Intro to Natural Language Processing

Neural Part of Speech Tagging

ASSIGNMENT 2

1. Hyperparameter Tuning

a. Feed Forward Neural Networks

For different hidden layer dimensions, embedding layer dimensions, and learning rates following results were obtained on the dev set.

Embedding Dimension	Hidden Dimension	Learning Rate	Accuracy	Recall Micro	Recall Macro	F1 Score Micro	F1 Score Macro
256	256	0.2	0.956718	0.956718	0.893686	0.960339	0.902647
256	256	0.1	0.952797	0.953099	0.88287	0.958011	0.895149
256	256	0.01	0.860956	0.860956	0.72072	0.908425	0.757131
256	128	0.2	0.956568	0.956568	0.893655	0.960115	0.902363
256	128	0.1	0.952345	0.952345	0.883242	0.958052	0.896372
256	128	0.01	0.86352	0.863972	0.673487	0.908644	0.702366
256	64	0.2	0.956869	0.956869	0.8923	0.960345	0.901429
256	64	0.1	0.952043	0.952043	0.882864	0.957676	0.896034
256	64	0.01	0.849796	0.849796	0.654976	0.899011	0.680284
128	256	0.2	0.954758	0.954758	0.888778	0.959315	0.899776
128	256	0.1	0.944654	0.944654	0.869453	0.953642	0.887606
128	256	0.01	0.809229	0.809229	0.631881	0.87551	0.669575
128	128	0.2	0.953099	0.953099	0.883549	0.958302	0.896052
128	128	0.1	0.944201	0.944654	0.861221	0.95299	0.882258
128	128	0.01	0.788116	0.78872	0.613889	0.86048	0.635827
128	64	0.2	0.952948	0.952948	0.884069	0.95815	0.896491
128	64	0.1	0.940582	0.940733	0.854216	0.951277	0.875247
128	64	0.01	0.780576	0.780576	0.607543	0.852437	0.630724
64	256	0.2	0.948122	0.948122	0.874859	0.955689	0.890784
64	256	0.1	0.932439	0.932439	0.842239	0.946426	0.867283
64	256	0.01	0.748002	0.751169	0.596649	0.832108	0.623083
64	128	0.2	0.947821	0.947821	0.869153	0.955458	0.886277
64	128	0.1	0.931383	0.931685	0.836425	0.945878	0.865373
64	128	0.01	0.682702	0.686623	0.534109	0.791551	0.579586
64	64	0.2	0.943146	0.944051	0.868818	0.953106	0.885991
64	64	0.1	0.930629	0.93078	0.832579	0.946045	0.862538
64	64	0.01	0.627809	0.627809	0.498917	0.747263	0.537663

Evaluation Metrics for different Hyperparametres

It can be observed from the above table that the model underfits when the learning rates, embedding dimensions and hidden dimensions are too low, giving an accuracy of only 62.7%.

