

Feed Forward Neural Network POS Tagging

3 Configurations:

p = p previous words, s = s successive words

1. where $p = 1, s = 1$

dev_accuracy = 0.9847328244274809

test_accuracy = 0.9891691982565051

2. where $p = 2, s = 2$

dev_accuracy = 0.9828902342721769

test_accuracy = 0.9856029586580373

3. where $p = 3, s = 4,$

dev_accuracy = 0.9792050539615689

test_accuracy = 0.983225465592392

Best configuration was where $p = s = 1$.

Hyperparameter Tuning on the best configuration:

Hyperparameters used:

1. hidden dimensions
2. number of hidden layers
3. embedding dimensions
4. number of layers
5. activation functions

Hyperparameter Tuning Results:

Embedding Dim	Hidden Dim 1	Hidden Dim 2	Activation	Dev Accuracy	Test Accuracy
100	512	128	ReLU	0.9843	0.9857
100	512	128	Tanh	0.9835	0.9896
100	512	64	ReLU	0.9853	0.9880
100	512	64	Tanh	0.9825	0.9886
100	256	128	ReLU	0.9842	0.9893
100	256	128	Tanh	0.9845	0.9898
100	256	64	ReLU	0.9849	0.9881
100	256	64	Tanh	0.9826	0.9871
200	512	128	ReLU	0.9842	0.9872
200	512	128	Tanh	0.9845	0.9877
200	512	64	ReLU	0.9838	0.9882
200	512	64	Tanh	0.9833	0.9878
200	256	128	ReLU	0.9842	0.9892
200	256	128	Tanh	0.9866	0.9905
200	256	64	ReLU	0.9838	0.9898
200	256	64	Tanh	0.9862	0.9902

Observations:

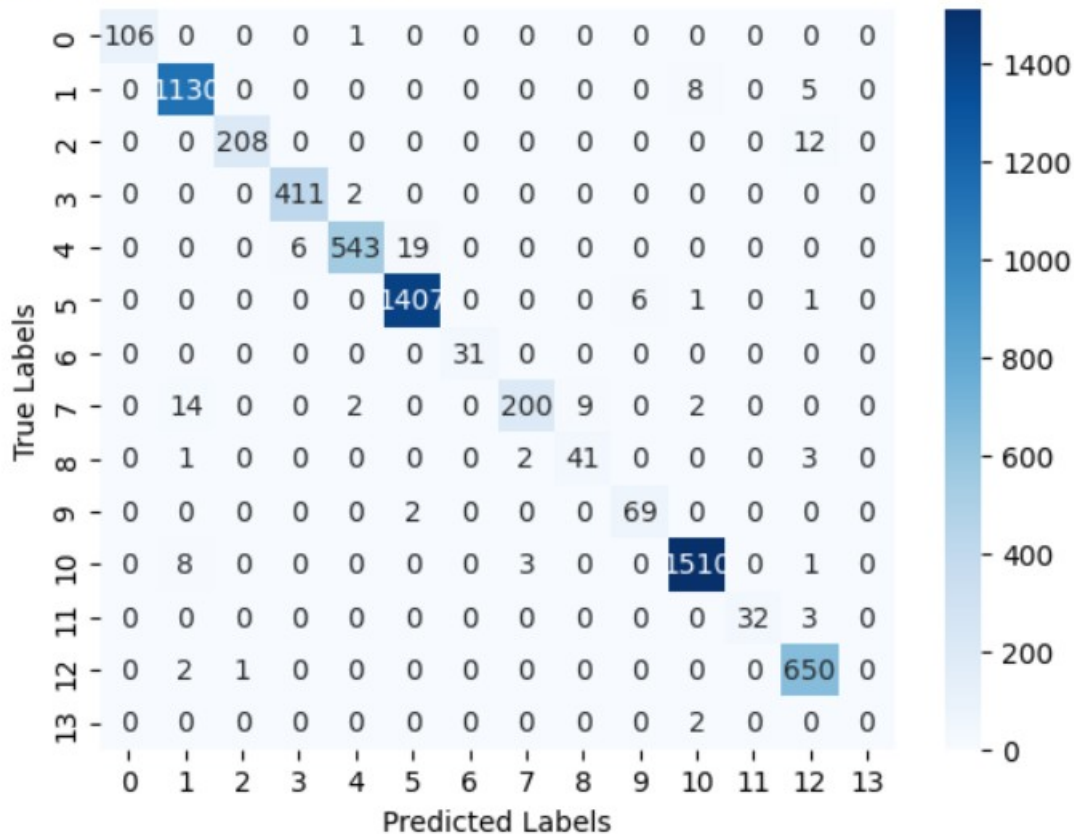
- 1. Hidden dimensions:** We can see that as we the number of hidden neurons increase from 256 to 512 in the first hidden layer and from 64 to 128, the dev and test accuracies also increase. But since our model already was giving high accuracy, the increase is not significant. The best test accuracy of 0.9905 we get, is for the combination of [512, 128] of hidden layer 1 and 2 respectively.
- 2. Number of hidden layers:** As we increase the number of hidden layers, the accuracy increases as it was able to learn more complex patterns.
- 3. Embedding dimensions:** Embedding dim of 100 and 200 were used. The model give better accuracies for 200 dim embedding infering that the increase in embedding dim is increases acurracy score.
- 4. Number of layers:** It is observed that as we increase more number of layers, the model is able to perform better.
- 5. Activation functions:** Used 2 different functions, ReLU and tanh. The model gives better accuracies when tanh was used compared to ReLU. The best configuration observed is [embedding_dim, hidden_dim1, hidden_dim2, activation] = [200, 256, 128, tanh] with a test accuracy of 0.9905.

