Neural Machine Translation Approach for Singlish to English Translation

Dinidu Sandaruwan[#1], Sagara Sumathipala[2], Subha Fernando[3]

Abstract— **Comprehension of "Singlish" (an alternative writing system for Sinhala language) texts by a machine had been a requirement for a long period. It has been a choice of many Sri Lankan's writing style in casual conversations such as small talks, chats and social media comments. Finding a method to translate Singlish to Sinhala or English has been tried for a couple of years by the research community in Sri Lanka and many of the attempts were tried based on statistical language translation approaches due to the challenge of finding a large dataset to use Deep Learning approaches. This research addresses the challenge of preparing a data set to evaluate deep learning approach's performance for the machine translation activity for Singlish to English language translation and to evaluate 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 ~0.26 M parallel sentence pairs with 50 K+ word vocabulary.**

Keywords— **Singlish, NMT, Language processing, Seq2seq, LSTM, Attention model, word embedding**

I. INTRODUCTION

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 popular writing system in the world today. It can be observed that this particular writing method has become an alternative writing system, especially in social media for some of the languages such as Hindi, Tamil, Urdu, Serbian and Bosnian. Many researchers are currently working on building models to analyze alternative writing systems as it has been trending in social media [1], [2]. In the Sri Lankan context, we have also seen that people tend to write Sinhala in Latin Script (English Alphabet) most of the times, and when they communicate with natives, and they call it "Singlish". The most significant issue of using Singlish is the unavailability of a standard way to write Singlish. In most of the occasions people use their own choice of ways to write Singlish. But it can be observed that the Singlish is a way of writing the Sinhala pronunciation with English alphabet.

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

Correspondence: Dinidu Sandaruwan[#1] (E-mail: iamdinidu@gmail.com)
Received: 03-05-2021 Revised:26-07-2021 Accepted: 28-07-2021

Dinidu Sandaruwan[#1], Sagara Sumathipala[2], Subha Fernando[3] are form Department of Computational Mathematics, University of Moratuwa Sri Lanka. (iamdinidu@gmail.com, sagaras@uom.lk, subhaf@uom.lk)

DOI : http://doi.org/10.4038/icter.v14i3.7230

II. 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 are included in the text. Furthermore, sometimes a Singlish word could be an English word already existing in its vocabulary with a different meaning. One of the challenges is to differentiate the words by language. Most of the times, it can be determined based on the context of the sentence and the position of it within the sentence.

- Diversity of writing

The Singlish writing pattern can be changed from person to person, based on how they spell Sinhala pronunciation in English alphabet. 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 an important requirement to develop a web crawler and additional supportive scripts to create a parallel dataset for training and testing. And also It is important to evaluate a language translation model for the translation activity for texts written in Singlish. Even though this has been tried for a couple of years, many of the attempts were based on statistical language translation approaches due to the challenge of finding a large dataset to use deep learning approaches [3].

III. 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.

IV. NEURAL MACHINE TRANSLATION

Before getting into the details, it might be worth describing the terms “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 and it is a way to learn more sophisticated functions to improve the accuracy of our probability estimates with less feature engineering. [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].

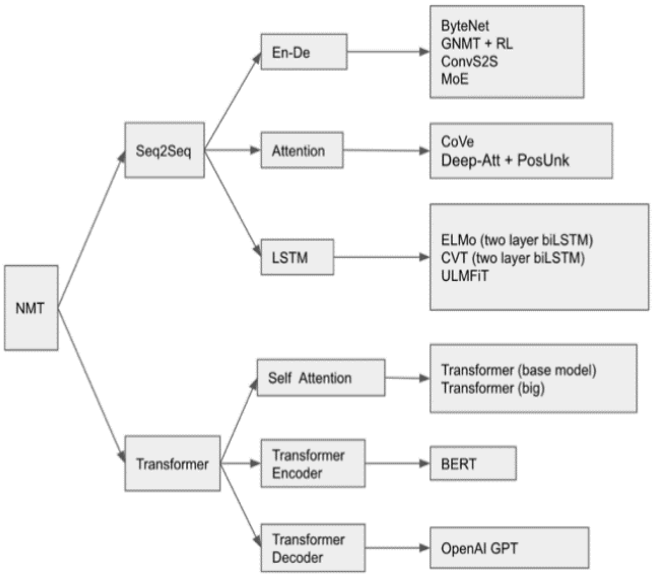

Figure 1: Classification of NMT models

As shown in the Figure 1, NMT can be categorized in to two main branches as “Seq2Seq” (sequence to sequence) and “Transformers”. When selecting an appropriate NMT approach for a particular language transtlation, It is important to discuss capabilities of each approach before making a decision to select one over the other.