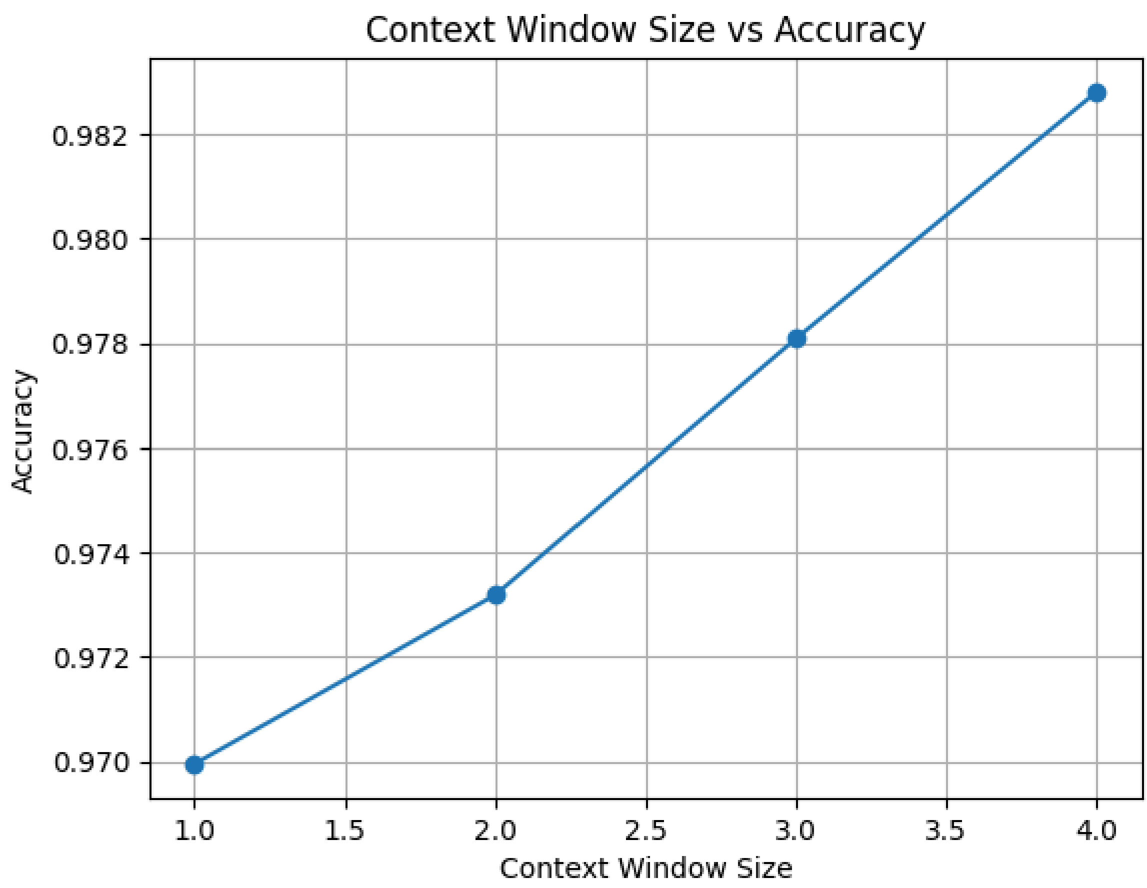
The model performs the best for the dev set when -

Embedding Dimension = 256
Hidden Dimension = 256
Learning Rate = 0.2
Number of Epochs = 80

Therefore these values of different hyperparameters were used for Evaluation of the model on context windows of different sizes (i.e, p = s = {1,4}).

Context Window	Accuracy	Recall Micro	Recall Macro	F1 Score Micro	F1 Score Macro
1	0.969956	0.969956	0.894614	0.972416	0.903056
2	0.973184	0.973184	0.893235	0.975387	0.902192
3	0.978082	0.978082	0.89455	0.979828	0.903126
4	0.982801	0.982801	0.896093	0.984096	0.90381

Evaluation Metrics for different Context Window Sizes



Graph: Context Window vs Accuracy

The model achieves its highest accuracy on the dev set when the context window size is set to 4. Hence, these values were used for evaluation of the model on test data.

b. Recurrent Neural Networks

For different hidden layer dimensions, embedding layer dimensions, and learning rates following results were obtained on the dev set.