Evaluation Metrics of best config $p = s = 1$,

```
Evaluation Metrics on train set
Evaluating: 100%|██████████| 1750/1750 [00:17<00:00, 101.85batch/s]
Accuracy: 0.9904613900648411
Precision: 0.99062147312443
Recall: 0.9904613900648411
F1 Score: 0.9904583826166543
Evaluation Metrics on dev set
Evaluating: 100%|██████████| 238/238 [00:00<00:00, 313.23batch/s]
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined: Labels 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 have no predicted samples
  _warn_prf(average, modifier, msg_start, len(result))
Accuracy: 0.9847328244274809
Precision: 0.9846565046841257
Recall: 0.9847328244274809
F1 Score: 0.9845278451358083
Evaluation Metrics on test set
Evaluating: 100%|██████████| 237/237 [00:00<00:00, 499.15batch/s]Accuracy: 0.9896975300488707
Precision: 0.9894888732377837
Recall: 0.9896975300488707
F1 Score: 0.9894902321728604
```

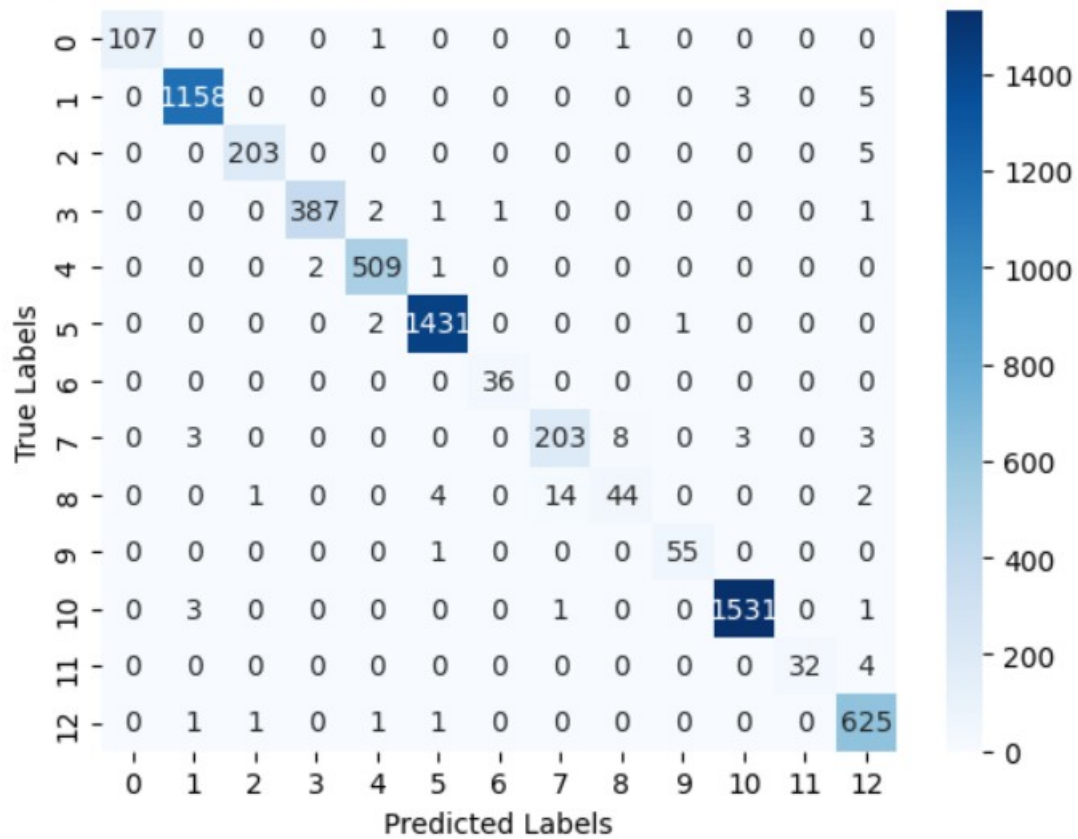
Dev set confusion matrix:

Dev set Confusion matrix

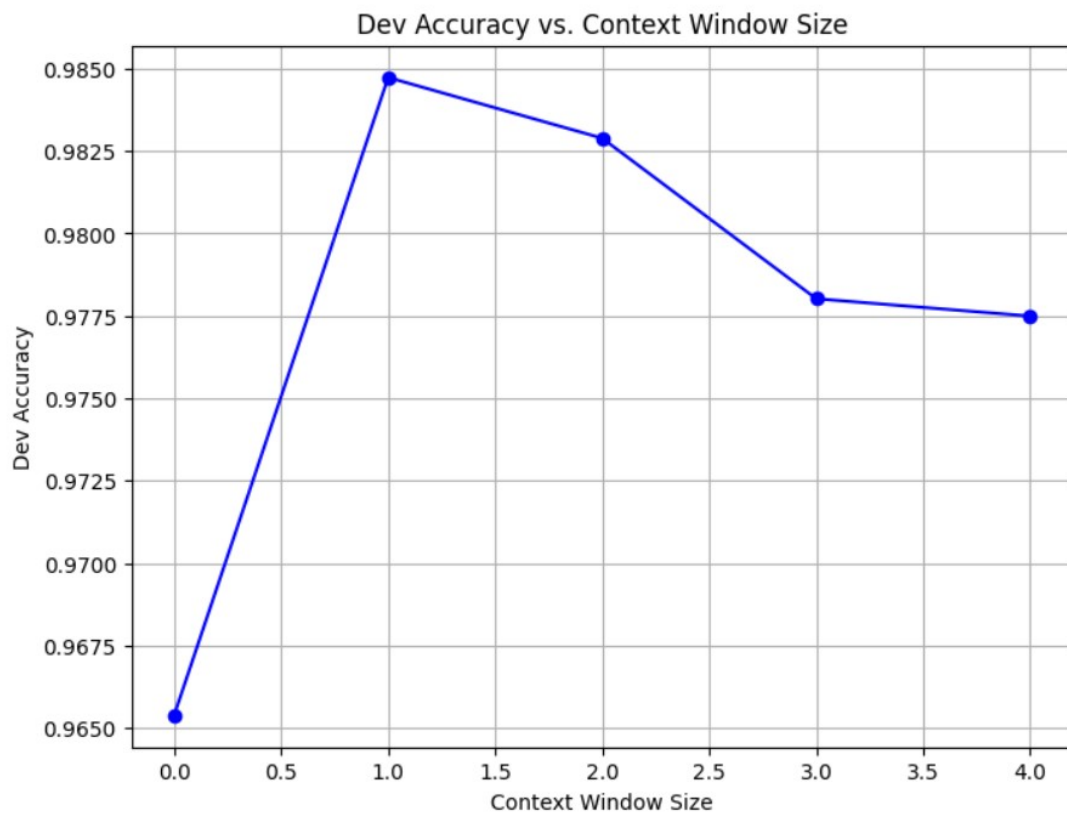


Test Set confusion matrix

Test set Confusion matrix



