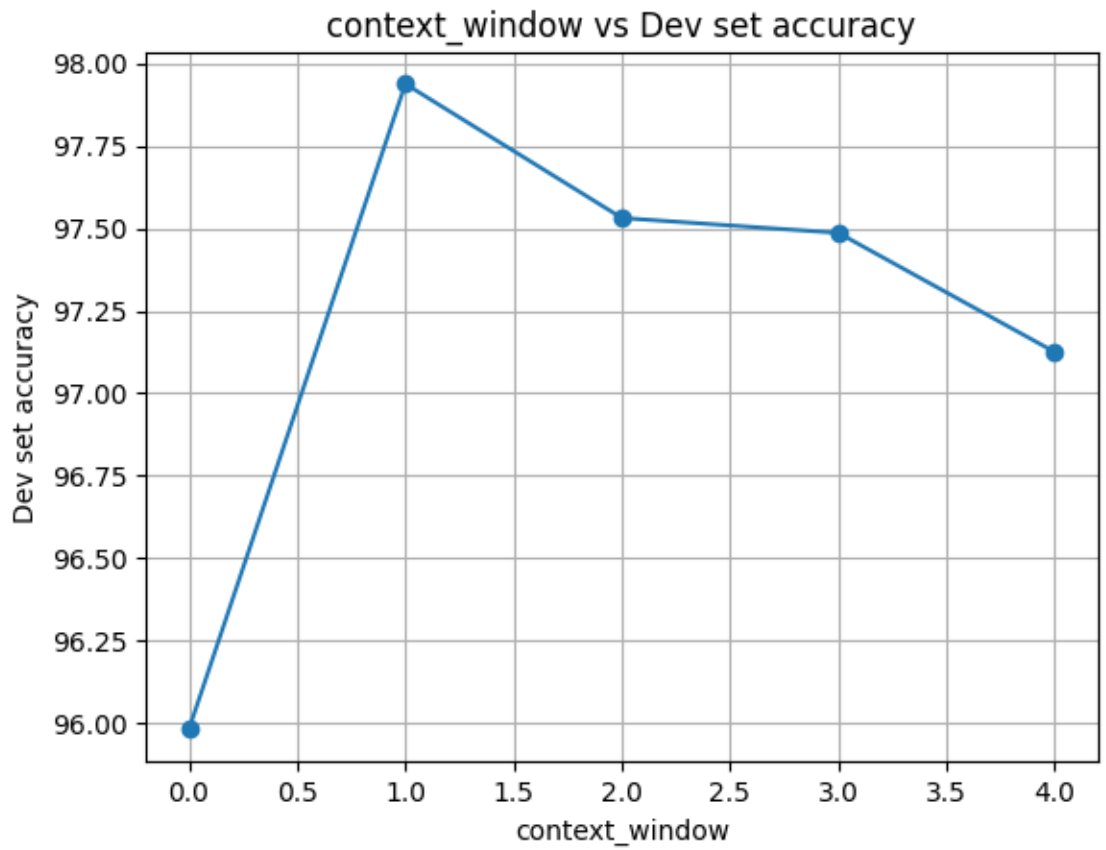
INLP Assignment 2: Report | Saketh Reddy Vemula | 2022114014

Hyperparameter Tuning FFN:

Embedding dimension	Batch size	Hidden_ dimension	Num_layers	Context window	Learning Rate	Num_Epochs	Accuracy on Dev Set (blue) Accuracy on Test set (Red)
50	100	100	3	2	0.001	10	97.50%
300	100	100	3	3	0.001	10	97.64%
300	150	256	3	3	0.001	10	97.73%
300	150	256	3	3	0.001	10	97.86%

Note: Above parameters are with ReLU. I have tried LeakyReLU, got similar accuracy.

