NLP with RNN, LSTM, Seq2Seq, and Transformer models

Seq2Seq Results – Default configuration

RNN

Training Loss: 4.2831
Training Perplexity: 72.4661
Validation Loss: 4.3379
Validation Perplexity: 76.5440

LSTM

Training Loss: 3.3645
Training Perplexity: 28.9199
Validation Loss: 3.4607
Validation Perplexity: 31.8399

RNN-with-Attention

Training Loss: 3.9729
Training Perplexity: 53.1373
Validation Loss: 3.9835
Validation Perplexity: 53.7070

LSTM-with-Attention

Training Loss: 3.3808
Training Perplexity: 29.3930
Validation Loss: 3.4736
Validation Perplexity: 32.2514

Seq2Seq Explanation (RNN vs LSTM)

Compare the RNN result to the LSTM result and explain why they differ.

RNN has bigger training loss and validation loss than LSTM. For training perplexity and validation perplexity, RNN also has larger value. Overall, LSTM performs better in Seq2Seq model.

This is because RNN has limitation on vanishing gradient problem. RNNs use a basic recurrent layer that takes the output of the previous time step as input to the current time step. The gradients will become too small to update the weights properly during backpropagation.

On the other hand, LSTMs use a more complex recurrent layer that includes an input gate, output gate, and forget gate. These gates allow the LSTM to selectively forget information from the past, which helps to mitigate the vanishing gradient problem and better train the model in sequential data.

Seq2Seq Explanation (RNN vs RNN-with-Attention)

Compare the RNN result to the RNN-with-Attention result and explain why they differ.

With attention, RNN has lower training loss, validation loss, training perplexity, and validation perplexity. RNN-with-attention performs better than RNN.

RNNs have limitations when it comes to handling long-term dependencies in the input sequences. Attention allows the model to focus on certain parts of the input sequence, so RNN with attention can selectively focus on parts of the input sequence which are relevant to the current prediction, instead of processing the entire sequence at once. This can help the model to long-term dependencies in the input sequences better and make more accurate predictions.

