A Novel Approach for Tamil - English translation and vice versa using RNN

R.Kasthuriraajan and Dr.S.Mahesan
Department of Computer Science, Faculty of Science, University of Jaffna
kasthuriraajan94@gmail.com

Abstract

This study focused on improving a better approach for Tamil-to-English translation and vice versa using RNN. End of the study a novel approach for Tamil-to-English translation and another for English-to-Tamil translation were found to build a Neural Machine Translation system. Here optimizers and bridges had an impact on performance. BLEU scores were used to measure the performance of the system. Finally, the best performing model for Tamil-to-English translation was obtained with a BLEU score of 8.13. The best performing model for English-to-Tamil translation was obtained with a BLEU score of 4.66 which outperforms Google translator that has the score of 4.06. It shows that models with less number of layers can perform better than a high number of layers in terms of computing power while using appropriate optimizers and bridging technologies.

Introduction

Nowadays people unavoidably needs to use machines for translation purposes. In order to meet this need, the studies on machine translation systems emerge in recent years. Big companies like Google, Microsoft also take much effort into building efficient machine translation systems. There are several machine translation techniques such as rule-based translation techniques, and statistical machine translation techniques. Even though Neural Machine Translation (NMT) is getting attention because of its accuracy and behaviour like human translation. So it is an active research topic all over the world. Google Neural Machine Translation (GNMT) system is a well known NMT system that was introduced in 2016 and used in Google translator by Google. More than 100 languages are supported by Google translator including Tamil and English. However, there is a need for many improvements in its performance. So this topic was

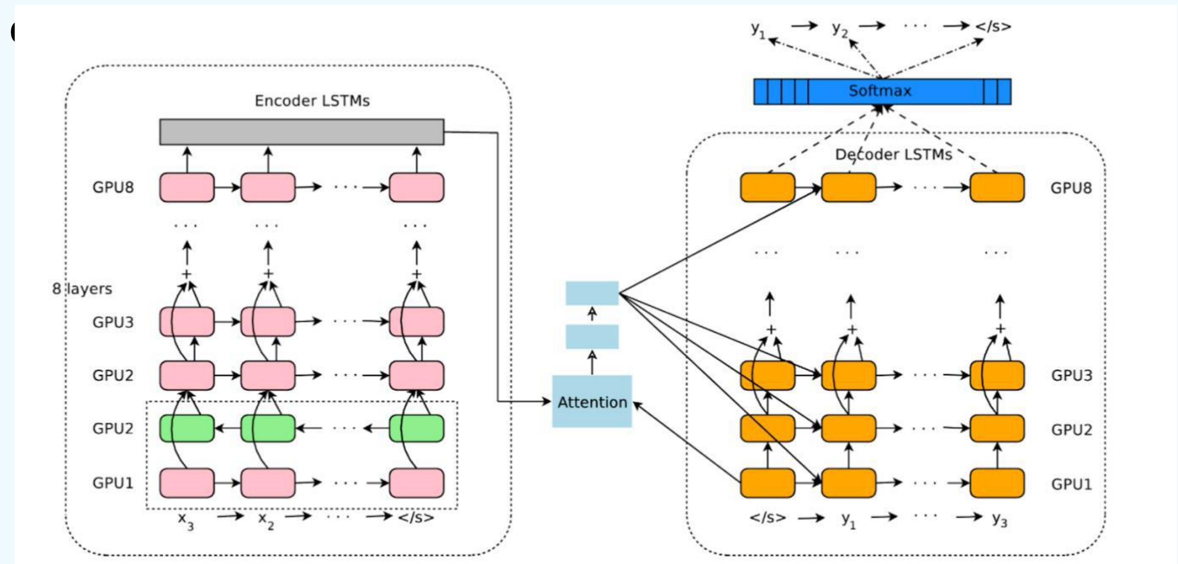

Figure01: GNMT architecture

Objective

The objective of this research project is to build a neural machine translation system for Tamil to English and vice versa using recurrent neural network (RNN).

Methodology

There are three major components in this study.

1)Pre-processing the dataset
2)Training Neural Machine Translation models with training dataset and validation dataset.
3)Testing the trained models with the testing dataset and obtain results.

