selection algorithms (TDA) called Feature Decay Algorithm (FDA) and Infrequent n-gram Recovery (INR) which have selected data by using the test set as seed and have retrieved those sentences that are comparatively closer to this seed than the others.

Feature Decay Algorithm (FDA) [14, 15]:
FDA uses n-grams of both the $test_{src}$ and the monolingual corpus to retrieve sentences from the large monolingual corpus that are most similar to the $test_{src}$.

- Takes a set of monolingual sentences K as input and the seed (in this experiment, the test set)
- Given K and the seed ($test_{src}$), FDA retrieves an organized sequence of sentences T from K.
  - Sentences are ordered according to the number of n-grams they share with the seed.
  - The higher the number of shared n-grams, the higher the preference.
- The algorithm initializes T as an empty sequence and iteratively selects one sentence $s \in K - T$ and appends it to T.
  - The sentence s selected at each step is based on the number of n-grams that s shares with the $test_{src}$.
- The score(s) of each sentence is computed as follows.

$$score(s) = \frac{\sum_{m \in n_s} 0.5^{K_n(m)}}{N} \tag{2.1}$$

  - n_s is the set of n-grams present in both $test_{src}$ and K. N is the number of words in s
  - $K_n(m)$ is the count of n-gram m in the sequence T
  - Including $K_n(m)$ in the formula makes the algorithm penalize n-grams which have been selected several times. Thus, favoring sentences containing new n-grams.

Incremental n-gram recovery algorithm (INR)[15]

- It extracts sentences containing n-grams (which are also found in the test set) that are considered infrequent.
  - Hence, words such as stop words are ignored.
- If the count of an n-gram in the selected pool of sentences is higher than a threshold, that n-gram does not contribute to the scoring of the sentence it contains.
- If the computed score of the sentence is larger than a threshold (which is selected by observing the scores of the sentences), the sentence is selected and added to the pool of selected sentences.
- The sentences in the candidate data W are scored according to the following equation.

$$score(s, W) = \sum_{ngr \in S_{test}} max(0, t - K_{SI+C}(ngr)) \tag{2.2}$$

  - t is the threshold that indicates whether an n-gram is frequent or not
  - $K_{SI+C}(ngr)$ is the count of the n-gram ngr in the selected pool C (an in-domain set SI is used for initialization).

As evident in Equation 2.1 and Equation 2.2, both FDA and INR penalize n-grams that are too frequent. FDA penalizes too frequent n-grams by diminishing their contribution to the score of the sentence whereas INR penalizes n-grams(recurrent ones) by not allowing them to contribute to the score of the sentence at all. Out of these two techniques, INR has performed better than FDA. The reason might be its strict penalizing approach.

2.3.2 Selecting sentences consisting of difficult to predict words

Apart from Transductive Data selection, Back-Translation can be further improved using another technique which selects data from a monolingual corpus by focusing on 'hard to predict' words. Fadaee et al. [22] have empirically shown that by selecting sentences containing difficult to predict words or words with high prediction losses, the translation quality of a system can be improved. Wang et al. [23] have also carried out experiments to select monolingual data by targeting difficult words. It has been observed that difficult to predict words mostly benefit from additional back-translated data. By oversampling sentences consisting of difficult-to-predict tokens the effectiveness of using back-translated data can be improved [22]. When it comes to difficult words with high prediction losses, by providing more sentences consisting of difficult words, the model's estimation has improved and the model's uncertainty in prediction has reduced [22].

2.3.3 Selecting target domain data

In addition to Transductive data selection and selecting sentences with difficult to predict words, selecting data closer to the target domain (domain of the training and test data) has also been proposed by previous research to improve Back-Translation. Sennrich et al. [10] have identified that selecting in-domain target monolingual data for BT reduces overfitting and better modeling of fluency. This caters to domain adaptation. Niu et al. [24] have applied cross-entropy difference to select pseudo-in-domain data from both in-domain and out-of-domain data. Artetxe et al. [19] have indicated that using domain-specific monolingual data to back-translate could help domain adaptation. Abdulmumin et al. [25] point out that fine-tuning a pre-trained model on in-domain data improves the quality of the BT model. The dynamic data selection proposed by Dou et al. [26], has selected subsets of sentences from a set of monolingual sentences by gradually shifting from selecting general domain data to target domain data at each training epoch.

2.4 Filtering

Filtering techniques aim at filtering out low-quality synthetic sentences that could degrade the performance of the NMT system. The filtering process involves computing the similarity score between a synthetic sentence and its monolingual counterpart using a sentence-level similarity metric. If this similarity score is above a certain threshold value, the sentence pair is added to the filtered synthetic parallel corpus. Previous research has used several sentence-level similarity metrics that fall under two broad categories: those that only use surface information of sentences when calculating the similarity and those that use distributed representations of sentences.

2.4.1 Sentence-level similarity metrics using surface information of sentences

- Sentence-level BLEU (Sent-BLEU): It is the BLEU score of each sentence pair; computed using the monolingual target sentence as the reference and synthetic target sentence as the candidate [27, 28].

  To generate the synthetic target sentence, synthetic source sentences are translated back to the target language. (This is called a round-trip translation)

2.4.2 Sentence-level similarity metrics using distributed representations of sentences

These metrics use the vector representations of words and sentences.

- Average alignment similarity (AAS): "The average cosine similarity between vectors of all words in monolingual and synthetic target sentences" [29]

- Maximum alignment similarity (MAS): "The average of the cosine similarity between the most similar word from the monolingual target sentence and each word from the synthetic target sentence and the cosine similarity between the most similar word from the synthetic target sentence and each word from the monolingual target sentence" [29]

- Sent-BiEmb: "The sentence-level cosine similarity between sentence embeddings of the synthetic source sentence and monolingual target sentence" [30]

2.4.3 Comprehensive analysis of different filtering techniques

Imankulova et al. [31] have used only sentence-level BLEU (Sent-BLEU) as the sentence-level similarity metric whereas Imankulova et al. [13] have extended this work([31]) by performing extensive experiments using three different sentence-level similarity metrics: Sent-BLEU, AAS, and MAS. In addition to those, they have used a sentence-level language model (sent-LM) to filter a corpus by taking only synthetic source sentences into account. All these sentence-level similarity metrics (Sent-BLEU, AAS, and MAS) use round-trip translations to generate synthetic target sentences. Xu et al. [30] have used both Sent-BLEU and Sent-BiEmb as sentence-level similarity metrics whereas Jaiswal et al. [32] have only used Sent-BiEmb.

Imankulova et al. [13] have used Word2vec model [33] to generate word embeddings to calculate AAS and MAS metrics. While Xu et al. [30] have used FastText [34] and VecMap model [35] to generate sentence embeddings, Jaiswal et al. [32] have used Multilingual Universal Sentence Encoder (MUSE) [36]. For each sentence, first, Xu et al. [30] have created word embeddings using FastText and obtained the sentence embedding vector by averaging the accumulated word vectors to form a single mean vector representation. Afterward, they used the VecMap model to ensure that the sentence embeddings of the monolingual target sentence and the respective synthetic source sentence are located in the same vector space.

Xu et al. [30] have shown that out of Sent-BiEmb and Sent-BLEU, the former is more effective than the latter for filtering noise in synthetic data. The reason could be that in Sent-BiEmb, "semantic information of words in both synthetic and monolingual sentences are considered by using both source and target word embeddings" whereas, in Sent-BLEU, only the target sentences are considered.

Even though Xu et al. [30] used filtered synthetic parallel data to augment the parallel data and eventually train the NMT model, Jaiswal et al. [32] used the filtered parallel data to fine-tune the open-domain base model (trained on the publicly available parallel corpus from different domains).

Both Imankulova et al. [31] and Imankulova et al. [13] have observed that filtering improves the performance of low-resource language pairs than high-resource language pairs. The improvements across different scoring metrics have been consistent with negligible differences for low-resource language pairs [13]. Hence, proving that filtering significantly impacts low-resource language pairs.

Arukgoda et al. [37] have introduced a method called incrementally filtered BT. In incrementally filtered BT, source → target and target → source translations are done in parallel. In other words, in this approach, iterative BT is performed in both translation directions simultaneously with an additional step of filtering synthetic parallel data and adding the filtered synthetic parallel corpus to the authentic parallel corpus. The performance improvement obtained in one translation direction is used to improve the performance of the NMT model in the other translation direction.

As we investigated more into the sentence-level similarity metrics which use vector representations, we came across different sentence embedding models. Here, we discuss a few sentence embedding models which claim to be giving the best results in different NLP tasks. These sentence embedding models have different architectures and it is necessary to do a comparative analysis of these techniques since one could get an idea of why various techniques create different sentence embeddings for the same sentence pair which are translations of each other.

2.4.4 Sentence Embedding techniques

Word embeddings are vector representations of words where words with similar meanings have similar representations. Many techniques have been introduced over the years to generate word embeddings. Word2Vec [38] is a pioneering technique in generating word embeddings that provides two models for computing word representations: skip-gram and CBOW (i.e continuous-bag-of-words). "The CBOW model predicts the current word from a window of neighboring context words whereas the skip-gram model uses the current word to predict the neighboring window of context words." FastText[1] [34] is another word embedding technique which is an extension of Word2Vec model. It works very well on a variety of languages by leveraging the morphological structure of the language. Furthermore, FastText incorporates subword information which has proven to be effective in morphologically rich languages. FastText has released pre-trained word vectors for 294 languages that have been trained on Wikipedia data.

However, FastText is not capable of building multilingual word representations (multilingual word vectors). Hence, to map two vectors generated by FastText (in two different languages) to the same Vector space, the VecMap model can be used. It is a framework to learn cross-lingual word embedding mappings[2] [35]. VecMap offers 4 main modes: Supervised, Semi-supervised, Identical, and Unsupervised.

LASER is a toolkit built for multilingual sentence representations which compute multilingual sentence embeddings for zero-shot cross-lingual transfer[3] [39]. "LASER generates embeddings for a set of languages together in a single shared space rather than having a separate model for each language." Similar sentences are mapped to close vectors despite the input language. Hence, LASER is known to be a model that generates language-independent sentence vectors. The latest version of LASER provides an encoder that has been trained in 93 languages and written in 28 different scripts. The same BiLSTM encoder is used to encode all these languages. Hence, it is not necessary to point out the input language.

[1] https://fasttext.cc/

[2] https://github.com/artetxem/vecmap

[3] https://github.com/yannvgn/laserembeddings

However, tokenization is language-specific.

LaBSE (Language-agnostic BERT Sentence Embedding) is a multilingual BERT embedding model proposed by Feng et al. [40] which generates language-agnostic cross-lingual sentence embeddings for 109 languages. It has been effective in low-resource languages as well, although no data were available for training. Since this is a multilingual embedding model it maps text from different languages into a shared embedding space like LASER. "This model is trained and optimized to generate similar vector representations for bilingual sentence pairs which are translations of each other." It has a dual encoder architecture where the source and target text are encoded separately using a shared transformer embedding network. This model can be differentiated from a word-level embedding model like FastText as this model takes word sequence into account rather than just individual words[4] [40].

XLM-RoBERTa (XLM-R)[5] is a transformer-based multilingual masked language model pre-trained on unlabeled text in 100 languages proposed by Conneau et al. [41]. XLM-R is a self-supervised model which performs well even for low-resource languages. It is capable of training a single model for many languages while withholding per-language performance. XLM-R is trained the same way as RoBERTa but on a large multilingual dataset. It only uses the Masked Language Modeling (MLM) objective like RoBERTa [42] avoiding the Translation Language Modeling (TML) objective (which is used by XLM [43]).

2.5 Tagged BT

Caswell et al. [16] have experimentally shown that Tagged BT outperforms standard BT and noised BT for both high-resource and low-resource language pairs. Tagged Back-Translation(TaggedBT) is prepending a <BT> tag to each synthetic sentence in the source language. Through Tagged BT, rather than adding noise to data, they have tried another way to signal the model that the source side of the synthetic parallel data is back-translated. This would eventually allow the

[4] https://tfhub.dev/google/LaBSE/1

[5] https://github.com/facebookresearch/XLM

model to treat the synthetic parallel data differently than the original/authentic parallel data [16, 17]. Furthermore, Marie et al. [17] have empirically shown that although Tagged BT prevents the translation quality of original texts from diminishing, Tagged BT also struggles with translationese (manually translated) texts. However, by tagging test sentences in addition to synthetic source sentences, the performance can be boosted for translationese texts [17].

Caswell et al. [16] have also shown that Iterative Tagged BT improves over Tagged BT with each iteration outperforming Iterative standard BT. Standard BT has not improved over each iteration. With Tagged BT, the model is given the liberty to bootstrap more effectively from the back-translated data whilst not being damaged by the quality concerns. However, standard BT models do not have the ability to distinguish between synthetic and authentic sentences and are often misled by the fluctuation of quality of BT data [16].

2.6 Other approaches

Apart from the above-mentioned approaches, we came across other techniques which have contributed to improving Back-Translation. In this section, we discuss and compare these various techniques which have empirically shown to improve Back-Translation.

2.6.1 Using both target-side and source-side monolingual data

Both Wu et al. [44] and Niu et al. [24] have used not only the target-side monolingual data but also source-side monolingual data in BT. Wu et al. [44] have experimentally shown that using target-side monolingual data only has performed better than using source-side monolingual data only. However, better results have been obtained by using a combination of both target-side and source-side monolingual data. Although Wu et al. [44] have used separate source → target and target → source NMT models to back-translate monolingual target sentences and monolingual source sentences respectively, Niu et al. [24] have used a bi-directional model to translate source and target monolingual data. This bi-directional NMT model has been trained on both directions of a language pair jointly which has reduced the overall computing resources significantly in comparison to training an individual model for each language direction.

By combining all synthetic parallel data (generated from source and target monolingual data) for bi-directional models, improvements have been observed over standard BT for low-resource settings. Nonetheless, for high-resource settings, no improvements have been observed by the bi-directional model over the uni-directional models [24]. Bi-directional models have consistently reduced the training time by 15 -30%.