Graph of context window vs dev set accuracies

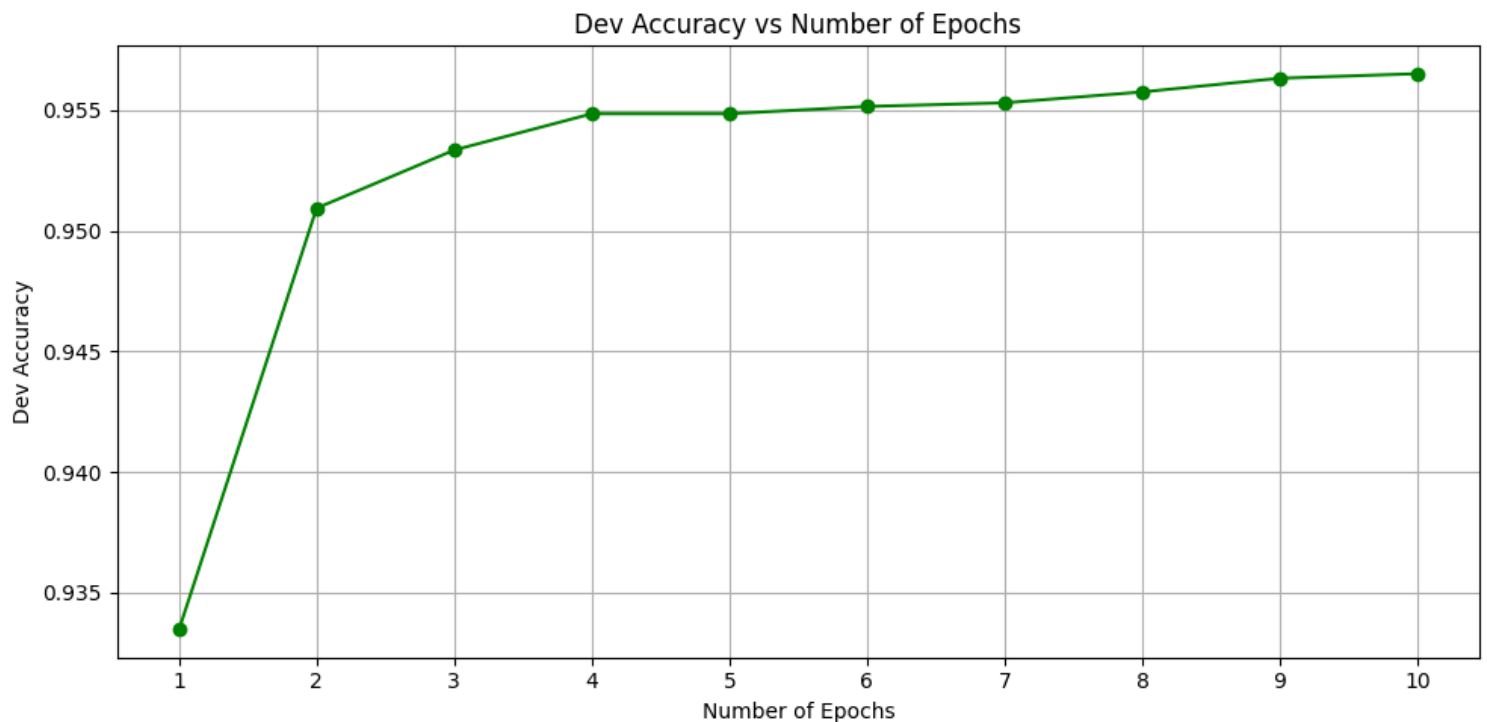


II. LSTMs

3 configurations:

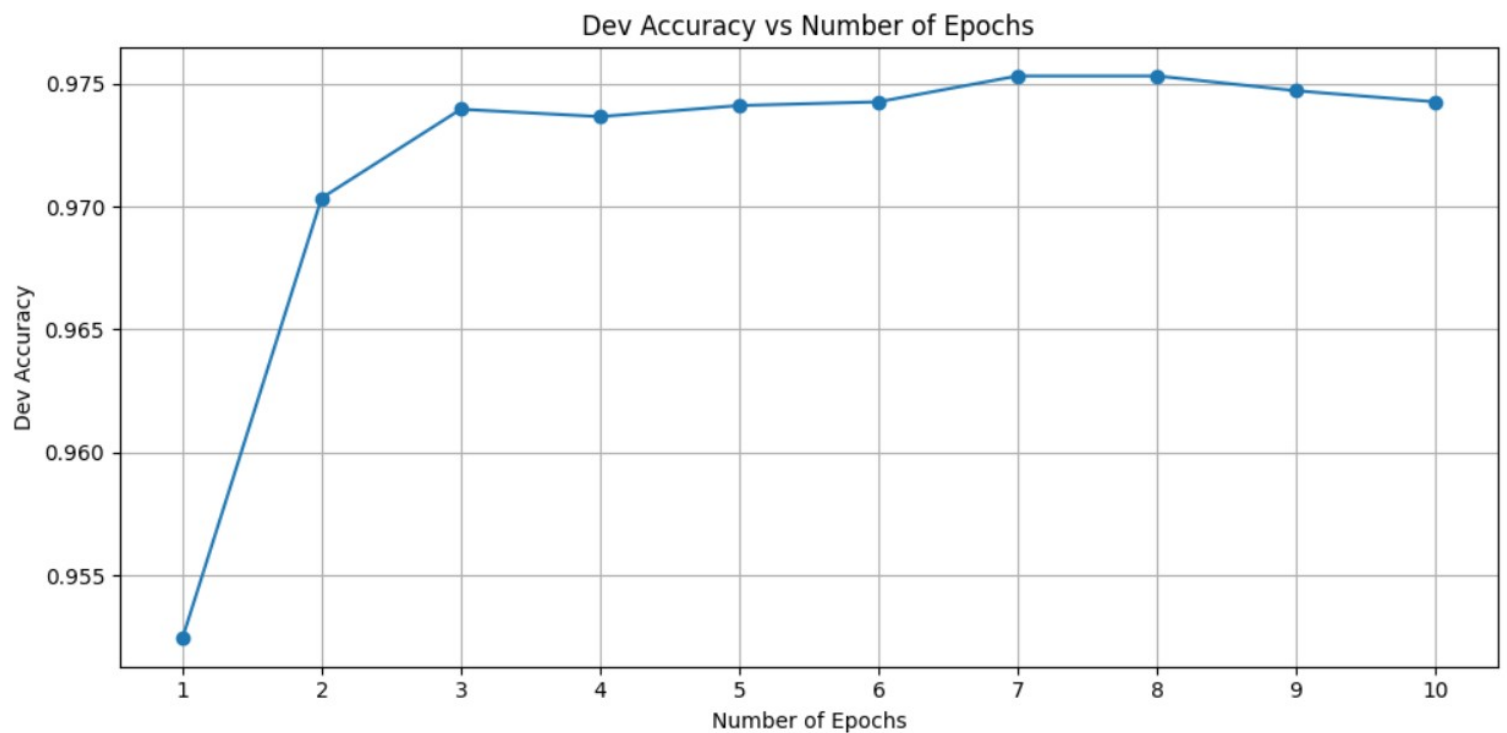
1. `bidirectional = False`, `embedding_dim = 100`, `hidden_dim = 128`,
activation function = `relu`

Epoch 1/10, Train Loss: 0.29760895058981235, Train Accuracy: 0.9098962079950673, Dev Accuracy: 0.9334738109572547
Epoch 2/10, Train Loss: 0.08763557096997218, Train Accuracy: 0.9705477340458329, Dev Accuracy: 0.9509331727874775
Epoch 3/10, Train Loss: 0.06359696162943426, Train Accuracy: 0.9770013359366971, Dev Accuracy: 0.9533413606261288
Epoch 4/10, Train Loss: 0.05453905282005636, Train Accuracy: 0.979488233480629, Dev Accuracy: 0.954846478025286
Epoch 5/10, Train Loss: 0.049355180538601794, Train Accuracy: 0.9807830644332546, Dev Accuracy: 0.954846478025286
Epoch 6/10, Train Loss: 0.04572752358149503, Train Accuracy: 0.9820984482581441, Dev Accuracy: 0.9551475015051174
Epoch 7/10, Train Loss: 0.04302514090147347, Train Accuracy: 0.9829616688932278, Dev Accuracy: 0.9552980132450332
Epoch 8/10, Train Loss: 0.04083418762550319, Train Accuracy: 0.9839276538896311, Dev Accuracy: 0.9557495484647802
Epoch 9/10, Train Loss: 0.03905054281945688, Train Accuracy: 0.9843592642071729, Dev Accuracy: 0.9563515954244431
Epoch 10/10, Train Loss: 0.037603000073115116, Train Accuracy: 0.9848730860137704, Dev Accuracy: 0.9560505719446116
Test Accuracy: 0.955775075987842



