NLP\_Assignment2\_Report

Name: Udrasht Pal

Roll NO: 2022201020

Question 3 and 4 answer combine below

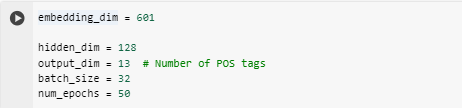
**3.1**

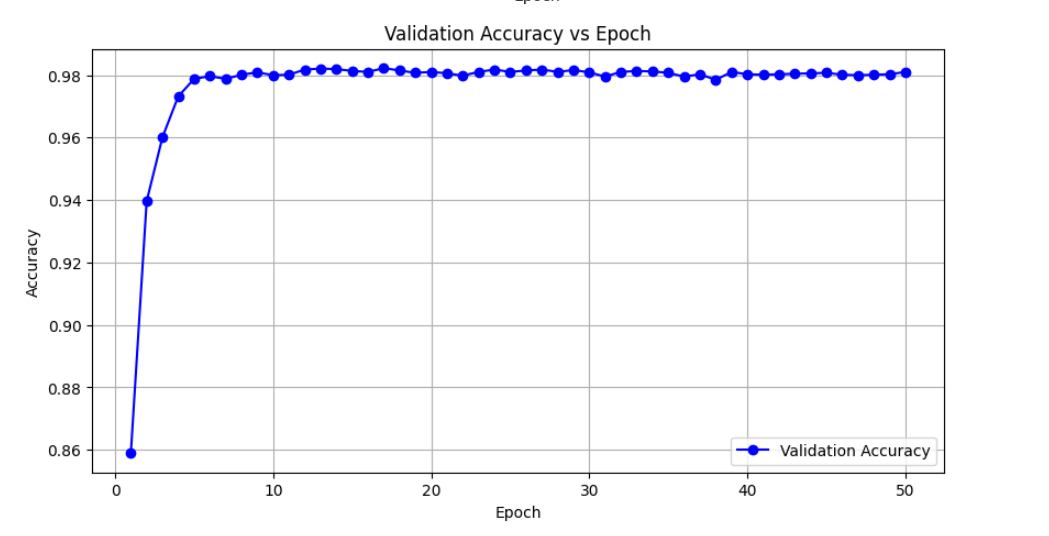
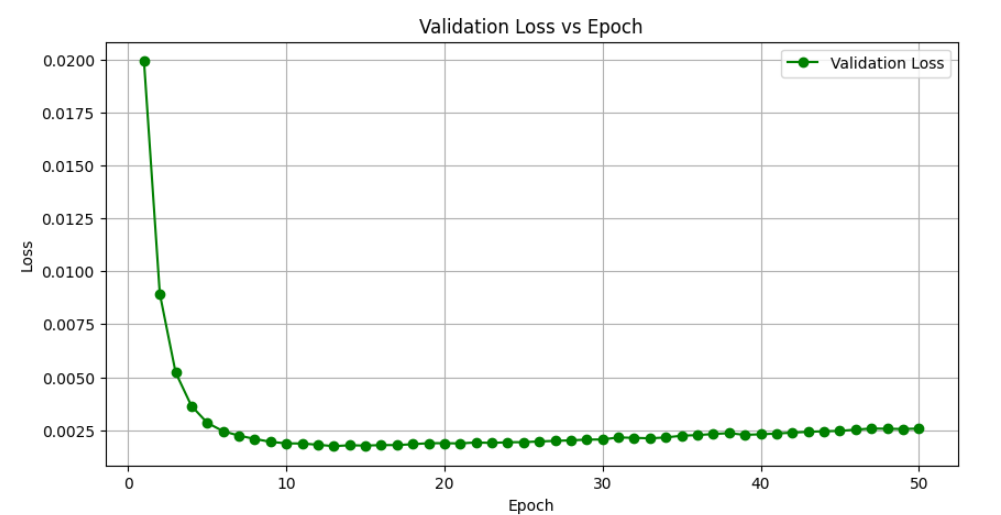
I explored various hyper-parameter and model architecture options. Multiple approaches were used to optimize hyper-parameters and model architecture.

