Jonathan Mandl 211399175

Danielle Hodaya Shrem 208150433

Part 1 – A Simple Window-Based Tagger

Architecture

For this part, we used the same hyperparameter configuration to both our POS and NER tasks, as it delivered strong performance on each.

Our neural network consists of a single hidden layer with 250 neurons and tanh activation.

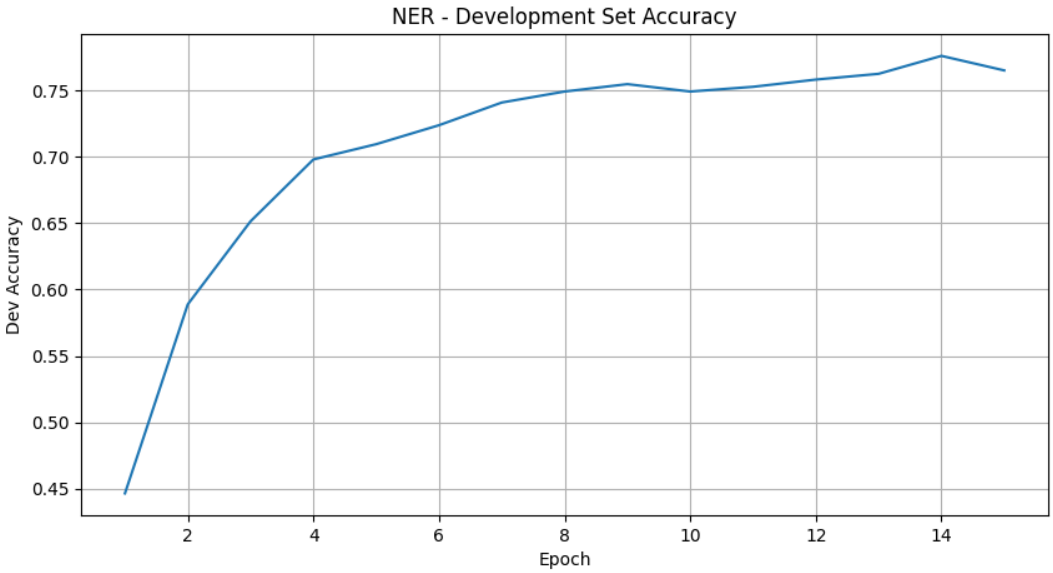
Our chosen hyperparameters were:

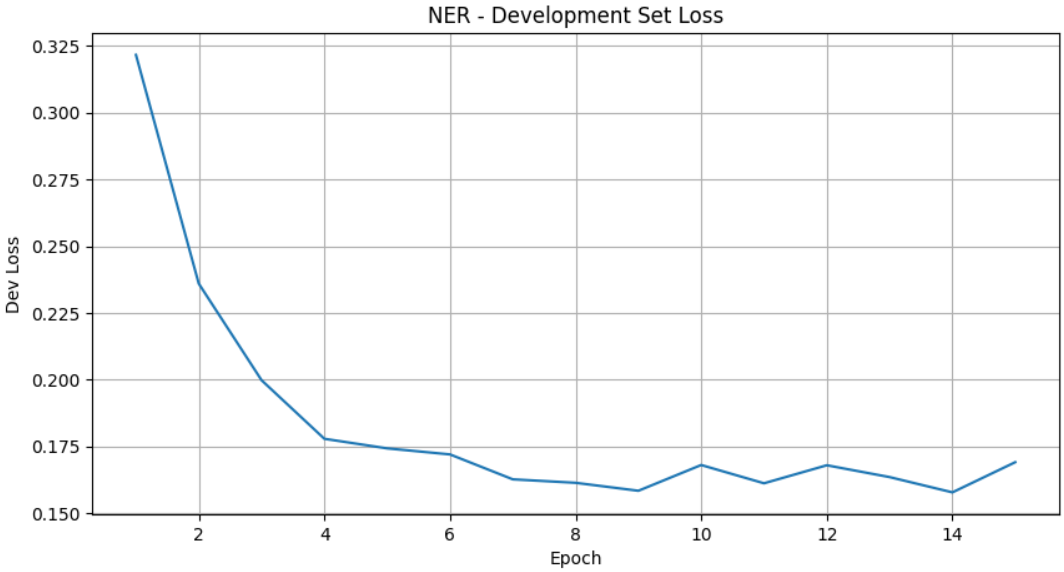
- **Learning rate:** 0.001
- **Epochs:** 15
- **Batch size:** 64