A. Seq2Seq (Sequence to sequence)

Sequence-to-sequence (seq2seq) models have been a great success for various NLP tasks such as machine translation, speech recognition, and text summarization. Seq2Seq is relatively a new paradigm, with its first published usage in 2014 [12]. At a high level, a seq2seq 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.

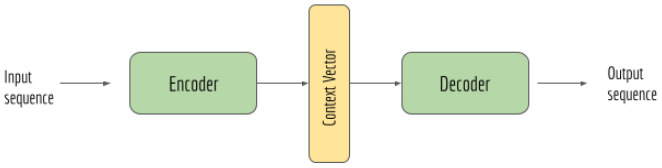

Figure 2 : Highlevel encoder decoder architecture

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.

B. *Transformer*

Seq2Seq models with Recurrent Neural Networks (RNNs) like Long Short Term Memory (LSTM) networks were becoming more popular and securing their well-earned reputation for quite a while until they were recently challenged by a new commer on the field of machine translation something called a Transformer [13]. It's a very innovative concept and is addressing two major weaknesses in RNNs:

- RNNs are not parallelizable as the output of the particular step mainly depends on the output of the previous step.
- RNNs struggle to maintain long-term language dependencies because it sees only the memory from the previous step.

In order to get a better understanding of the Transformer model, Let's look at the architecture of a Transformer model which is built and trained to do a translation of a given sentence in a particular language to another language. Note that here we will not be discussing all the bits and pieces of the Transformer model, but just enough to understand how it differs from the Seq2Seq model.

The transformer is also an encoder-decoder model. Encoder and decoder consist of several layers. Each layer is made up of two types of sub-layers such as self-attention and fully connected layers. In addition, the decoder layer must contain a Softmax layer, since it must generate probabilities for the vocabulary of the target language for each position. The self-attention layer is the revolutionary concept of the transformer model as it allows the model to look at all other words while processing a single word in the sequence. There's no specialty around the fully connected layer as simply it takes the outputs of the separate self-attention layer and creates a hidden representation for each word using a fully connected layer.

As we can see, none of the sublayers have sequential computations and repeating units waiting for the output of the previous step like it does with LSTM. This reduces the need for the model to maintain a memory state like in LSTMs. The transformer can thus calculate the outputs for all time steps at the same time. As we can see, at some point the self-attention sublayer also sees all the other inputs. Because of this, it becomes trivial to have long-term dependencies on long chunks of text.

By comparing Seq2Seq model with Transformer model, Its very obvious that the Transformer model addresses couple of main drawbacks in Seq2Seq. And it's been consistently shown that Transformer models almost always outperform sequential models. But the question is can this transformer models be applied in the context of every machine translation environment as it is known to be a heavy model which does not support in low resource environments. The original transformer models are quite large. And also it requires comparatively a large data set which is again challaging to be used in low resource languages.

Eg:- BERT (Bidirectional encoder representation of transformers)

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 model deals with 300M parameters and that limits the ability to use this model in low resource environments.

But simple seq2seq models with LSTMs can be used at a fraction of memory of this massive models occupy. Seq2Seq models are in other hand easy to prototype and understand. With compre to a transoformer model, setting up a seq2seq model is comparatively earier. If the focus is to do a feasibility study of a machine translation for a language pair which previously have not tested with any NMT approach, Seq2Seq model is a much better approach to get started with.

C. *Seq2Seq model with Attention Mechanism*

After the further study of Seq2Seq model, Found some advance features that can be used along with the Seq2Seq models to improve the performance of the model. the **"attention mechanism"**, which was first introduced by Bahdanau [6], then later refined it further by Luong in 2015 and others [15]. The main idea of the attention mechanism is to create direct links between target and source by "paying attention" to relevant source content during translation. The attention alignment metrix is a byproduct of the attention mechanism which is an easy way of visualizing the alignment between the source and target sentences.

