PROJECT REPORT

[Machine Translation between Specific Languages]

Members:

K213309 – Mohammad Yehya

K213349 – Talha Bilal

K213297 – Hasan Iqbal

Background

The motivation of this project was to construct a translation system based on deep learning architectures to translate English texts to French. The reason we took French in particular was because of the relative of the dataset (230000+ phrases). The dataset is obtained from http://www.manythings.org/anki/fra-eng.zip.

Methodology

Firstly, the dataset was split into 3 groups: train, dev, test sets. Only the train and dev sets are used during the training process. The dev set is used as a validation set after training. The test set is held out until the evaluation process. Since the dataset comprises of over 230,000 phrases, it is deemed not practical to use it all for this small project. The model was trained for 40 epochs with a batch size of 256. The model structure is based on an Encoder-Decoder architecture. The inputs are encoded using a fixed size word embedding, which is passed to a decoder which decodes it word by word to generate a probability value for each target word in the prediction. All the dataset were loaded from and encoded into sequences using the keras Tokenizer. Each English sentence is used as input and each French sentence is used as output. Each sequence was padded to the longest sentence in its language. The output is further one-hot encoded using to_categorical, as the model will predict the probability of each of the target word in the translation during prediction. Here is the summary of the models (both encoder and decoder):

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, None, 81)	0
lstm (LSTM)	[(None, 64), (None, 64), (None, 64)]	37,376

Total params: 37,376 (146.00 KB)
Trainable params: 37,376 (146.00 KB)
Non-trainable params: 0 (0.00 B)

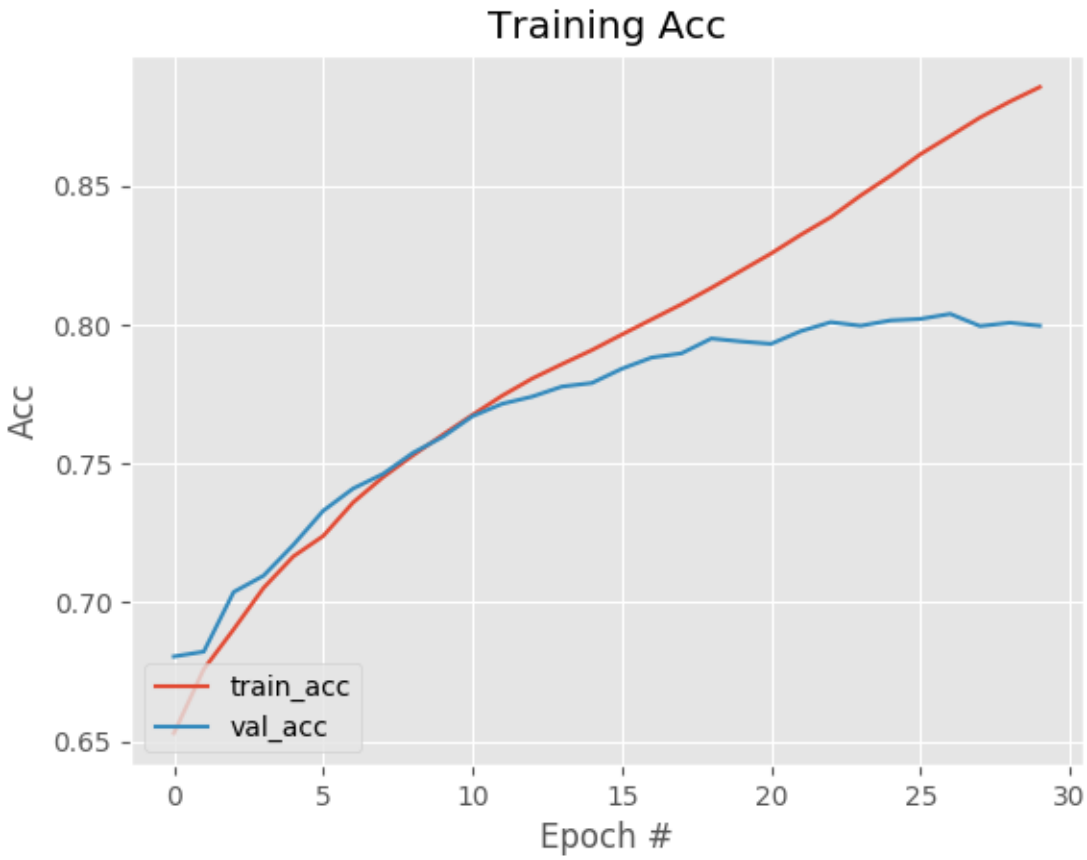
WARNING:absl:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.