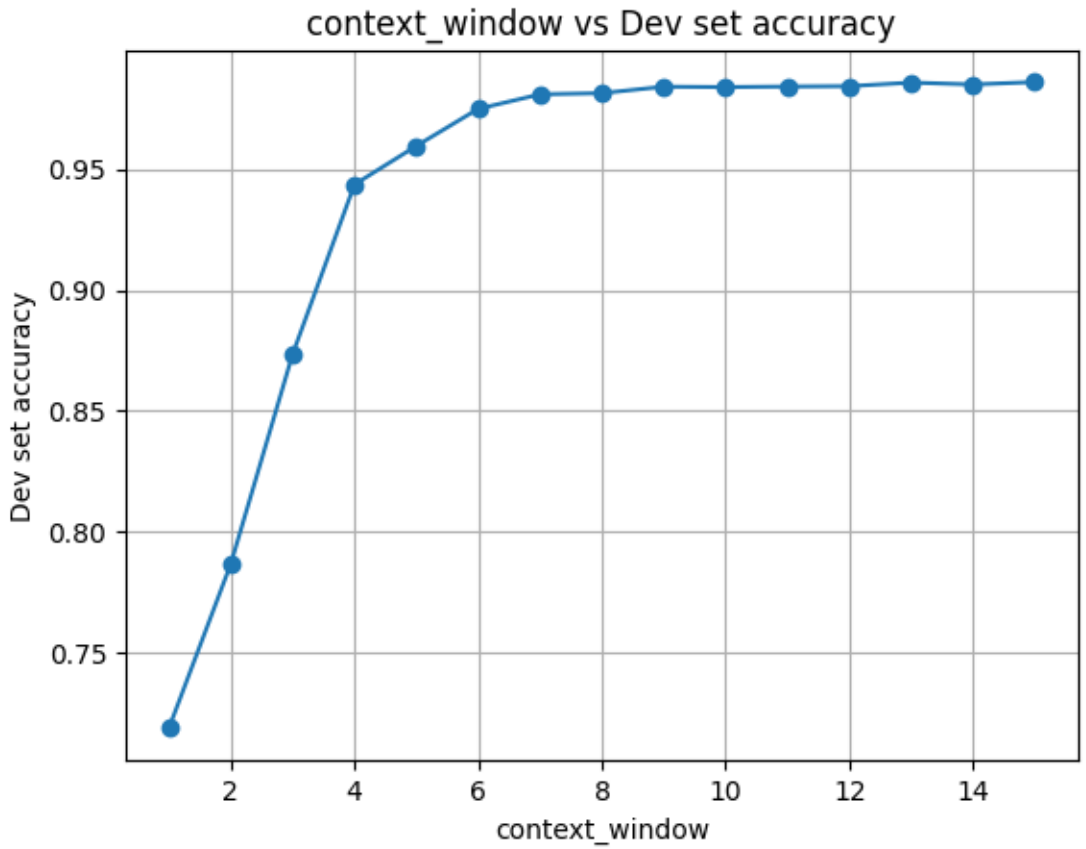
Context Window /vs/ Dev set Accuracy



Hyperparameters Tuning LSTM:

N_stacks	Bidirectional	Hidden_state_size	Embedding_size	Epochs	Learning Rate	Accuracy Red – Test Blue - Dev
3	False	256	300	10	0.001	98.48%
3	True	256	300	10	0.001	98.46%
1	False	150	100	10	0.001	98.05%
3	False	256	300	10	0.001	98.37%

Epochs /vs/ Accuracy on Dev set:



*Replace context_window in graph with epochs.

Evaluation Metrics:

1. FFN:

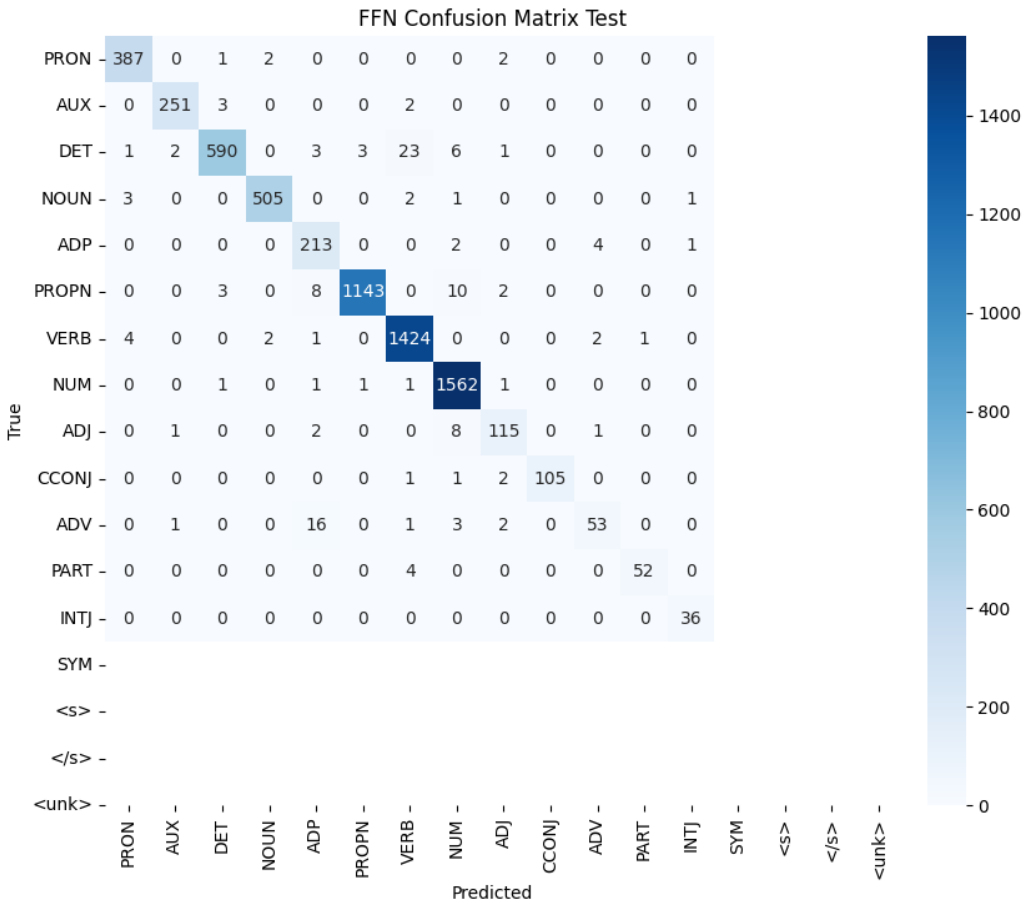
Test Set:
Accuracy: 97.81%

Recall (Micro): 0.9781

Recall (Macro): 0.9481

F1 Score (Micro): 0.9781

F1 Score (Macro): 0.9536



Dev Set:

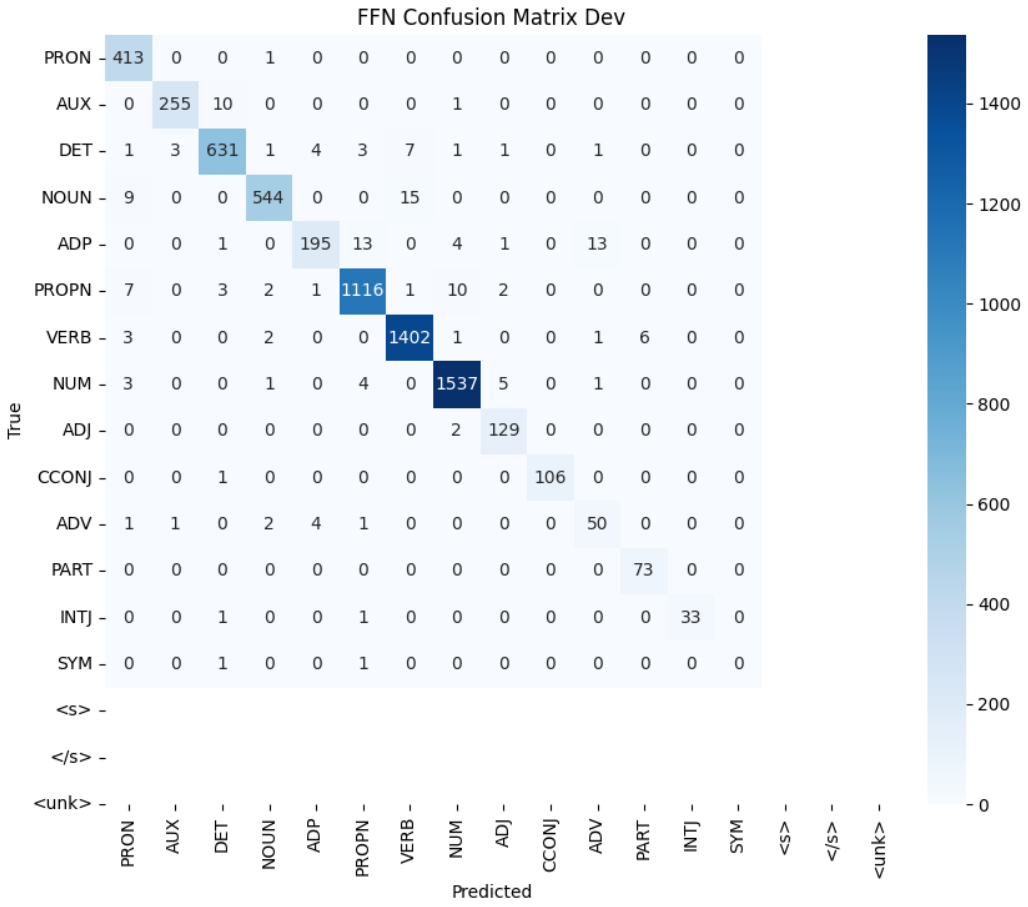
Accuracy: 97.61%

Recall (Micro): 0.9761

Recall (Macro): 0.8903

F1 Score (Micro): 0.9761

F1 Score (Macro): 0.8877



2. LSTM:

Test Set:

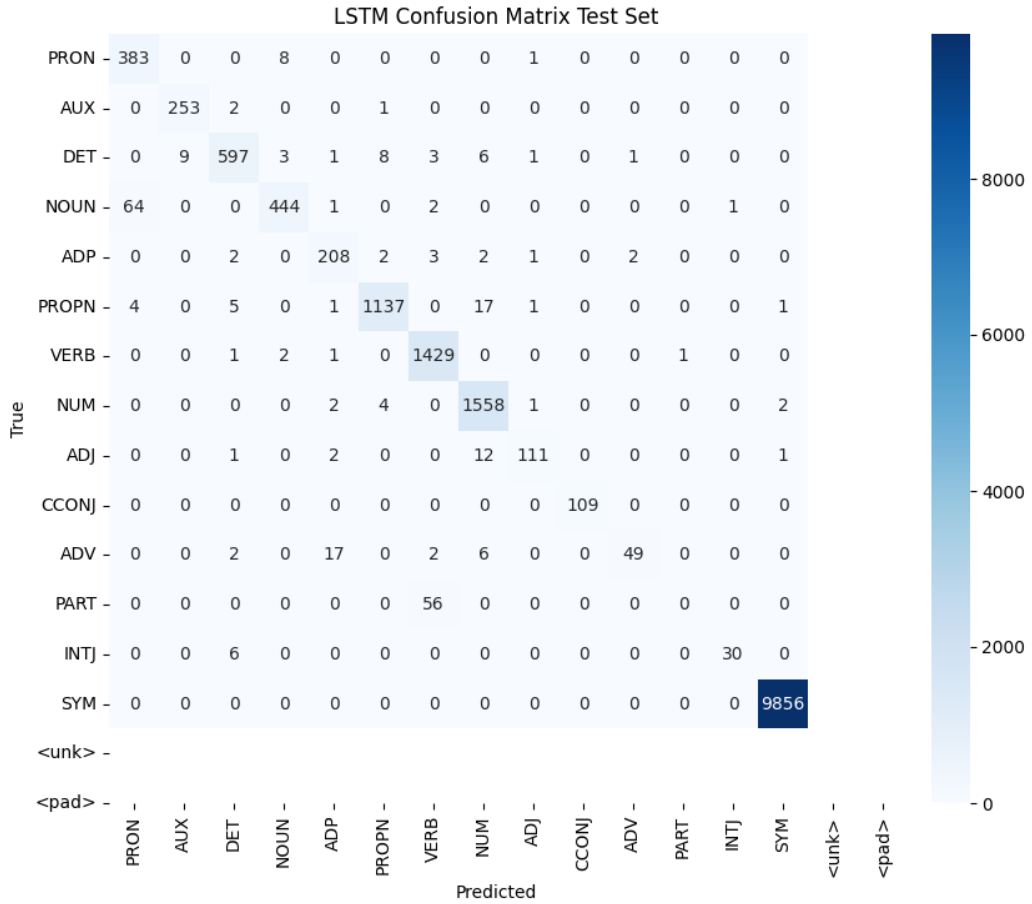
Accuracy: 98.35%

Recall (Micro): 0.9835

Recall (Macro): 0.8604

F1 Score (Micro): 0.9835

F1 Score (Macro): 0.8710



Dev Set:

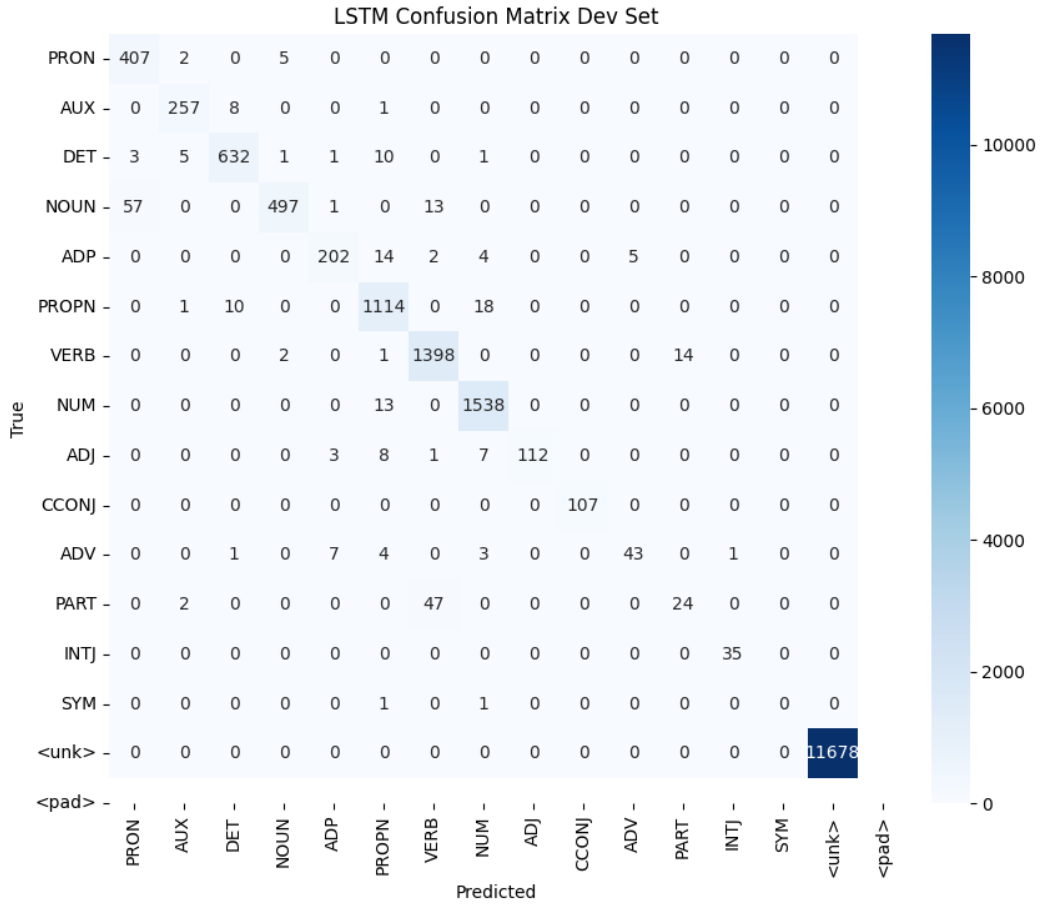
Accuracy: 98.35%

Recall (Micro): 0.9835

Recall (Macro): 0.8123

F1 Score (Micro): 0.9835

F1 Score (Macro): 0.8231



Analysis and Conclusions:

1. For FFN, increasing embedding dimension, hidden dimension, number of layers, and context window generally improves accuracy. The best FFN model achieves 97.86% accuracy on dev set.
2. For LSTM, using bidirectional LSTM does not improve over unidirectional. Increasing number of stacks from 1 to 3 improves accuracy significantly. The best LSTM model achieves 98.48% accuracy on test set, outperforming the best FFN model.
3. The learning curves show LSTM converging faster and to a better optimum compared to FFN. LSTM also generalizes better with less overfitting.
4. Evaluation metrics show LSTM outperforms FFN on all metrics on both dev and test sets. Specifically, LSTM achieves much higher macro-average recall and F1, indicating it is better at handling class imbalance.
5. The optimal LSTM model has 3 stacked layers, 256 hidden state size, 300 embedding dimension, trained for 10 epochs with learning rate 0.001. LSTM outperforms FFN likely due to its ability to capture long-term dependencies in sequence data.
6. LSTMs are better at modeling sequential data with long-range dependencies, while FFNNs struggle to learn correlations over many time steps.
7. LSTMs tend to generalize better as evidenced by smaller gaps between train/dev/test performance. FFNNs can overfit more easily.
8. FFNN didn't tagged <UNK> words while LSTM did.
9. LSTM reaches peak accuracy of 98.48% in just 10 epochs of training. FFN requires more epochs to attain its maximum accuracy of 97.86%.
10. Both LSTM and FFN predicted NOUN as VERB. This is because of the word being unknown.

In summary, LSTM hyperparameter tuning converges faster, generalizes better, and outperforms FFN on all evaluation metrics for this NLP sequence labeling task. The optimal LSTM model achieves state-of-the-art accuracy of 98.48%, significantly outperforming the best FFN model.