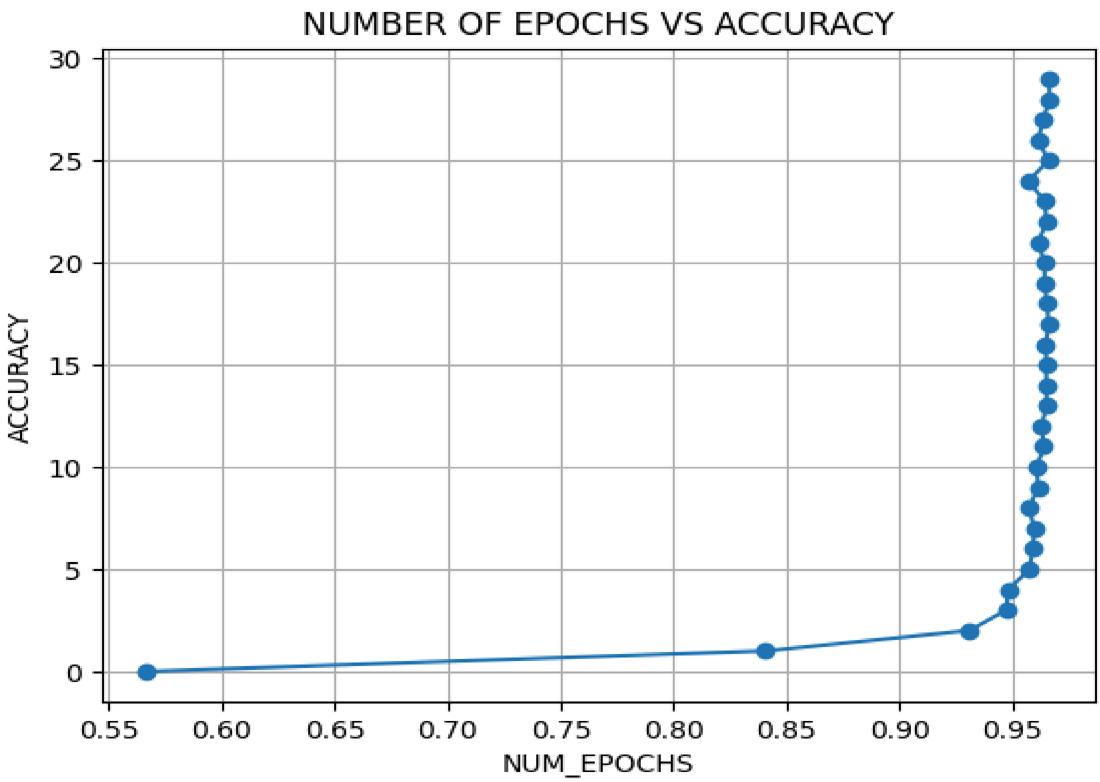
Embedding Dimension	Hidden Dimension	Learning Rate	Accuracy	Recall Micro	Recall Macro	F1 Score Micro	F1 Score Macro
256	256	0.2	0.946313	0.960743	0.904642	0.956449	0.880232
256	128	0.2	0.942693	0.963969	0.958527	0.958046	0.935646
128	64	0.2	0.925049	0.965346	0.956921	0.950865	0.922802
128	64	0.1	0.956568	0.972613	0.967318	0.965596	0.953271
128	128	0.1	0.962751	0.973848	0.971159	0.969867	0.958176

Evaluation Metrics for different Hyperparametres

It can be observed for LSTM, the model overfits when the Embedding Dimensions, Hidden Dimensions and Learning Rates are high and it was observed to perform the best on the dev set when -

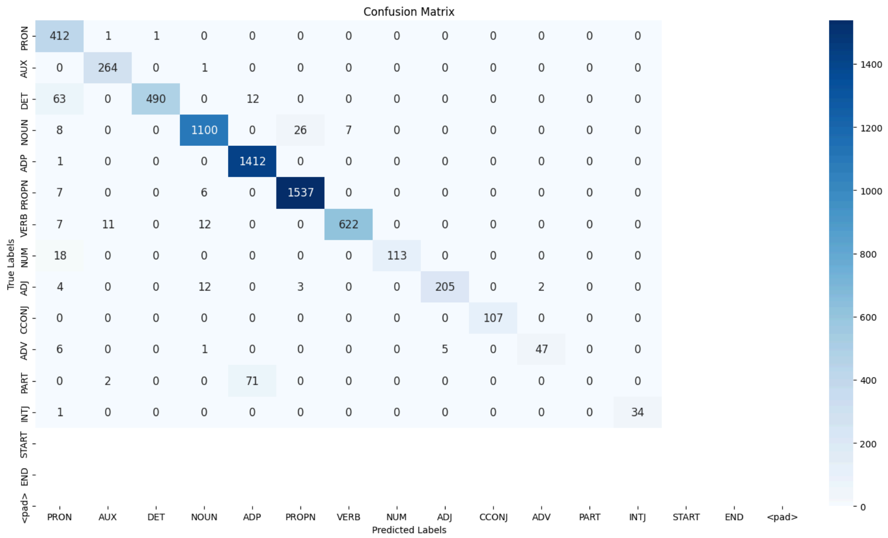
Embedding Dimension = 128
Hidden Dimension = 128
Learning Rate = 0.1
Number of Epochs = 30

These values were then used to analyse models performance through the number of epochs.

