Homework 4

Question 1.

1.a) Why do we need embedding?

Embeddings are low dimensional vector representation. There are two main reason we need embedding.

Embedding reduces the dimensionality of categorical variables. If we use one hot encoding and we have dictionary of 5000 words. Each word will be represented by a vector containing 5000 integers. Except one, all the vectors values will be zero. One hot encoding scheme is computationally inefficient for big datasets. It also meaningfully represents categories in the transformed space. Embeddings also meaningfully represent categories in the transformed space. The vectors of each embedding get updated during network training. Similarities between words can be found in a multi-dimensional space.

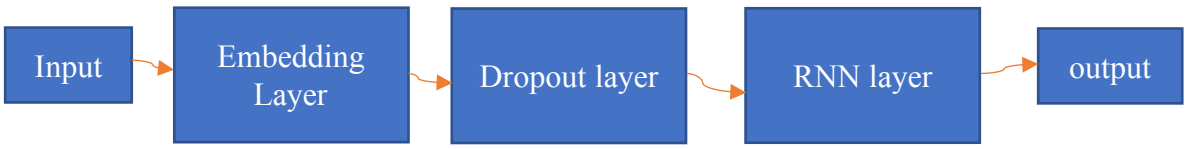
1.b) Explain the idea of gated models and the main differences between GRU and LSTM.

Vanilla RNN suffers from short term memory, it's difficult from model to carry information from earlier time steps to later time steps. This problem is termed is gradient vanishing problem. To overcome this problem, gated models are used. These gated models contain memory cells and gates which carry information through timestamps and control and regulate the flow of information to the cell states.

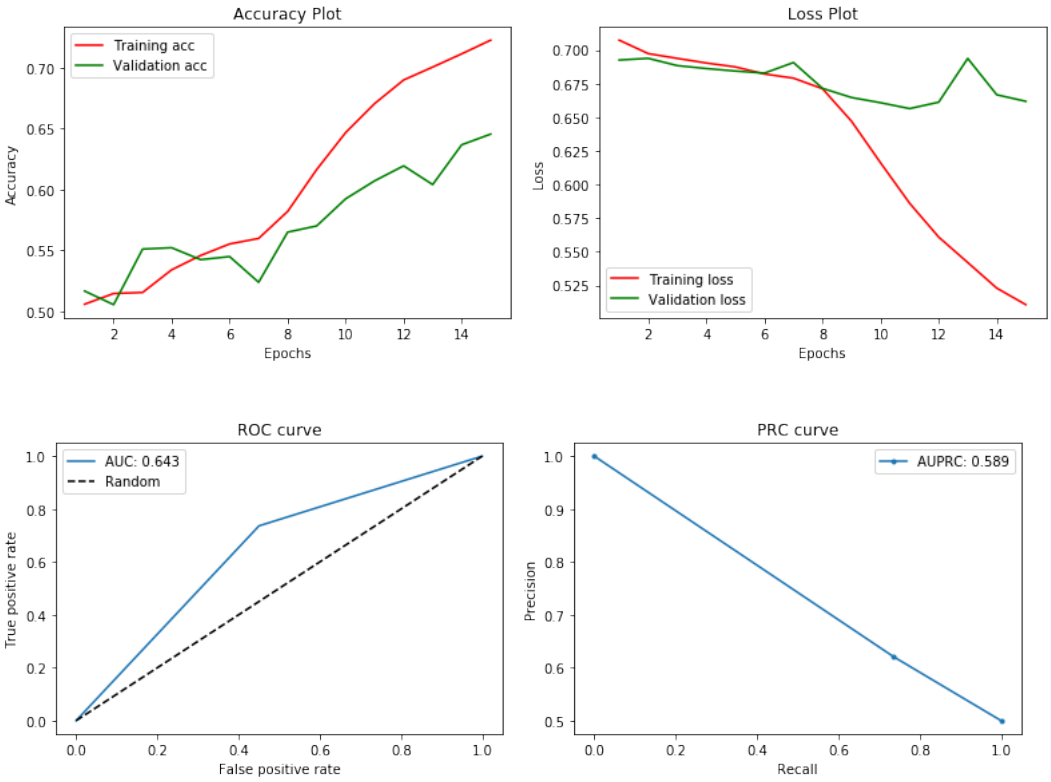
Two important gated models are GRU and LSTM. Both models contain memory cell, gates and retain information. LSTM has three gates, input gate, forget gate and output gate. The input gate controls that how much of the state of new cell to keep. Forget gate controls the forgetnes of the current memory. The output gate controls the state of cell that is to be forwarded to next layer. On other hand GRU has a reset gate and an update gate. Reset gate reside between the activations of preceding and the following candidates and its purpose is to forget previous state. The update gate controls how much of the activation is to use in updating the cell state. GRUs disclose the entire state of the cell to other units while LSTMs regulates the content of memory that is to be disclosed.