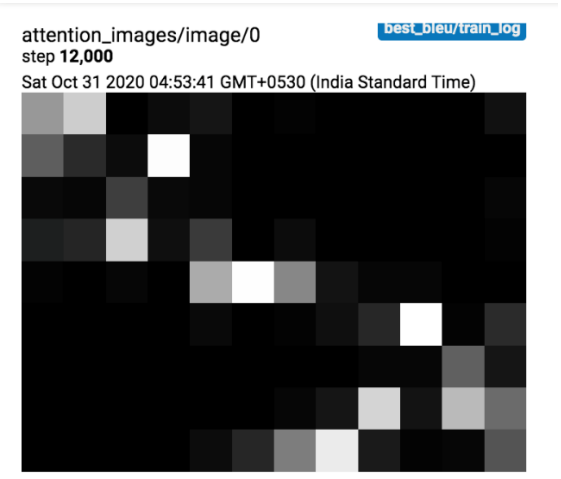

Figure 3 : Alignments between source and target sentences (Singlish to English)

It is important to look at some other popular machine transtalions other than the two main branches we discussed so far in this paper. One out of other approaches which known to be popular was the Meta Learning approach. MetaMT is one of the models that is popular among low resource language translations.

D. 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 a training set of 0.21 M sentence pairs while Fi-En results in 20.20 with 2.63 M pairs.

After going through all positives and negatives of the existing machine translation approaches, We decided to select Seq2Seq approach with attention mechaniusm to test this Singlish to English translation activity.

V. APPROACH

The 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. The 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].

Preparing a dataset was one of the biggest challenges of this research since there are not many resources available for Singlish. To prepare the required dataset, we selected IWSLT'15 English-Vietnamese parallel data set published by Stanford University [17], obtained English sentences in the dataset to generate Sinhala translation with Google translator API and prepared the Singlish data set parallel to the original English sentences with Google pronunciation API. Once the data is prepared, the additional script was developed to clean the data generated from the pronunciation API. Even though this is a dataset generated synthetically, it is close enough to the way how people write in Singlish. However, one can choose to write in this way if he thinks about how Sinhala pronunciation can be written with English letters. Following example 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.

VI. DESIGN AND IMPLEMENTATION

In this design, a deep multi-layer RNN is considered, which consists of bidirectional LSTMs as recurrent units. An example of such a model is depicted in Figure 2. In this example, shows how the source sentence **"mama gedara yanawa"** is translated into a target sentence **"I go home"**. At a high level, the neural machine translation model consists of two recurrent neural networks, the encoder RNN consumes the input source words without any predictions. The decoder RNN, on the other hand, processes the target sentence by predicting the next words.

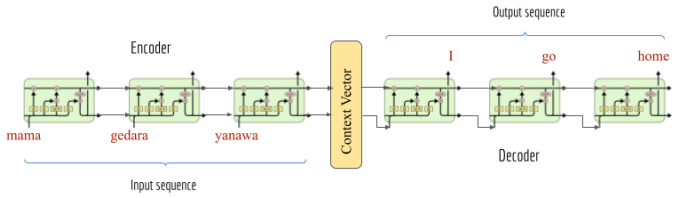

Figure 4 : seq2seq NMT high level architecture

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

Embedding :

Given the categorical nature of a word, the model must first 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.

Encoder :

Then 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)
```

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.