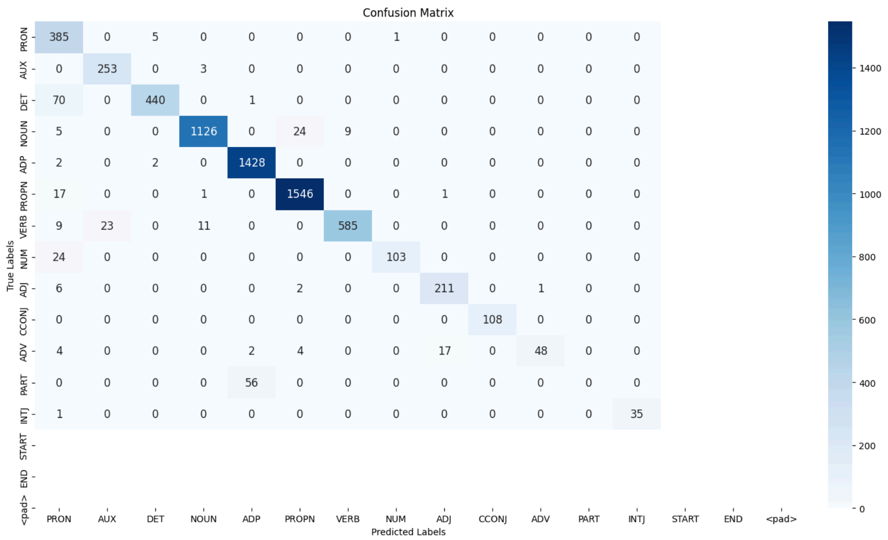


2. Evaluation Metrics

a. Feed Forward Neural Networks



Confusion Matrix for Feed Forward Neural Network on Dev Data

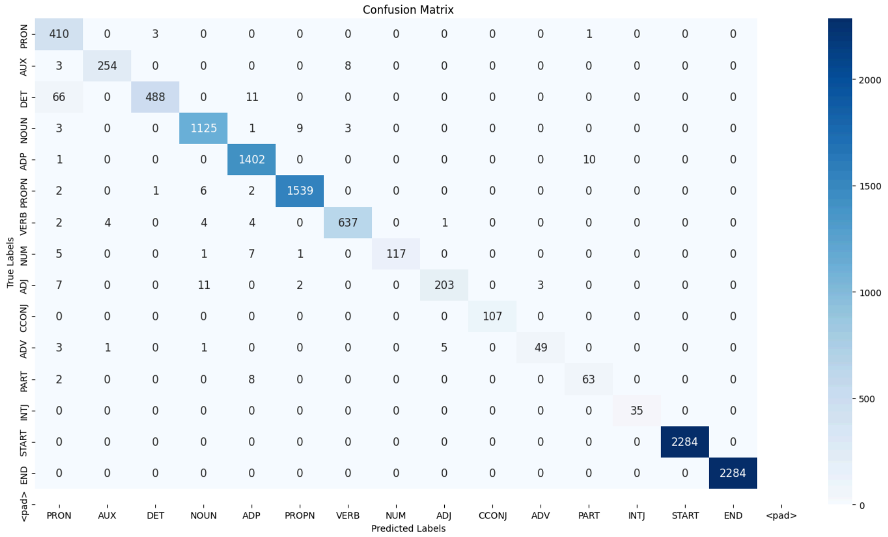


Confusion Matrix for Feed Forward Neural Network on Test Data

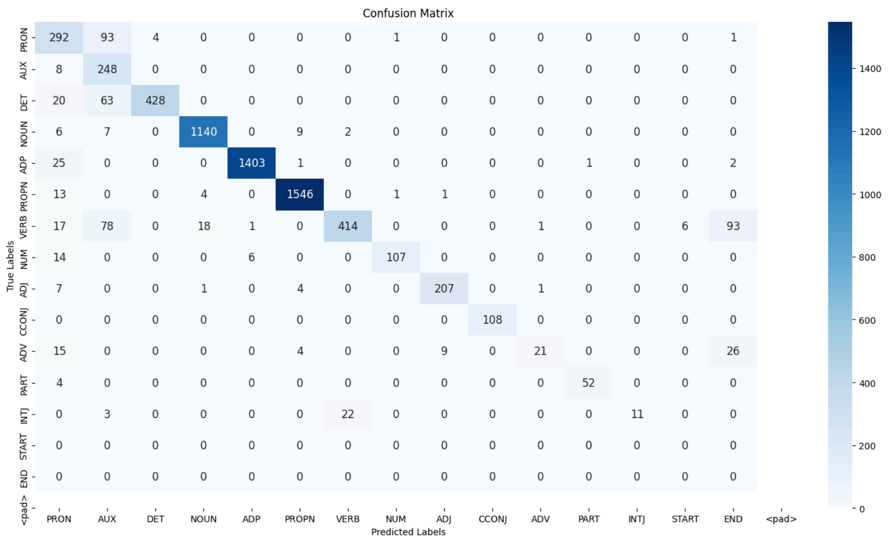
```
Accuracy : 0.9541787182219515
Micro-average Recall: 0.9541787182219515
Micro-average F1-score: 0.95936328154894
Macro-average Recall: 0.8812673893391904
Macro-average F1-score: 0.8920083413009009
```

Scores of Different Evaluation Metrics on Test Data (for FFNN)

b. Recurrent Neural Networks



Confusion Matrix for LSTM on Dev Data



Confusion Matrix for LSTM on Dev Data

```
Accuracy : 0.981516206804179
Micro-average Recall: 0.9851387645478962
Micro-average F1-score: 0.9838615941705039
Macro-average Recall: 0.9704296803441175
Macro-average F1-score: 0.9621537083497247
```

Scores of Different Evaluation Metrics on Test Data (for LSTM)

3. Analysis

- **Context Window** - In the case of Feed Forward Neural Networks, a larger context window helps in capturing more information about the word, thus increasing the accuracy of the model. However, a very large context window may introduce noise which leads to overfitting and it also increases the computational complexity of the model.
- **Hidden Dimension** - Increasing the hidden dimension allows the model to learn more complex patterns from the training data. But if the size is not regularised properly, it can lead to overfitting. Smaller hidden dimensions might result in underfitting.
- **Embedding Dimension** - Higher embedding dimensions capture more semantic information, improving accuracy of the model. However, very high dimensions can lead to overfitting.
- **Learning Rates** - The learning rate determines the step size during optimization of the model. Exceptionally low learning rates cause slow convergence (the model then requires more epochs to get trained) while high learning rates cause the optimizer to overshoot the minimum of the loss function. Instead of converging to the optimal solution, the optimization process may oscillate or diverge, leading to poor performance.
- **Number of Epochs** - The number of epochs can affect the balance between underfitting and overfitting. If the number of epochs is too low, the model may underfit the training data, meaning it fails to capture the underlying patterns in the data. If the number of epochs is too high, the model may overfit the training data, meaning it learns to memorise the training examples instead of generalising to unseen data.
  The number of epochs can affect how quickly the model converges to an optimal solution. With a sufficient number of epochs, the model parameters may converge to values that minimise the loss function. However, if the number of epochs is too low, the model may not have enough time to converge, resulting in suboptimal performance.
- **Evaluation Metrics (recall, f1 score)** - The model shows higher micro averages (recall and f1 score) in comparison to macro averages, because there is a class imbalance in the dataset and micro-averaged metrics give more weight to larger classes, while macro-averaged metrics treat all classes equally.