Seq2Seq Results – Best model

The best model after hyper-parameter tuning

Best model

Training Loss: 3.2333

Training Perplexity: 25.3635

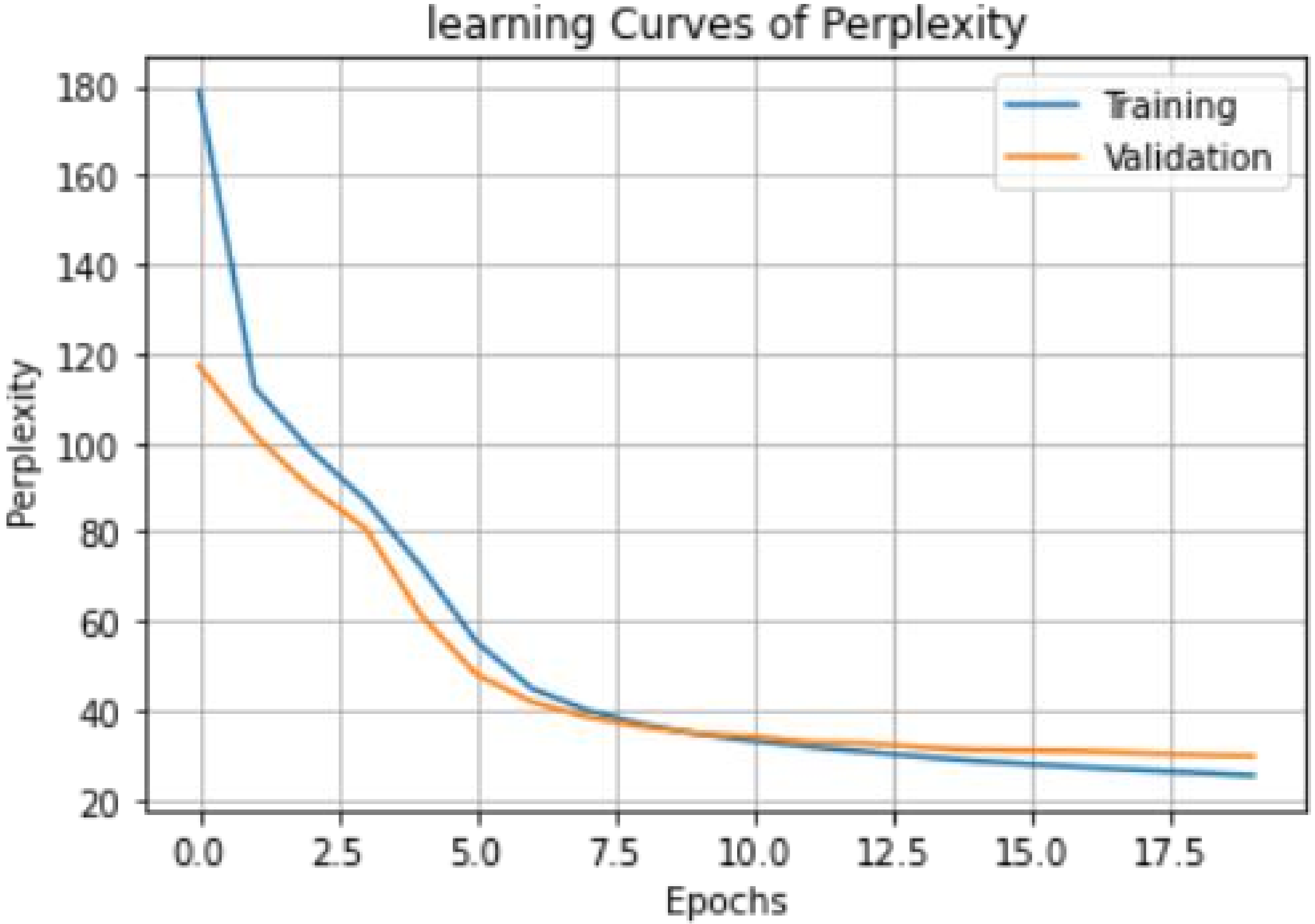
Validation Loss: 3.3918

Validation Perplexity: 29.7207

List your best model hyper-parameter values including model type: LSTM

encoder_emb_size = 32	encoder_hidden_size = 64	encoder_dropout = 0.2
decoder_emb_size = 32	decoder_hidden_size = 64	decoder_dropout = 0.2
learning_rate = 0.005	model_type = LSTM without attention	EPOCHS = 20

Seq2Seq Best model Learning Curves (Perplexity)



Seq2Seq Explanation – Best model

Explain the details of the best model.

I'm using LSTM model without attention, and set the learning rate to be 0.005. I set the encoder/decoder embbding size to be 32, and set the encoder/decoder hidden size to be 64. I kept the dropout rate for both encoder and decoder to be 0.2. The epoch number is still 20.

I'm using LSTM model because it can mitigate the vanishing gradient problem and better train the model in sequential data. I didn't implement attention because, from the default configuration model comparison, I noticed the LSTM model with attention has worse performance, which may due to the model doesn't need to focus on particular things or attention adds complexity to the model and lends to overfitting. I set the learning rate to be 0.005 because this learning rate has the best performance during my training experiments. The embedding size and hidden size should be large enough to capture the relevant features of the input data, but not so large that the model becomes too complex and overfits the data, so I set them to be 32 amd 64.

Transformer Results

Default configuration	Best model
Training Loss: 2.5637	Training Loss: 2.9061
Training Perplexity: 12.9834	Training Perplexity: 18.2848
Validation Loss: 3.2128	Validation Loss: 3.1734
Validation Perplexity: 24.8492	Validation Perplexity: 23.8875

