NEURAL MACHINE TRANSLATION APPROACH FOR SINGLISH TO ENGLISH TRANSLATION

H.G.Dinidu Sandaruwan

(189396L)

Degree of MSc in Artificial Intelligence

Department of Computational Mathematics

University of Moratuwa
Sri Lanka

February 2021

NEURAL MACHINE TRANSLATION APPROACH FOR SINGLISH TO ENGLISH TRANSLATION

H.G.Dinidu Sandaruwan
(189396L)

Thesis submitted in partial fulfilment of the requirement of the degree of MSc in Artificial Intelligence

Department of Computational Mathematics

University of Moratuwa
Sri Lanka

February 2021

Declaration

I declare that this is my own research dissertation and this does not incorporate without acknowledgement any material previously published for a Degree or a Diploma in any other university or institute of higher education and to the best of my knowledge, awareness and belief it does not include any content that previously published or composed by another person except where the acknowledgement is made in the text. Here, I also agree to photocopy and interlibrary loan of my dissertation (if accepted), and provide its title and abstract to external organizations.

Name of the Student H.G.Dinidu Sandaruwan	Signature: Date: 08th February 2021
Supervised by Dr. Subha Fernando	Signature Date: 08th February 2021
Supervised by Dr. Sagara Sumathipala	Signature Date: 08th February 2021

Abstract

This dissertation is for a research that aimed at proposing a language model to translate texts written in Singlish to English. Singlish is an alternative writing system for Sinhala language that uses Latin scripts (English Alphabet) instead of using native Sinhala alphabet. This had been a requirement for long period, since many Sri Lankans use this writing method to write product reviews, social media posts and comments etc. This has been tried since couple of years by many research students but the main challenge was to find a proper data set to evaluate deep learning models for this Natural Language Processing (NLP) task. Hence, traditional statistic, rule-based models has been proposed with less data. This research addresses the challenge of preparing a data set to evaluate a deep learning approach for this machine translation activity and also to evaluate a seq2seq Neural Machine Translation (NMT) model. The proposed seq2seq model is purely based on the attention mechanism, as it has been used to improve NMT by selectively focusing on parts of the source sentence during translation. The proposed approach can achieve 24.13 BLEU score on Singlish-English by seeing ~0.15 M parallel sentence pairs with ~50 K word vocabulary.

Keywords:
Singlish, NMT, Language processing, seq2seq, Attention model, word embedding.

Dedication

I dedicate my dissertation to my family and many friends. A special feeling of gratitude is extended to my loving parents whose thoughts of encouragement supported me in reaching this milestone. I will always appreciate the support of my friends for all the things they have done on behalf of me and their valuable thoughts. Last but not least, I dedicate this work with heartfelt gratitude to my supervisors and the staff of University of Moratuwa for their help, specially to the academic staff for providing me guidance throughout this research.

Above all, I am ever indebted, like every other Sri Lankan, to all those citizens who supported Free Education with their taxes and to all those students who fought with their life to protect Free Education in Sri Lanka.

Acknowledgement

First and foremost, I acknowledge my supervisors Dr. Sagara Sumathipala and Dr. Subha Fernando for their insight for making my research successful. Without the support from supervisors, it would undoubtedly be a real challenge for me to make this research a success.

I would like to thank University of Moratuwa for giving an opportunity to carry out this research and its continuing support during the research. I would like to extend my gratitude for all the academic and non-academic staff of the University of Moratuwa and my colleagues for their generous support, comments and encouragement throughout the project. I am ever grateful to all the expert and novice meditators who were involved in this research project.

Table of Contents

List of Figures ..... **viii**
List of Tables ..... **viii**
List of Abbreviation ..... **ix**
Introduction **1**
1.1 Prolegomena 1
1.2 Background and Motivation 1
1.3 Aim and Objectives 2
1.3.1 Aim 2
1.3.2 Objectives 2
1.4 Problem Definition 2
1.4.1 Challenges 2
1.5 Summary 3
Literature Review **4**
2.1 Introduction 4
2.2 Traditional Machine Translations 4
2.3 Neural Machine Translation 5
2.3.1 Seq2Seq Approaches 6
2.3.2 Attention Mechanism 6
2.3.3 BERT 6
2.3.4 MetaMT 7
2.5 Summary 7
Methodology **8**
3.1 Introduction 8
3.2 Approach 8
3.2.1 Hypothesis 8
3.2.2 Inputs and Outputs 8
3.2.3 Process 9
3.4 Design 9
3.5 Implementation 10
3.5.1 Technology adaption 10
3.5.2 Embedding 10
3.5.3 Encoder 11
3.5.4 Decoder 11

3.5.5 Attention 11
3.5.6 Beam Search 13
Evaluation 14
4.1 Training details 14
4.3 Training Speed 15
Conclusion and Further Work 16
5.1 Conclusion 16
5.2 Discussion and Further Work 16
References 18
Appendix A 20

List of Figures

Figure 1 : Classification of NMT models 5
Figure 2 : En-De Architecture 9
Figure 3 : Neural machine translation 10
Figure 4 : Attention levels 12
Figure 5 Tensor Board dev/test bleu score graphs (x-steps, y-bleu score) 14
Figure 6 : Attention images for training and best bleu 15