2.6.2 Noised Back-Translation

Adding noise to Back-Translated data has also been tried out by Edunov et al. [45] and Caswell et al. [16]. Both research have used noised beam BT where they have added noise to beam search outputs. Edunov et al. [45] have empirically shown that noised beam BT outperforms the original parallel data only model, and standard BT using pure beam and greedy methods. Furthermore, Caswell et al. [16] have experimentally shown that Noised BT outperforms standard BT in high-resource settings but fails to do so in low-resource settings. The reason for noise + beam to work well is that noisy source sentences make target translations harder to predict which eventually helps to learn [45].

Wu et al. [44] have also shown that adding noise to synthetic parallel sentences improves the performance over standard BT. One of the key differences from the aforementioned research ([45, 16]) has been using both source-side and target-side synthetic data. Hence they have not only added noise to source-side synthetic sentences but also to source-side monolingual sentences. Their standard BT model has also been trained on original parallel data and synthetic parallel data consisting of both source-side and target-side synthetic sentences (also their respective monolingual sentences). Furthermore, by fine-tuning the noised models with clean synthetic data combined with original parallel data, the performance has further improved.

2.6.3 Sampling

Edunov et al. [45] have empirically shown that sampling is better than beam and greedy methods to generate synthetic source sentences. This claim has been confirmed by Wang et al. [23] who have observed that sampling is better than beam. Not only beam and greedy, sampling too has outperformed the original parallel data only model as well. The reason why sampling performs better than pure beam and greedy methods is that sampling better approximates the data distribution. Hence providing a better training signal than the others [45]. In addition to that, the results have shown that sampling is more effective than beam in high-resource settings whereas beam is more effective in resource-poor settings.

2.6.4 Using a pivot language

Currey et al. [46] have factually proven that adding pivot-language monolingual data improves zero-resource NMT performance. Zero-resource translation has started from a multilingual NMT system and has improved the zero-shot direction using the synthetic parallel corpus. They have back-translated the monolingual pivot data into both language A and language B using the multilingual NMT model. For each sentence in the monolingual corpus C, its translations in language A and language B have been obtained and the synthetic parallel corpus $A' \leftrightarrow B'$ has been constructed. Models obtained by training on original parallel data combined with the generated synthetic parallel corpus (using monolingual data in pivot language) have outperformed the best direct translation model for a rather high-resource language pair. In addition to that, models obtained by only fine-tuning them using original parallel data combined with synthetic parallel corpus have also improved over the performance of the best direct translation model [46].

2.6.5 Training the model on synthetic data and fine-tuning on authentic data

Abdulmumin et al. [25] have outperformed standard BT by training the model on the synthetic data and fine-tuning the model on authentic data. This approach has aimed to enable the model to learn efficiently through pre-training and fine-tuning. Moreover, this model converges earlier than other models, thus requiring less time to train.

2.6.6 The impact of the size of the monolingual corpus on Back-Translation

Some of these research has investigated the impact of the size of the synthetic data on the performance of the machine translation model. Hence, they have conducted experiments with different sizes of synthetic data to determine the effectiveness of adding more/less synthetic data. Various other factors also contribute to the impact of the size of the synthetic data. Hence, different research has witnessed distinct results. However, in most cases, when the size of the synthetic data has been much larger than the size of the authentic data, the performance of the machine translation system has dropped.

Xu et al. [30], Abdulmumin et al. [25] and Fadaee et al. [22] have empirically shown that the ratio between synthetic to authentic data depends on the language pair as well. Xu et al. [30] have trained models for a low-resource language pair where models which have been trained on authentic to synthetic ratio 1:5 have outperformed the ratio 1:1 by a large margin. The 1:10 ratio of real-to-synthetic data has performed best in the experiments. Abdulmumin et al. [25] have also observed that performance improves up to a ratio between 1:5 for a different low-resource language pair.

However, Fadaee et al. [22] have shown that the model trained on a 1:4 ratio of authentic to synthetic data has achieved the best results whereas the model trained on a 1:10 ratio has performed not well compared to the former. These models were trained for high-resource language pairs. Poncelas et al. [18]

have also observed that the performance has diminished after the ratio between authentic to synthetic was 1:2 for a high-resource language pair.

Abdulmumin et al. [25] have observed that for Tagged BT, the scores have gradually risen from a ratio of 1:1 between authentic to synthetic data to a ratio of 1:3. And then, the performance dropped slightly when the ratio increased up to 1:5. These experiments have been conducted for a low-resource language pair. This shows that even though in Tagged BT, authentic and synthetic data have been explicitly differentiated, the model might not have been able to distinguish between authentic and synthetic data completely while training.

These results brought us to the conclusion that there is no definite ratio between authentic to synthetic training data and it depends on the language pair, the domain match (whether the authentic data are in the same domain as synthetic data), and the underlying NMT architecture.

2.7 Summary

Even though the aforementioned existing literature have improved Back-Translation using various approaches, none of them has combined 3 or more of these techniques. Iterative BT [12, 19, 20], Filtering [30, 31, 13, 32] and Transductive Data selection algorithms [14, 15] have not been combined together by any of these previous research to improve Back-Translation. In addition to those, none of the research has combined Tagged BT/Iterative Tagged BT [16, 17] and Filtering [30, 31, 13, 32]. None of these previous research on Filtered BT has done a comprehensive analysis of multilingual sentence embedding techniques like LASER, LaBSE, and FastText+VecMap.

Chapter 3

Methodology

We identified Filtering, Iterative Back-Translation, Data selection and Tagged BT as four vital contributors to further improving Back-Translation. Previous research has used these approaches individually to improve Back-Translation. However, none of the previous research has combined three or more of these methods to enhance the performance of Back-Translation. Hence, we conducted experiments on these methods separately and then combined them together to evaluate their effectiveness. First, we formed vanilla BT models for Sinhala → English and English → Sinhala translation directions. Then, we constructed Iterative BT models by iteratively executing the Back-Translation process in both Sinhala → English and English → Sinhala translation directions.

We also combined Back-Translation with filtering (various experiments were carried out with different multilingual embedding techniques to generate sentence embeddings) and built Filtered BT models for English ↔ Sinhala. We also did a comprehensive analysis of different sentence embedding techniques used for filtering. Next, we combined Iterative BT with filtering and formed Iterative Filtered BT models for English ↔ Sinhala. Finally, we combined Data selection with Iterative Filtered BT using Transductive Data selection algorithms on both monolingual Sinhala corpus and monolingual English corpus. In addition to the above-mentioned experiments, we also generated NMT models using Tagged BT by prepending a tag to the synthetic source sentences. We further improved the performance of the models obtained by Tagged BT by executing Iterative Tagged BT, Tagged BT with filtering and Iterative Tagged BT with filtering. Our research process is depicted in Figure 3.1.

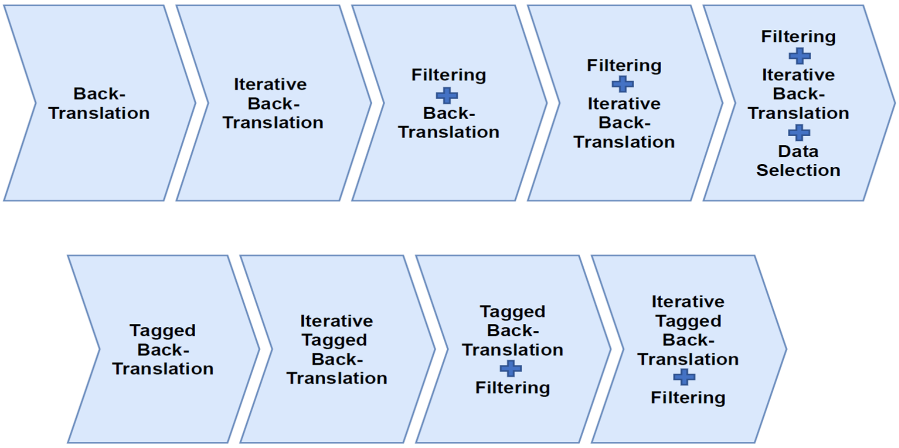

Back-Translation
Iterative Back-Translation
Filtering + Back-Translation
Filtering + Iterative Back-Translation
Filtering + Iterative Back-Translation + Data Selection
Tagged Back-Translation
Iterative Tagged Back-Translation
Tagged Back-Translation + Filtering
Iterative Tagged Back-Translation + Filtering

Figure 3.1: Research process

3.1 Vanilla Back-Translation

First, we trained NMT models on the original parallel data in Sinhala → English translation direction and English → Sinhala translation direction. To generate synthetic sentences in the English language, we took the monolingual Sinhala corpus and translated it to English using Sinhala → English NMT model. Then the generated synthetic English sentences and their respective monolingual Sinhala sentences were used to construct a synthetic parallel corpus which was added to the already existing authentic parallel corpus. Next, using this augmented parallel corpus, an NMT model in English → Sinhala translation direction was trained. Thus, creating the vanilla BT model for English → Sinhala translations.

To generate a Back-Translated NMT model for Sinhala → English direction, the monolingual English sentences were translated to Sinhala using the English → Sinhala NMT model trained only on authentic parallel data. The generated synthetic Sinhala sentences and their monolingual English counterparts were used to construct a synthetic parallel corpus. These parallel data were added to the authentic parallel data to form an augmented parallel corpus. Then a Sinhala → English NMT model was trained on the augmented parallel data. Thus, building the vanilla BT model for Sinhala → English translations.

In addition to the aforementioned vanilla BT models trained on in-domain synthetic and authentic parallel corpora, we also tried to determine the impact of the size of the synthetic parallel corpus on the NMT model, we experimented with different sizes of monolingual corpora. Since we couldn't find many monolingual sentences in the 'official government document domain' we used News data crawled from the web. By using monolingual corpora in the News domain, we study the impact of domain adaptation in Back-Translation. Since the authentic parallel data were from the 'official government document domain' and monolingual corpora were from a different domain we got to investigate the impact of the combination of in-domain and out-of-domain data on Back-Translation.

3.2 Iterative Back-Translation

For the Sinhala-English language pair, the monolingual English corpus was back-translated using the English → Sinhala NMT model, and the monolingual Sinhala corpus was back-translated using the Sinhala → English NMT model. The generated synthetic corpora, which contained 'monolingual English sentences and synthetic Sinhala sentences' were added to the authentic parallel corpus to train the NMT model in Sinhala → English translation direction. The other corpora consisting of 'monolingual Sinhala sentences and synthetic English sentences' were added to the authentic parallel corpus to train the NMT model in English → Sinhala translation direction. This process was performed iteratively on the Back-translated NMT models obtained from the previous iteration in both translation directions until no improvements were observed in the BLEU scores for both translation directions.

3.3 Filtered BT

After Back-Translation, the generated synthetic Sinhala sentences and their respective monolingual English sentences were filtered to form the filtered synthetic parallel corpus. For Sinhala → English translation direction, the Filtered Back-Translated NMT model was trained on augmented parallel corpus constructed by combining authentic parallel data along with the generated filtered synthetic parallel corpus.

Likewise, for English → Sinhala translation direction, English synthetic sentences and their Sinhala monolingual counterparts were filtered to form a filtered synthetic parallel corpus which was then added to the original parallel corpus to construct an augmented parallel corpus. This augmented parallel corpus was used to train the NMT model in the English → Sinhala direction.

To filter sentence pairs, the 'Sent-BiEmb' sentence-level similarity metric was used as proposed by Xu et al. [30] since it has proven to perform better than the 'Sent-BLEU' similarity metric. Even though Xu et al. [30] used FastText embeddings along with the VecMap model, we also used the LASER embeddings [39] and LaBSE embeddings [40] as they are both multilingual embedding models pre-trained with a large data set and support the Sinhala language. Moreover, we experimented with 6 threshold values to find the best threshold value for each MT model.

We used FastText embeddings to create word embeddings of each word in a sentence and then the sentence vector was obtained by averaging the accumulated word vectors. These sentence embeddings of English and Sinhala languages did not share the same vector space. To locate them in the same vector space we used the VecMap model as used by Xu et al. [30]. Since LASER and LaBSE are multilingual sentence embedding models we did not have to use the VecMap model to map the sentence embeddings of Sinhala and English languages into the same vector space. To compute the similarity score between the two sentence embeddings, cosine similarity was used. If the cosine similarity between the sentence embeddings of the synthetic source sentence and the respective monolingual target sentence was above a certain threshold, the sentence was selected for the filtered synthetic parallel corpus. Cosine similarity between two vectors S_x and S_y is computed according to the following formula.

$$Score(S_x, S_y) = \frac{S_x \cdot S_y}{|S_x| \cdot |S_y|} \tag{3.1}$$

Filtered BT for some sample data is depicted in Figure 3.2. It demonstrates the process for English → Sinhala translation direction.