- Recurrent Neural Network (RNN) was selected to build neural machine translation models in this research project.
- A publicly available Tamil-to-English parallel corpus from various domains (EnTam V2) which was compiled by Loganathan Ramasamy was used for this study.
- Byte Pair Encoding (BPE) was selected to learn encoding and applied to all datasets except the target test data. Vocabulary was created from source and target datasets. All training and validation datasets were changed into torch tensors.
- In neural machine translation systems, the encoder-decoder mechanism is used to translate language pairs. First source language is encoded by RNN encoders and then RNN decoders decode them into target language.
- Long Short Term Memory (LSTM) was used in this research experiment to overcome the Long term Dependency Problem. Two layers of bidirectional LSTMs (Bi-LSTM) were selected as encoder with 500 hidden layers and two layers of LSTMs were selected as decoders.
- Two optimization methods were experimented here. One is ***adam*** with learning rate 0.001 and the other one is ***sgd*** with learning rate 1.0. A bridge is an additional layer between an encoder and decoder that defines how information is passed from encoder to decoder. Here two models were trained with bridge and two models without bridge.
- Finally, the translated sentences were compared with target test data. Using the BLEU scoring system, the accuracy of each model was measured and compared with each other.

Figure02 :Basic Architecture of the System

Experimental Setup

In this research project, OpenNMT-py was used to do experiments. This is a research-friendly Pytorch port of OpenNMT which is an open source (MIT license) ecosystem for neural machine translation and neural sequence learning.

The following models were created with the help of OpenNMT-py. Word vector size for source and target was defined as 500. Every model was trained with 2 layers Bidirectional RNN, 2 layers of RNN decoders, 500 hidden layers, 100,000 training steps and ***mlp*** attention type with the specific attributes mentioned below. After every 5000 steps, the trained models were saved.

Table 01:Models

	Model 1 (EnTa)	Model 2 (EnTa)	Model 1 (TaEn)	Model 2 (TaEn)
RNN type	LSTM	LSTM	LSTM	LSTM
Optimizer	adam	sgd	adam	sgd
Bridge	False	False	True	True

Table 02:Data set

Data Set	
Training Data	166,871 Sentences
Testing Data	2,000 Sentences
Validation Data	1,000 Sentences

Results

The test data were translated using the trained models mentioned above with the help of OpenNMT-py. The translated corpus was then evaluated using BLEU scoring method. BLEU is measured out of 100 and the better performance will have higher score. According to the BLEU score, the best performing trained models among which were saved after every 5000 steps, were selected in each model. The test data were also translated by Google Translator and evaluated with the BLEU scoring method.

Table 03: English-to-Tamil translation results

Model	BLEU Score
LSTM+mlp+adam(EnTa)	4.59
LSTM+mlp+sgd(EnTa)	4.66
GNMT(Google Translator)	4.06

Table 04:Tamil-to-English translation results

Model	BLEU Score
LSTM+mlp+adam+bridge (TaEn)	8.13
LSTM+mlp+sgd+bridge (TaEn)	7.81
GNMT(GoogleTranslator)	21.16

The OpenNMT-py frame work is available at
https://github.com/OpenNMT/OpenNMT-py

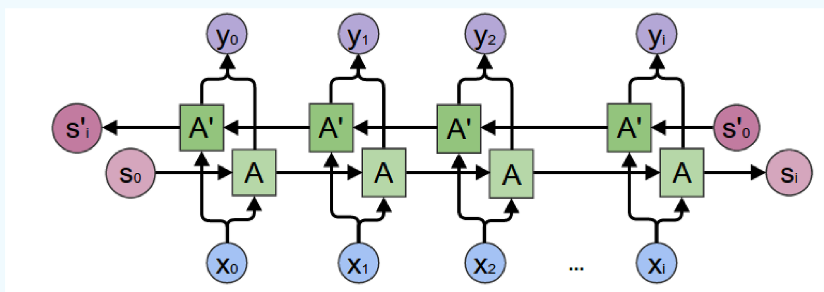

Figure03 : A bidirectional RNN

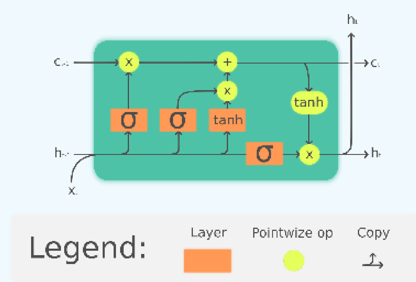

Figure04 :An LSTM Cell

Discussion & Conclusion

Some Neural Machine Translation systems use more layers in their model and a big parallel corpus for training (E.g. GNMT uses 8 layers). But here only two layers were used with a limited number of parallel corpora and a better performance was gained.

Our best performing English-to-Tamil translation model gained a BLEU score of 4.66 and Tamil-to-English gained a BLEU score of 8.13.

We could thus conclude that an NMT system can be implemented using this technique with low resources

References

1. Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144, 2016.
2. Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio,Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473v7 [cs.CL], last revised 19 May 2016.
3. Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, Yoshua Bengio.On the Properties of Neural Machine Translation: Encoder-Decoder Approaches.arXiv:1409.1259 [cs.CL], 7 Oct 2014.