I experiment the model by changing the dimension of hidden layer, learning rate, loss function

A few of the possible analyses are displayed below

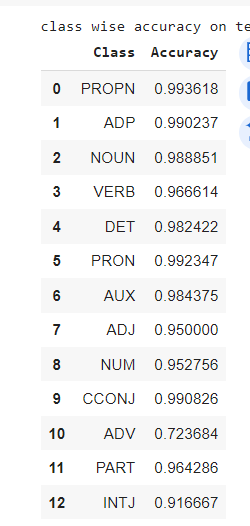
Hyper-parameters Tuning

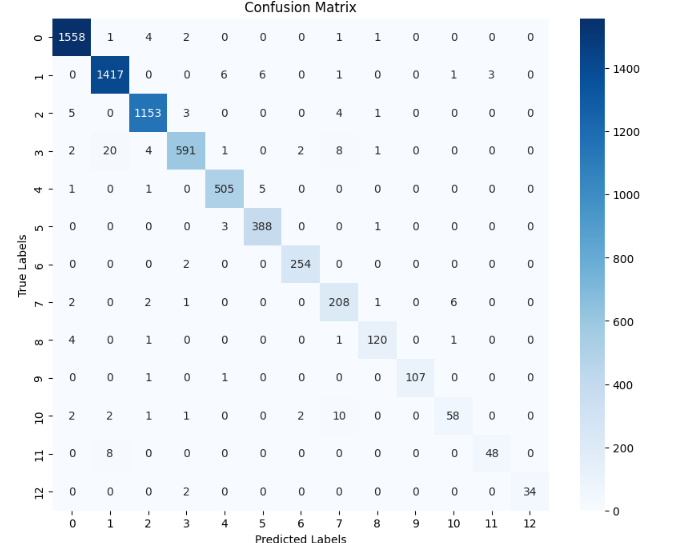


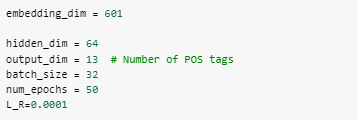


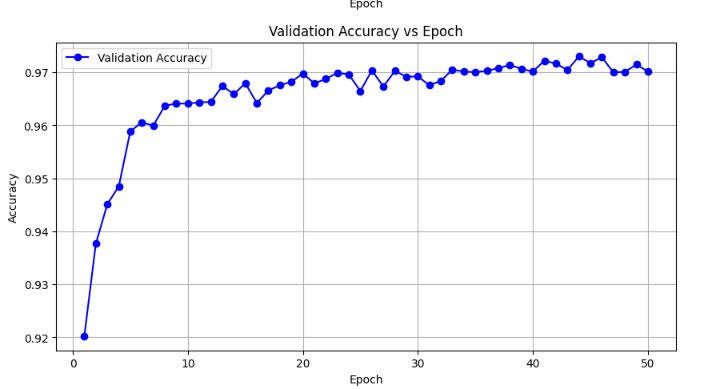
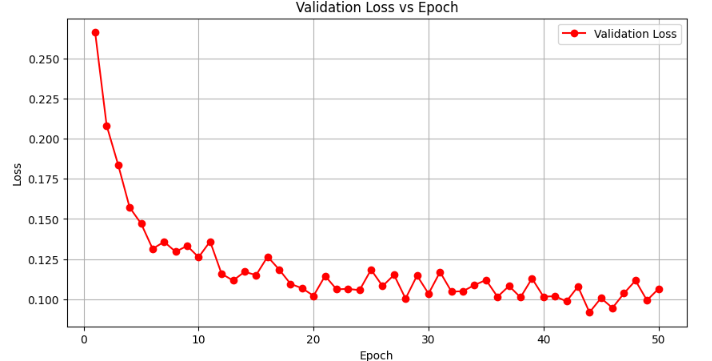
Loss vs epoch graph for dev set Accuracy vs epoch graph for dev set

On test data set

r



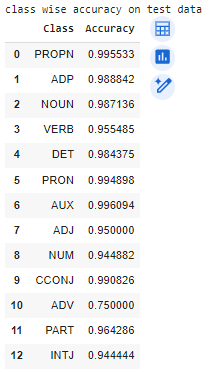
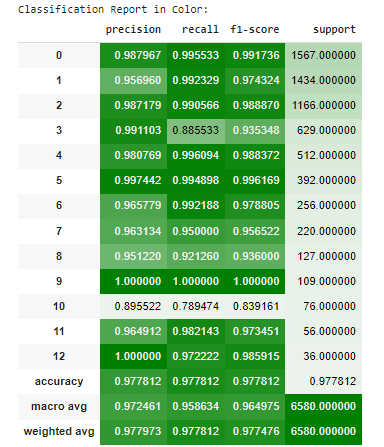