1.c) Vanilla RNN.

A model of vanilla RNN was trained. The model has embedding layer, dropout layer and a single recurrent neural network layer. The following figure shows architecture of the model.



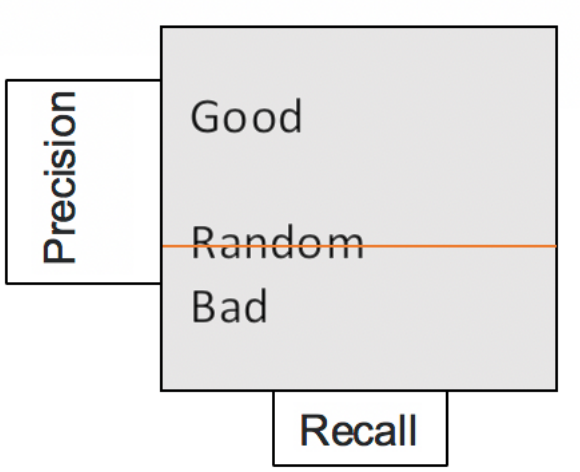
The model was trained only for 15 epochs. As beyond 15 epochs we are getting overfitting. The training accuracy of this model is 72 and validation accuracy (test accuracy) is 64. The Training loss is 0.51 while validation loss is 0.66. Plots of model accuracy (train and test), model loss (train and test), receiver operating characteristic curve (ROC) and precision recall curve (PRC) are shown below.



1.d). In Classification, if a fraction k of cases as is classified as positive then, because of the randomness, the equal fraction k of cases which is positive will be classified positive (true positives), and equal fraction k of cases which is negative will be classified positive (false positives).
So, we get a straight line in the ROC curve because the true positive rate and the false positive rate are the same.

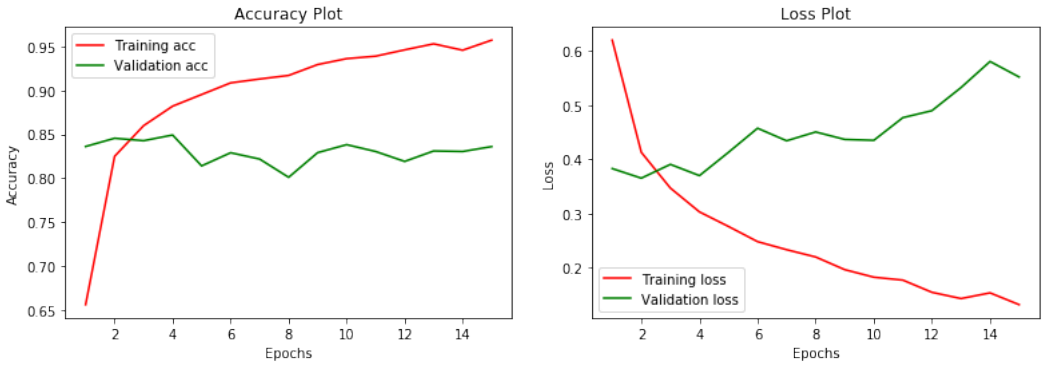
1.e). In PRC case a classifier with the random guess shows a horizontal line. The precision/recall space is divided into two parts. The upper area shows