List of Tables

Table 1 BLEU Score table for training and testing 14

List of Abbreviation

NLP – Natural Language Processing

NMT – Neural Machine Translation

MT – Machine Translation

RBMT – Rule Based Machine Translation

BERT – Bidirectional Encoder Representations from Transformers

BLEU – Bilingual Evaluation Understudy

RNN – Recurrent Neural Network

LSTM – Long Short Term Memory

SGD – Stochastic Gradient Decent

TF-IDF – Term Frequency-Inverse Document Frequency

Chapter 01

Introduction

1.1 Prolegomena

Language comprehension is a challenging task for computers. Subtle nuances of communication that human toddlers can understand still confuse the most powerful machines. Still, Natural Language Processing (NLP) models struggled to differentiate words based on context. Most of the languages use their own alphabet for writing and we call it as a writing system for any language. We have seen Latin (Roman) script is being used to write many modern-day languages. It is the most used writing system in the world today. Also, it is the official script for nearly all the languages of Western Europe and of some Eastern European languages. It is also used in languages such as Turkish, Vietnamese, Malay language, Somali, Swahili and Tagalog. It can be observed that this particular writing method has become an alternative writing system specially in social media for some of the languages such as Hindi, Urdu, Serbian and Bosnian. This process is defined as code mixed language writing systems. Many researchers are currently working on building models to analyze texts that are written using code mixing as it has been trending in social media [1], [2]. In Sri Lankan context, we have also seen that people tend to write Sinhala in Latin Script (English Alphabet) most of the times when they communicate with those who understand Sinhala and they call it Singlish. The main reason for this phenomenon which is to use Latin scripts as an alternative to their native alphabet, is the keyboard support for Latin scripts and, it is easy for them to use English keyboard rather than using a native keyboard. Singlish is very popular for messaging and also many sub flavors of Singlish can be seen in different social levels mainly based on the age, education level and profession. The most significant issue of using Singlish is the non-availability of a standard method to write in Singlish and most of the people use their own choice of ways to write in Singlish.

1.2 Background and Motivation

In Sri Lankan context, use of social media is rapidly increasing with people's adaption to the technology and their interest of using a Smart Technological device as the main stream of communication and as the source of gaining social knowledge. During the period from January 2019 to January 2020, internet users have increased from 34% to 47% which is 10.1 million of users and, the number of active social media users has risen from 27% to 30% which is 6.40 million of users out of the total population of 21.37 million of people in Sri Lanka. Majority of the social media users in Sri Lanka select their native language (mainly Sinhala or Tamil) for communication. However, a lesser number of users out of them, use the native alphabet itself for writing and the rest of them use English alphabet to write Sinhala or Tamil texts. It has now become a new writing system not only in Sri Lanka but also in many countries in south Asian region.

The motivation for this research comes with the inability to interpret the texts written with alternative writing systems like Singlish in certain circumstances. For an example, many social media platforms give you an option to translate the texts written in different languages to English if you feel you don't understand. But there are no options available to translate something written in an alternative language such as Singlish, Tanglish as those languages are not recognized as standard languages. On the other hand, especially in the countries which this type of writing system is popular struggle to analyze social media data as there are no language models implemented to analyze.

1.3 Aims and Objectives

1.3.1 Aim

The research aims to develop a neural machine translation model with deep neural network architecture to translate Singlish to English and evaluate the performance.

1.3.2 Objectives

- To critically review the existing machine translation models and approaches uses deep learning techniques.
- To evaluate deep learning approaches and techniques to build neuro machine translation model.
- To propose a neuro machine translation model for Singlish to English language translation.
- To construct a parallel corpus for Singlish-English language pair to train the proposed neural machine translation model.
- To construct a standard vocabulary for Singlish language.
- To train and validate the model with the constructed parallel corpus.

1.4 Problem Definition

The main challenge we have currently is to analyze texts written in Singlish. This has been identified as a challenge for most of the online businesses that need to analyze customer feedbacks and reviews of their products or services. There is a requirement of developing a new machine translation model to translate texts written in Singlish. This has been tried for couple of years by many researchers, but the main challenge was to find a proper data set to evaluate deep learning models for this NLP task. Hence, traditional statistic, rule-based models has been proposed with less data [1], [3]. This research addresses the challenge of preparing a data set to evaluate a deep learning approach for this machine translation activity and also to evaluate a seq2seq Neural Machine Translation (NMT) model.

1.4.1 Challenges

- Code mixed nature

Most significantly, when looking at a text written in Singlish, it can be observed that a mixture of English and Sinhala words is included in the text. And sometimes Singlish word also can be an English word. The first objective is to differentiate the words by language, having based on the context of the text and the position.

- Diversity of writing

The Singlish writing pattern can be changed from person to person, based on how they spell Sinhala sounds in English. That is also a challenge that needs attention to resolve.

- Lack of availability of resources

There is no publicly available parallel dataset to be used in a deep learning approach as of now for Singlish and English; it is required to develop a web crawler and additional supportive scripts to create a parallel dataset for training and testing.

1.5 Summary

With the popularity of using Latin scripts as an alternative writing system in Sinhala, online businesses struggle to use their business-intelligent tools to analyze texts written in Singlish especially when people write reviews, comments, feedbacks as it can only be understood by a human who can understand Sinhala. This has been identified as a challenge for most of the online businesses that need to analyze customer feedbacks and reviews of their products or services. This research is an effort of building a new Language Model to understand and interpret texts written in Singlish. This research is mainly scoped at proposing a good approach to translate text written in Singlish to English among many machine translation techniques. Literature review has been done to identify the ideal approach to fit in to morphological mapping between Singlish and English, and also based on the availability of data (parallel corpus).

