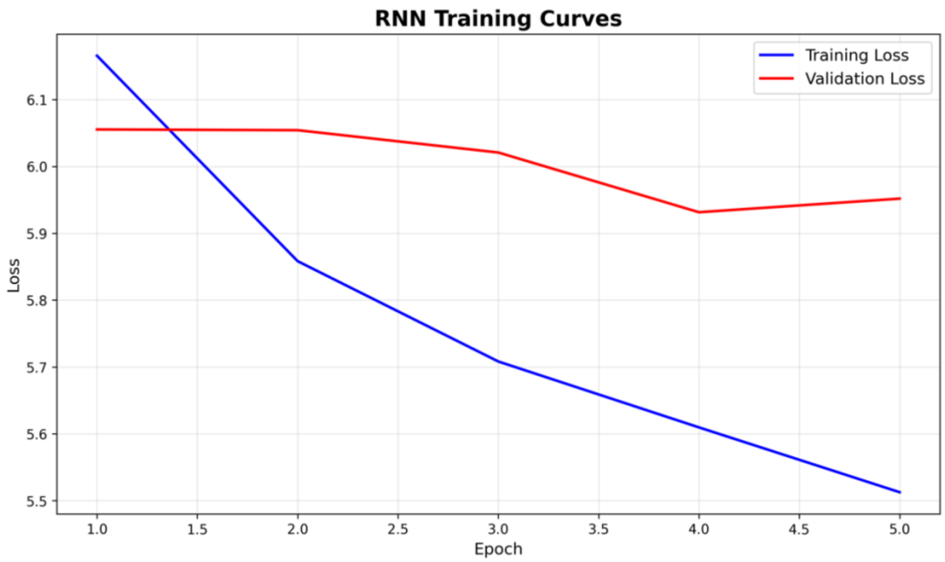
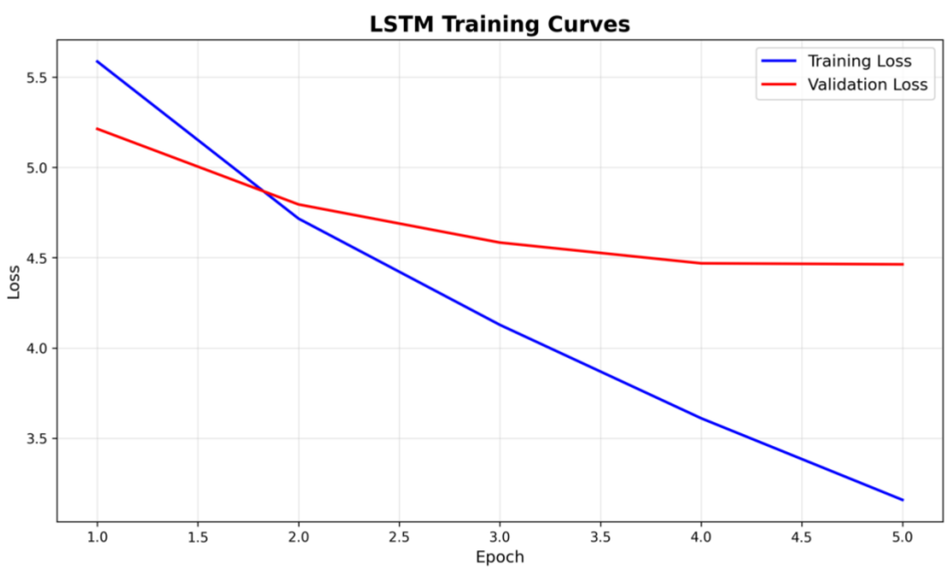
PROBLEM 1 Recurrent NN : implement Encoder and Decoder

(a) nn.RNN



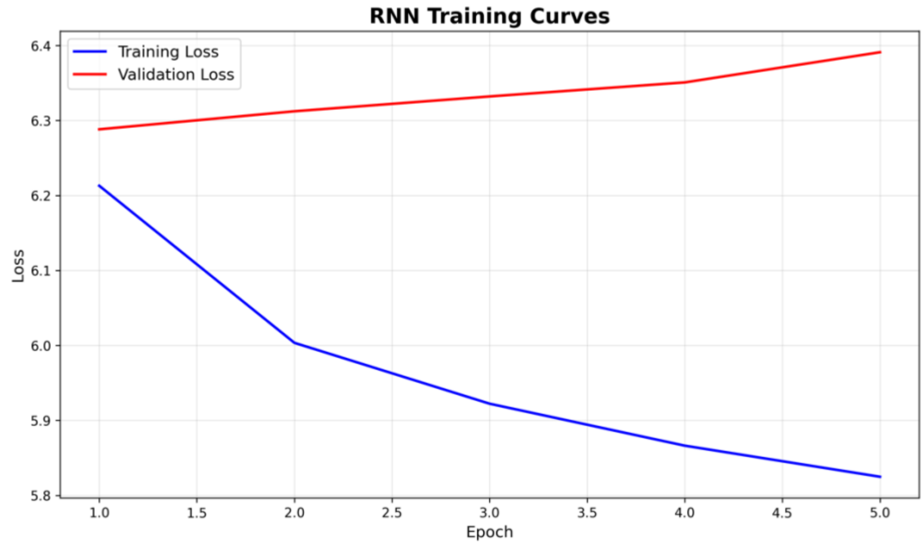

```
rnn BLEU Score: 2.91
rnn BERT F1 Score: 0.1359393447637558
```