Model: "functional_5"

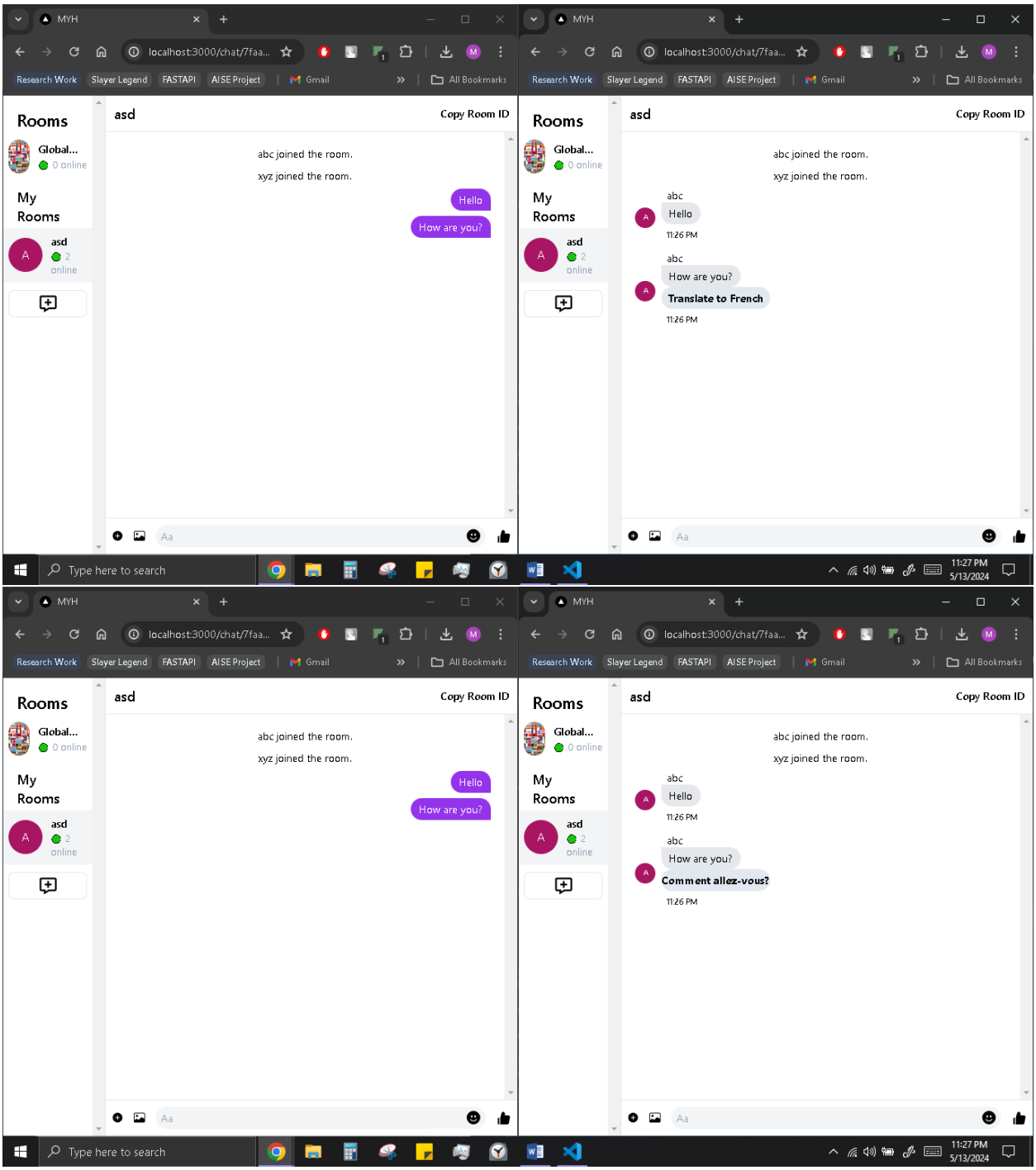
Layer (type)	Output Shape	Param #	Connected to
input_layer_1 (InputLayer)	(None, None, 103)	0	-
input_layer_2 (InputLayer)	(None, 64)	0	-
input_layer_3 (InputLayer)	(None, 64)	0	-
lstm_1 (LSTM)	[(None, None, 64), (None, 64), (None, 64)]	43,008	input_layer_1[0][0], input_layer_2[0][0], input_layer_3[0][0]
dense (Dense)	(None, None, 103)	6,695	lstm_1[0][0]