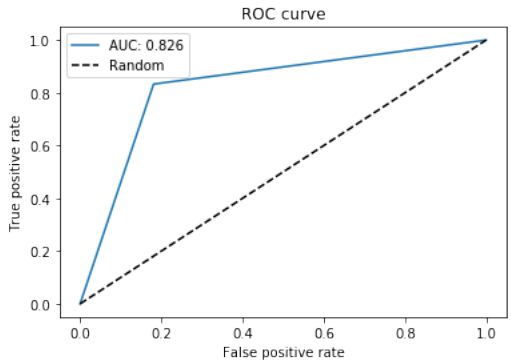
good performance level, the lower area shows bad performance level. Figure below shows the performance of random classifier.

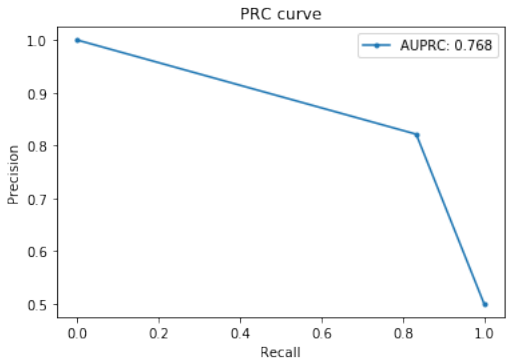


1.f). The model architecture which we used for Long and Short Term Memory (LSTM) and Gate recurrent unit (GRU) is the same as vanilla RNN. The motive behind using the same architecture was to see which model is performing better.

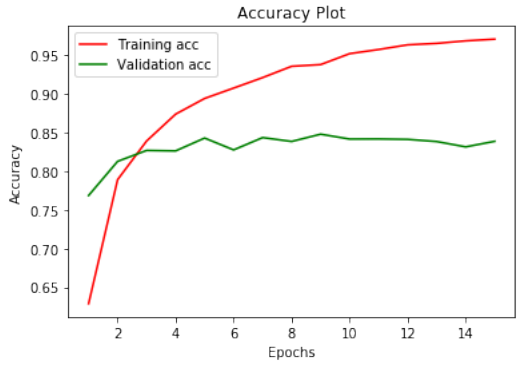
- First, we will discuss the LSTM. This model was trained for 15 epochs. The training accuracy of this model is 95 and validation accuracy (test accuracy) is 83. In terms of accuracy the LSTM outperforms the Vanilla RNN. The Training loss is 0.13 while validation loss is 0.55. if we look at loss plot we also see some overfitting in the model. The loss in this model is also very low as compared to Vanilla RNN. The receiver operating characteristic curve (ROC) value is 0.826 while precision recall curve (PRC) value is 0.768. Plots of model accuracy (train and test), model loss (train and test), ROC and PRC are shown below.