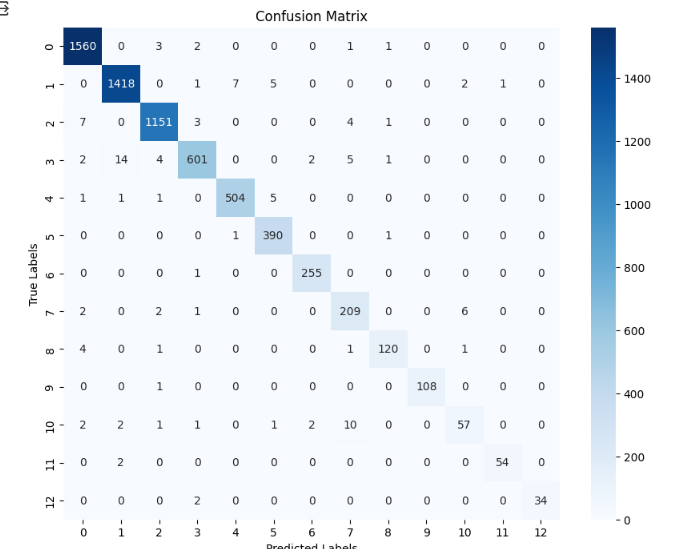


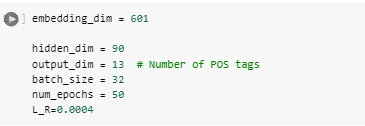
Loss vs epoch graph for dev set Accuracy vs epoch graph for dev

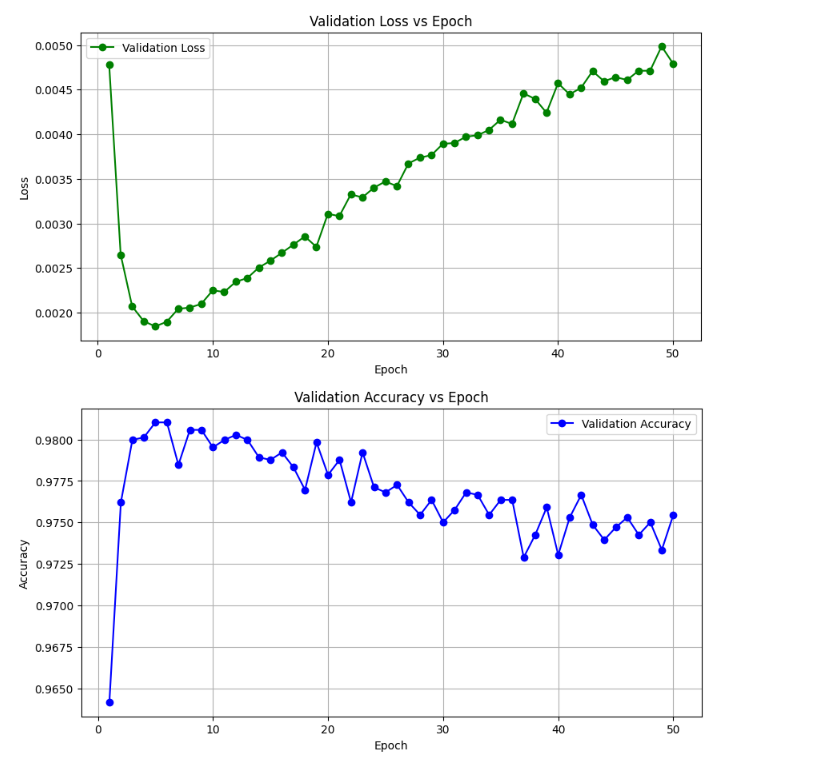
**This model perform better as compare to other model in FFNN**

On test set



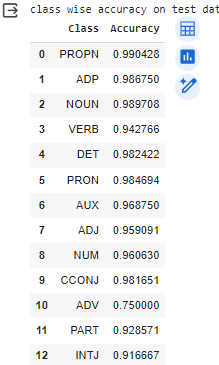
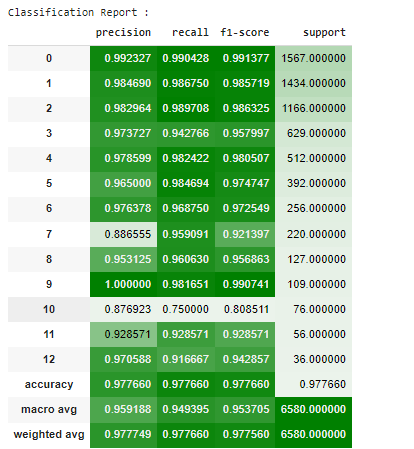