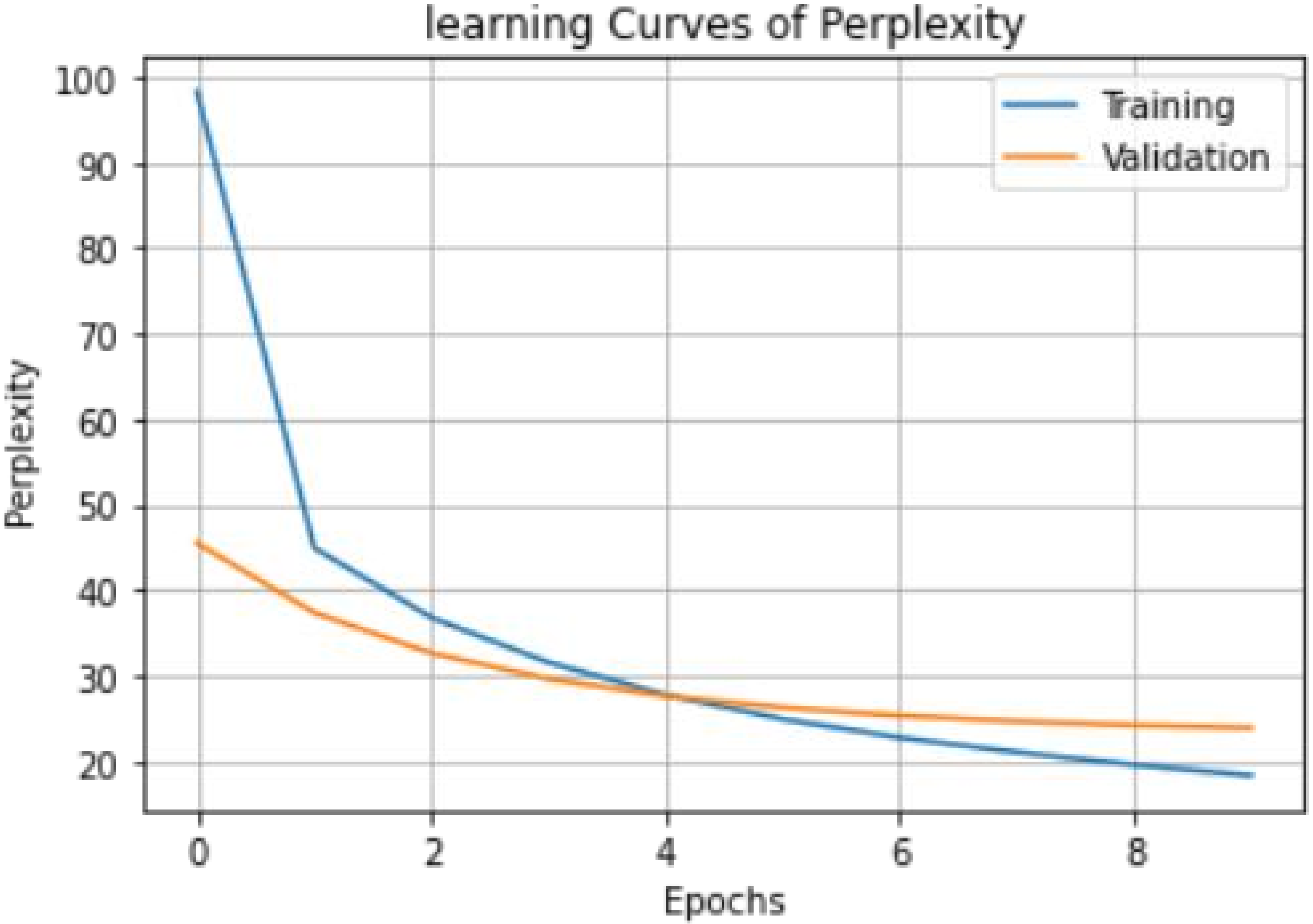
List your best model hyper-parameter values:

learning_rate = 0.0005

Batch_size = 128

EPOCHS = 10

Transformer Best model Learning Curves (Perplexity)



Transformer Explanation – Best model

Hyper-parameter and how to improve the model performance.

I set the learning rate to be 0.0005 and the batch size to be 128. Using 10 epochs to train the model.

I experimented with different learning rates and observed that the 0.0005 learning rate has the model's performance on the validation set. The learning is relatively small, but it can help prevent the model from diverging and overshooting the minimum of the loss function. And it helps to make small and gradual updates to the parameters. This can help the model to generalize better to new data and avoid overfitting. I kept the batch size at 128, which is relatively large. It can lead to faster training times and better generalization since the model is exposed to more diverse examples during each training step. This can help prevent overfitting and improve the model's prediction.

Transformer Translation (Best model) Results

Table 1

Input sentence	Back translation
'<sos>', 'a', 'man', 'in', 'an', 'orange', 'hat', 'starring', 'at', 'something', '.', '\n', '<eos>'	'<sos>', 'a', 'man', 'with', 'an', 'orange', 'hat', 'something','something', 'something', 'something', '<eos>'
'<sos>', 'a', '<unk>', 'terrier', 'is', 'running', 'on', 'lush', 'green', 'grass', 'in', 'front', 'of', 'a', 'white', 'fence', '.', '\n', '<eos>'	'<sos>', 'a', '<unk>', 'and', 'is', 'running', 'across', 'across', 'in', 'in', 'of', 'of', 'white', 'white', '.', '\n', '<eos>'
'<sos>', 'a', 'girl', 'in', 'karate', 'uniform', 'breaking', 'a', 'stick', 'with', 'a', 'front', 'kick', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'man', 'in', 'a', 'a', 'is', 'a', 'a', 'a', 'a', 'a', '.', '.', '<eos>'
'<sos>', 'people', 'are', 'fixing', 'the', 'roof', 'of', 'a', 'house', '.', '\n', '<eos>',	'<sos>', '<unk>', 'are', 'the', 'game', 'of', 'of', 'a', '.', '.', '\n', '<eos>'
'<sos>', 'a', 'group', 'of', 'people', 'standing', 'in', 'front', 'of', 'an', 'igloo', '.', '\n', '<eos>'	'<sos>', 'a', 'group', 'of', 'people', 'are', 'in', 'in', 'front', 'of', '.', '.', '<eos>'
'<sos>', 'a', 'guy', 'works', 'on', 'a', 'building', '.', '\n', '<eos>'	'<sos>', 'a', 'man', 'is', 'working', 'a', 'a', '.', '.', '\n', '<eos>'
'<sos>', 'a', 'man', 'in', 'a', 'vest', 'is', 'sitting', 'in', 'a', 'chair', 'and', 'holding', 'magazines', '.', '\n', '<eos>',	'<sos>', 'a', 'man', 'in', 'a', 'jacket', 'sits', 'on', 'on', 'a', 'and', 'and', 'a', '.', '\n', '<eos>'
'<sos>', 'a', 'mother', 'and', 'her', 'young', 'song', 'enjoying', 'a', 'beautiful', 'day', 'outside', '.', '\n', '<eos>'	'<sos>', 'a', 'woman', 'and', 'her', 'small', 'the', 'enjoying', 'a', 'a', 'in', '.', '.', '.', '<eos>'
'<sos>', 'a', 'woman', 'holding', 'a', 'bowl', 'of', 'food', 'in', 'a', 'kitchen', '.', '\n', '<eos>'	'<sos>', 'a', 'woman', 'in', 'a', 'a', 'a', 'a', 'a', 'a', '.', '.', '.', '.', '<eos>'