2. `bidirectional=True`, `embedding_dim = 100`, `hidden_dim = 128`,
`activation function = ReLU`

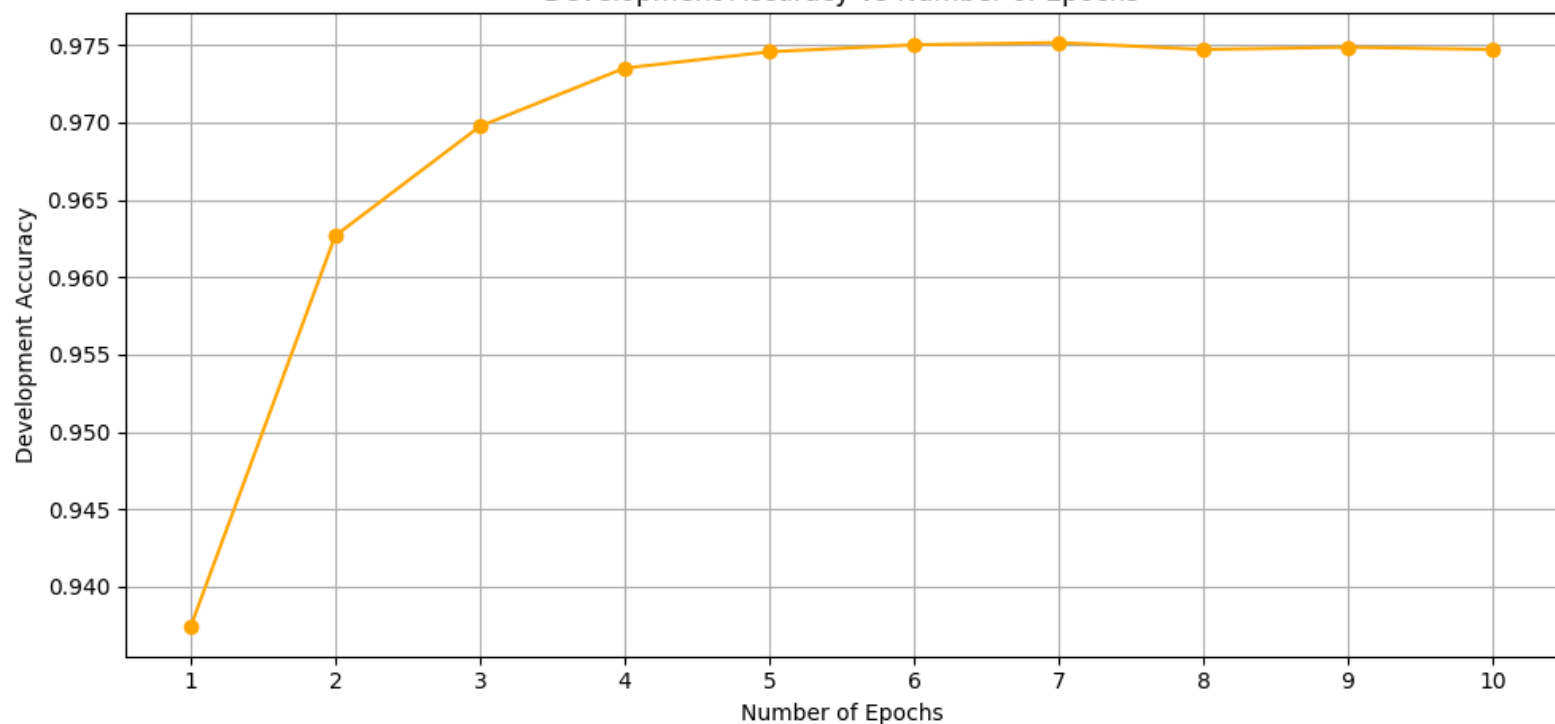
Epoch 1/10, Train Loss: 0.27313665058677494, Train Accuracy: 0.9194121878532525, Dev Accuracy: 0.9524382901866345
Epoch 2/10, Train Loss: 0.06843052229357875, Train Accuracy: 0.9791182817798787, Dev Accuracy: 0.9703491872366045
Epoch 3/10, Train Loss: 0.04569067336677271, Train Accuracy: 0.9857979652656459, Dev Accuracy: 0.9739614689945816
Epoch 4/10, Train Loss: 0.03697571603734155, Train Accuracy: 0.9878326996197718, Dev Accuracy: 0.9736604455147502
Epoch 5/10, Train Loss: 0.031842221027846215, Train Accuracy: 0.9889836604665502, Dev Accuracy: 0.9741119807344973
Epoch 6/10, Train Loss: 0.028113175259739383, Train Accuracy: 0.9899907512074813, Dev Accuracy: 0.974262492474413
Epoch 7/10, Train Loss: 0.02512098084435017, Train Accuracy: 0.9907923132257733, Dev Accuracy: 0.975316074653823
Epoch 8/10, Train Loss: 0.02262644368535122, Train Accuracy: 0.9915527694995375, Dev Accuracy: 0.975316074653823
Epoch 9/10, Train Loss: 0.02054450740233665, Train Accuracy: 0.9921899085397184, Dev Accuracy: 0.9747140276941602
Epoch 10/10, Train Loss: 0.01873950346378706, Train Accuracy: 0.9928681533244271, Dev Accuracy: 0.974262492474413