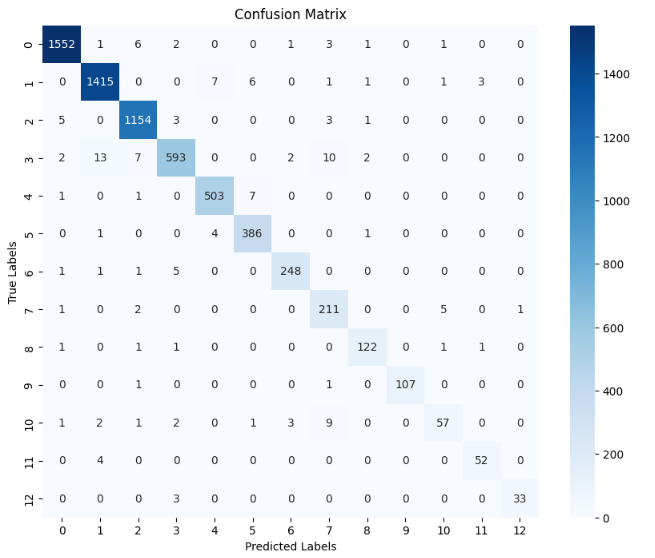




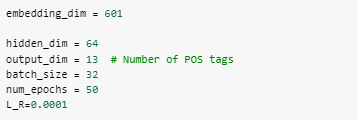
This model perform worst hear the loss value is increase and accuracy decrease

On test data

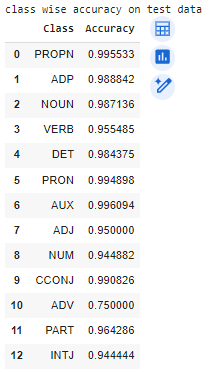
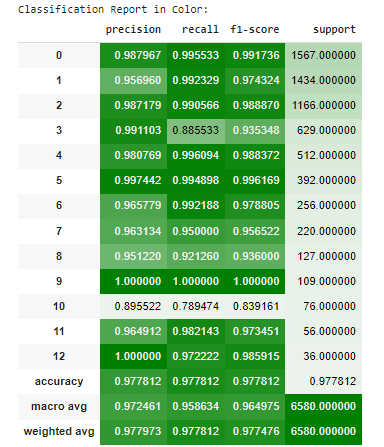


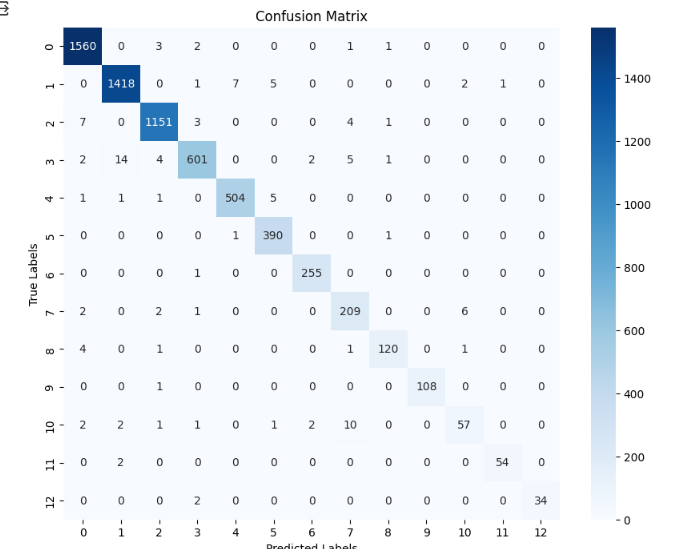


**Conclusion**: On below parameters my model perform best in my best knowledge with test set accuracy 97.5% accuracy on test set and 97% on dev set