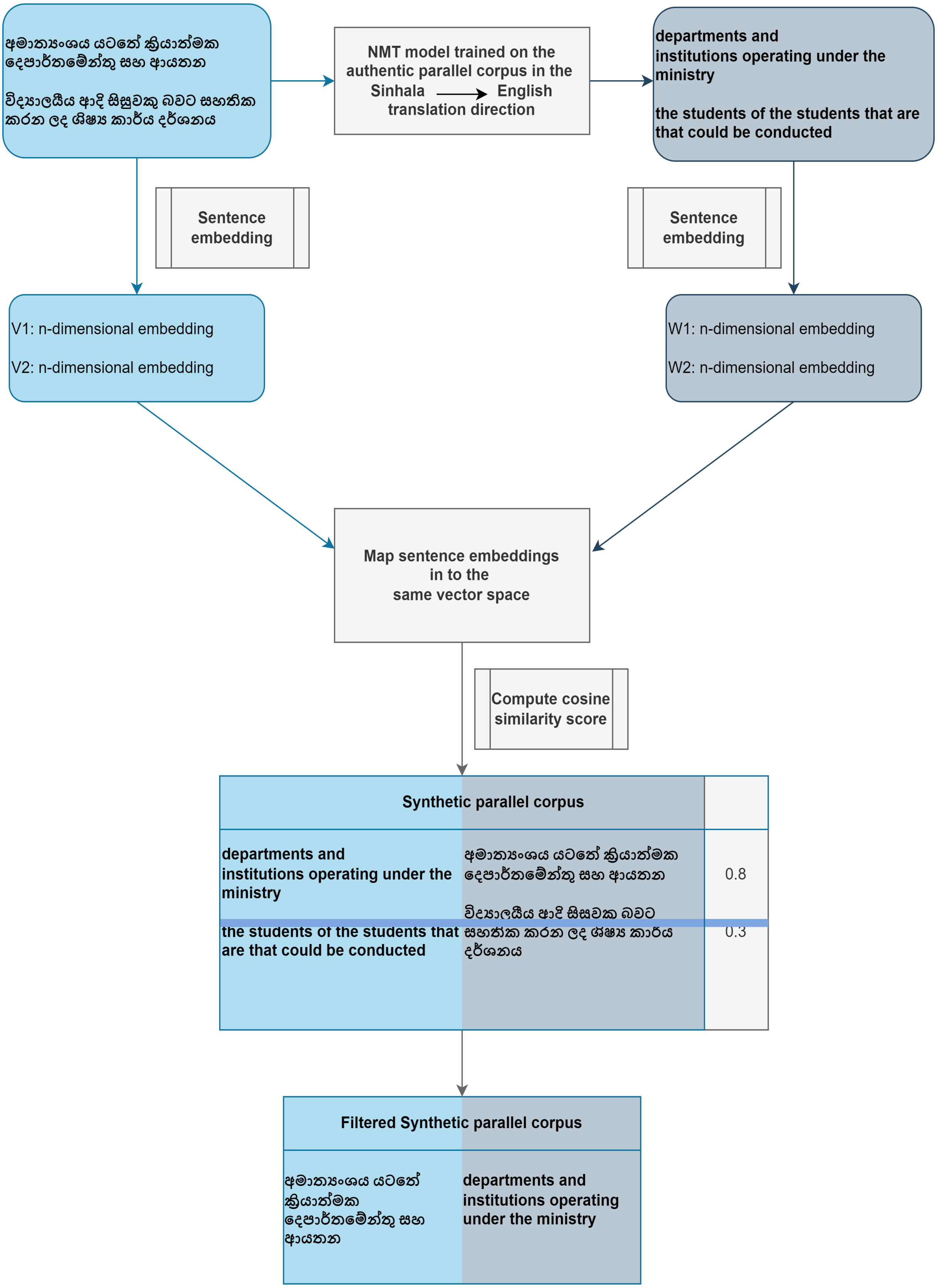

අමාත්‍යංශය යටතේ ක්‍රියාත්මක දෙපාර්තමේන්තු සහ ආයතන
විද්‍යාලයීය ආදි සිසුවකු බවට සහතික කරන ලද ශිෂ්‍ය කාර්ය දර්ශනය
NMT model trained on the authentic parallel corpus in the Sinhala → English translation direction
departments and institutions operating under the ministry
the students of the students that are that could be conducted
Sentence embedding
Sentence embedding
V1: n-dimensional embedding
V2: n-dimensional embedding
W1: n-dimensional embedding
W2: n-dimensional embedding
Map sentence embeddings in to the same vector space
Compute cosine similarity score
Synthetic parallel corpus
departments and institutions operating under the ministry
අමාත්‍යංශය යටතේ ක්‍රියාත්මක දෙපාර්තමේන්තු සහ ආයතන
0.8
the students of the students that are that could be conducted
විද්‍යාලයීය ආදි සිසුවක බවට සහතික කරන ලද ශිෂ්‍ය කාර්ය දර්ශනය
0.3
Filtered Synthetic parallel corpus
අමාත්‍යංශය යටතේ ක්‍රියාත්මක දෙපාර්තමේන්තු සහ ආයතන
departments and institutions operating under the ministry

Figure 3.2: En → Si Filtered BT process

3.3.1 Iterative Filtered Back-Translation

Iterative Filtered BT is the combination of Iterative BT and filtering [37]. With iterative filtering, we filter the back-translated data and train the NMT system with the filtered synthetic parallel corpus iteratively. In other words, the back-translated data are filtered using the filtering algorithm and then the filtered parallel data are added to the original parallel corpus at each iteration. The difference between Iterative BT and Iterative Filtered BT is that in Iterative Filtered BT, before adding to the original parallel data at each iteration, the synthetic parallel sentences are filtered to form a filtered parallel corpus.

We chose the best NMT model from the models trained on original parallel data along with filtered synthetic parallel data which were filtered using FastText+VecMap, LASER, and LaBSE embeddings. These models were trained on data filtered by different threshold values. Then starting from the picked model, Iterative Filtered BT was performed while filtering data by the same threshold value as the initial best model. Hence, at each iteration, different pairs of sentences (monolingual target and synthetic source sentences) were filtered out. Assuming that the performance of NMT models improves at each iteration, the quality of the synthetic data also improves. We execute Iterative Filtered BT until no improvements in the performance of the NMT models were observed.

3.4 Data selection

Transductive data selection algorithms were used for data selection. Feature Decay Algorithm (FDA) was implemented to pick sentences with a score above a certain threshold where sentences are scored based on the n-grams (here we used unigrams) common in both the training data (seed) and the particular sentence. The higher the number of n-grams common with the seed, the higher the score of the sentence. However, the scoring formula penalizes n-grams that are too frequent in the selected pool of sentences.

Even though Poncelas et al. [15] used a test set as the seed for both INR and FDA, we used the training set as the seed as we assumed that the former introduces a bias since the same test set is used for evaluation. Moreover, since online processing has given better results for Poncelas et al. [15], we also used online processing where the training data (authentic parallel data) in the source language was translated to the target language using source → target NMT model. These translated data were used as the seed set.

Before feeding into FDA and INR algorithms, we translated the Sinhala training data to English and the English training data to Sinhala. Then, we fed the Sinhala monolingual corpus with the translated English training data (now in Sinhala) and the English monolingual corpus along with translated Sinhala training data (now in English) to the TDAs. For each monolingual sentence, if the score of the sentence is greater than the threshold, the sentence was picked from the selected pool of sentences.

3.5 Iterative Filtered BT with Data selection

After Iterative Filtered BT was over, we pick the best model for each translation direction from the models obtained at each iteration. Then we translated the source-side training data to the target language using the picked best models. After obtaining the translated corpora as the seed sets(in target language), we ran the FDA and INR algorithms for both Sinhala and English monolingual corpora. Then the monolingual Sinhala sentences selected by the TDAs were translated to the source language using the best Sinhala → English Iterative Filtered Back-Translated NMT model. Then the generated synthetic parallel corpus was added to authentic parallel data to construct the augmented parallel corpus which was used to train the NMT model in English → Sinhala translation direction.

To train a finer model in the other translation direction, monolingual English sentences selected by the TDAs were translated to the Sinhala language using the best English → Sinhala Iterative Filtered BT model. Then these synthetic parallel sentences were added to the authentic parallel data to form an augmented parallel corpus. Next, Sinhala → English NMT model was trained on the augmented parallel data. The NMT models obtained after this entire process for English →

Sinhala and Sinhala → English translation directions are the models attained by combining Iterative BT, Filtering, and Data Selection.

3.6 Tagged Back-Translation

Tagged Back-Translation is the process of prepending a tag to the synthetic sentences after Back-Translation. By tagging the synthetic sentences, Caswell et al. [16] states that a signal to the model is sent that the source side is back-translated. Thus, allowing the decoder to distinguish between authentic and synthetic parallel data.

After Back-translating the monolingual Sinhala corpus and monolingual English corpus, the obtained synthetic English and Sinhala corpora were tagged. Tagged synthetic sentences were obtained by adding the tag $< BT >$ before each sentence in the synthetic corpus. An example of a tagged synthetic sentence is shown in Table 3.1.

Synthetic source sentence	*Tagged Synthetic source sentence*
survey for lease agreement plan	<BT> survey for lease agreement plan

Table 3.1: **Tagged synthetic sentence**

3.6.1 Iterative Tagged Back-Translation

Iterative Tagged Back-Translation is, iteratively executing the Tagged BT process until no improvements can be observed in both translation directions. We tagged synthetic Sinhala and English sentences after they were back-translated from monolingual English and Sinhala sentences. Then, when the tagging was complete, the tagged synthetic source sentences and their respective monolingual target sentences were added to the authentic parallel corpus to construct an augmented parallel corpus. These newly formed parallel data were then used to train new Sinhala → English and English → Sinhala NMT models.

It can be described that the Iteration-n Tagged-BT model is trained on Back-Translated data generated by the Iteration-(n-1) Tagged-BT model. This entire process was executed iteratively till no improvements were observed in the performance of the Sinhala → English and English → Sinhala NMT models.

3.6.2 Tagged BT with Filtering

After back-translating the monolingual Sinhala corpus, we filtered the generated synthetic parallel corpus before tagging the synthetic English sentences. Then, we added the tagged synthetic English sentences and their respective monolingual Sinhala sentences to the authentic parallel corpus. Finally, we trained the English → Sinhala NMT model with the augmented parallel corpus. For the opposite translation direction, after back-translating the monolingual English corpus we filtered the generated synthetic parallel sentences before tagging the synthetic Sinhala sentences. Next, we added the tagged synthetic Sinhala sentences and their respective monolingual English sentences to the authentic parallel corpus which was used to train the NMT model in Sinhala → English translation direction. We used LASER embeddings as the sentence embedding technique with different threshold values.

3.6.3 Iterative Tagged BT with Filtering

The main difference between this method and Iterative Tagged BT is, filtering the generated synthetic parallel corpus before adding the tags to the synthetic source sentences.

First, we pick the threshold value which gave the best model for Tagged BT with Filtering for each translation direction. We used it as the threshold value in our proceeding experiments. At each iteration after monolingual target sentences were back-translated to the source language, we filtered the generated synthetic parallel corpus using the picked threshold and tagged the filtered synthetic source sentences. Then the generated tagged synthetic source sentences and their respective monolingual target sentences were added to the original parallel corpus. This augmented parallel corpus was used to train a model in source → target translation direction.

The above-mentioned process was executed on both monolingual Sinhala and English data where at each iteration, monolingual Sinhala sentences were translated by the Sinhala → English NMT model in the previous iteration and vice versa. This iteration process was carried out until no improvement could be observed in both Sinhala → English and English → Sinhala translation directions.

Chapter 4

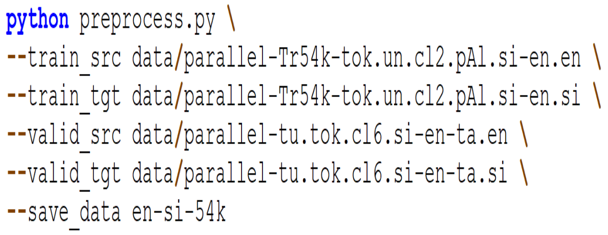
EXPERIMENTS

4.1 Setup

We used OpenNMT-py[1] for our experiments on Google colab pro with T4 and P100 GPUs with access to high-memory VMs. We used the BLEU score as our evaluation metric.

4.2 Baseline NMT model

We used an encoder-decoder network with a 2-layer bi-directional Long-Short Memory Network (LSTM) as the encoder and an LSTM as the decoder. For each experiment, we pre-processed both the source and target side training data following the pre-processing steps in OpenNMT-py to create word dictionaries. Then, using the generated dictionaries, serialized files were created for training and development sets. In addition to that, we tuned the hyper-parameters with the development set. The script we used for pre-processing in English → Sinhala translation direction is depicted in Figure 4.1.



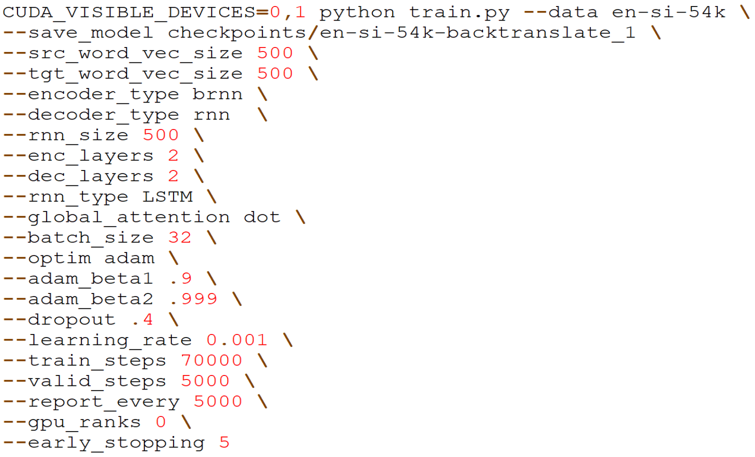
```
python preprocess.py \
--train_src data/parallel-Tr54k-tok.un.cl2.pAl.si-en.en \
--train_tgt data/parallel-Tr54k-tok.un.cl2.pAl.si-en.si \
--valid_src data/parallel-tu.tok.cl6.si-en-ta.en \
--valid_tgt data/parallel-tu.tok.cl6.si-en-ta.si \
--save_data en-si-54k
```

Figure 4.1: Pre-processing script

After that, we trained the network with an early stopping criteria with a patience of 5 valid steps. During the inference phase, we used a beam search of 5 beams. For ensembling, we used both checkpoint ensembling and model ensembling by training 4 models; saving checkpoints for each model. A part of

[1] https://github.com/OpenNMT/OpenNMT-py

the script we used for training (for only one model saving checkpoints) is depicted in Figure 4.2. Then, we selected the top 4 models from the saved checkpoints based on their results on the validation set; which were used as the **Ensemble** model.



```
CUDA_VISIBLE_DEVICES=0,1 python train.py --data en-si-54k \
--save_model checkpoints/en-si-54k-backtranslate_1 \
--src_word_vec_size 500 \
--tgt_word_vec_size 500 \
--encoder_type brnn \
--decoder_type rnn  \
--rnn_size 500 \
--enc_layers 2 \
--dec_layers 2 \
--rnn_type LSTM \
--global_attention dot \
--batch_size 32 \
--optim adam \
--adam_beta1 .9 \
--adam_beta2 .999 \
--dropout .4 \
--learning_rate 0.001 \
--train_steps 70000 \
--valid_steps 5000 \
--report_every 5000 \
--gpu_ranks 0 \
--early_stopping 5
```

Figure 4.2: Training script

The Baseline NMT models in English → Sinhala and Sinhala → English translation directions are trained as mentioned above. Since Back-Translation does not require the existing model architecture to change, all the other models were trained and evaluated the same way. The only difference was the training data used to train each model since the synthetic parallel data changed with each technique and the translation direction.