Chapter 2

Literature Review

2.1 Introduction

As discussed in the introduction, there is a rapid trend that people use alternative writing systems to communicate in social media and there is a need of interpreting or translating the texts written in those alternative writing systems. The machine translation techniques can be used to translate those texts in to a language that most people can understand. This chapter is mainly to review the language translation techniques that are proven to be work well with different languages and identify the best and feasible techniques to implement a language model to translate Singlish to English.

2.2 Traditional Machine Translations

Machine translation is a subfield of computational linguistics, which studies how to use software to translate text from one natural language to another .The machine translation has been evolved and rapidly developed since 1950s to now, with different approaches and techniques [4]. The development of machine translation can be seen in three main branches of Rule-based, Statistic-based and Neuro machine translation. In Rule-based machine translations, we can see there are three main approaches as Direct Systems (Dictionary based machine translation) map input to output with basic rules.

The RBMT (rule-based machine translation) system uses morphological and syntactic analysis, while the bilingual RBMT system (Interlingua) uses abstract meaning. But there are so many shortcomings in this approach. In Earlier days, the difficulty of finding good dictionaries and development of a dictionary was also costly and yet certain linguistic information still needs to be processed manually. And also, the interaction of rules, ambiguities and idioms in large-scale systems are difficult to deal with. Again, it fails to adapt to new domains. When compared with the Rule-based approach, Statistic-based approach has significant improvements as Statistical MT performs better when large and qualified corpora are available. The translation is fluent, which means it reads well and therefore meets user expectations. However, the translation is neither predictable nor consistent. The training of high-quality corpus is automated, and the cost is low. However, the training on the universal language corpus (i.e., texts other than specified domains) is poor. In addition, statistical MT requires a lot of hardware to build and manage large-scale translation models. But statistical machine translation techniques are being used for many low resource languages [5]. However, the common issue of this type of traditional machine translation is that to build the model can be seen as the need of expertise knowledge of both the source and target language.

2.3 Neural Machine Translation

Neural Machine Translation (NMT) is the latest method of machine translation and is said to produce much more accurate translations than statistical machine translation methods [6]. NMT is based on the neural network model and sends information to different "layers" for processing before output. NMT uses deep learning techniques for self-learning to translate text based on existing statistical models. It helps to build a translation model without having an expert knowledge about the languages. Also, self-learning leads to a faster translation with a quality output compared to the statistical method of machine translation. NMT uses algorithms to learn language rules on its own. The most notable advantage of NMT is its speed and quality. Many researchers say that the NMT is the way of the future, and there is no doubt that the process will continue to improve its capabilities. Figure 1 shows the visualization of some famous NMT models and the various changes suggested by the researchers over time [7]–[11].