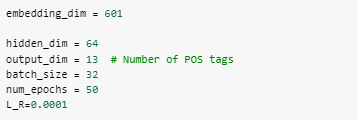
On test set

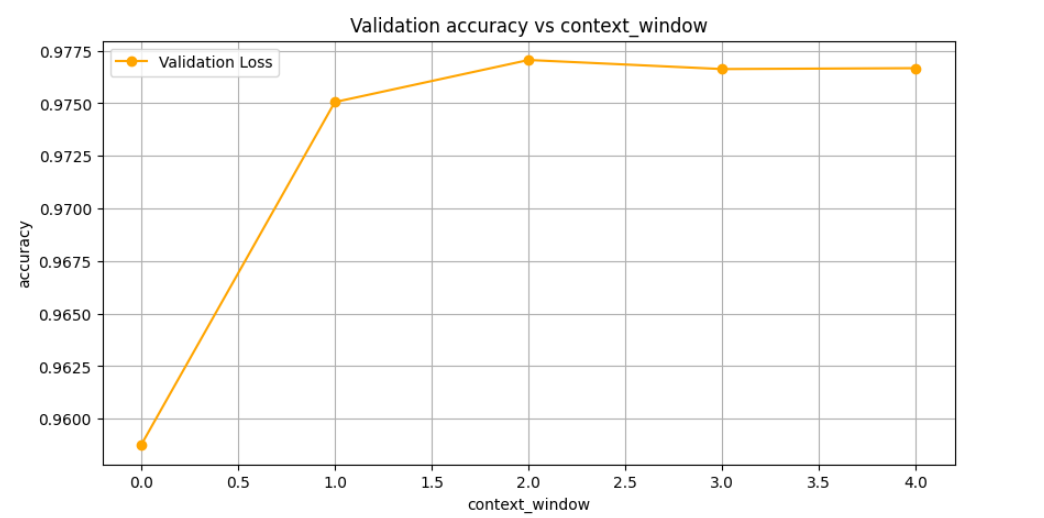




So I draw accuracy vs context\_window graph on dev set

On context\_window size =2 (p=2 , s=2) model perform best





Accuracy vs context\_window graph

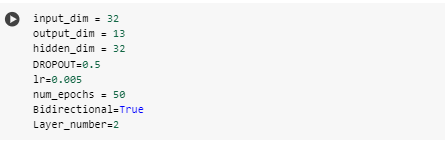
3.2

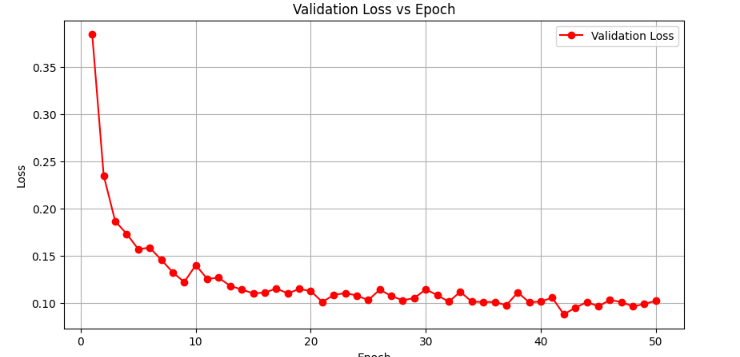
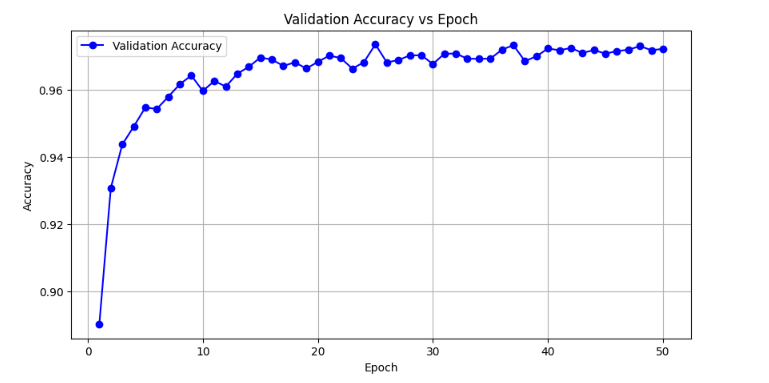
RNN🡪 LSTM

I explored various hyper-parameter and model architecture options. Multiple approaches were used to optimize hyper-parameters and model architecture.