3. `bidirectional = true`, `embedding dim = 100`, `hidden dim = 512`,
`activation = relu`

Training configuration: `embedding_dim=100`, `hidden_dim=512`, `activation_function=relu`
Epoch 1: Train Loss: 0.28800013523957846, Train Accuracy: 0.9464186620080156, Dev Accuracy: 0.9378386514148104
Epoch 2: Train Loss: 0.09007485440287721, Train Accuracy: 0.968697975542082, Dev Accuracy: 0.962673088500903
Epoch 3: Train Loss: 0.05732953555319042, Train Accuracy: 0.9761792210461412, Dev Accuracy: 0.9697471402769416
Epoch 4: Train Loss: 0.04330384112032295, Train Accuracy: 0.9796115507142124, Dev Accuracy: 0.9735099337748344
Epoch 5: Train Loss: 0.03563495332470134, Train Accuracy: 0.9813379919843798, Dev Accuracy: 0.9745635159542444
Epoch 6: Train Loss: 0.030691794184228124, Train Accuracy: 0.9823450827253108, Dev Accuracy: 0.9750150511739916
Epoch 7: Train Loss: 0.02699386998121182, Train Accuracy: 0.9831055389990752, Dev Accuracy: 0.9751655629139073
Epoch 8: Train Loss: 0.0240987911835682, Train Accuracy: 0.9839687596341589, Dev Accuracy: 0.9747140276941602
Epoch 9: Train Loss: 0.021716403124178985, Train Accuracy: 0.9844209228239647, Dev Accuracy: 0.9748645394340758
Epoch 10: Train Loss: 0.01969424058263888, Train Accuracy: 0.9848525331415066, Dev Accuracy: 0.9747140276941602
Test Accuracy: 0.9759878419452888

Development Accuracy vs Number of Epochs



Best configuration is the 3rd one, `bidirectional = true`, `embedding dim = 100`, `hidden dim = 128`, `activation = relu`, with test accuracy of 0.9777

Hyperparameter Tuning:

hidden_dim	embedding_dim	activation	test_accuracy
128	100	relu	0.97750759878
128	100	tanh	0.97598784194
256	100	relu	0.97720364741
256	100	tanh	0.97583586626
512	100	relu	0.97598784194
512	100	tanh	0.97370820668
128	200	relu	0.97325227963
128	200	tanh	0.97462006079
256	200	relu	0.97507598784
256	200	tanh	0.97462006079
512	200	relu	0.97431610942
512	200	tanh	0.97215532811

Observations:

1. Hidden dimensions: we can see that as dimensions increase there is a slight decrease in the test accuracy and the model takes a lot more time to train than with a dimension of 128.