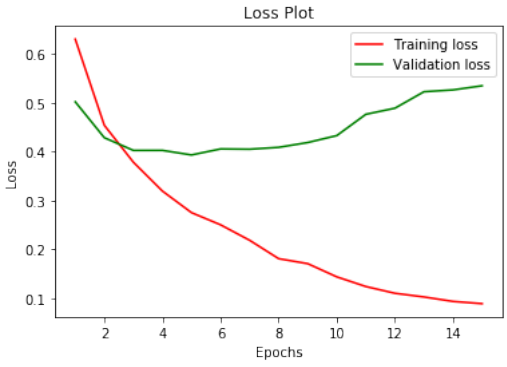


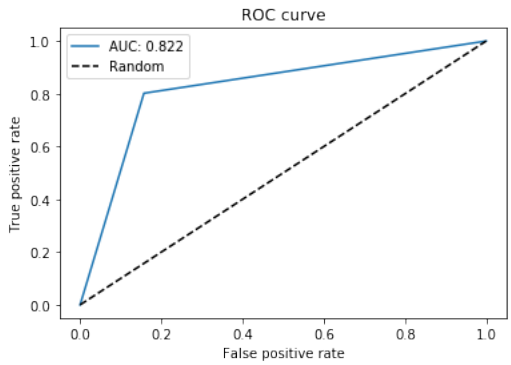


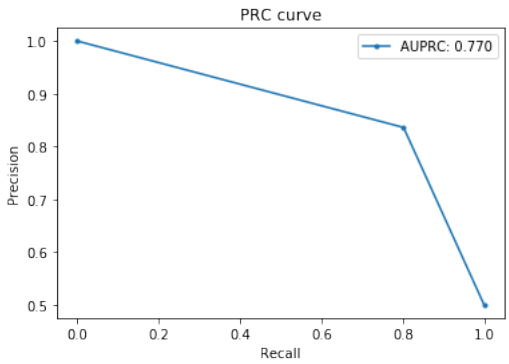


- The second model with the same architecture which we will compare with Vanilla RNN is Gated Recurrent unit. The GRU model was also trained for 15 epochs. The training accuracy of this model is 97 and validation accuracy (test accuracy) is 83. It is the almost the same as LSTM but far better than Vanilla RNN. In terms of accuracy the GRU also outperforms the Vanilla RNN. The Training loss is 0.08 while validation loss is 0.53. here in GRU model we also see slight overfitting if we look at the green line in the loss plot. The loss in this model is also very low as compared to Vanilla RNN. The receiver operating characteristic curve (ROC) value is 0.822 while precision recall curve (PRC) value is 0.77. Plots of GRU model accuracy (train and test), model loss (train and test), ROC and PRC are shown below.