A few of the possible analyses are displayed below

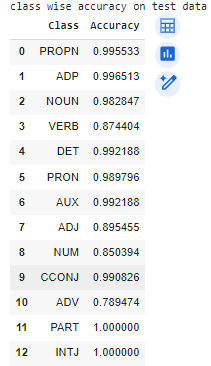
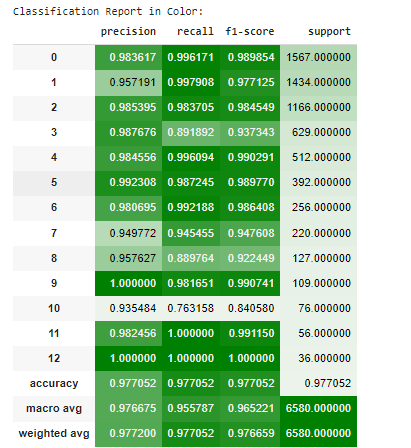
Hyper-parameters Tuning

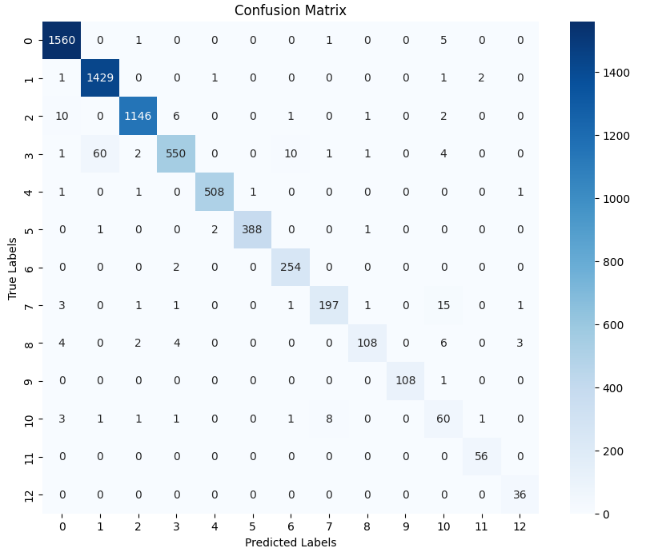


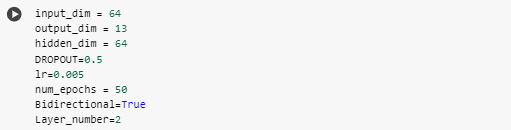


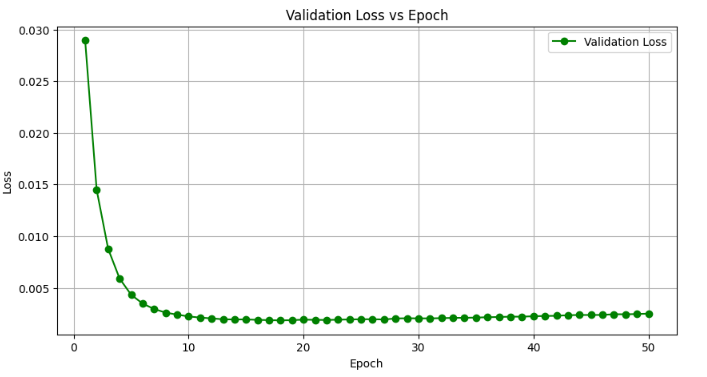
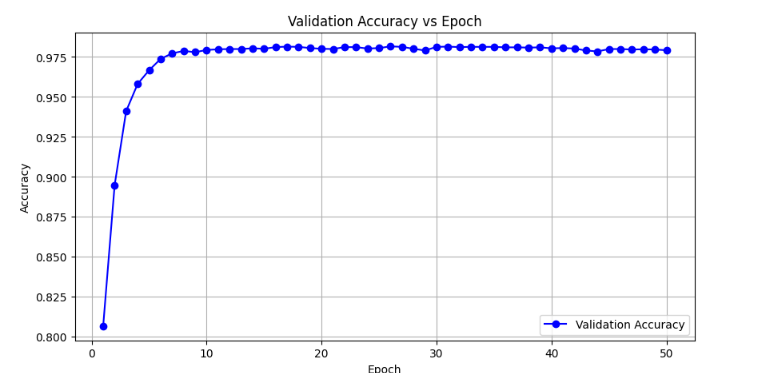
Accuracy vs epoch graph for dev set Loss vs epoch graph for dev set