We used three sentence embedding techniques for Filtered Back-Translation.

1. The first sentence embedding technique we used was FastText+VecMap. The first step in generating sentence embeddings was to generate word embeddings using FastText. We trained new FastText models using our Sinhala and English monolingual corpora. To train the models, we picked the skip-gram model over CBOW since it is better at capturing semantic relationships [47]. Moreover, we used hierarchical softmax for model training as it speeds up the training process. We set the dimensionality to 300 because it captures more information rather than setting it to a smaller value. We used 12 worker threads with 5 iterations through each corpus to

train each model.

(a) To map the sentence embeddings to the same vector space since FastText is not a multilingual embedding model, we used the VecMap model. We used the Unsupervised mode since we did not have a seed dictionary and did not want to rely on identical words.

2. The second sentence embedding technique we used was LASER. We used 'laserembeddings' to generate sentence embeddings for English and Sinhala languages. We directly used the pre-trained embedding model since it has been trained on both Sinhala and English data and the model has performed very well in multilingual similarity search for high-resource and low-resource languages alike. To generate sentence embeddings, we fed Sinhala and English sentences which were translations of each other to the LASER embedding model along with their language tags 'si' and 'en', and it generated the sentence embedding pair as the output.

3. The third and the last sentence embedding technique we used was LaBSE. We used the pretrained LaBSE embedding model because it has been trained on both English and Sinhala corpora and it has outperformed LASER on certain NLP tasks such as cross-lingual text retrieval. The sentence embeddings for Sinhala and English languages were generated separately.

4.3 Data

The data we used for the experiments demonstrate extremely low-resource (i.e. parallel data in Sinhala-English languages are very scarce) and domain-specific settings. We used parallel training, validation and test data in the 'official government documents' domain [48].

We constructed a monolingual English corpus and a monolingual Sinhala corpus in the same domain as parallel corpora. After back-translating these monolingual corpora, we pre-processed them by removing pairs that contain text in the same language on both the source and target sides, as well as pairs where

	Size
Train	74,468
Validation	1,623
Test	1,603

Table 4.1: **Parallel Data for Si-En**

the same sentence was present on both the source and target sides. Sentences with empty translations were also removed. Moreover, sentences containing only special characters or numbers were removed as well. After pre-processing, we obtained new synthetic parallel corpora (Synthetic Sinhala - Monolingual English and Synthetic English - Monolingual Sinhala) for both translation directions. We used the monolingual sentences in the target languages from the obtained parallel corpora to construct a new monolingual English corpus and a new monolingual Sinhala corpus which we used in all our experiments.

Language	**Condition**	**Domain**	**Size**
Sinhala	Raw	in-domain	53,735
English	Raw	in-domain	53,093
Sinhala	Pre-Processed	in-domain	44,115
English	Pre-Processed	in-domain	42,773

Table 4.2: **Monolingual Data**

In addition to the in-domain monolingual corpora, we also constructed out-of-domain monolingual corpora from the News data crawled from the web. The English and Sinhala monolingual corpora generated from these crawled documents were comparable. We used these monolingual corpora to investigate the impact of the authentic to synthetic data ratio on Back-Translation. Hence, the size of the monolingual Sinhala and English corpora changed according to the ratio values.

4.4 Experimental details

The vanilla BT models were obtained by first back-translating the monolingual English and Sinhala data into source languages. Then we combined the obtained

Language	Ratio	Domain	Size
Sinhala / English	1:1	out-of-domain	74,468
Sinhala / English	1:2	out-of-domain	148,936
Sinhala / English	1:3	out-of-domain	223,404
Sinhala / English	1:4	out-of-domain	297,872

Table 4.3: **Monolingual Data for different ratios in the News domain**

synthetic parallel corpora with the authentic parallel data and trained the NMT models in Sinhala → English and English → Sinhala translation directions with the generated augmented parallel corpora.

To evaluate the effectiveness of Iterative BT, we iterated the entire Back-Translation process in both Sinhala → English and English → Sinhala translation directions until no improvements were observed in the BLEU scores for both the translation directions.

For data selection, we used FDA and INR algorithms with a threshold value of 0.7 for both algorithms. The threshold was picked after observing the scores of all the sentences in the monolingual corpora for both FDA and INR algorithms. To combine Data Selection with Iterative BT, the selected data were back-translated by the best model obtained so far with Iterative BT. Finally, we trained NMT models on the obtained synthetic parallel corpora combined with authentic parallel data.

We used filtering using LASER embeddings, LaBSE embeddings and FastText embeddings combined with VecMap model for different threshold values: 0.1, 0.3, 0.4, 0.45, 0.5, and 0.7. To combine filtering with Iterative BT, we performed Iterative Filtered BT with the threshold value which gave the best results in the previous experiment. To combine Iterative Filtered BT with Data Selection, first, we ran FDA and INR algorithms on monolingual Sinhala and English corpora. Then the selected sentences were back-translated by the best model obtained so far with Iterative Filtered BT.

We conducted experiments on the ratio between authentic to synthetic data; 1:1, 1:2, 1:3, and 1:4 for the data extracted from the News data sets. Then we picked the models which performed the best for Sinhala → English and English → Sinhala translation directions. We pick the Sinhala and English monolingual corpora which were used to train the best models obtained in these experiments for future experiments. We conducted filtering using LASER embeddings for different threshold values: 0.1, 0.3, 0.4, 0.45, 0.5, and 0.7 on the synthetic parallel data obtained by back-translating the monolingual data picked based on the results of the previous experiments. To combine Iterative BT, filtering, and Data selection, we used FDA and INR algorithms with a threshold of 0.7 on monolingual News data.

Chapter 5

RESULTS AND DISCUSSION

5.1 In-domain data

5.1.1 Vanilla BT

Sinhala → English

As evident in Table 5.1, vanilla Back-Translation improves over the Baseline NMT system proving that Back-Translation enhances the performance of NMT systems in low-resource domain-specific (since we use in-domain monolingual English data) settings. An improvement of +1.07 BLEU was observed over the Baseline NMT model. One of the main reasons for the vanilla BT model to outperform the Baseline NMT model is the monolingual English corpus we used. These sentences were neither too long nor too short. Hence the translations of these sentences were of higher quality.

	Model	**Size**	**BLEU**
Baseline	Ensemble	74468	24.42
Back-Translation	Ensemble	42773 + 74468	**25.49**

Table 5.1: **Si→En** Vanilla BT

English → Sinhala

As we can see in Table 5.2, vanilla BT did not improve over the performance of the Baseline NMT model. It was lagging by -1.67 BLEU points contradicting our claim that Back-Translation improves the performance of Machine Translation models.

The reason for this performance drop was the monolingual Sinhala corpus used. Compared to the English monolingual corpus, the Sinhala monolingual corpus consisted of longer sentences that did not get translated correctly from the

Baseline NMT model. Thus, generating low-quality synthetic sentences. These synthetic sentences contained several repetitions of a correctly translated phrase of a long sentence.

	Model	**Size**	**BLEU**
Baseline	Ensemble	74468	22.85
Back-Translation	Ensemble	44115 + 74468	**21.18**

Table 5.2: **En→Si** Vanilla BT

For an example consider the following translated synthetic sentences: *"the accounting liabilities of the supreme court court of appeal high court complex high court complex 25 high court complex 25 high court complex 25 high court complex high court complex magistrate apos s court complex magistrate apos s court complex magistrate apos s court nuwara eliya magistrate apos s court etc ."* and *"the total cost of the people living in the areas of lives and property of the people living in the areas of lives living in the areas of lives and property of the people living in the areas of lives living in the dry season of the people living in the district is rs 413.9 million ."*

5.1.2 Iterative BT

Sinhala → English

As we can observe in Table 5.3, Iterative BT improves over vanilla BT slightly by a BLEU score of +0.14 in the first iteration itself. But then it drops in the next 2 iterations and picks up again slightly. Since the results kept fluctuating (no gradual improvement), we terminated the iteration process after 4 iterations. As presented in Table 5.3, FDA and INR algorithms combined with Iterative BT improved over the best Iterative BT model. FDA algorithm combined with Iterative BT improved over vanilla BT by +1.0 BLEU points and the Back-Translated model at Iteration-1 by +0.86 BLEU points whereas only a slight gain of +0.19 BLEU points from vanilla BT was observed by INR algorithm combined with Iterative BT.

	Model	Initial	Itr-1	Itr-2	Itr-3	Itr-4	FDA	INR
Baseline	Ensemble	24.42						
Back-Translation	Ensemble	25.49	**25.63**	23.6	23.62	25.01	**26.49**	25.82

Table 5.3: Iterative BT with Data selection for **Si** → **En**

English → Sinhala

As evident in Table 5.4, Iterative BT improved over both the Baseline NMT and vanilla BT models in the 2^{nd} iteration with a slight gain of +0.18 BLEU points from the Baseline NMT model. After the 2^{nd} iteration, the performance dropped again. Hence, Iterative BT was conducted only for 4 iterations. When the Transductive data selection algorithm FDA was combined with Iterative BT, an improvement of +0.27 BLEU points over the Baseline NMT model and a significant rise of +1.94 BLEU points over the vanilla BT model was observed. However, as visible in Table 5.4, INR combined with the Back-Translated model at iteration-2 (the best model obtained through Iterative BT) failed to improve over the performance of the same model(at iteration-2).

	Model	Initial	Itr-1	Itr-2	Itr-3	Itr-4	FDA	INR
Baseline	Ensemble	22.85						
Back-Translation	Ensemble	21.18	21.82	**23.03**	22.31	22.53	**23.12**	22.89

Table 5.4: Iterative BT with Data selection for **En** → **Si**

5.1.3 Filtered Back-Translation

Sinhala → English

As visible in Table 5.5, filtering with LASER embeddings did not improve over the vanilla BT model. It was a very slight drop of -0.07 BLEU points which was almost negligible. But the reason for this drop could be the quality of the synthetic parallel corpus. The sentence embeddings for the Sinhala language generated by LASER embeddings might not be as accurate as one expected them to be, since Sinhala is a low-resource language and they have trained their models with a smaller Sinhala dataset compared to languages like English, German, French, and Spanish.

			LASER		**FastText + VecMap**		**LaBSE**	
	Model	**Threshold**	**Size**	**BLEU**	**Size**	**BLEU**	**Size**	**BLEU**
Baseline	Ensemble		24.42					
Back-Translation	Ensemble		25.49					
Filtering	Ensemble	0.7	14864 + 74468	24.95	15342 + 74468	24.82	7137 + 74468	24.24
	Ensemble	0.5	25503 + 74468	24.91	23040 + 74468	25.47	25214 + 74468	25.3
	Ensemble	0.45	25767 + 74468	**25.42**	23794 + 74468	**27.66**	25432 + 74468	**26.11**
	Ensemble	0.4	25868 + 74468	23.87	24354 + 74468	25.7	25555 + 74468	24.25
	Ensemble	0.3	25929 + 74468	25.17	25076 + 74468	25.65	25831 + 74468	23.09
	Ensemble	0.1	25941 + 74468	24.82	25667 + 74468	27.52	25941 + 74468	25.34

Table 5.5: **Si→En** Filtered BT with different threshold values and embedding techniques

As shown in Table 5.5, using FastText embeddings combined with the VecMap model for filtering has enhanced the performance of the NMT system significantly. A +2.17 gain in BLEU score from the vanilla BT model and a +3.24 BLEU gain over the Baseline NMT model were observed when the threshold value was 0.45 which was the highest score obtained through Filtered BT out of all the embedding techniques. A threshold of 0.1 also gave a +2.03 BLEU increment over the vanilla BT model.

Filtered BT using LaBSE embeddings also slightly improved over the performance of the vanilla BT model as evident in Table 5.5. A +0.62 gain in the BLEU score was observed over the vanilla BT model for a threshold value of 0.45. Other models with different threshold values failed to perform better than the vanilla BT model.

When we observe Figure 5.1, we can see that the highest BLEU score was obtained by the model trained on parallel data filtered using FastText embeddings combined with the VecMap model. Averaging FastText word embeddings to get the sentence embedding and then using the VecMap model to map them to the same vector space generated more accurate English and Sinhala sentence embeddings than the other two techniques. The Filtered BT model using LaBSE embeddings also performed better than the vanilla BT model and LASER, indicating that the generated sentence embeddings were more accurate than LASER and less accurate than the combined method of FastText and VecMap.

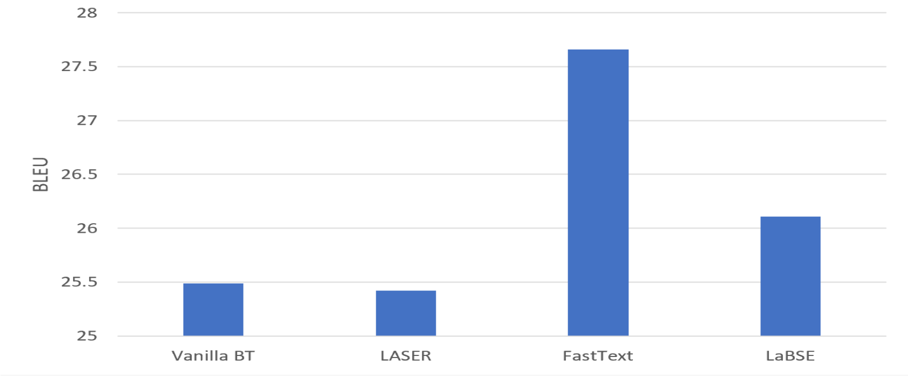


Figure 5.1: **Si→En** vanilla BT model and the best Filtered BT models from each embedding technique

Furthermore, as evident in Figure 5.2, different embedding techniques perform distinctly with each threshold value. For FastText + VecMap embeddings, the threshold value to give the highest BLEU score was 0.45 and the lowest was 0.7. This makes perfect sense since a threshold value of 0.7 filterers out the majority of sentences in the corpus and outputs $1/4^{th}$ of the entire original corpus. For LaBSE embeddings, the highest BLEU score was obtained with a threshold value of 0.45 and the lowest was obtained with a threshold value of 0.3. For LASER embeddings, the best performance was observed when the threshold value was 0.45 and the worst performance was observed at 0.4. Figure 5.2 makes one understand that there isn't a perfect threshold value and it varies with the embedding technique as well as the monolingual corpora we use. Hence, the performance of these models does not vary according to a certain pattern with different threshold values.