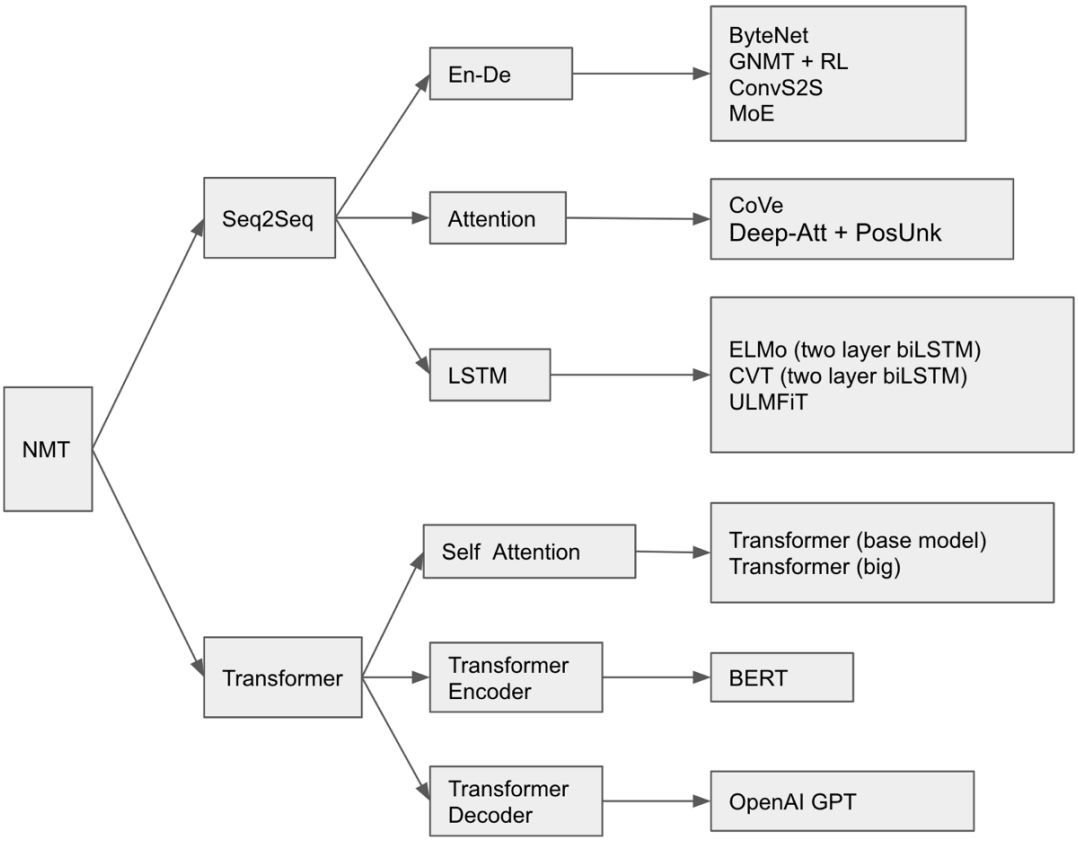

Figure 1 : Classification of NMT models

2.3.1 Seq2Seq Approaches

Sequence-to-sequence (seq2seq) models have been a great success for various NLP tasks such as machine translation, speech recognition, and text summarization. Sequence-to-sequence (Seq2Seq) is relatively a new paradigm, with its first published usage in 2014 [12]. At a high level, a sequence-to-sequence model consists of two recurrent neural networks, one as encoder and the other one as the decoder. The encoder is responsible for processing each item in the input sequence and converges the information it collects in to a separate vector called context vector. Once the input sequence is fully processed by the encoder, it passes the context vector to the decoder. Decoder starts producing the output sequence item by item. Some of the seq2seq (NMT) models consist of a standard, two-layer, bidirectional LSTM encoder with an attention layer and, two-layer unidirectional LSTM decoders. In terms of performance, such models look better than the standard encoder-decoder architecture.

2.3.2 Attention Mechanism

Another exciting study was conducted by Google Brain in 2017 with the goal of creating a new simple network architecture called "Transformer", based solely on attention mechanisms, completely eliminating recurrent and convolution nature of the network. The paper is named "*Attention Is All You Need*" [13]. The authors have shown that sequential nature can be captured by using only the attention mechanism without any use of LSTMs or RNNs. It was an influential article with a gripping headline and opened the doors for a new era in machine translation. In the past, recurrent neural networks were a highly recommended architecture for machine translations. This article surprised everyone by introducing Transformer, a network without having a recurrent nature that only used attention with a couple of other components.

2.3.3 BERT

In 2019, Google AI again introduced a new language model for natural language processing with a revolutionary attention engine called **BERT** [14], or Bidirectional Encoder Representations from Transformers. By design, the model can see the context from both left and right sides of each word of a given sentence. This research suggests a pre-trained model that does not require significant architectural modifications to be applied to specific NLP tasks (Jawahar et al., 2019). BERT is a direct successor to GPT. Train a large language model for free text and then optimize certain tasks without making changes to the network architecture. Compared to GPT, the main difference and improvement of BERT is that the training is bidirectional. The model learns to predict both left and right contexts.

2.3.4 MetaMT

MetaMT is a meta learning method proposed as a solution for low resource languages [16]. NMT model with a new word embedding transition technique for fast domain adaptation. Splits parameters in the model into two groups: **model parameters** and **meta parameters**. Domain adaptation of the machine translation model to low-resource domains using multiple translation tasks on different domains. It proposes a new training strategy based on meta-learning to update the model parameters and meta parameters alternately. Tr-En translation experiment results BLEU score 13.74 with training set of 0.21 M sentence pairs while Fi-En results 20.20 with 2.63 M pairs.

2.5 Summary

With the review of neural machine translation approaches, BERT outperforms other en-de NMT models and which doesn't require any substantial architecture modifications to be applied to specific NLP tasks. Comparative to normal seq2seq models, Large data set is required for pre training of transformer-based models to perform well. Performance of meta learning is high for low resource languages but comparatively lower than recurrent en-de models with attention for large datasets.

Chapter 3

Methodology

3.1 Introduction

Based on the findings and the critical review on language translation techniques under neural machine translation, some techniques have been highlighted that can be put together to build a language model for Singlish. This chapter presents the methodology to solve the defined problem. This chapter has been structured to discuss the proposed solution.

3.2 Approach

3.2.1 Hypothesis

Seq2Seq neural machine translation topology with attention mechanism can be used to construct a new language model for Singlish (languages with very different morphology and syntax) to English translations. Inspiration behind this hypothesis is coming along with the outperforming results of related NMT models which associated with seq2seq topology and attention mechanism. It has shown successful results in low resource languages like Vietnam, Urdu etc [17].

3.2.2 Inputs and Outputs

Inputs:

Source word sequence (Singlish)
ie: - "mama gedara yanawa"

Outputs:

Target word sequence (English)
ie: - "I go home"

Training data set:

Finding a dataset was one of the biggest challenges of this research since there are not many resources available for Singlish. This research required a parallel corpus with at least 0.2 M sentence pairs and over 50K word of vocabulary. In order to prepare the required dataset, A web crawler is developed on top of scrapy framework and used google translator and pronunciation APIs. Once the data is collected, additional script was developed to clean the data generated from pronunciation API. Since this a generated dataset, it is entirely similar to the way how people actually

write in Singlish. But one can choose to write in this way if he really thinks about how Sinhala pronunciation can be written with English letters. Following sample sentence pair shows the process of generating the dataset.

Scraped: “There was a time in my life where we had a very troubled experience in our family”

Translated: “අපේ පවුල තුළ අපට බොහෝ කරදරකාරී අත්දැකීම් ඇති කාලයක් මගේ ජීවිතයේ තිබුණි”

Pronunciation: “apē pavula tuḷa apaṭa bohō karadarakārī atdækīm æti kālayak magē jīvitayē tibuṇi”

Processed: “ape paula tula apata boho karadarakare atdakem ati kalayak mage jewitaye tibuni”

With this approach, A parallel corpus of 0.26 M language pairs (Singlish-English), 65 K Singlish and 49 K English word vocabulary was generated for training and ~1.5 K languages pairs were prepared for each testing and validation.

