**BiLSTM Tagger Report**

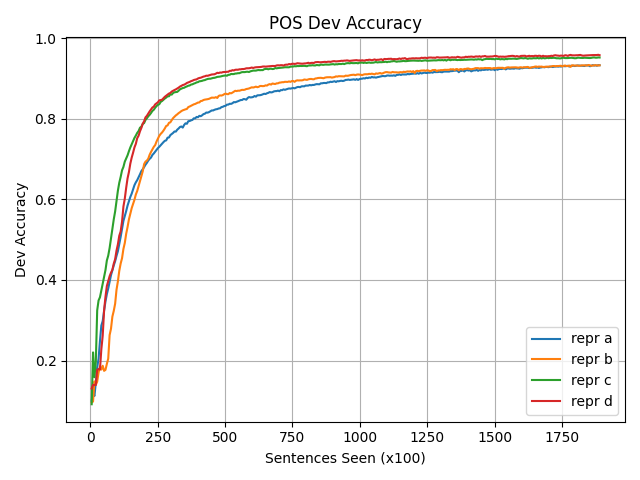
**Model Description**

We implemented a 2-layer BiLSTM-based sequence tagger to predict linguistic labels (POS or NER) over input sequences. The model includes the following architecture:

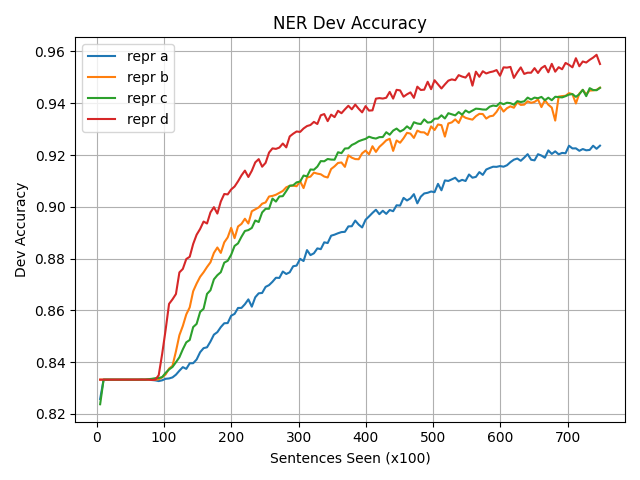
* **Embedding Layer**: Word-level embeddings of 30 dimensions.
* **Optional Character/Subword Embeddings**:
  + **(b)** Char-BiLSTM over character embeddings (dim 15), followed by a bidirectional LSTM.
  + **(c)** Sum of word, prefix, and suffix embeddings (dim 30).
  + **(d)** Concatenation of (a) and (b) passed through a projection layer.
* **Two BiLSTM layers**: Each with 50 hidden units per direction.
* **Dropout**: 0.3 after the second BiLSTM.
* **Classifier**: Linear layer over the BiLSTM output.
* **Loss**: CrossEntropyLoss with ignore\_index=-100 for padding.
* **Optimizer**: Adam, learning rate=0.001.

Training was done for 5 epochs with batch size 64.

**Graphs**

**POS Dev Accuracy**

**NER Dev Accuracy**



The x-axis shows the number of sentences seen (in hundreds), and the y-axis shows the dev accuracy. Every 500 sentences, dev accuracy was evaluated.

**Best Model Selection**

Based on the dev accuracy curves:

* **Best POS model**: Representation **d** with final dev accuracy **0.9591**
* **Best NER model**: Representation **d** with final dev accuracy **0.9587**

**Final Predictions**

The following models were used to generate the final test predictions:

python bilstmPredict.py d model\_pos\_d.pt pos/test > test4.pos

python bilstmPredict.py d model\_ner\_d.pt ner/test > test4.ner

These predictions are saved in test4.pos and test4.ner respectively.

**Notes**

* The model files (\*.pt) and accuracy logs (\*.pt.accuracies.csv) are saved and used to track training progress.
* Evaluation was computed on dev set only, blind test accuracy is not available.