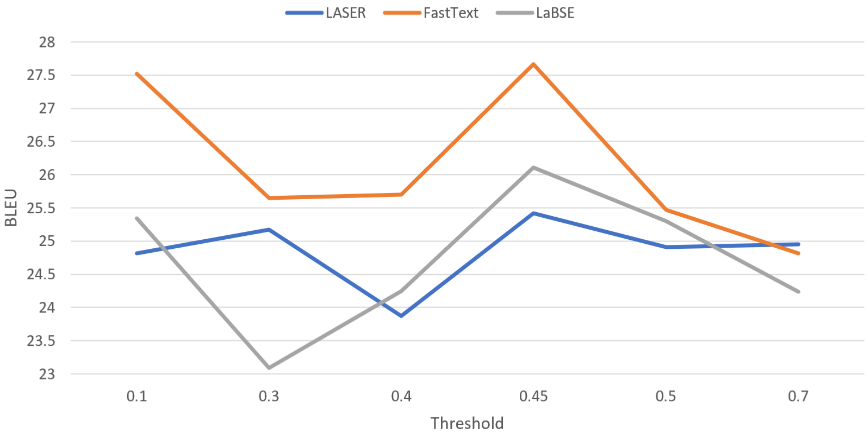


Figure 5.2: **Si→En** Filtered BT for different threshold values

English → Sinhala

As we can see in Table 5.6, Filtered BT with LASER embeddings improved over vanilla BT for all the threshold values. The best performance was observed for a threshold value of 0.45 which obtained a significant gain of +2.0 BLEU points over the vanilla BT model and a very slight improvement of +0.33 BLEU points over the Baseline NMT model.

			LASER		**FastText + VecMap**		**LaBSE**	
	Model	**Threshold**	**Size**	**BLEU**	**Size**	**BLEU**	**Size**	**BLEU**
Baseline	Ensemble		22.85					
Back-Translation	Ensemble		21.18					
Filtering	Ensemble	0.7	15243 + 74468	21.91	17182 + 74468	23.19	8246 + 74468	23.46
	Ensemble	0.5	29279 + 74468	22.76	26559 + 74468	23.18	29413 + 74468	22.9
	Ensemble	0.45	29743 + 74468	**23.18**	27493 + 74468	22.75	29632 + 74468	22.59
	Ensemble	0.4	29966 + 74468	22.81	28152 + 74468	21.86	29734 + 74468	23.13
	Ensemble	0.3	30083 + 74468	22.64	29046 + 74468	22.76	29991 + 74468	**23.7**
	Ensemble	0.1	30109 + 74468	22.63	29734 + 74468	**23.27**	30109 + 74468	23.39

Table 5.6: **En→Si** Filtered BT with different threshold values and embedding techniques

Furthermore, as evident in Table 5.6, using FastText embeddings combined with the VecMap model for Filtered BT improved the performance of NMT models significantly. A gain of +2.09 BLEU points from the vanilla BT model was observed when the threshold value was 0.1 which was the highest gain obtained out of all the threshold values with FastText + VecMap embeddings. Furthermore, this model outperformed the Baseline NMT model by +0.42 BLEU points. All the other Filtered BT models with different threshold values also outperformed the vanilla BT model.

In addition to Filtered BT with LASER and FastText + VecMap embeddings, Filtered BT with LaBSE embeddings also improved over the vanilla BT model significantly as presented in Table 5.6. A gain of +2.52 BLEU points over the vanilla BT model was observed for a threshold of 0.3 which was the highest gain obtained out of all the threshold values and embedding techniques. The best Filtered BT model also outperformed the Baseline NMT model by +0.85 BLEU points.

As we can see in Figure 5.3, Filtered BT outperformed the vanilla BT as well as Baseline NMT (Table 5.6) for the English → Sinhala translation direction. As evident in the chart, the highest gain was obtained using LaBSE whereas FastText + VecMap and LASER have lagged slightly behind. However, we observe that FastText + VecMap and LaBSE have performed better than LASER which we observed in Sinhala → English translation direction as well.

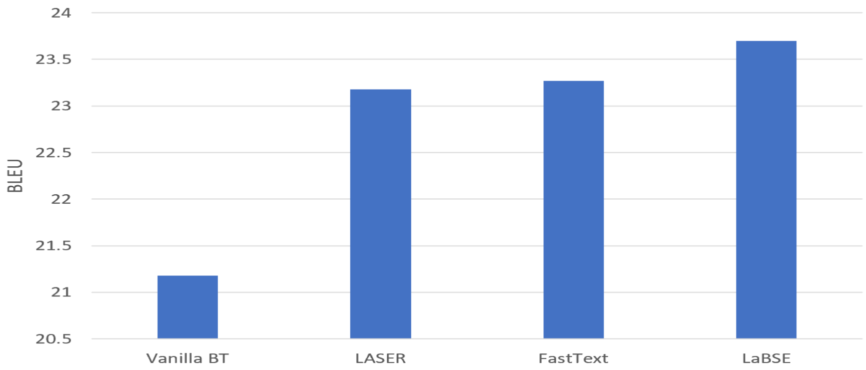


Figure 5.3: **En→Si** vanilla BT model and the best Filtered BT models from each embedding technique

As we mentioned previously, the synthetic parallel data we form for the English → Sinhala translation direction is of poor quality compared to the synthetic parallel data formed for the other translation direction. For Sinhala → English translation direction, the best Filtered BT model was obtained for a threshold value of 0.45 for all three embedding techniques. However, for the English → Sinhala translation direction, it doesn't hold. For FastText + VecMap the threshold value was 0.1, for LaBSE it was 0.3 and for LASER it was 0.45. Hence, these results are not as comparable to making a constructive decision on the performance. Also, the results only showed slight differences between each embedding technique.

As shown in Figure 5.4, the performance of the Filtered BT models change according to the threshold and the sentence embedding technique. For the approach of FastText combined with VecMap, the highest BLEU score was obtained at the 0.1 threshold value and the lowest at 0.4. For LASER, the best performance was achieved when the threshold value was 0.45 and the worst performance was at the threshold value of 0.7. With LABSE embeddings, the highest was at 0.3 and the lowest at 0.45. The threshold value of the best model depends on both the embedding technique and the monolingual corpora we use.

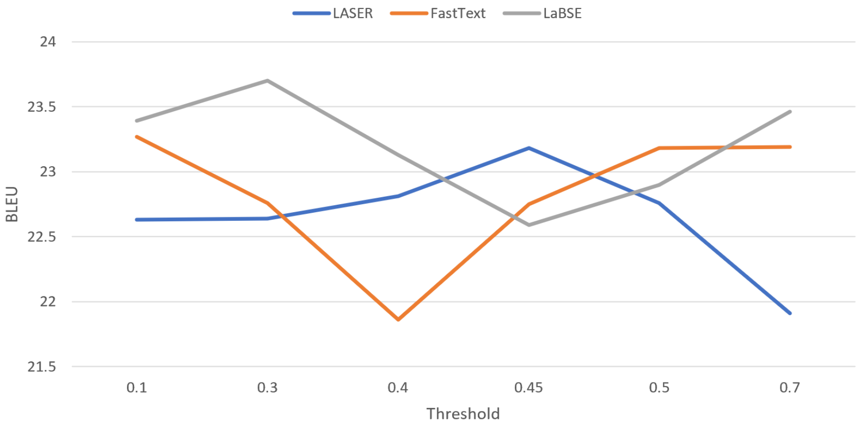
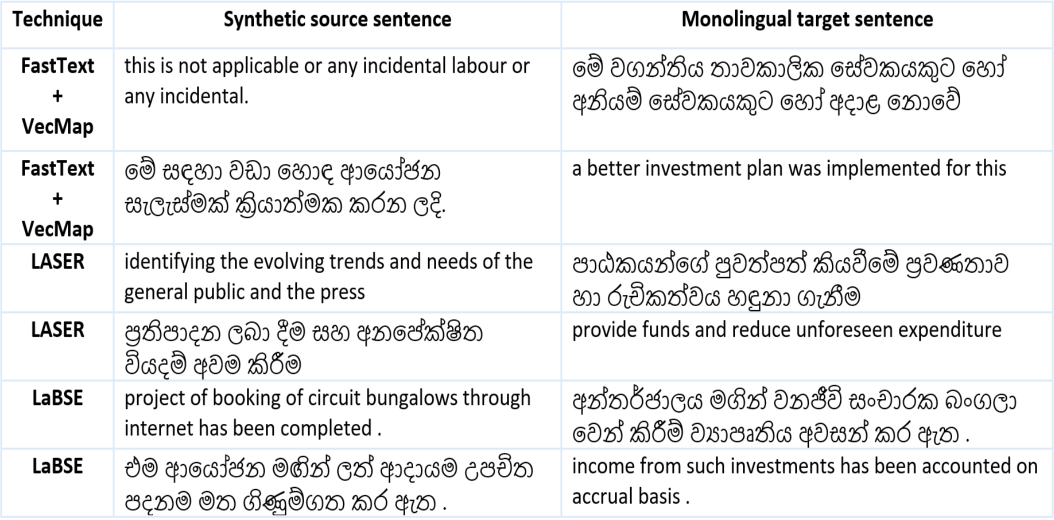


Figure 5.4: **En→Si** Filtered BT for different thresholds

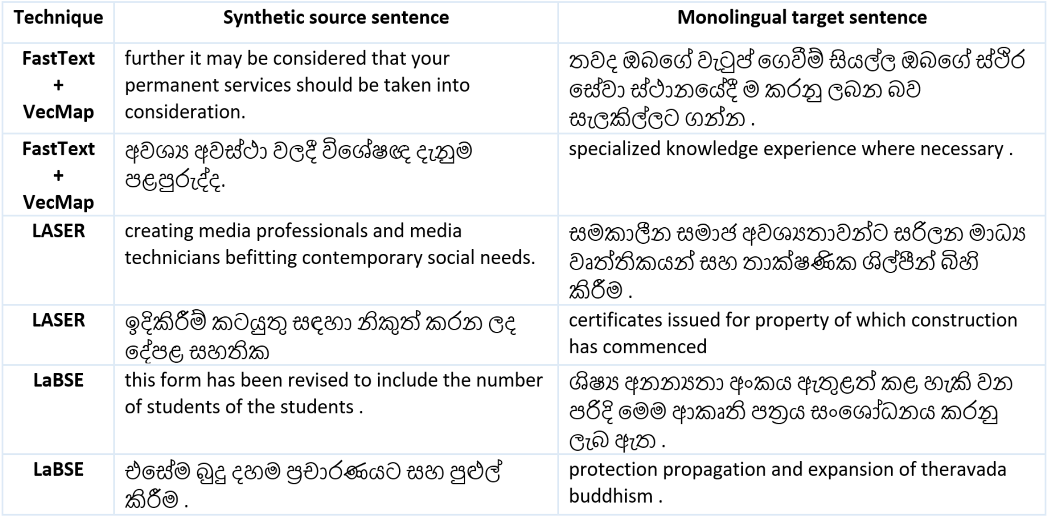
Even the way different embedding techniques change with different threshold values is very distinct. This is evident in Figure 5.4. By observing the chart we can see how different the lines representing each embedding model are. However, LABSE has performed better than the other two, since not at any threshold value, has a model that obtained a BLEU score of less than 22.5.

A few examples of sentence pairs picked and filtered out through filtering by different sentence embedding techniques are presented in Figure 5.5 and Figure 5.6 respectively.



Technique	Synthetic source sentence	Monolingual target sentence
FastText + VecMap	this is not applicable or any incidental labour or any incidental.	මේ වගන්තිය තාවකාලික සේවකයකුට හෝ අනියම් සේවකයකුට හෝ අදාළ නොවේ
FastText + VecMap	මේ සඳහා වඩා හොඳ ආයෝජන සැලැස්මක් ක්‍රියාත්මක කරන ලදි.	a better investment plan was implemented for this
LASER	identifying the evolving trends and needs of the general public and the press	පාඨකයන්ගේ පුවත්පත් කියවීමේ ප්‍රවණතාව හා රුචිකත්වය හඳුනා ගැනීම
LASER	ප්‍රතිපාදන ලබා දීම සහ අනපේක්ෂිත වියදම් අවම කිරීම	provide funds and reduce unforeseen expenditure
LaBSE	project of booking of circuit bungalows through internet has been completed .	අන්තර්ජාලය මගින් වනජීවී සංචාරක බංගලා වෙන් කිරීම් ව්‍යාපෘතිය අවසන් කර ඇත .
LaBSE	එම ආයෝජන මඟින් ලත් ආදායම උපචිත පදනම මත ගිණුම්ගත කර ඇත .	income from such investments has been accounted on accrual basis .