3.2.3 Process

NMT model consists of two main recurrent neural networks: The **encoder** RNN only consumes the input source word sequence without any prediction. On the other hand, the **decoder** processes the target sentence while predicting the next word. This simply means that the *encoder* converts the source sentence into a "meaning" vector, and then passes it through the *decoder* to generate a translation.

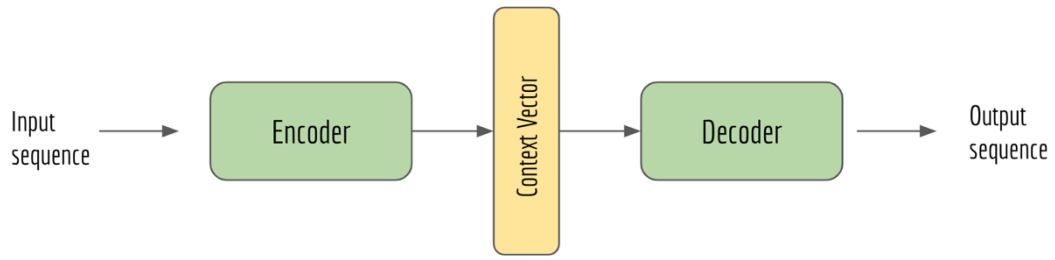

Figure 2 : En-De Architecture

3.4 Design

In this design, we consider a deep multi-layer RNN, which contains a bidirectional LSTM as a recurrent unit. An example of this model is shown in Figure 3. In this example, It shows how the model works to translate a source sentence **"mama gedara yanawa"** into a target sentence **"I go home"**. As describes in the process

section in a high level, the encoder simply consumes the input source without making any prediction and then the decoder, on the other side, processes the output sentence while predicting the next best probable words.

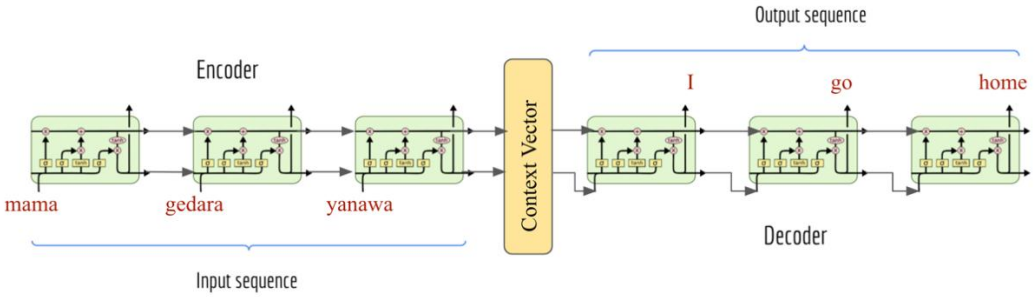

Figure 3 : Neural machine translation

An example of a deep recurrent architecture used to translate the source sentence **"mama gedara yanawa"** into the target sentence **"I go home"** is proposed.

3.5 Implementation

3.5.1 Technology adaption

TensorFlow NMT implementation is being backed by Google AI team since 2017 and makes a room for researchers to build competitive translation models from the scratch [18]. In this research also TensorFlow has been used as the framework for the implementation.

3.5.2 Embedding

Given the categorical nature of a word, the model must first find the source and target embeddings to retrieve the corresponding word representation. In order for the embedding layer to work, first select a vocabulary for each language. Usually, the vocabulary size V is selected, and only the most frequent V words are considered unique. All other words will be converted to "unknown" tokens, and all words will get the same embedding. Embedding weights (one set for each language) are usually learned during training.

```
# Embedding Implementation
embedding_encoder = variable_scope.get_variable( "embedding_encoder", [src_vocab_size,
embedding_size], ...)

encoder_emb_inp = embedding_ops.embedding_lookup(embedding_encoder, encoder_inputs)
```

Similarly, we can embed the decoder and decoder word sequences and construct the embedding later of the network. Note that you can choose to use pre-trained word representations (such as word2vec or Glove vectors) to initialize the embedding weights. Usually, if there is a lot of training data, we can learn these embeddings from scratch.

3.5.3 Encoder

Once retrieved, the word embedding is fed as input to the main network, which is composed of two multi-layer RNNs-the source language encoder and the target language decoder. In principle, these two RNNs can share the same weight. However, in practice, two different RNN parameters are used very oftenly for this type of models as it performs better when fitting large training data sets. The *encoder* RNN uses zero vectors as its starting states and is built as follows:

```
# Encoder RNN cell implementation
encoder_cell = tf.nn.rnn_cell.BasicLSTMCell(num_units)

# Run Dynamic RNN
encoder_outputs, encoder_state = tf.nn.dynamic_rnn(encoder_cell, encoder_emb_inp,
sequence_length=source_sequence_length, time_major=True)
```