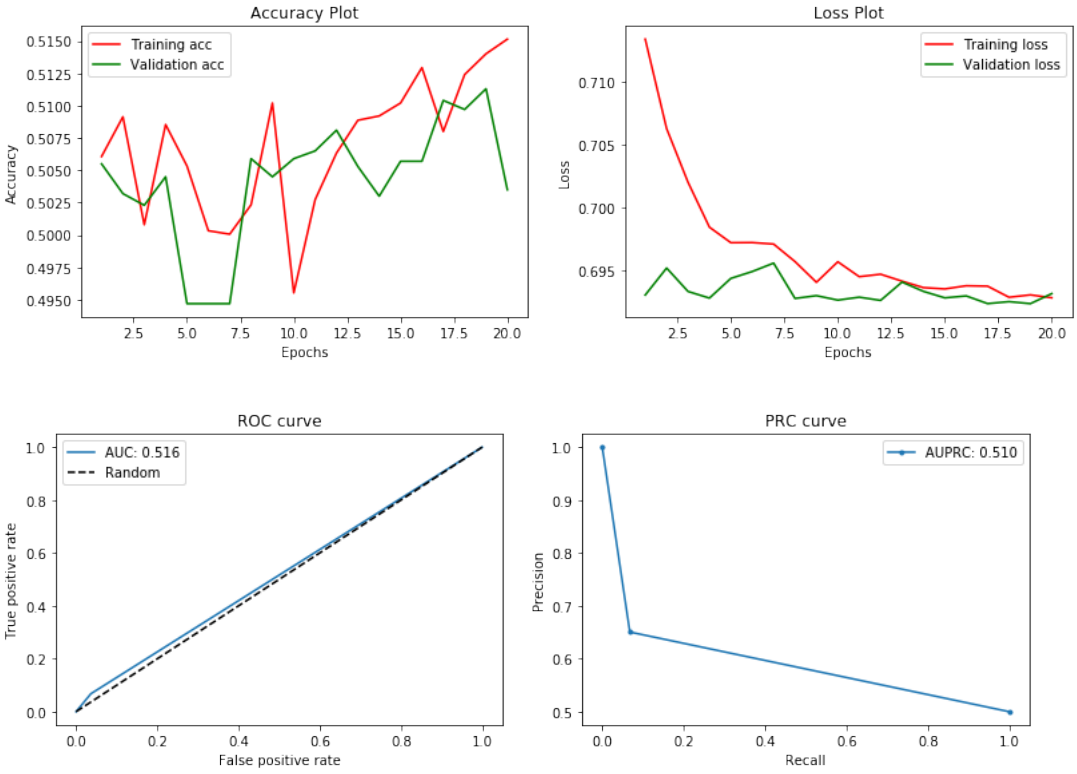




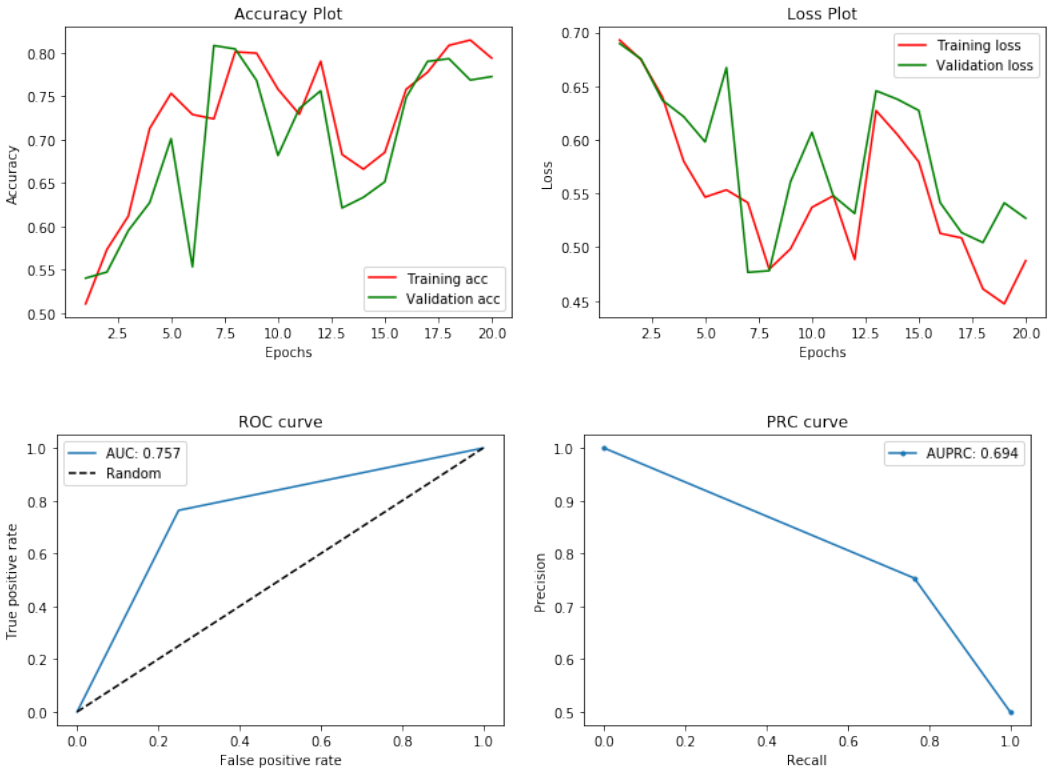
1.g). Maximum length of each review from 120 words to 256 words.

I kept the models in question c and f the same except that I changed the length of each review from 120 to 256. Also, I trained all of the models up to 20 epochs, to see whether the model performance is improved or not. Initially using 120 length we trained the models for 120 epochs.

