Artificial Intelligence Final Report Assignment 問題 3 (Problem 3)
レポート解答用紙（Report Answer Sheet）

Group Leader
学生証番号（Student ID):18521068
名前(Name):Trần Bình Luật

Group Members
学生証番号（Student ID):18521284
名前(Name):Huỳnh Ngọc Quân

学生証番号（Student ID):18521285
名前(Name):Nguyễn Minh Quân

問題 3 （Problem 3)のレポート

# I. Introduction

## 1. Machine translation

Machine translation is an automatic translation from one language to another. The benefit of machine translation is that it is possible to translate a large corpus of text instantly. Given a sequence of text in a source language as an input, the output is a translated sentence in the target language. But there is no best translation of that text. This is because of the flexibility of human language. This makes the challenge of automatic machine translation difficult, maybe one of the most difficult in artificial intelligence.

There are two main types of machine translation: Statistical machine translation, or SMT for short, and Neural machine translation, or NMT for short. SMT finds the most probable translation in the target language by some of the formulas which are difficult to understand. In this work, we focus on NMT, which has been proven to be much more effective than its predecessor. NMT attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation.

## 2. Dataset

Created by Stanford in 2015, the IWSLT 15 English-Vietnamese Sentence pairs for translation., in Multi-Lingual language. The dataset was split into 3 parts: train, test, validation. The distribution of each part is train set with 133,317 sentences, validation set with 1553 sentences, test set with 1268 sentences. Here is an example of a sentence pair for translation.

| | |
|---|---|
| **English** | Rachel Pike : The science behind a climate headline. |
| **Vietnamese** | Khoa học đằng sau một tiêu đề về khí hậu. |

Table 1. An example in IWSLT15

## II. Method

### 1. Architecture

We will use Bi-LSTM as an **Encoder** and **Decoder** in **Seq2Seq model** to solve this problem. Bi-LSTM effectively increase the amount of information available to the network, improving the context available to the algorithm and this is why we chose it.

A Bi-LSTM architecture typically contains two single LSTM networks that are used simultaneously and independently to model the input sequence in two directions: left-to-right (forward LSTM) and right-to-left (backward LSTM). It means the model will take information from the past and future.

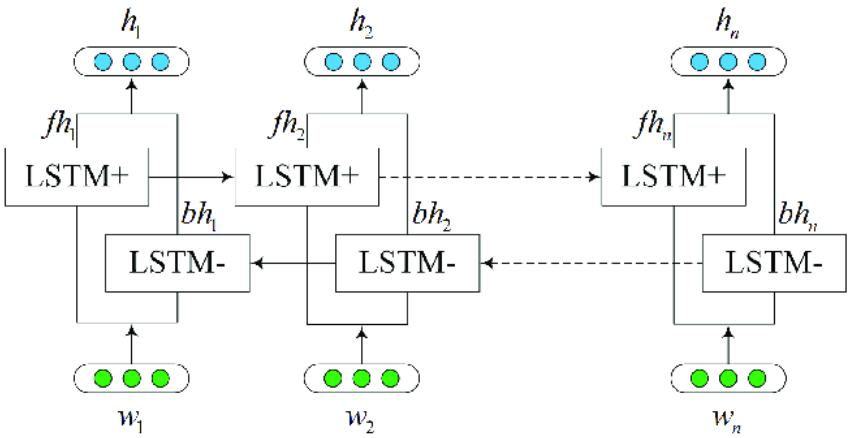


Figure 1. A basic architecture of Bi-LSTM (Source: ResearchGate)

The Encoder will encode the sentence word by words into an indexed of vocabulary or known words with index, and the decoder will predict the output of the coded input by decoding the input in sequence and will try to use the last input as the next input if it is possible. With this method, it is also possible to predict the next input to create a sentence. Each sentence will be assigned a token to mark the end of the sequence. At the end of prediction, there will also be a token to mark the end of the output. So, from the encoder, it will pass a state to the decoder to predict the output.