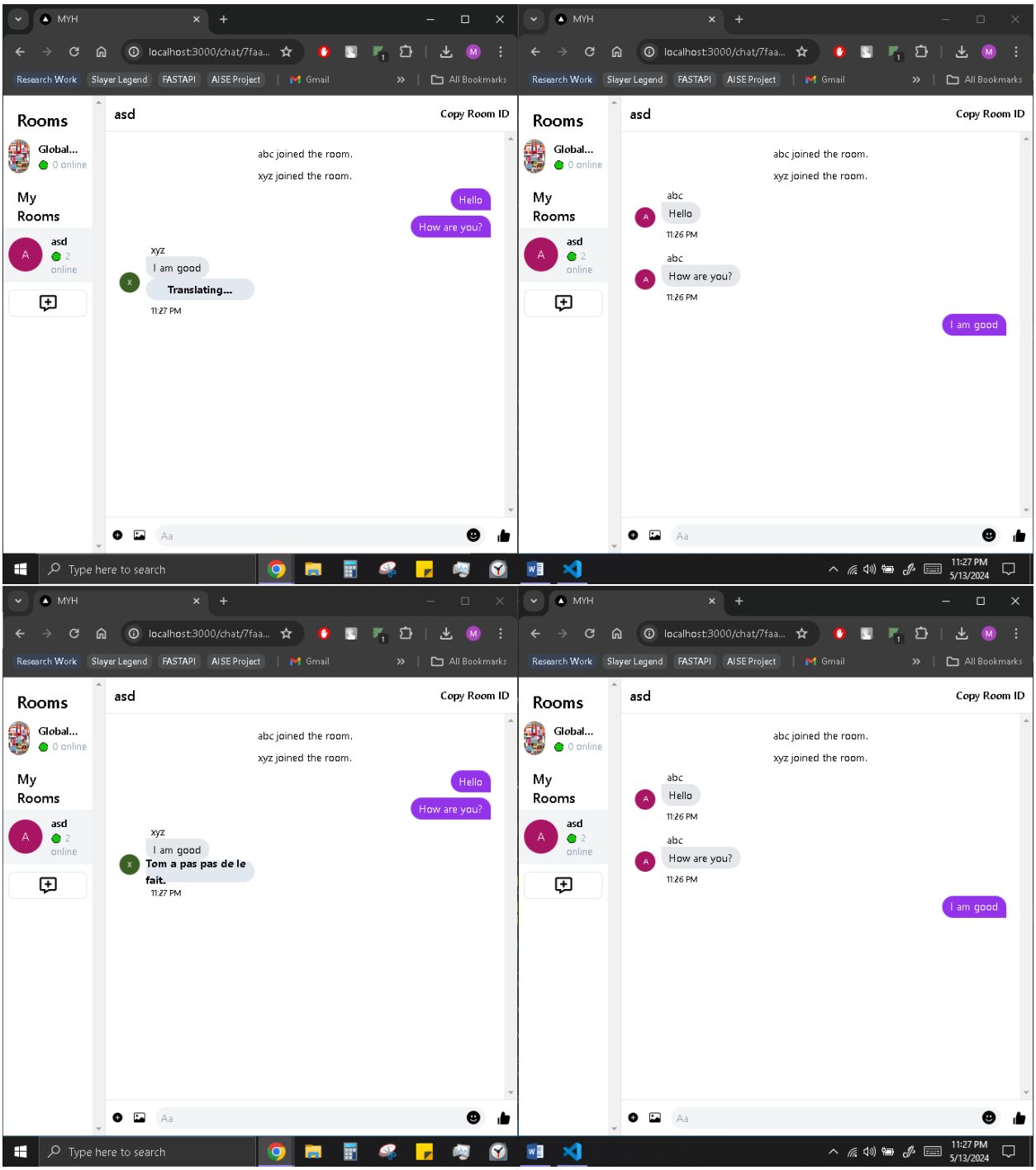
Results

Due to the lack of compute resources (i.e. GPU), our training was limited and as such couldn't achieve a desirable output. Even when we tried using the resources of Google Collaborator, it repeatedly crashed its runtime execution due to the RAM being instantly depleted. Therefore, we opted for a local solution which ran on CPU usage. Here is the plot of training accuracy:



To better visualize the results, we implemented our model into a real life use case, which is a chat room which has the option to translate sentences to French. Here are some screenshots:





Future Improvements

One thing our model is lacking, is an attention mechanism. The largest reason of our model not being able to translate most sentences correctly is largely due to this fact. This would allow us to create deeper models to increase the representational capacity of the network.

Another room for improvement can be seen in the hyper parameter values as they can be further optimized for a better result.