(b) nn.LSTM

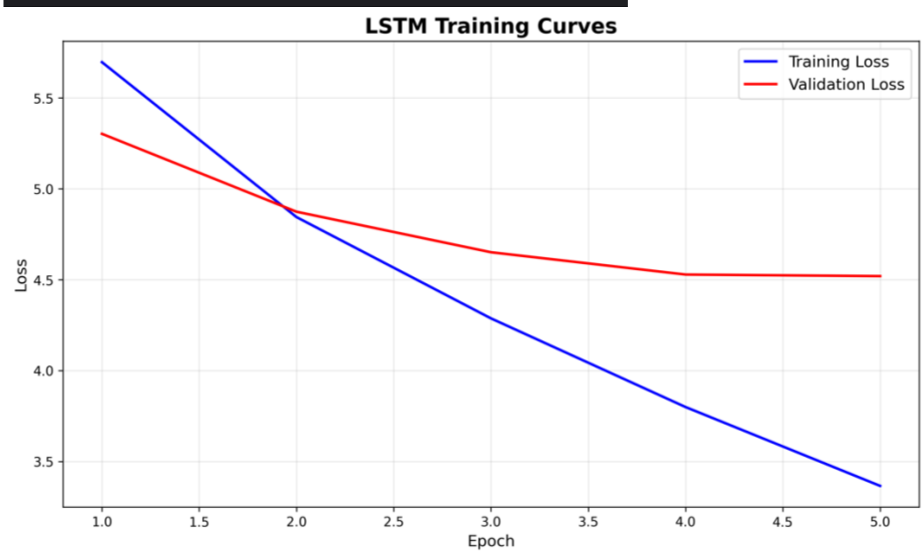



```
lstm BLEU Score: 2.29
lstm BERT F1 Score: 0.2530685365200043
```

PROBLEM 2 Recurrent NN : Implement the RNN and LSTM

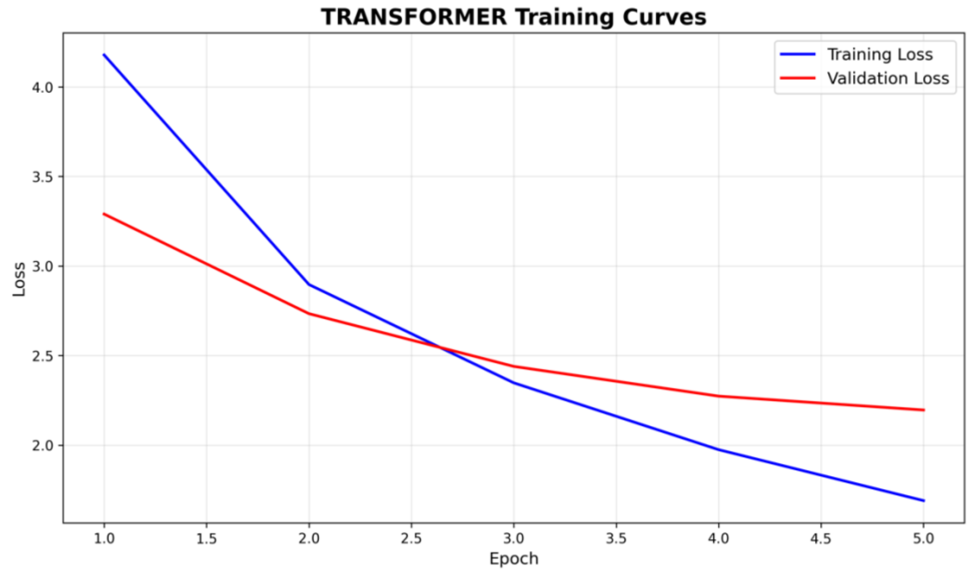



```
rnn BLEU Score: 2.57
rnn BERT F1 Score: 0.0746447741985321
```




```
lstm BLEU Score: 2.92
lstm BERT F1 Score: 0.22969160974025726
```

PROBLEM 3 Transformer NN




```
transformer BLEU Score: 2.98
transformer BERT F1 Score: 0.42821386456489563
```

Results Summary

Model	Implementation	BLEU Score	BERT F1 Score
RNN	Built-in	2.91	0.1359
LSTM	Built-in	2.29	0.2531
RNN	Hand-implemented	2.57	0.0746
LSTM	Hand-implemented	2.92	0.2297
Transformer	-	2.98	0.4282

Model comparison analysis:

Transformer achieved highest BLEU (2.98) and BERT F1 (0.4282) scores with superior training convergence. Built-in RNN outperformed built-in LSTM in BLEU score (2.91 vs 2.29) but had much lower semantic understanding (BERT F1: 0.14 vs 0.25).**Training Issues**: Hand-implemented RNN showed severe overfitting with increasing validation loss.Transformer architecture demonstrates clear superiority. Among recurrent models, LSTMs provide better semantic understanding despite variable BLEU performance.

Challenges Faced

Training takes too much time and resources.