3.5.4 Decoder

The decoder also needs to access the source information. A simple way to implement it is to initialize it with the last hidden state encoder_state of the encoder. In Figure 2, the hidden state at the source word "yanawa" is passed to the decoder side. However, the last hidden state would depend more on the last word and no previous words have taken into consideration. Here the attention mechanism comes in to play.

```
# Decoder RNN cell implementation
decoder_cell = tf.nn.rnn_cell.BasicLSTMCell(num_units)

# Helper
helper = tf.contrib.seq2seq.TrainingHelper(decoder_emb_inp, decoder_lengths, time_major=True)

# Decoder
decoder = tf.contrib.seq2seq.BasicDecoder(decoder_cell, helper, encoder_state,
output_layer=projection_layer)

# Dynamic decoding
outputs, _ = tf.contrib.seq2seq.dynamic_decode(decoder, ...)
logits = outputs.rnn_output
```

3.5.5 Attention

Attention to the human mind means giving attention to a particular aspect. Conventional methods like TF-IDF give more importance to particular words

(according to TF-IDF value) but are not able to see the sequential information. The whole idea is to check whether we can combine the best of both worlds.

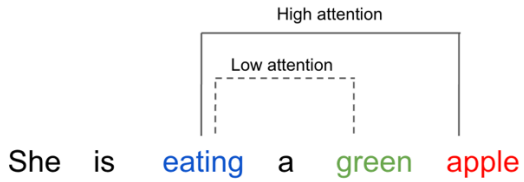

Figure 4 : Attention weights

In Figure 3, described how the decoder takes the last hidden state of the encoder. The results were not good. Therefore, to produce a better translation, all the hidden states have to be considered. This importance is decided by the scores generated by this attention mechanism as an aggregated context vector.

Note that the calculation occurs at each decoder time step. It includes the following steps:

1. Compare the current hidden state of the target with all source states to get the attention weight.
2. Based on the attention weight, the context vector is calculated using the weighted average of the source state.
3. Combine both the context vector and the current target hidden state to produce the attention vector.
4. Feed the attention vector as input to the next time step (input feed). This process can be summarized by the following equation

The first three steps can be summarized by the equations below:

$$\alpha_{ts} = \frac{\exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \quad \text{[Attention weights]} \quad (1)$$

$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s \quad \text{[Context vector]} \quad (2)$$

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W_c}[\boldsymbol{c}_t; \boldsymbol{h}_t]) \quad \text{[Attention vector]} \quad (3)$$

$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \boldsymbol{W} \bar{\boldsymbol{h}}_s & \text{[Luong's multiplicative style]} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W_1} \boldsymbol{h}_t + \boldsymbol{W_2} \bar{\boldsymbol{h}}_s\right) & \text{[Bahdanau's additive style]} \end{cases} \quad (4)$$

3.5.6 Beam Search

The straight forward way to generate output sequences is to use a greedy algorithm. Picking the token with the highest probability, and moving on to the next. However, it can often lead to sub-optimal output sequences. Computationally also, it is inefficient.

One recommended way to deal with this issue is to use Beam Search. Beam Search uses breadth-first search algorithm to build its search graph, but only keeps top N nodes (beam-size) at each level in the search tree. The next level will then be expanded from these N nodes. Itis still a greedy search algorithm, but a lot less greedy than the previous one as its search space is larger. In this research, we tested with N=10 as the beam size.

Chapter 4

Evaluation

4.1 Training details

A bidirectional encoder (one bidirectional layer of the encoder) is used to train a 2-layer LSTM with 512 units, and the embedding size is 512. LuongAttention (scale = True) is used with dropout keep_prob of 0.8. All parameters are uniform. learning rate of 1.0 has been used, as shown below, training for 12K steps (~12 epochs); after 8K steps, the learning speed will be halved in every 1K step.

4.2 Results

Below summary shows the averaged results of 2 models. measured the translation quality in terms of BLEU scores [19].

Systems	Test2020 (dev)	Test2020 (test)
NMT (greedy)	19.24	21.27
NMT (beam=10)	21.80	24.13

Table 1 BLEU Score table for training and testing

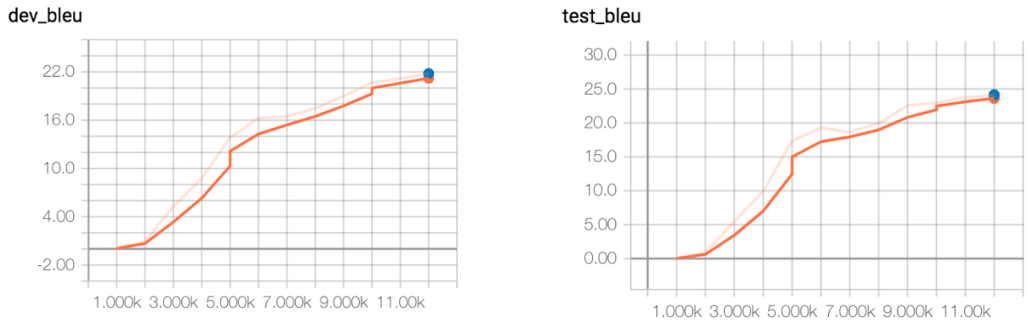

Figure 5 TensorBoard dev/test bleu score graphs (x-steps, y-bleu score)

Figure 6 : Attention images for training and best bleu

4.3 Training Speed

(0.283s step-time, 18.3K wps) for *2.6 M* sentences on a Macbook i7 with 2.2 GHz 6 core cpu and 8 GB memory. Here, step-time means the time taken to run one mini-batch (of size 128). For wps, we count words on both the source and target.