On test set



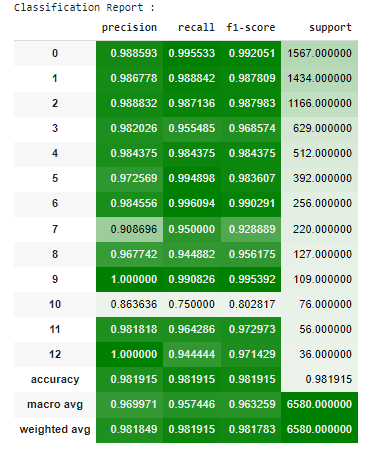


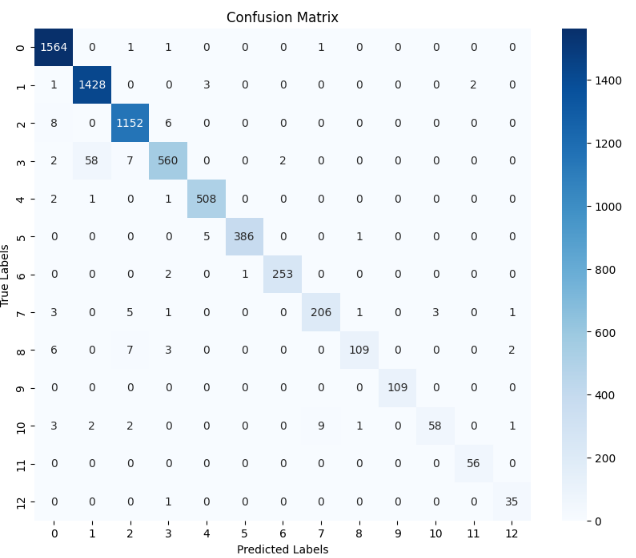


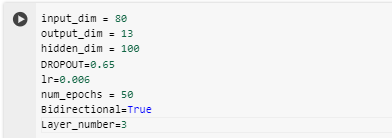


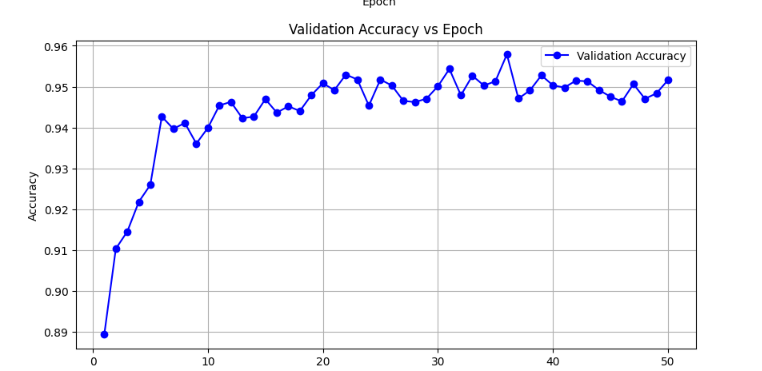
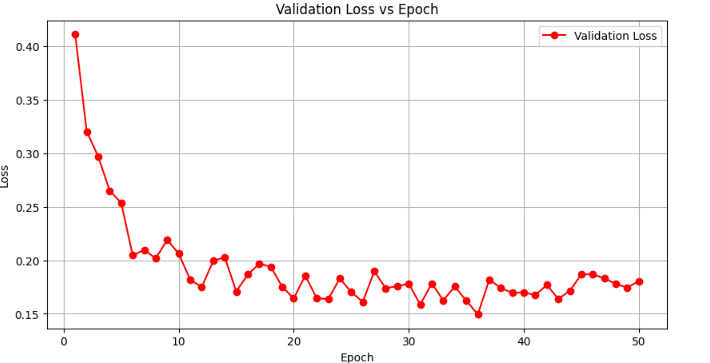
Accuracy vs epoch graph for dev set Loss vs epoch graph for dev set