Figure 5.5: Sentence pairs picked by different embedding techniques



Technique	Synthetic source sentence	Monolingual target sentence
FastText + VecMap	further it may be considered that your permanent services should be taken into consideration.	තවද ඔබගේ වැටුප් ගෙවීම් සියල්ල ඔබගේ ස්ථිර සේවා ස්ථානයේදී ම කරනු ලබන බව සැලකිල්ලට ගන්න .
FastText + VecMap	අවශ්‍ය අවස්ථා වලදී විශේෂඥ දැනුම පළපුරුද්ද.	specialized knowledge experience where necessary .
LASER	creating media professionals and media technicians befitting contemporary social needs.	සමකාලීන සමාජ අවශ්‍යතාවන්ට සරිලන මාධ්‍ය වෘත්තිකයන් සහ තාක්ෂණික ශිල්පීන් බිහි කිරීම .
LASER	ඉදිකිරීම් කටයුතු සඳහා නිකුත් කරන ලද දේපළ සහතික	certificates issued for property of which construction has commenced
LaBSE	this form has been revised to include the number of students of the students .	ශිෂ්‍ය අනන්‍යතා අංකය ඇතුළත් කළ හැකි වන පරිදි මෙම ආකෘති පත්‍රය සංශෝධනය කරනු ලැබ ඇත .
LaBSE	එසේම බුදු දහම ප්‍රචාරණයට සහ පුළුල් කිරීම .	protection propagation and expansion of theravada buddhism .

Figure 5.6: Sentence pairs filtered-out by different embedding techniques

5.1.4 Iterative Filtered BT with Data Selection

Sinhala → English

As evident in Table 5.7, Iterative Filtered BT (with LASER) using 0.45 as the threshold value improved over vanilla BT with the best model at Iteration-1 giving a gain of +1.4 BLEU points. We observe that the results drop in the 2^{nd} and the 3^{rd} iterations and rise again in the 4^{th} iteration; a pattern we also observed in Iterative BT. When Transductive data selection algorithms FDA and INR were combined with Iterative Filtered BT (using LASER embeddings) further enhancements in the performance were observed. When the FDA algorithm was combined with iterative filtering, we observed a +3 BLEU score improvement (the highest gain) over the Baseline NMT model and a significant gain of +1.93 BLEU points over the vanilla BT model. However, INR combined with the best Iterative Filtered BT model failed to improve the performance of the latter.

	Model	**Embeddings**	**Threshold**	**Initial**	**Itr-1**	**Itr-2**	**Itr-3**	**Itr-4**	**FDA**	**INR**
Baseline	Ensemble			24.42						
Back-Translation	Ensemble			25.49	25.63	23.6	23.62	25.01	**26.49**	25.82
Filtering	Ensemble	LASER	0.45	25.42	**26.89**	26.01	25.79	26.34	**27.42**	26.89
	Ensemble	FastText + VecMap	0.45	**27.66**	24.88	25.97	26.32	24.76	25.71	25
	Ensemble	LaBSE	0.45	**26.11**	24.43	25.97	25.07	24.71	24.91	25.72

Table 5.7: Iterative Filtered BT (different embedding techniques) with Data selection for **Si → En**

Contrary to our expectations, Iterative Filtered BT (with FastText + VecMap) failed to improve Filtered BT with a threshold value of 0.45 (which was chosen since it gave the best results out of all the threshold values). We can observe in Table 5.7, Transductive data selection algorithms FDA and INR combined with Iterative Filtered BT failed to improve over the initial Filtered BT model with a threshold value of 0.45.

Iterative Filtered BT (with LaBSE) started dropping from the 1^{st} iteration for the threshold value of 0.45 and failed to rise to the performance of the initial

Filtered BT model with the same threshold value. Furthermore, as we can see in Table 5.7, FDA and INR algorithms when combined with Iterative Filtered BT, failed to improve over the initial Filtered BT model with a threshold value of 0.45.

As evident in Figure 5.7, Iterative Filtered BT with different embedding techniques displays the fluctuating patterns we observe with Iterative BT. Hence, we stopped at the 4^{th} iteration. We can observe that with LASER embeddings, performance improves with iterations and keeps fluctuating. Iterative Filtered BT has performed better than Filtered BT with LASER embeddings. However, with FastText + VecMap and LaBSE embeddings, it has been quite the opposite.

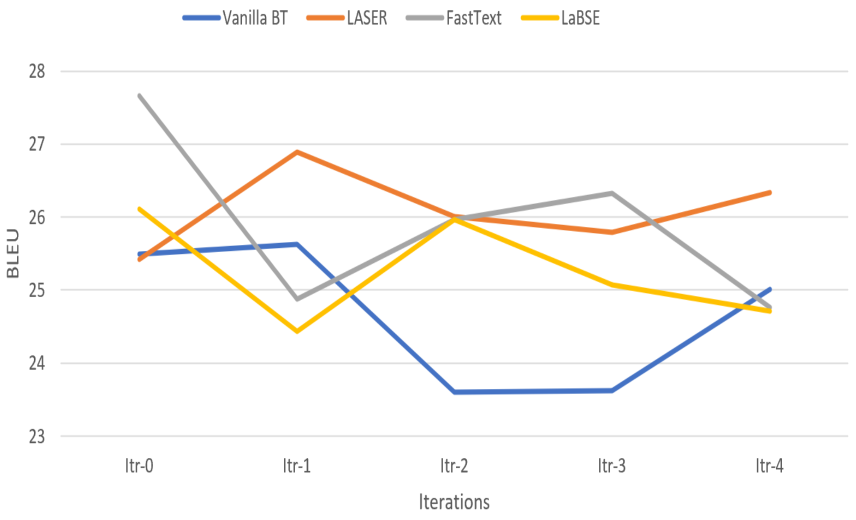


Figure 5.7: **Si→En** Iterative BT and Iterative Filtered BT

As seen in Figure 5.8, FDA combined with Iterative Filtered BT with LASER embeddings outperforms all the other models. The chart shows that this combination performs better than other approaches. However, as we can see in the chart, Iterative Filtered BT with FastText + VecMap and LaBSE combined with FDA and INR failed to improve over the initial Filtered BT model.

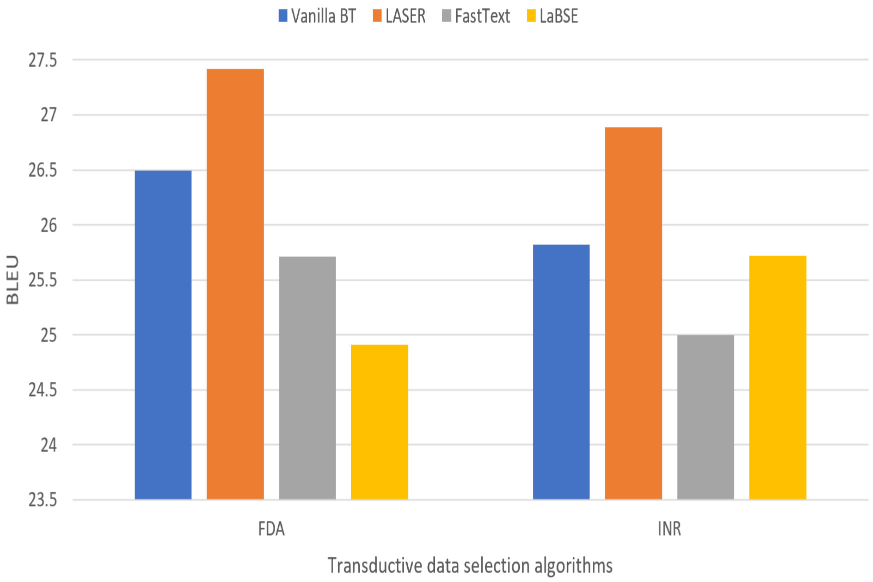


Figure 5.8: **Si**→**En** FDA and INR combined with Iterative BT and Iterative Filtered BT

English → Sinhala

As shown in Table 5.8, Iterative Filtered BT with LASER embeddings (with a 0.45 threshold value) improved the performance slightly over the Baseline NMT model. A gain of +2.15 BLEU points over the vanilla BT model was observed at Iteration-3. Using the FDA algorithm along with Iterative Filtered BT (with LASER), improved over the vanilla BT model with a gain of +2.22 BLEU points (the highest gain). However, the other Transductive data selection algorithm INR combined with Iterative Filtered BT failed to improve over the best model at Iteration-3.

	Model	**Embeddings**	**Threshold**	**Initial**	**Itr-1**	**Itr-2**	**Itr-3**	**Itr-4**	**FDA**	**INR**
Baseline	Ensemble			22.85						
Back-Translation	Ensemble			21.18	21.82	23.03	22.31	22.53	**23.12**	22.89
Filtering	Ensemble	LASER	0.45	23.18	23.08	22.7	**23.33**	22.15	**23.4**	22.9
	Ensemble	FastText + VecMap	0.1	**23.27**	22.79	22.98	23.14	22.78	22.6	23.02
	Ensemble	LaBSE	0.3	**23.7**	22.78	23.12	22.56	23.47	21.96	22.4

Table 5.8: Iterative Filtered BT (different embedding techniques) with Data selection for **En** → **Si**

Contrary to our expectations, Iterative Filtered BT with FastText + VecMap failed to improve over the initial Filtered BT with the same threshold value. Furthermore, as we can see in Table 5.8 when we combined FDA and INR algorithms with Iterative Filtered BT (with FastText + VecMap), they failed to improve over the Filtered BT model with a threshold value of 0.1.

As we can see in Table 5.8, Iterative Filtered BT with LaBSE embeddings, started dropping from the 1^{st} iteration for the threshold value of 0.3 and failed to pick up to the performance of the initial Filtered BT model with the same threshold. Moreover, both the FDA and INR algorithms combined with Iterative Filtered BT (with LaBSE) failed to improve the performance of the initial Filtered BT model with a threshold value of 0.3.

As it can be seen in Figure 5.9, Iterative Filtered Back-Translation has only been effective with LASER as the sentence embedding technique. FastText + VecMap and LaBSE had strong initial models and iterating the same process a few times failed to beat these initial models. However, LASER has outperformed its initial model at the 3^{rd} iteration since the model at iteration-2 in Sinhala → English translation direction was strong. As visible in the chart, the performance of the models kept fluctuating with each iteration.

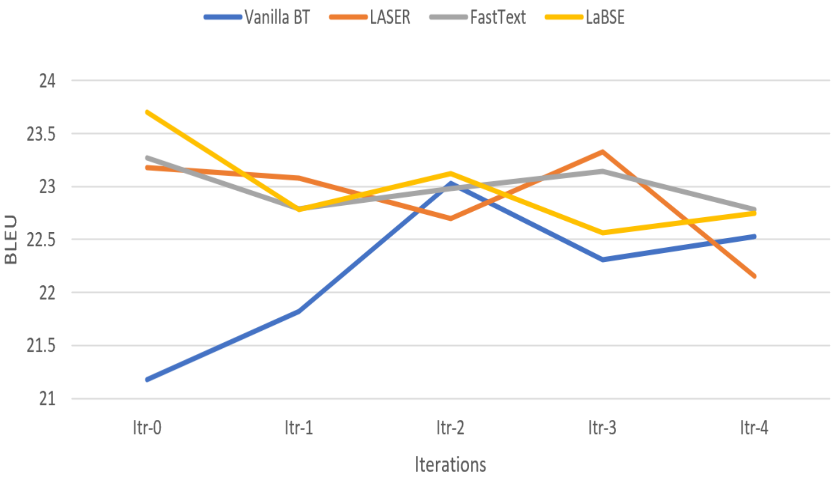


Figure 5.9: **En→Si** Iterative BT and Iterative Filtered BT

Out of all the models, the best one so far has been the initial Filtered BT model with the LaBSE embedding technique. But these differences between Filtered BT and Iterative Filtered BT were very slight. Hence, the reasons for the slight changes can not be interpreted.

Figure 5.10 shows that Iterative Filtered BT with LASER embedding technique achieved the highest BLEU combined with FDA. Even though the Iterative Filtered BT model with FastText + VecMap gave the best results when combined with INR, it did not perform better than the initial Filtered BT model (with FastText + VecMap). Hence, by comparing the charts we can say that LASER is the only sentence embedding technique used in Iterative Filtered BT, combined with Transductive data selection algorithms which performed better than its initial Filtered BT model.

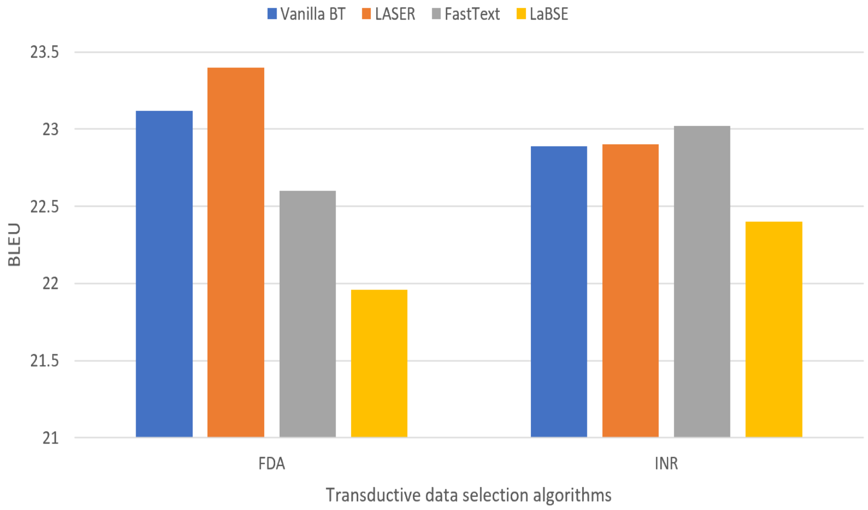
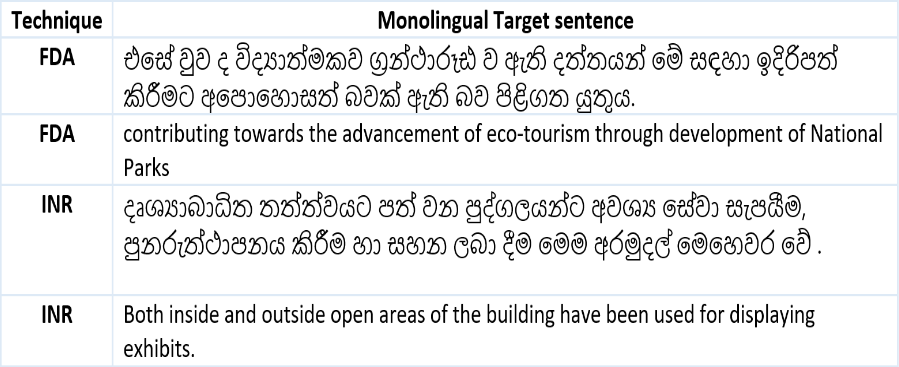


Figure 5.10: **En→Si** FDA and INR combined with Iterative BT and Iterative Filtered BT

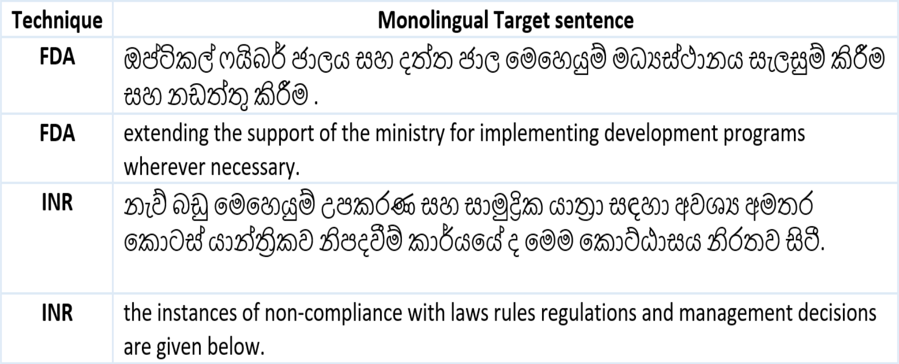
Iterative BT also combined with FDA and INR outperformed the vanilla BT by a significant margin. Transductive Data Selection algorithms combined with Iterative Filtered BT models constructed using LaBSE and FastText + VecMap embedding techniques failed to outperform their initial models because they were very strong to beat. LaBSE and FastText + VecMap embedding techniques with only Iterative Filtered BT, failed to improve over the initial model. Since we combined a weaker Iterative Filtered BT model with FDA and INR, the models obtained through the combination were also weaker than the initial Filtered BT model.