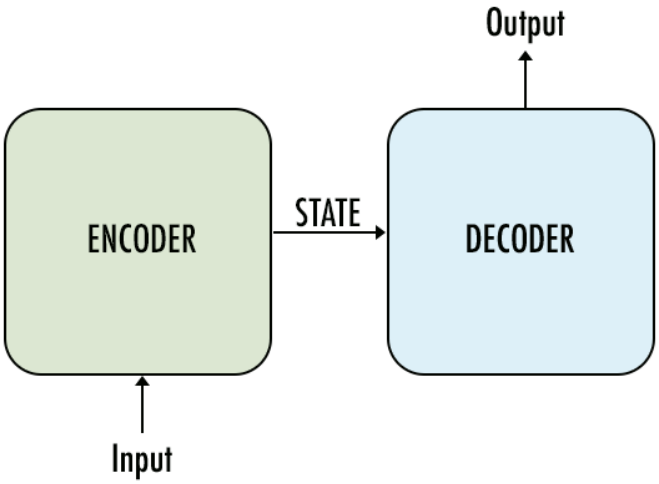


Figure 2. Encoder and Decoder. (Source: Seq2Seq Model )

Decoder will decode the input from the encoder output. It will try to predict the next output and try to use it as the next input if it's possible. The Decoder consists of an Embedding layer, Bi-LSTM layer, and a Linear layer. The embedding layer will make a lookup table for the output and passed into a Bi-LSTM layer to calculate the predicted output state. After that, a Linear layer will help to calculate the activation function to determine the true value of the predicted output.

### 2. Configurations

Our configuration is based on LAB 6. We adjust LSTM to Bi-LSTM and set the numbers of Bi-LSTM layers to 3. The hidden size is 512, dropout is 0.5 in both Encoder and Decoder, all of them are the same as LAB 6. But we change the dimensions of embedding from 256 to 512. We decrease the batch size to 64 for saving GPU, if not the Colab will crash because of 3 Bi-LSTM layers. In addition, we add learning rate decay of 20% per 5 epochs for a better minimalization step. Finally, 20 epoch is enough.

## III. Experiments

When we started to solve this problem, we first adapted the LSTM to Bi-LSTM to improve the context (2 layers of Bi-LSTM). It worked, increasing the Bleu score from 5.5 up to 8.4. We thought a multilingual word embeddings might help and decided to use MUSE from Facebook Research to replace the embedding available from Pytorch. However, it seems that we did something wrong so the results didn't increase but even got a little worse. We tried adjusting the batch size, numbers of feature up and down, but the results didn't change too much. We then tried to adjust the numbers of Bi-LSTM layers from 2 to 3 and Colab crashed because the batch size was at 128. We didn't realize that and proceeded to add learning rate decay. Reducing

the learning rate after many epochs improves a little, about 0.3 bleu. Adjusting the dropout only makes the results worse so 0.5 is the best choice. Things did not go well because, after many experiments, the results were still not as expected. But we believe our model can reach 10 and keep going. And finally, after realizing that reducing batch size can save quite a bit of GPU, we reduced it to 64 and increased the number of Bi-LSTM layers to 3. The model now achieves 9.7 bleu. Changing the embedding dimension from 256 to 512 we got 10 Bleu on epoch 20th. We also tried increasing the number of Bi-LSTM to 4 but the results did not improve. We also tried increasing the number of epochs to 30 or even 50 and found that the loss decrease but the bleu did not increase, the model was overfitted. We, therefore, decided to keep 20 epochs.

## IV. Conclusions Future Work

In this task, actually, we encountered a lot of problems. We can easily leverage some available models out there on the internet. But we think it is more important to understand the fundamentals, know how to fine-tune the model to achieve good results. This is also the foundation for us to easily use more modern things in the future like attention or transformer. This task gave us a lot of inspiration because we were able to solve one of the most difficult problems in NLP with our own hands. We definitely study more and more about NMT in the future.