On test Data



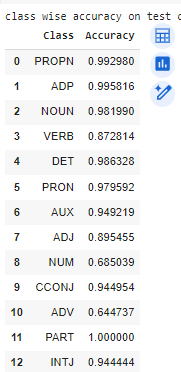


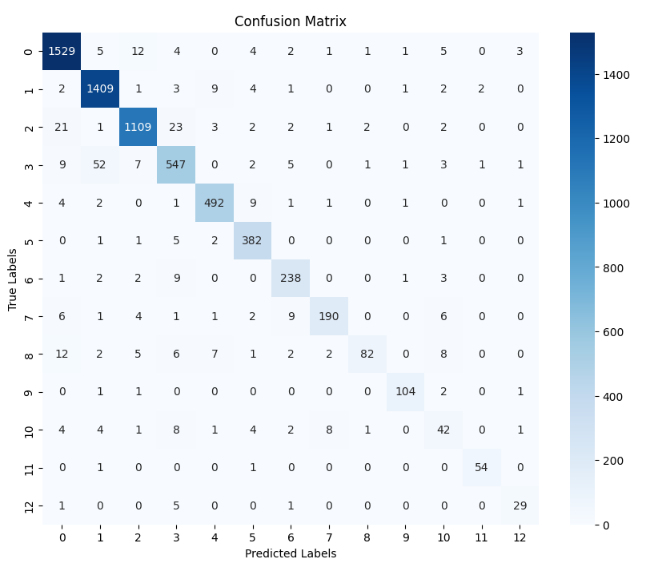




Accuracy vs epoch graph for dev set Loss vs epoch graph for dev set

On test data

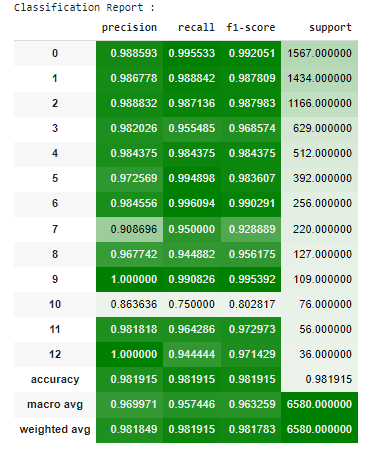




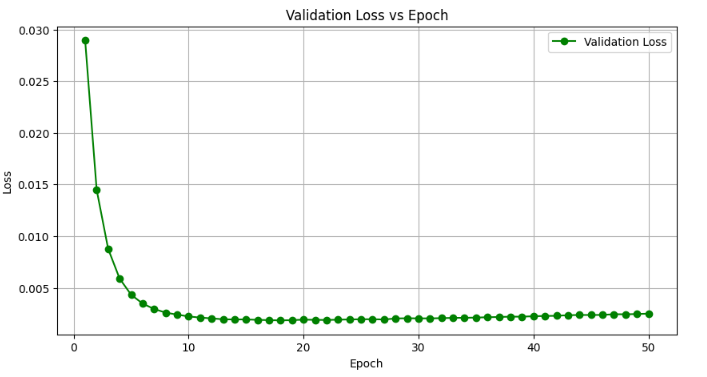
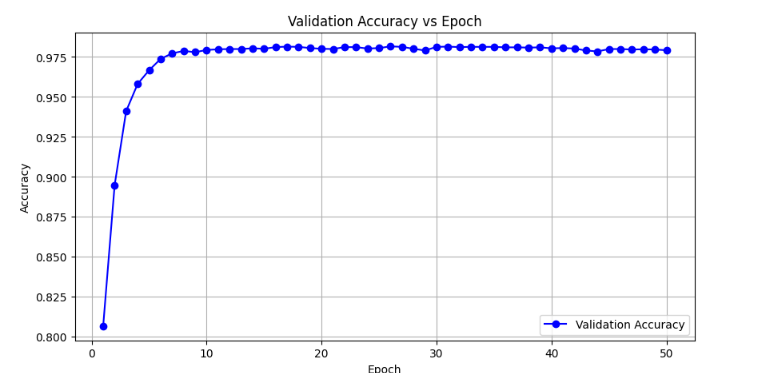
**Result:**

After following the approach discussed above , I was able to achieve accuracy above 98.5% on test set.

Below are the classification report on test set



The above figures show different measurements like precision, recall, and F1-score, which balance precision and recall by calculating their harmonic mean. Below, there are plots that show the validation accuracy and loss during training. These plots analyze how well the model performs with each training epoch.



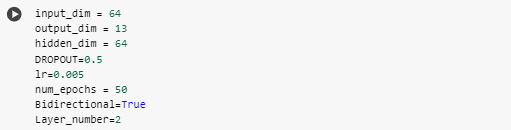
Accuracy vs epoch graph for dev set Loss vs epoch graph for dev set

The results demonstrate that as the number of epochs increase, there is a reduction in overall loss and an increase in accuracy. This indicates that the model has been effectively trained.

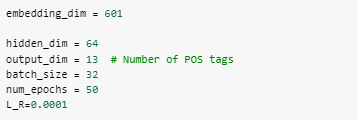
**5 Analysis:**

According to my analysis the RNN(LSTM) perform better Post tagging then FFNN because it can capture sequential dependencies in the input data, which is crucial for understanding language structure.

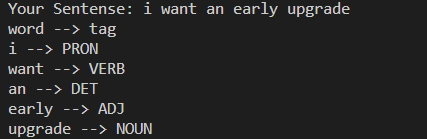
LSTM model: test set accuracy 98.5% on



FFNN model: test set accuracy 97.5% approx. on



Some input output sample for my FFNN model



Below are the some input output result of my model on RNN LSTM