Chapter 5

Conclusion and Further Work

5.1 Conclusion

It can be seen from the data evaluation, that we have achieved a great success with a 24.13 BLEU score for Singlish-English translation. Finding a data set for this research was one of the key pain-points for many researchers in Sri Lanka, especially in the attempt of trying out a deep learning approach for Singlish-English translation. An alternative way is provided in this research to generate a pretty decent dataset for this translation activity. This is only a very initial stage of the research domain of analyzing, translating alternative writing systems used in Sri Lanka. The biggest achievement is putting a step ahead to use latest and greatest deep learning approaches in machine translation domain. The good news is that this research opens up doors to several new research paths to carry forward the model development and improvement for Singlish-English machine translation.

5.2 Discussion and Further Work

The data set used for this research has been a synthetic generation using an existing trained language model. As mentioned in the thesis, the data set is not in exact alignment with how people write Sinhala pronunciation in English alphabet. But it's fair to make an argument that only the spellings will be different if we make this data set to align with how people actually write the pronunciation since this data set is in a good shape when it's come to grammar. Theoretically we shouldn't be expecting a huge improvement for this model if the dataset is improved only on spelling as we haven't used a character level word embedding for this model. But it's worth to improve the data set to align with how people actually write.

The other area that needs improvement to this model is how this model handles unseen word in translation. Translation is an open-vocabulary problem and we can see the problem in many angles. Names, numbers are morphologically simple, but open word classes. Following is one instance where this problem occurs when numbers are included in the source sentence.

Source :

"2002 de mema madhyasthana satun 45,000 k laba gat atara in 37,000 k kurullan wiya"

Translated :

"In <unk> <unk> these centers were given <unk> animals <unk> and <unk> of them were birds"

For an example, the abovementioned source sentence was translated with <unk> tag everywhere as numbers were not included in the vocabulary. It has simply ignored the word and replaced out-of-vocabulary words with <unk>. We can consider this problem as an area for the future improvements. There are some existing techniques such as back-off model which replace rare words with <unk> in training time and when system produce <unk>, align it to source word and translate that with back-off model [20]. Another approach that we can consider is using sub word model with bite pair encoding [21].

References

[1] R. Singh, N. Choudhary, and M. Shrivastava, “Automatic Normalization of Word Variations in Code-Mixed Social Media Text,” p. 11.

[2] K. Sreelakshmi, B. Premjith, and K. P. Soman, “Detection of Hate Speech Text in Hindi-English Code-mixed Data,” *Procedia Comput. Sci.*, vol. 171, pp. 737–744, 2020, doi: 10.1016/j.procs.2020.04.080.

[3] R. Weerasinghe, “A Statistical Machine Translation Approach to Sinhala-Tamil Language Translation,” p. 7.

[4] J. Hutchins, “Machine Translation: History,” in *Encyclopedia of Language & Linguistics*, Elsevier, 2006, pp. 375–383.

[5] W. P. Pa, Y. K. Thu, A. Finch, and E. Sumita, “A Study of Statistical Machine Translation Methods for Under Resourced Languages,” *Procedia Comput. Sci.*, vol. 81, pp. 250–257, 2016, doi: 10.1016/j.procs.2016.04.057.

[6] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” *ArXiv14090473 Cs Stat*, May 2016, Accessed: Nov. 01, 2020. [Online]. Available: http://arxiv.org/abs/1409.0473.

[7] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, “Neural Machine Translation in Linear Time,” *ArXiv161010099 Cs*, Mar. 2017, Accessed: Jun. 05, 2020. [Online]. Available: http://arxiv.org/abs/1610.10099.

[8] Y. Wu *et al.*, “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,” *ArXiv160908144 Cs*, Oct. 2016, Accessed: Jun. 07, 2020. [Online]. Available: http://arxiv.org/abs/1609.08144.

[9] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional Sequence to Sequence Learning,” *ArXiv170503122 Cs*, Jul. 2017, Accessed: Jun. 07, 2020. [Online]. Available: http://arxiv.org/abs/1705.03122.

[10] J. Zhou, Y. Cao, X. Wang, P. Li, and W. Xu, “Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation,” *ArXiv160604199 Cs*, Jul. 2016, Accessed: Jun. 05, 2020. [Online]. Available: http://arxiv.org/abs/1606.04199.

[11] N. Shazeer *et al.*, “Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer,” *ArXiv170106538 Cs Stat*, Jan. 2017, Accessed: Jun. 07, 2020. [Online]. Available: http://arxiv.org/abs/1701.06538.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” p. 9.

[13] A. Vaswani *et al.*, “Attention Is All You Need,” *ArXiv170603762 Cs*, Dec. 2017, Accessed: May 17, 2020. [Online]. Available: http://arxiv.org/abs/1706.03762.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *ArXiv181004805 Cs*, May 2019, Accessed: May 17, 2020. [Online]. Available: http://arxiv.org/abs/1810.04805.

[15] G. Jawahar, B. Sagot, and D. Seddah, “What Does BERT Learn about the Structure of Language?,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 3651–3657, doi: 10.18653/v1/P19-1356.