A few examples of monolingual target sentences selected and rejected by Transductive data selection algorithms (FDA and INR) are demonstrated in Figure 5.11 and Figure 5.12 respectively.



Technique	Monolingual Target sentence
FDA	එසේ වුව ද විද්‍යාත්මකව ග්‍රන්ථාරූඪ ව ඇති දත්තයන් මේ සඳහා ඉදිරිපත් කිරීමට අපොහොසත් බවක් ඇති බව පිළිගත යුතුය.
FDA	contributing towards the advancement of eco-tourism through development of National Parks
INR	දෘශ්‍යාබාධිත තත්ත්වයට පත් වන පුද්ගලයන්ට අවශ්‍ය සේවා සැපයීම, පුනරුත්ථාපනය කිරීම හා සහන ලබා දීම මෙම අරමුදල් මෙහෙවර වේ .
INR	Both inside and outside open areas of the building have been used for displaying exhibits.

Figure 5.11: Monolingual target sentences selected by FDA and INR algorithms



Technique	Monolingual Target sentence
FDA	ඔප්ටිකල් ෆයිබර් ජාලය සහ දත්ත ජාල මෙහෙයුම් මධ්‍යස්ථානය සැලසුම් කිරීම සහ නඩත්තු කිරීම .
FDA	extending the support of the ministry for implementing development programs wherever necessary.
INR	නැව් බඩු මෙහෙයුම් උපකරණ සහ සාමුද්‍රික යාත්‍රා සඳහා අවශ්‍ය අමතර කොටස් යාන්ත්‍රිකව නිපදවීම් කාර්යයේ ද මෙම කොට්ඨාසය නිරතව සිටී.
INR	the instances of non-compliance with laws rules regulations and management decisions are given below.

Figure 5.12: Monolingual target sentences rejected by FDA and INR algorithms

5.1.5 Tagged Back-Translation

Sinhala → English

As shown in Table 5.9, Tagged BT failed to improve over the vanilla BT model by lagging by -1.92 BLEU points. However, Iterative Tagged BT managed to enhance the performance with each iteration giving the best results at iteration-3. A slight gain of +0.39 BLEU points from the vanilla BT model was observed by the best model.

	Model	**Initial**	**Itr-1**	**Itr-2**	**Itr-3**	**Itr-4**	**Itr-5**	**Itr-6**	**Itr-7**	**Itr-8**
Baseline	Ensemble	24.42								
Back-Translation	Ensemble	25.49	**25.63**	23.6	23.62	25.01				
Tagged BT	Ensemble	23.57	24.46	24.41	**25.88**	25.72	24.21	24.81	25.62	25.31

Table 5.9: Tagged BT and Iterative Tagged BT for **Si** → **En**

As evident in Table 5.10, Tagged BT combined with filtering (with LASER embeddings) exceeds the performance of the vanilla BT model by +0.87 BLEU points and the Baseline NMT model by +1.94 BLEU points.

	Model	**Threshold**	**Initial**	**Itr-1**	**Itr-2**	**Itr-3**	**Itr-4**	**Itr-5**	**Itr-6**
Baseline	Ensemble		24.42						
Back-Translation	Ensemble		25.49	**25.63**	23.6	23.62	25.01		
Filtering	Ensemble	0.7	26.35						
	Ensemble	0.5	**26.36**	26.21	24.83	25.19	26.18	23.05	25.65
	Ensemble	0.45	23.85						
	Ensemble	0.4	24.01						
	Ensemble	0.3	25.74						
	Ensemble	0.1	24.43						

Table 5.10: Tagged BT with filtering and Iterative Tagged BT with filtering (LASER) for **Si** → **En**

However, when we iterated the process up to 6 iterations, no improvements were observed over the initial Filtered model at any iteration. The performance kept fluctuating but never picked up to the performance of the initial Filtered model with a threshold value of 0.5. The reason could be that the initial Tagged BT + Filtering model was too strong. Thus, iterating the process deteriorated the performance.

English → Sinhala

As evident in Table 5.11, Tagged BT outperformed the vanilla BT model by +1.29 BLEU points. This proves Caswell et al. [16]' s claim that Tagged BT improves BT. However, it has failed to outperform the Baseline NMT model which can be justified on the grounds; that Tagged BT improved the vanilla BT model by a significant amount and it was bouncing back from the fall the system took with vanilla BT. Furthermore, Iterative Tagged BT managed to improve the performance with each iteration giving a gain of +1.91 BLEU points over the vanilla BT model and a slight gain of +0.24 BLEU points over the Baseline NMT model at iteration-4.

	Model	**Initial**	**Itr-1**	**Itr-2**	**Itr-3**	**Itr-4**	**Itr-5**	**Itr-6**	**Itr-7**	**Itr-8**
Baseline	Ensemble	22.85								
Back-Translation	Ensemble	21.18	21.82	**23.03**	22.31	22.53				
Tagged BT	Ensemble	22.47	22.48	23.01	23.07	**23.09**	22.32	22.6	22.77	22.88

Table 5.11: Tagged BT and Iterative Tagged BT for **En** → **Si**

The Iterative Tagged BT model in Sinhala → English translation direction at the 3^{rd} iteration obtained the best results. This caused the monolingual Sinhala sentences to be Back-Translated properly generating a fine synthetic parallel corpus which was used to train the NMT model at the 4^{th} iteration in the English → Sinhala direction. Hence, the best model was obtained at iteration-4.

As we can see in Table 5.12, Tagged BT with filtering outperformed both the Tagged BT model and the vanilla BT model by +0.34 BLEU points and +1.63 BLEU points respectively.

	Model	**Threshold**	**Initial**	**Itr-1**	**Itr-2**	**Itr-3**	**Itr-4**	**Itr-5**	**Itr-6**
Baseline	Ensemble		22.85						
Back-Translation	Ensemble		21.18	21.82	**23.03**	22.31	22.53		
Filtering	Ensemble	0.7	22.58						
	Ensemble	0.5	22.27						
	Ensemble	0.45	**22.81**	23	22.38	22.81	**23.04**	22.36	22.64
	Ensemble	0.4	22.79						
	Ensemble	0.3	22.57						
	Ensemble	0.1	22.29						

Table 5.12: Tagged BT with filtering and Iterative Tagged BT with filtering (LASER) for **En** → **Si**

However, it failed to improve over the Baseline NMT model. When we iterated the process (Tagged BT + Filtering), the performance fluctuated with each iteration. Even though there was no steady rise or drop in the performance with each iteration, the best performance was observed at the 4^{th} iteration. A gain of +0.19 BLEU points was observed over the Baseline NMT model and a gain of +1.86 BLEU points was observed over the vanilla BT model.

5.2 Out-of-domain data

5.2.1 Selecting the best authentic to synthetic data ratio

As evident in Table 5.13 the best performances for both Sinhala → English and English → Sinhala translation directions were observed by the models trained with a ratio of 1:2. For Sinhala → English direction, a gain of +0.93 BLEU points over the Baseline NMT model was observed. However, the model lagged behind the in-domain vanilla BT model by a slight drop of -0.14 BLEU points. For English → Sinhala translation direction, a slight gain of +0.57 BLEU points was observed over the Baseline NMT model and a significant gain of +2.24 BLEU points was observed over the in-domain vanilla BT model.

Ratio	**Sinhala → English**	**English → Sinhala**
	BLEU	BLEU
1:1	24.13	22.61
1:2	**25.35**	**23.42**
1:3	23.81	21.51
1:4	23.17	20.89

Table 5.13: Performance with different authentic to synthetic data ratios

As evident in Figure 5.13, the performance of Back-Translated NMT models change based on the size of the synthetic parallel corpus for both Sinhala → English and English → Sinhala translation directions. This experiment was conducted using out-of-domain News data since we did not have large in-domain monolingual corpora. As we can see in the chart, the best performance was obtained when the ratio between authentic to synthetic parallel sentences was 1:2 for both the translation directions. Then, when the ratio was 1:3, the performance took a drastic drop which further went down when the ratio was 1:4.

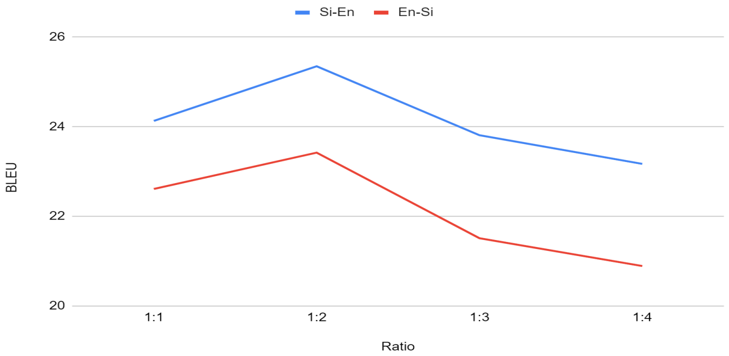


Figure 5.13: Different authentic to synthetic parallel data ratio (News data)

5.2.2 Filtered BT and Iterative Filtered BT with Data Selection

As evident in Table 5.14, the best result for Filtered BT was given by the threshold value of 0.4 for both Sinhala → English translation direction and English → Sinhala translation direction. For Sinhala → English translation direction, a gain of +0.36 BLEU points was observed over the Basic BT model whereas for English → Sinhala translation direction, the best model lagged behind the Basic BT model by -0.05 BLEU points. The reason could be the decrease in the size of the synthetic parallel corpus after filtering.

As we can see in Table 5.15, Iterative Filtered BT improved over Filtered BT with each iteration for both Sinhala → English and English → Sinhala translation directions. A gain of +1.04 BLEU points over the initial Filtered BT model and a gain of +1.40 BLEU points over the Basic BT model was observed in the Sinhala → English translation direction at the 6^{th} iteration. For the English → Sinhala translation direction, a gain of +0.74 BLEU points over the initial Filtered BT model and a gain of +0.69 BLEU points over the Basic BT model were observed at the 6^{th} iteration. For all the experiments LASER embedding technique was used to generate sentence embeddings for filtering.

However, as we saw in Table 5.15, Transductive Data selection algorithms FDA and INR combined with Iterative Filtered BT failed to outperform the best models obtained by Iterative Filtered BT for both Sinhala → English and English → Sinhala translation directions.

			Si → En		En → Si	
	Model	**Threshold**	**Size**	**BLEU**	**Size**	**BLEU**
Baseline	Ensemble		74468	24.42	74468	22.85
Back-Translation	Ensemble		148936 + 74468	**25.35**	148936 + 74468	**23.42**
Filtering	Ensemble	0.7	66387 + 74468	24.32	69077 + 74468	22.72
	Ensemble	0.5	145111 + 74468	25.08	143443 + 74468	22.82
	Ensemble	0.45	145331 + 74468	24.93	143537 + 74468	22.71
	Ensemble	0.4	145410 + 74468	**25.71**	143608 + 74468	**23.37**
	Ensemble	0.3	145467 + 74468	25.41	143680 + 74468	22.99
	Ensemble	0.1	145485 + 74468	25.14	143714 + 74468	23.07

Table 5.14: **Filtering with different thresholds with LASER as the sentence embedding technique.**

Si → En											
	Threshold	**Initial**	**Itr-1**	**Itr-2**	**Itr-3**	**Itr-4**	**Itr-5**	**Itr-6**	**Itr-7**	**FDA**	**INR**
Baseline		24.42									
Back-Translation		25.35									
Filtering	0.4	**25.71**	25.35	25.42	25.82	25.8	26.32	**26.75**	26.34	25.13	25.72
En → Si											
	Threshold	**Initial**	**Itr-1**	**Itr-2**	**Itr-3**	**Itr-4**	**Itr-5**	**Itr-6**	**Itr-7**	**FDA**	**INR**
Baseline		22.85									
Back-Translation		23.42									
Filtering	0.4	**23.37**	23.41	23.4	23.94	23.65	23.27	**24.11**	23.59	22.87	23.32

Table 5.15: **Iterative Filtered BT with Data selection (All the models are Ensemble models)**

5.3 Discussion

5.3.1 In-domain data

The effectiveness of Back-Translation depends on the target-side monolingual corpus used. We observed that the drop in performance in the English → Sinhala Back-Translated model was due to the poor quality of the Sinhala monolingual corpus we used. Due to the lack of quality in the monolingual corpus used, the synthetic corpus generated after Back-Translation transpired to be of rather poorer quality. This resulted in a below-par synthetic parallel corpus which was then combined with the authentic parallel corpus to train the NMT model. Thus, we could state that the quality of the monolingual corpus affects the performance of Back-Translation systems strongly.

