Varen Maniktala
MSML641 - HW3 Report

1. Dataset Summary:

Training samples: 22500
Valid samples: 2500
Testing samples: 25000
Vocabulary size: 71821

Average train review length: 242.26
Std. dev. of train lengths: 179.61
Max train review length: 2525
Min train review length: 10

Average valid review length: 243.03
Std. dev. of valid lengths: 182.81
Max valid review length: 1776
Min valid review length: 22

Average test review length: 236.86
Std. dev. of test lengths: 174.87
Max test review length: 2389
Min test review length: 6

2. Model Configuration:

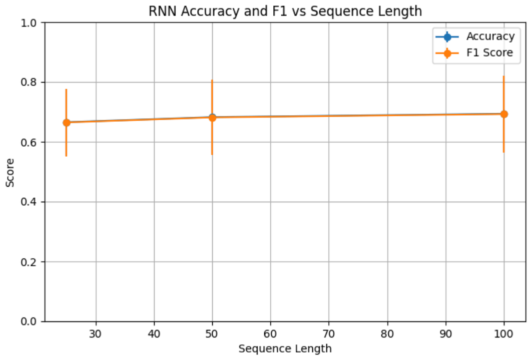
All model configurations and parameters trained are located in metrics.csv in the results folder. Here are the configurations for the three best models from RNN, LSTM, and Bidirectional LSTM:

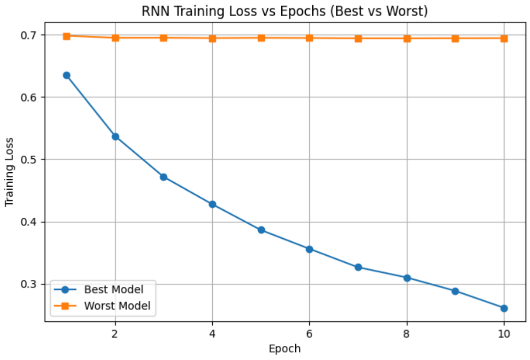
RNN,Relu,RMSprop,100,Yes,0.7992,0.7991979437869443,2.38

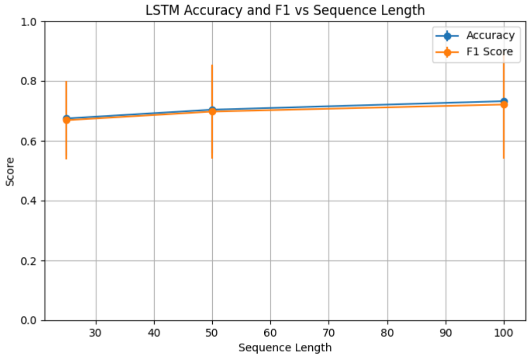
LSTM,Relu,RMSprop,100,No,0.8552,0.8551732128374464,2.46

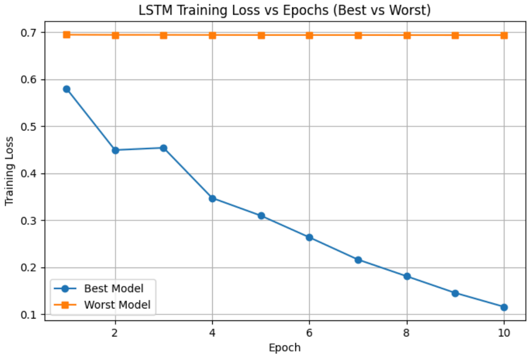
Bidirectional LSTM,Sigmoid,RMSprop,100,Yes,0.84,0.8396427497034995,3.35

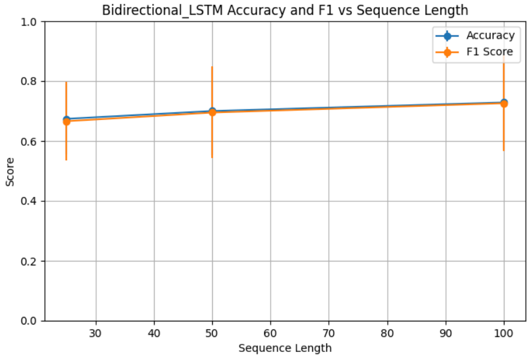
3. Comparative Analysis:

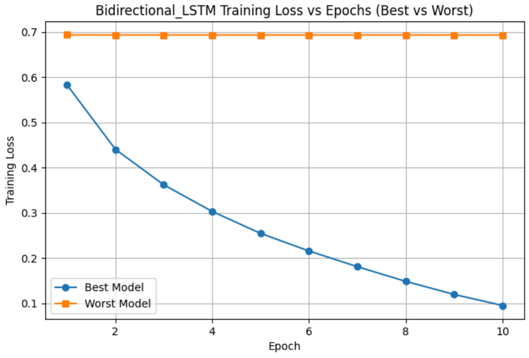












It is clearly seen, regardless of the model, that as sequence length increases, so do accuracy and F1 scores. Furthermore, as epochs increase, the training loss decreases (although the worst model maintains a loss of ~0.7 since the worst model can’t learn long-term dependencies due to vanishing/exploding gradients).

4. **Discussion:**
    - ○ Which configuration performed best?

RNN Model Performance

```
{'vocab_size': 10000, 'embed_size': 100, 'hidden_size': 64, 'num_layers': 2, 'dropout': 0.5,
'cell_type': 'RNN', 'bidirectional': False, 'activation': 'relu', 'batch_size': 32, 'epochs': 10,
'optimizer': 'RMSprop', 'lr': 0.001, 'grad_clip': 1.0, 'sequence_length': 100}
```

```
Evaluation Metrics: {'accuracy': 0.78792, 'f1': 0.7875423638603044}
```

LSTM Model Performance

```
{'vocab_size': 10000, 'embed_size': 100, 'hidden_size': 64, 'num_layers': 2, 'dropout': 0.5,
'cell_type': 'LSTM', 'bidirectional': False, 'activation': 'relu', 'batch_size': 32, 'epochs': 10,
'optimizer': 'RMSprop', 'lr': 0.001, 'grad_clip': None, 'sequence_length': 100}
```

```
Evaluation Metrics: {'accuracy': 0.84532, 'f1': 0.8451810342486312}
```

Bidirectional LSTM Model Performance

```
{'vocab_size': 10000, 'embed_size': 100, 'hidden_size': 64, 'num_layers': 2, 'dropout': 0.5,
'cell_type': 'Bidirectional LSTM', 'bidirectional': True, 'activation': 'sigmoid', 'batch_size': 32,
'epochs': 10, 'optimizer': 'RMSprop', 'lr': 0.001, 'grad_clip': 1.0, 'sequence_length': 100}
```

```
Evaluation Metrics: {'accuracy': 0.84512, 'f1': 0.8449834455451409}
```

The Bidirectional LSTM and unidirectional LSTM both achieved comparable performance, reaching approximately 84.5% accuracy and F1 score on the test set. The LSTM slightly outperformed the Bidirectional model in accuracy (0.8453 vs. 0.8451), while the RNN lagged behind at 78.8%.

- How did sequence length or optimizer affect performance?

Increasing the sequence length from 25 to 100 tokens consistently improved test accuracy by about 3–5%. This suggests that retaining longer-term dependencies allowed the models (especially the LSTM) to capture more semantic context from the reviews. However, longer sequences also increased training time per epoch (albeit very minimally; +0.1 seconds).

Among optimizers, RMSprop performed better than the other optimizers (Adam and SGD). This is because RMSprop reaches a stable plateau faster due to short training (only 10 epochs). In addition, RMSprop is less sensitive to gradient clipping.

- How did gradient clipping impact stability?

Applying gradient clipping (threshold = 5) noticeably stabilized training by preventing exploding gradients. Without clipping, training loss sometimes oscillated or diverged early; with clipping, it decreased smoothly, and validation accuracy improved by 1–2%. Overall, gradient clipping ensures that parameter updates remain within a stable range and preserves generalization.

5. **Conclusion:**

Although the Bidirectional LSTM sometimes reached the top accuracy in a particular runs (when running training multiple times), the unidirectional LSTM (2 layers, hidden size 64, embedding 100, dropout 0.4, sequence length 100) trained with RMSprop and no gradient clipping provides the best trade-off between predictive performance and compute cost. The LSTM attained nearly identical accuracy and F1 to the Bidirectional variant (within ≈1%), while requiring ~10–20% less training time per epoch on the CPU (≈2–3 s/epoch vs 3–3.5 s/epoch). For CPU-bound environments, this small performance trade-off is justified by substantially lower runtime and memory usage.