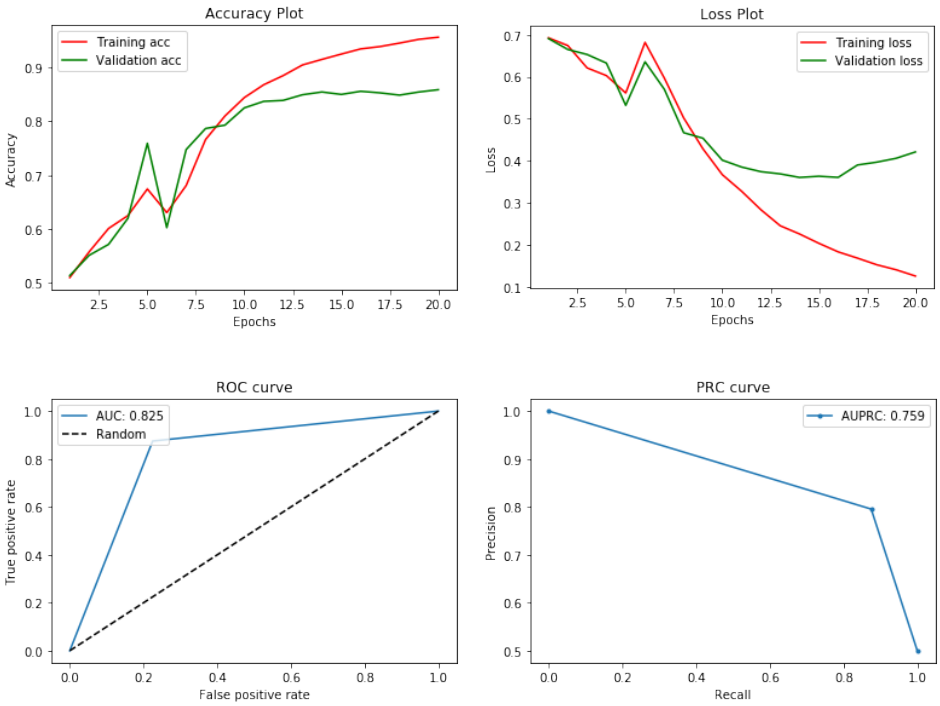
- First, I trained the Vanilla RNN. In plots below, we see there is fluctuations in both training and testing lines in the accuracy plot. The model seems unstable and is not training well. The training accuracy dropped to 0.51 and testing accuracy to 0.503. In the loss plot the Training loss is decreasing but training loss is fluctuating and is almost the same throughout the training. From the ROC plot we see that the model is behaving almost the same as random classifier as AUC value is 0.51. The PRC value also dropped to 0.51.



- The LSTM model was then trained changing the length to 256. Both accuracy and loss have huge fluctuations. If we look at trend it seem there is no overfitting till this point. The training accuracy dropped to 0.79 and testing accuracy dropped to 0.77. There is also a change in training and validation loss. The ROC and PRC values also dropped significantly. The ROC dropped from 0.82 to 0.75b while PRC from 0.76 to 0.51. These change in model performance can be seen from the plots below.

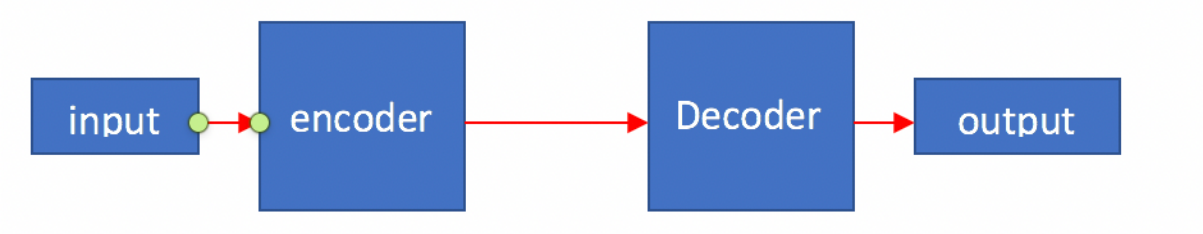


- The final model which we trained with 256 length is GRU. Compared to LSTM there is not much fluctuations but there are slight fluctuations in the beginning in comparison when 128 review length was used. The is not much change in accuracy and loss, the train accuracy decreased by 0.2 but the test accuracy has improved. The ROC has and PRC is also the same as earlier version of itself.

