CSE 654 HW4 171044052 REPORT

My methods during this homework:

1)Firstly I cleaned punctuations,special characters, and made all my words to lower characters.

2)Then I merged my Ottoman Turkish file and Türkiye Turkish file line by line with tab. For example my first line is talebeler mektepe gitti öğrenciler okula gitti. Then I saved them in a text file and read them fro it to my raw dataset.I saved may sentences char by char.

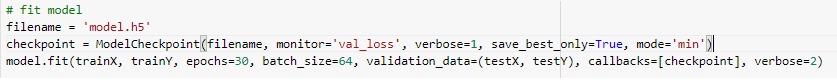
3)Firstly I created my dataset with 1000 then 5000 and one more 5000 for training.I used 900 for training and 100 for test, 4500 training and 500 for test.

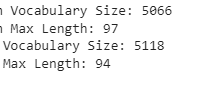
4)I used Keras Tokenize class to map words to integers, as needed for modeling. I separated tokenizer for the Ottoman Turkish sequences and the Türkiye Turkish sequences.Then I encoded Ottoman Turkish and Türkiye Turkish sequences and created train and test values.

5)Then I started define the model.I used encoder-decoder LSTM model on this problem.

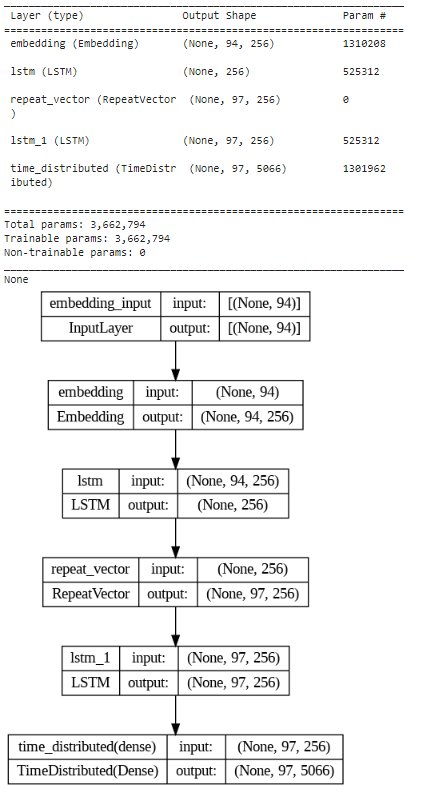
6)Finally I trained my model.For 1000 length dataset I trained the model for 30 epochs and a batch size of 64 examples. For first 5000 length dataset I trained the model for 10 epochs and a batch size of 64 examples.For second 5000 length dataset I trained the model for 30 epochs and a batch size of 64 examples

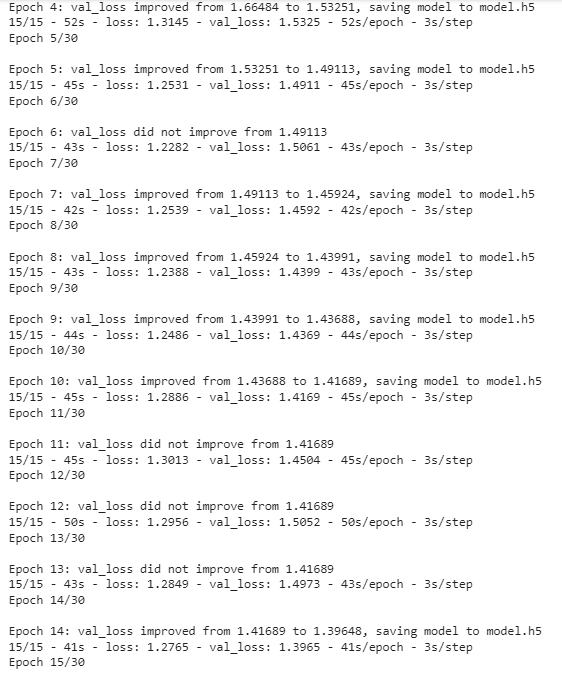
Results for 1000 length dataset:



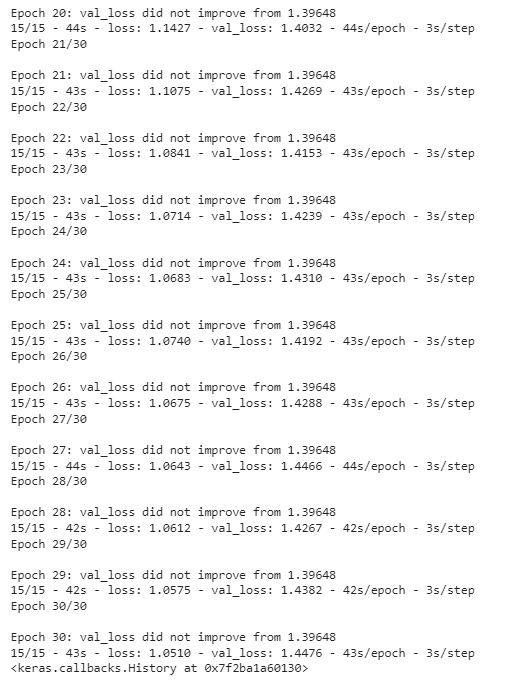


Plot of Model

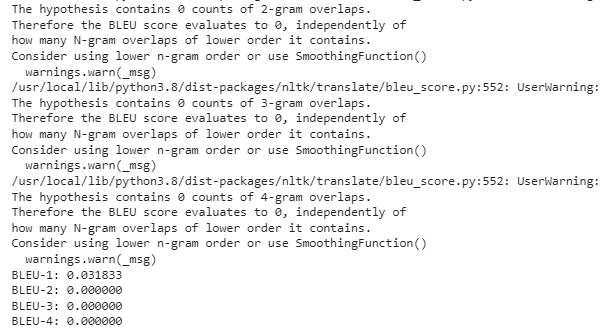


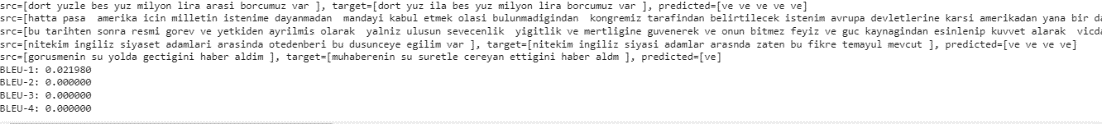


Epochs:

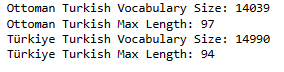


BLEU Scores And Results

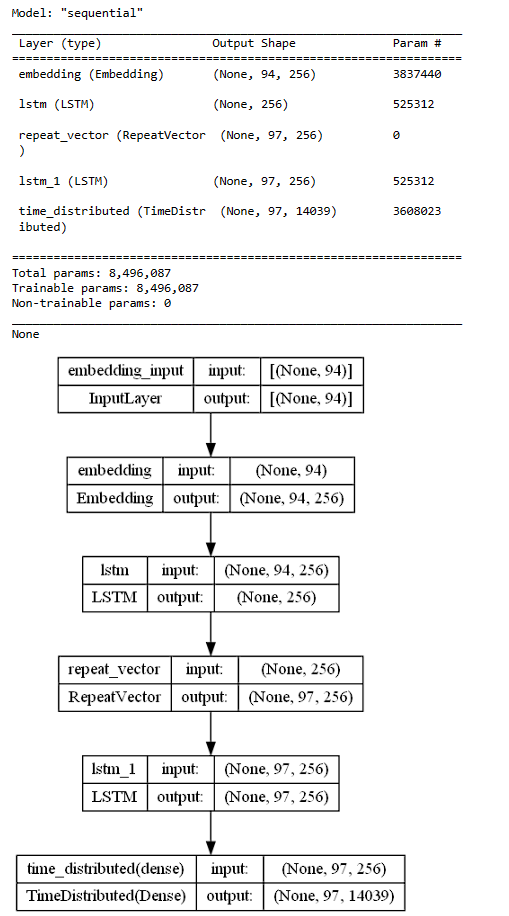




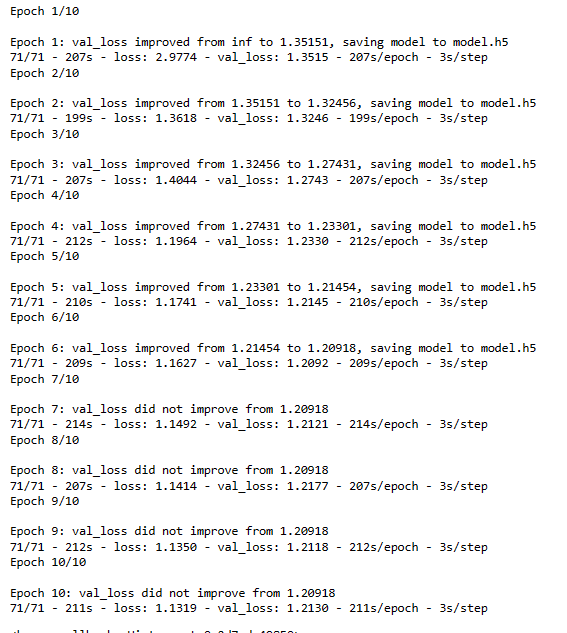
Results for 5000 length dataset:



Plot of Model



Epoch:

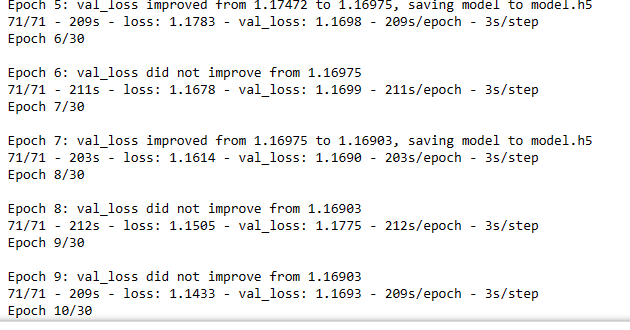
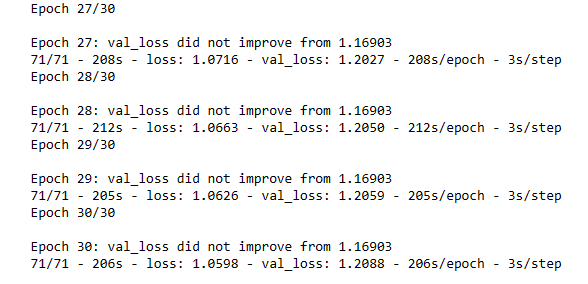


BLEU Scores And Results



Results for 5000 length dataset and 30 Epoch:

Only changing thing is epoch count.



BLEU Scores And Results



**Results**

My translation is not good. I could not train my model with huge dataset because of lack of good computer. I can train maximum 5000 length dataset. Above of 5000 I get ram error.

**My Method Details:**

I use an encoder-decoder LSTM model on this problem. In this architecture, the input sequence is encoded by a front-end model called the encoder then decoded word by word by a backend model called the decoder.

The function define\_model() below defines the model and takes a number of arguments used to configure the model, such as the size of the input and output vocabularies, the maximum length of input and output phrases, and the number of memory units used to configure the model.

The model is trained using the efficient Adam approach to stochastic gradient descent and minimizes the categorical loss function because we have framed the prediction problem as multi-class classification.