```
# 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
```

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.

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.

In Figure 4, 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.

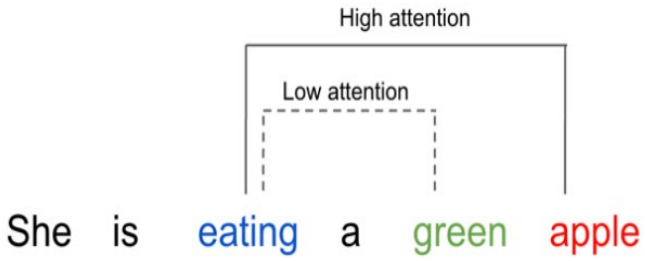

Figure 5 : Attention Mechanism.

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 [13], [18].

Attention Weights :

$$a_{ts} = \frac{exp\left(score\left(h_t, \bar{h}_s\right)\right)}{\sum_{s'=1}^{s} exp\left(score(h_t, h_{s'})\right)}$$

Context Vector :

$$c_t = \sum_s \alpha_{ts} \bar{h}_s$$

Attention Vector :

$$a_t = f(c_t, h_t) = tanh(W_c[c_t; h_t])$$

Score Function :

$$score\left(h_t, \bar{h}_s\right) = \begin{cases} h_t W \bar{h}_s \\ v_a \tanh\left(W_1 h_t + W_2 \bar{h}_s\right) \end{cases}$$

Here, the function **score** is used to compare the target hidden state $\boldsymbol{h_t}$ with each of the source hidden states $\bar{\boldsymbol{h}}_s$, and the result is normalized to produced attention weights (distribution over source positions). There are various choices of the scoring function, popular scoring functions include the multiplicative and additive forms given in **score function**. After calculation use the attention vector $\boldsymbol{a_t}$ to derive logit and softmax loss. This is similar to the hidden state of the target in the top layer of the Vanilla Seq2seq model. The function **f** can also take other forms.

BEAM Search

Finally, 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. But the problem is, there is a high chance of leading it to sub-optimal output sequences. Computationally also, it is inefficient. One recommended way to deal with this issue is to use Beam Search [18]. 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. It is still a greedy search algorithm, but a lot less greedy than the previous one as its search space is larger. While greedy search can provide us with quite reasonable translation quality, beam search decoders can further improve performance. The idea of beam search is to better explore the search space for all possible translations by keeping a small group of the best candidates when we translate. The beam size is called the beam width. The minimum beam width, for example a size of 10, is usually sufficient. In this research, we tested with N=10 as the beam size.

VII. EVALUATION

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

TABLE I
BLEU RESULTS

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

(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, the step-time refers to the time it takes to run a mini-batch (size 128). For WPS, we count words on both the source and target.

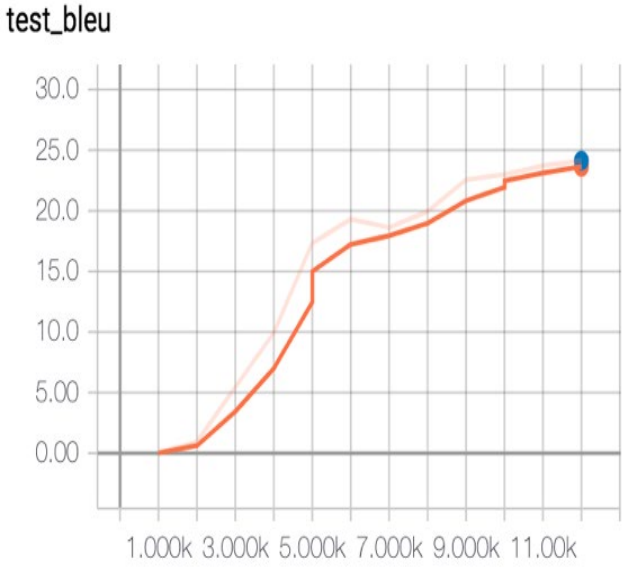

Figure 6 : Test BLEU Score Graph

VIII. CONCLUTION

It can be observed from the evaluation of the results that we have achieved significant 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 decent dataset for this translation activity. The main problem with the approach of generating sysnthetic data set is, This translation works only for the Singlish sentences written in a particular scheme. But in real scenario, people use different schemes to wite words in Singlish. In other words, People write the same word with different spellings. This scenario can be consider as an unseen scenario for a given word. The word has not been seen in the training phase nor included in the vocabulary. This is a similar scenario of misspelled word.

Eg:- ඉදිරියට = Idiriyata, Idiriata.

In this particular scenario the model thinks this as two different words. If the model is trained with both the scinarios with samples and if both the words included in the vocabulary, the model will give the correct translation. Otherwise it will give correct translation for a one scenario and while giving the word prediction as unknown for the other scenario.

Scinario 1 (Seen word) :

Singlish : Eyala idiriyata ena wada gana katha kara

English : They talked about the work ahead

Scinario 2 (Unseen word) :

Singlish : Eyala idiriata ena wada gana katha kara
English : They talked about the work <unknown>

Handling unseen word or misspelled word in this translation model will potentially solve the above problem. Then the model will consider "Idiriyata" and "Idiriata" as the same word.

This is an initial stage of the research domain of analyzing, translating alternative writing systems used in Sri Lanka. The most significant achievement is putting a step ahead to use the latest and greatest deep learning approaches in the machine translation domain. The good news is that this research opens up doors to several new research paths to continue forward by addressing the limitations of handling different schemes of writing styles and the model development and improvement for Singlish-English machine translation.

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] A. M. Silva and R. Weerasinghe, "Example Based Machine Translation for English-Sinhala Translations," p. 10.

[4] J. Hutchins, "Machine Translation: History," in *Encyclopedia of Language & Linguistics*, Elsevier, 2006, pp. 375–383. doi: 10.1016/B0-08-044854-2/00937-8.

[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] T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp. 1412–1421. doi: 10.18653/v1/D15-1166.

[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.