2. Embedding dimensions: we can see that there is a slight decrease in the accuracy for 200 compared to 100.

3. Activation function: Using relu showed better results than using tanh.

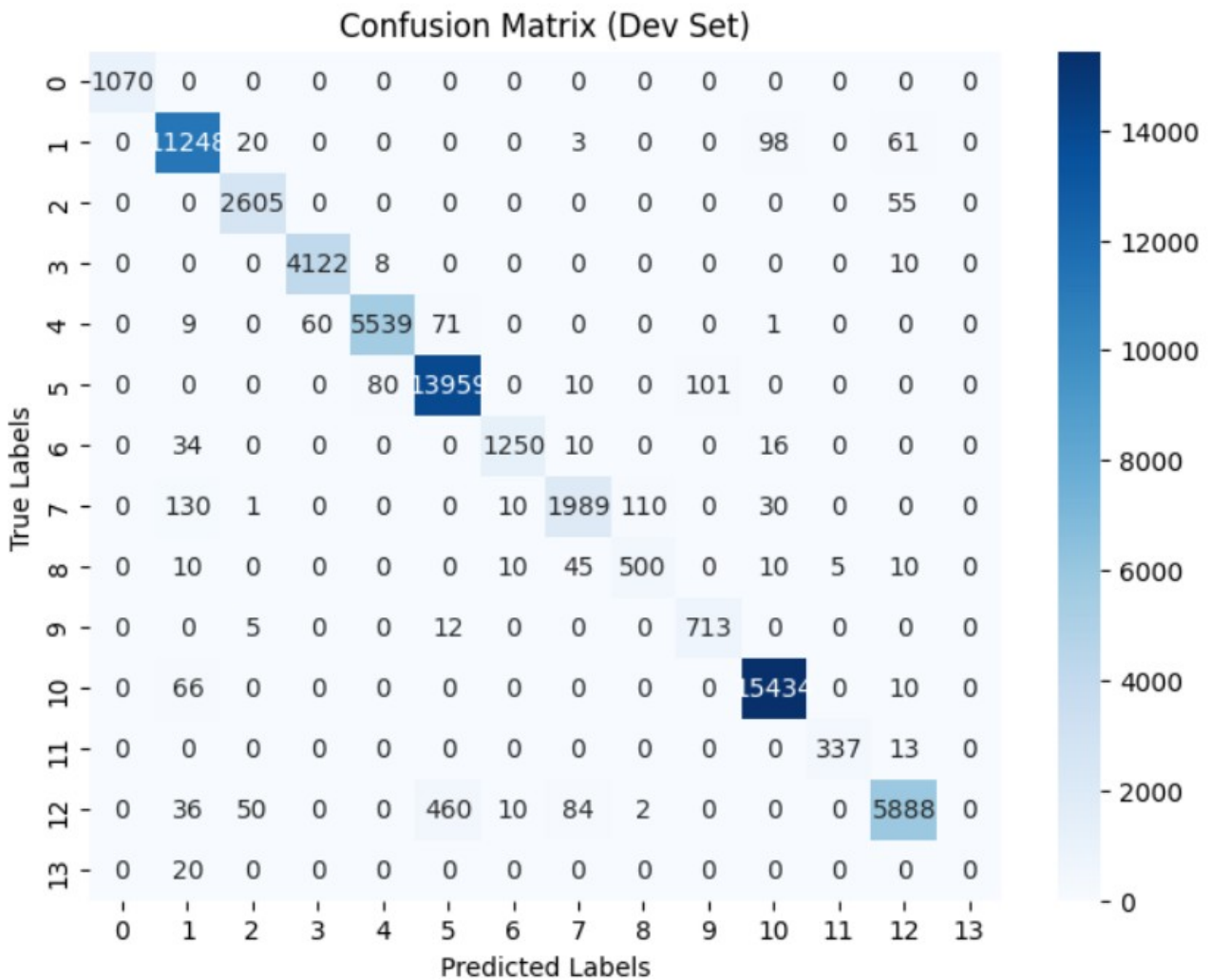
Best configuration, evaluation metrics:

Test Accuracy: 0.9762917933130699

Precision: 0.9764465781141688,

Recall: 0.9762917933130699,

F1-score: 0.9761323262170662



Confusion Matrix (Test Set)