Seq2Seq (Best model) Translation Results

Table 2

Input sentence	Back translation
'<sos>', 'a', 'man', 'in', 'an', 'orange', 'hat', 'starring', 'at', 'something', '.', '\n', '<eos>'	'<sos>', 'a', 'man', 'in', 'a', 'blue', 'shirt', 'is', 'a', '.', '.', '\n', '<eos>'
'<sos>', 'a', '<unk>', 'terrier', 'is', 'running', 'on', 'lush', 'green', 'grass', 'in', 'front', 'of', 'a', 'white', 'fence', '.', '\n', '<eos>'	'<sos>', 'a', 'man', 'is', 'is', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', '.', '.', '\n', '<eos>'
'<sos>', 'a', 'girl', 'in', 'karate', 'uniform', 'breaking', 'a', 'stick', 'with', 'a', 'front', 'kick', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'man', 'in', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', '.', '.', '\n', '<eos>'
'<sos>', 'people', 'are', 'fixing', 'the', 'roof', 'of', 'a', 'house', '.', '\n', '<eos>',	'<sos>', 'people', 'are', 'a', 'a', 'a', 'a', 'a', '.', '.', '\n', '<eos>'
'<sos>', 'a', 'group', 'of', 'people', 'standing', 'in', 'front', 'of', 'an', 'igloo', '.', '\n', '<eos>'	'<sos>', 'a', 'group', 'of', 'people', 'are', 'in', 'front', 'a', 'a', '.', '.', '\n', '<eos>'
'<sos>', 'a', 'guy', 'works', 'on', 'a', 'building', '.', '\n', '<eos>'	'<sos>', 'a', 'man', 'is', 'on', 'a', 'a', '.', '.', '\n', '<eos>'
'<sos>', 'a', 'man', 'in', 'a', 'vest', 'is', 'sitting', 'in', 'a', 'chair', 'and', 'holding', 'magazines', '.', '\n', '<eos>',	'<sos>', 'a', 'man', 'in', 'a', 'white', 'shirt', 'and', 'a', 'a', 'a', 'a', '.', '.', '\n', '<eos>'
'<sos>', 'a', 'mother', 'and', 'her', 'young', 'song', 'enjoying', 'a', 'beautiful', 'day', 'outside', '.', '\n', '<eos>'	'<sos>', 'a', 'woman', 'and', 'a', 'young', 'are', 'in', 'a', 'a', 'a', '.', '.', '\n', '<eos>'
'<sos>', 'a', 'woman', 'holding', 'a', 'bowl', 'of', 'food', 'in', 'a', 'kitchen', '.', '\n', '<eos>'	'<sos>', 'a', 'woman', 'on', 'a', 'a', 'a', 'a', 'a', 'a', '.', '.', '\n', '<eos>'

Compare Seq2Seq to Transformer (Best models)

Compare the Seq2Seq best model results to the Transformer best model results both quantitatively and qualitatively and explain the differences.

Based on the result, the Transformer model outperformed the Seq2Seq model in this task. The translation from Transformer makes more sense and has more quality translated sentences. The Transformer model can handle longer sentences more efficiently, and it can capture complex dependencies between words and phrases, which is difficult for the Seq2Seq model. Additionally, the Transformer model allows for parallelization during training, that's why it can train the model faster.

Overall, while both models have their strengths and weaknesses, the Transformer model performs better in the language translation task.