Model choices

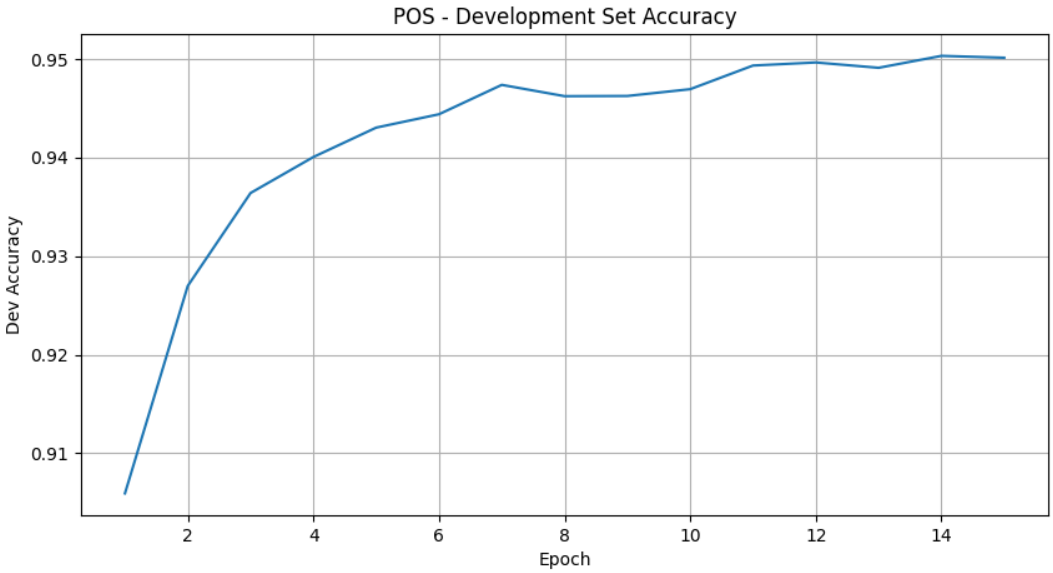
To handle words that appear in the development set but not in training, we added a special <UNK> token to our embedding matrix. During training, we randomly masked 15% of tokens—replacing them with the <UNK> token - so that the model learns a useful representation for unknown words. To handle context windows that include sentence boundaries, we padded each input with <PAD> tokens on both sides (equal to the context size) and likewise included <PAD> in our embedding matrix.

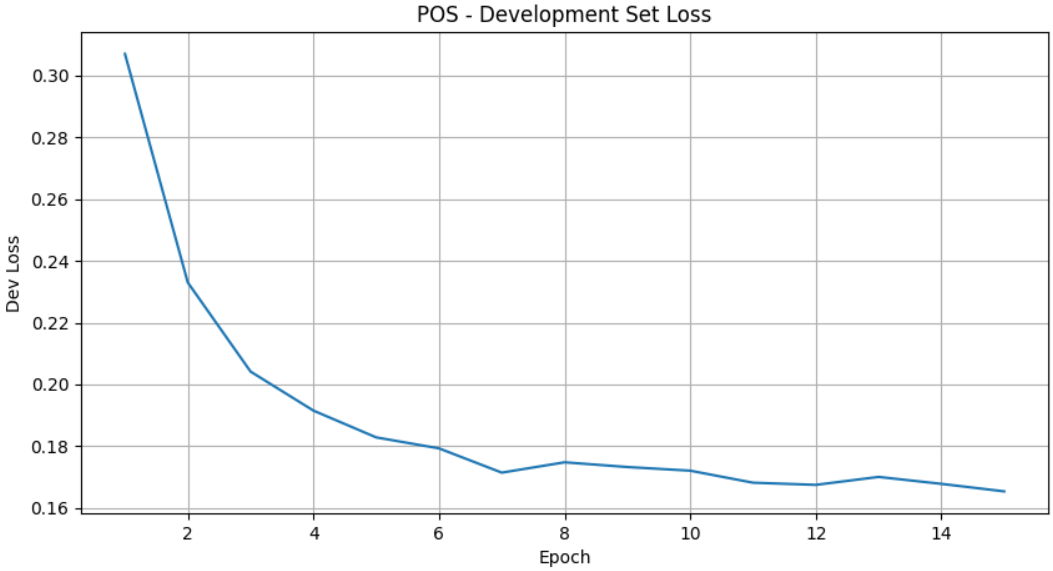
Task -NER


NER - Development Set Accuracy
Dev Accuracy
0.75
0.70
0.65
0.60
0.55
0.50
0.45
2
4
6
8
10
12
14
Epoch


NER - Development Set Loss
Dev Loss
0.325
0.300
0.275
0.250
0.225
0.200
0.175
0.150
2
4
6
8
10
12
14
Epoch

Task - POS


POS - Development Set Accuracy
Dev Accuracy
0.95
0.94
0.93
0.92
0.91
2
4
6
8
10
12
14
Epoch


POS - Development Set Loss
Dev Loss
0.30
0.28
0.26
0.24
0.22
0.20
0.18
0.16
2
4
6
8
10
12
14
Epoch