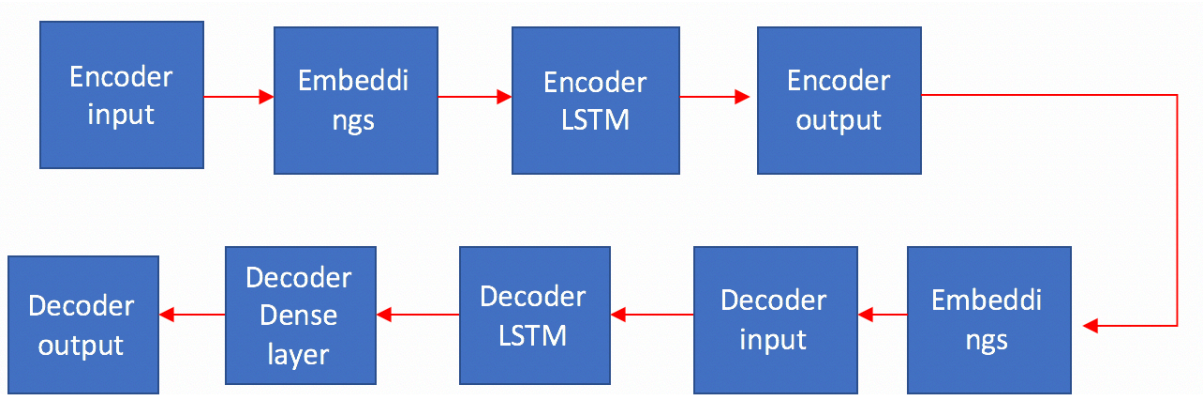


Question 2.

1.a) For any type of machine learning task we first preprocess the data. In our case both of our files contain paired sentences, so we don't need any terrible preprocessing. In our method we just read the files we convert characters to lower case and removed any quotation marks, if there is any, we also removed numbers and extra spaces from our files. Moreover, we attached a start and end token to the target sequences. Using NLTK library we also removed stop words. We also tokenized the words. After that we created the vocabulary for both English and French, vocabulary size and maximum sized of word. For Sequence to sequence model is mostly built architecture is encoder and decoder architecture. Below figure shows how an encoder and decoder architecture works.



The encoder and decoder module contain LSTM or GRU. In our architecture we have used LSTM model in both encoder and decoder. Below figure shows our seq2seq model. Our model takes the English file as input and fed it into an encoder which in our case has embedding layer and LSTM layer. The encoder encodes the input. the output of the encoder is then passed to the decoder, which consist of three layers, embedding layer, an LSTM and a dense layer. We train this Architecture back to back for translation from English to French.



The evaluation of this type of system is very difficult. For example. A person translates a sentence to "I am good" and another person translate the same

sentence “I am fine” or “I am feeling well”. All the sentences are correct. We cannot classify any of the sentence as correct or incorrect. To evaluate such a system, we need humans. Only humans can say that whether a particular translation is correct or not. To evaluate this system, we need human resources. One way to evaluate is to generate random sentences and look to its translation.

Some of the translation are given below.

```
new jersey is sometimes quiet during autumn , and it is snowy in april .
new jersey est parfois calme pendant l' automne , et il est neigeux en avril .
the banana is their favorite fruit , but the pear is our favorite .
la banane est leur fruit préféré, mais la poire est notre préféré.
the united states is sometimes busy during january , and it is sometimes warm in
november .
les états-unis est parfois occupé en janvier , et il est parfois chaud en novembre
china is sometimes dry during october , but it is never beautiful in spring .
La Chine est parfois sèche en octobre, mais elle n’est jamais belle au printemps.
```

2.b. Please find the translated file in attachment.