We observed in Iterative BT, that the performance of the NMT model does not gradually improve or decline with each iteration. Rather the performance fluctuates. Since Iteration is done in Sinhala → English and English → Sinhala translation directions simultaneously the performance of the NMT model at a particular iteration depends on the condition of the NMT model in the opposite translation direction in the previous iteration. For example, if the NMT model at 2^{nd} iteration in the English → Sinhala translation direction performed poorly, the performance of the NMT model at 3^{rd} iteration in Sinhala → English translation direction may deteriorate and vice versa. This is the reason for the performance of the NMT models at each iteration to fluctuate without gradually increasing or decreasing.

In Filtered BT, we observed that FastText + VecMap performed better than LASER and LaBSE sentence embedding techniques. The main reason for this is the VecMap model. We can presume that VecMap maps the sentence embeddings into the same vector space more accurately than LASER and LaBSE since VecMap is specifically implemented for that purpose. Hence, VecMap is better than LASER and LaBSE at mapping embeddings of sentences which are translations of each other, to points in the same neighborhood.

Contrary to our expectations, for both translation directions, the Iterative Filtered BT models with LaBSE and FastText+VecMap failed to beat their initial Filtered BT models. The reason could be that both these techniques have generated two strong initial Filtered BT models which can not be exceeded by iteratively training NMT models. Artetxe et al. [19] also observed that the final NMT system learned through Iterative BT has performed weaker than the initial system used for warmup. Hence, they also claimed that iterative BT can even degrade the performance when the initial system is very strong.

In addition to the previous observation, we also noticed that the performance of the Iterative Filtered BT models didn't improve when combined with FDA and INR algorithms. Since Iterative BT can degrade the performance of the NMT model when the initial system is strong, the weaker Iterative Filtered BT model could generate low-quality synthetic data. Sentences picked by FDA and INR were Back-Translated by these weak models. Hence, the NMT models generated by training on these low-quality synthetic data perform poorly than the initial Filtered BT models.

With Tagged BT, contrary to our expectations, the performance dropped from the baseline NMT model and the vanilla BT model (only for Sinhala → English direction). The tag is used to signal the model that the synthetic parallel sentences are different from authentic parallel sentences. Caswell et al. [16] claimed that the word-for-word translation bias in BT data is usually incorporated into the BT model after training with synthetic parallel data. However, by Tagged BT, the model has learned how to decode parallel text without having to manually break this translation bias. The reason for the Tagged BT model falling behind the vanilla BT model could be the unacquainted tag preceding every synthetic sentence. Nevertheless, with Iterative Tagged BT, the performance improved. Although the tag was unfamiliar to the model initially, with each iteration, it helps the model to distinguish between authentic and synthetic parallel data. Hence, the performances of the Iterative Tagged BT models outperform the initial Tagged BT model.

5.3.2 Out-of-domain data

We observed that the performance started to drop when the ratio between synthetic to authentic parallel corpora grew. The reason could be the models favoring out-of-domain parallel data since they were way larger than the in-domain parallel data. So, when in-domain validation and test data were used to evaluate the models, the performance dropped as the balance between in-domain and out-of-domain data was leaning more toward out-of-domain data.

Not only the domain mismatch but also the quality of the synthetic parallel data caused the drop in the results when the synthetic parallel corpus was increased in size. Since we used the Baseline NMT models to Back-Translate the monolingual Sinhala and English corpora, some words and phrases foreign to the NMT models might not have gotten translated accurately. When the synthetic parallel corpus was too large, it would have contained these inaccurate translations in large amounts causing the quality of the corpus to be poor. Eventually, these low-quality data affect the performance of the NMT system adversely.

The best model for English → Sinhala translation direction was obtained by Iterative Filtered BT for out-of-domain monolingual data. The main reason is the monolingual Sinhala corpus we used. Unlike for in-domain experiments, the monolingual corpus we used for out-of-domain experiments contained proper sentences. Both out-of-domain monolingual Sinhala and monolingual English corpora were comparable unlike in the in-domain scenario. Hence, the synthetic English sentences generated at each iteration leading up to the 4^{th} iteration progressed with each iteration. This contributed to the performance gain of the NMT models at each iteration.

However, we observed that FDA and INR combined with the best models obtained through Iterative Filtered BT failed to outperform the initial Filtered BT models. The reason for the drop is the domain mismatch between the seed (which was the original parallel data) and the monolingual News data. The number of sentences picked from the monolingual corpora was low since the scores assigned to each sentence were low (because common n-grams in the seed and monolingual corpus were sparse) with the threshold value being 0.7. Hence, the size of the monolingual corpora was smaller compared to previous experiments; causing the performance of the models to deteriorate. Even though Poncelas et al. [15] claimed that FDA and INR help domain adaptation, it seems not to hold in this case.

5.4 Best 5 models obtained for each translation direction

Table 5.16 presents the top 5 models we obtained for Sinhala → English translation direction with their gains over the Baseline NMT model.

Rank	*Model description*	*BLEU*
1	**Filtered BT** with FastText embeddings combined with VecMap for in-domain data. Threshold value = 0.45	**27.66** (+3.24)
2	**Iterative Filtered BT + FDA** with LASER embeddings for in-domain data. Threshold value = 0.45	**27.42** (+3)
3	**Iterative Filtered BT** with LASER embeddings for in-domain data. Threshold value = 0.45	**26.89** (2.47)
4	**Iterative Filtered BT** with LASER embeddings for out-of-domain data. Ratio = 1:2 , Threshold value = 0.4	**26.75** (+2.33)
5	**Iterative BT + FDA** for in-domain data.	**26.49** (+2.07)

Table 5.16: **Best models for Si → En**

We present the top 5 models we obtained for English → Sinhala translation direction in Table 5.17 with the gain of each model from the Baseline NMT model.

Rank	*Model description*	*BLEU*
1	**Iterative Filtered BT** with LASER embeddings for out-of-domain News data. Ratio = 1:2 , Threshold value = 0.4	**24.11** (+1.26)
2	**Filtered BT** with LaBSE embeddings for in-domain data. Threshold value = 0.3	**23.7** (+0.85)
3	**Iterative Filtered BT + FDA** with LASER embeddings for in-domain data. Threshold value = 0.45	**23.4** (+0.55)
4	**Iterative Filtered BT** with LASER embeddings for in-domain data. Threshold value = 0.45	**23.33** (+0.48)
5	**Filtered BT** with FastText embeddings combined with VecMap for in-domain data. Threshold value = 0.1	**23.27** (+0.42)

Table 5.17: **Best models for En → Si**

Chapter 6

CONCLUSION AND FUTURE WORK

It is evident that Back-Translation improves translation performance in extremely low-resource domain-specific settings when a large monolingual corpus is used. The generated synthetic sentences tend to contain errors due to the sub-optimal nature of the NMT system used to translate the monolingual corpus. We have identified iterative BT, data selection, filtering, and tagged BT as approaches to alleviate this problem. A considerable gain in the BLEU score can be observed when filtering and iterative BT are combined. Furthermore, combining filtering, iterative BT, and data selection give the best results and a significant improvement for both Sinhala → English translation direction and English → Sinhala translation direction. In addition to these techniques, tagged BT combined with filtering and Iterative Tagged BT improved over the vanilla BT model and the Baseline NMT model. However, tagged BT alone failed to improve over both the vanilla BT and Baseline NMT models for Sinhala → English translation direction and also failed to improve over the Baseline NMT model for English → Sinhala translation direction.

LASER and LaBSE are competent sentence embedding models out of which LaBSE proved to be better at determining the semantic similarity for bilingual sentence pairs which are translations of each other. FastText combined with the VecMap model demonstrated to outperform both LaBSE and LASER for Sinhala → English translation direction. By comparatively analyzing these embedding techniques, we identified that VecMap maps the sentence embeddings of different languages into the same vector space better than LaBSE and LASER since VacMap is specifically built for that purpose.

The size of the monolingual corpus depends on the domains of both the monolingual and the parallel data and the language pairs involved. If both the monolingual data and parallel data are in the same domain for a low-resource language pair, increasing the size of the monolingual data helps improve the performance. However, even if the domains of the monolingual and parallel training data are the same, for high-resource language pairs, increasing the size of the monolingual corpus can degrade the performance of the NMT model. The reason is the sub-optimal nature of the synthetic sentences which causes the models trained on large amounts of parallel data to depreciate. When the domains of the monolingual and parallel training data mismatch for low-resource language pairs, increasing the ratio between the authentic to synthetic parallel data deteriorate the performance of the NMT model. Hence, we have gathered that the size of the synthetic parallel data depends on the domains of both the monolingual and the parallel training data. In addition to that, it also depends on the language pair.

In the future, we plan to construct larger in-domain Sinhala and English monolingual corpora to use for the BT experiments in Sinhala↔English translation directions. We plan to conduct experiments with Sinhala ↔Tamil and English↔Tamil translations as well. We also plan to use other techniques such as Sampling, and using both source side and target side monolingual data to improve BT for Sinhala↔English, Sinhala ↔Tamil and English↔Tamil. In addition, we plan to publish a journal paper on a survey on Back-Translation.

References

[1] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1700–1709, 2013.

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

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

[4] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[5] Jonas Gehring, Michael Auli, David Grangier, and Yann N Dauphin. A convolutional encoder model for neural machine translation. *arXiv preprint arXiv:1611.02344*, 2016.

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

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

[8] Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. Automatic speech recognition for under-resourced languages: A survey. *Speech communication*, 56:85–100, 2014.

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

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

[11] Franck Burlot and François Yvon. Using monolingual data in neural machine translation: a systematic study. *arXiv preprint arXiv:1903.11437*, 2019.

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

[13] Aizhan Imankulova, Takayuki Sato, and Mamoru Komachi. Filtered pseudo-parallel corpus improves low-resource neural machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(2):1–16, 2019.

[14] Alberto Poncelas, Gideon Maillette de Buy Wenniger, and Andy Way. Data selection with feature decay algorithms using an approximated target side. *arXiv preprint arXiv:1811.03039*, 2018.

[15] Alberto Poncelas, Gideon Maillette de Buy Wenniger, and Andy Way. Adaptation of machine translation models with back-translated data using transductive data selection methods. *arXiv preprint arXiv:1906.07808*, 2019.

[16] Isaac Caswell, Ciprian Chelba, and David Grangier. Tagged back-translation. *arXiv preprint arXiv:1906.06442*, 2019.

[17] Benjamin Marie, Raphael Rubino, and Atsushi Fujita. Tagged back-translation revisited: Why does it really work? In *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics, pages 5990–5997, 2020.

[18] A Poncelas, D Shterionov, A Way, GM de Buy Wenniger, and P Passban. Investigating backtranslation in neural machine translation. arxiv 2018. *arXiv preprint arXiv:1804.06189*.

[19] Mikel Artetxe, Gorka Labaka, Noe Casas, and Eneko Agirre. Do all roads lead to rome? understanding the role of initialization in iterative back-translation. *arXiv preprint arXiv:2002.12867*, 2020.

[20] Idris Abdulmumin, Bashir Shehu Galadanci, and Abubakar Isa. Iterative batch back-translation for neural machine translation: A conceptual model. *arXiv preprint arXiv:2001.11327*, 2019.

[21] Ryan Cotterell and Julia Kreutzer. Explaining and generalizing back-translation through wake-sleep. *arXiv preprint arXiv:1806.04402*, 2018.

[22] Marzieh Fadaee and Christof Monz. Back-translation sampling by targeting difficult words in neural machine translation. *arXiv preprint arXiv:1808.09006*, 2018.

[23] Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. Improving back-translation with uncertainty-based confidence estimation. *arXiv preprint arXiv:1909.00157*, 2019.

[24] Xing Niu, Michael Denkowski, and Marine Carpuat. Bi-directional neural machine translation with synthetic parallel data. *arXiv preprint arXiv:1805.11213*, 2018.

[25] Idris Abdulmumin, Bashir Shehu Galadanci, and Aliyu Garba. Tag-less back-translation. *arXiv preprint arXiv:1912.10514*, 2019.

[26] Zi-Yi Dou, Antonios Anastasopoulos, and Graham Neubig. Dynamic data selection and weighting for iterative back-translation. *arXiv preprint arXiv:2004.03672*, 2020.

[27] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, 2004.

[28] Chin-Yew Lin and Franz Josef Och. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 501–507, 2004.

[29] Yangqiu Song and Dan Roth. Unsupervised sparse vector densification for short text similarity. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1275–1280, 2015.

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

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

[32] Nikhil Jaiswal, Mayur Patidar, Surabhi Kumari, Manasi Patwardhan, Shirish Karande, Puneet Agarwal, and Lovekesh Vig. Improving nmt via filtered back translation. In *Proceedings of the 7th Workshop on Asian Translation*, pages 154–159, 2020.

[33] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[34] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[35] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *arXiv preprint arXiv:1805.06297*, 2018.

[36] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*, 2020.

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

[38] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

[39] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, 2019.

[40] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*, 2020.

[41] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

[42] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[43] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

[44] Lijun Wu, Yiren Wang, Yingce Xia, QIN Tao, Jianhuang Lai, and Tie-Yan Liu. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4198–4207, 2019.

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

[46] Anna Currey and Kenneth Heafield. Zero-resource neural machine translation with monolingual pivot data. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 99–107, 2019.

[47] TH Muneeb, Sunil Sahu, and Ashish Anand. Evaluating distributed word representations for capturing semantics of biomedical concepts. In *Proceedings of BioNLP 15*, pages 158–163, 2015.

[48] Aloka Fernando, Surangika Ranathunga, and Gihan Dias. Data augmentation and terminology integration for domain-specific sinhala-english-tamil statistical machine translation. *arXiv preprint arXiv:2011.02821*, 2020.