[16] J. Gu, Y. Wang, Y. Chen, K. Cho, and V. O. K. Li, “Meta-Learning for Low-Resource Neural Machine Translation,” *ArXiv180808437 Cs*, Aug. 2018, Accessed: Jun. 21, 2020. [Online]. Available: http://arxiv.org/abs/1808.08437.
[17] M.-T. Luong and C. D. Manning, “Stanford Neural Machine Translation Systems for Spoken Language Domains,” p. 4.
[18] G. Neubig, “Neural Machine Translation and Sequence-to-sequence Models: A Tutorial,” *ArXiv170301619 Cs Stat*, Mar. 2017, Accessed: Nov. 01, 2020. [Online]. Available: http://arxiv.org/abs/1703.01619.
[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL ’02*, Philadelphia, Pennsylvania, 2001, p. 311, doi: 10.3115/1073083.1073135.
[20] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, “On Using Very Large Target Vocabulary for Neural Machine Translation,” *ArXiv14122007 Cs*, Mar. 2015, Accessed: Nov. 03, 2020. [Online]. Available: http://arxiv.org/abs/1412.2007.
[21] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, “Addressing the Rare Word Problem in Neural Machine Translation,” *ArXiv14108206 Cs*, May 2015, Accessed: Nov. 03, 2020. [Online]. Available: http://arxiv.org/abs/1410.8206.

Appendix A

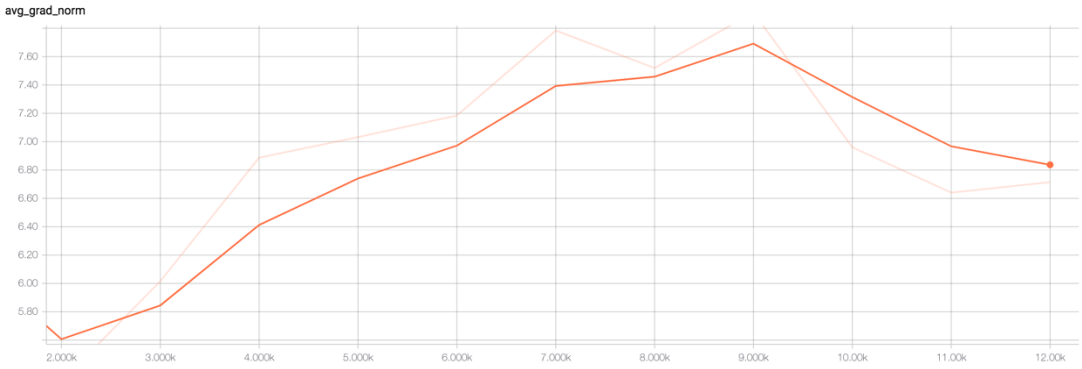

Figure 7 average grad norm over training iterations

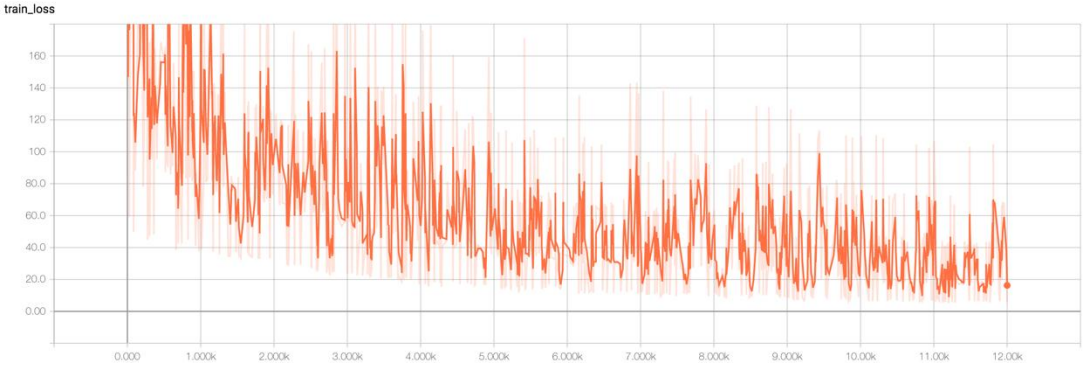

Figure 8 training loss over training iterations

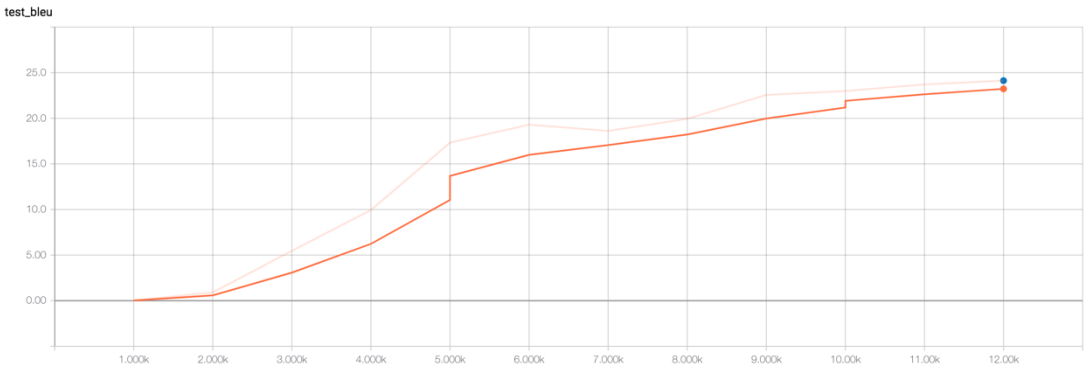

Figure 9 test bleu score variation over training iterations