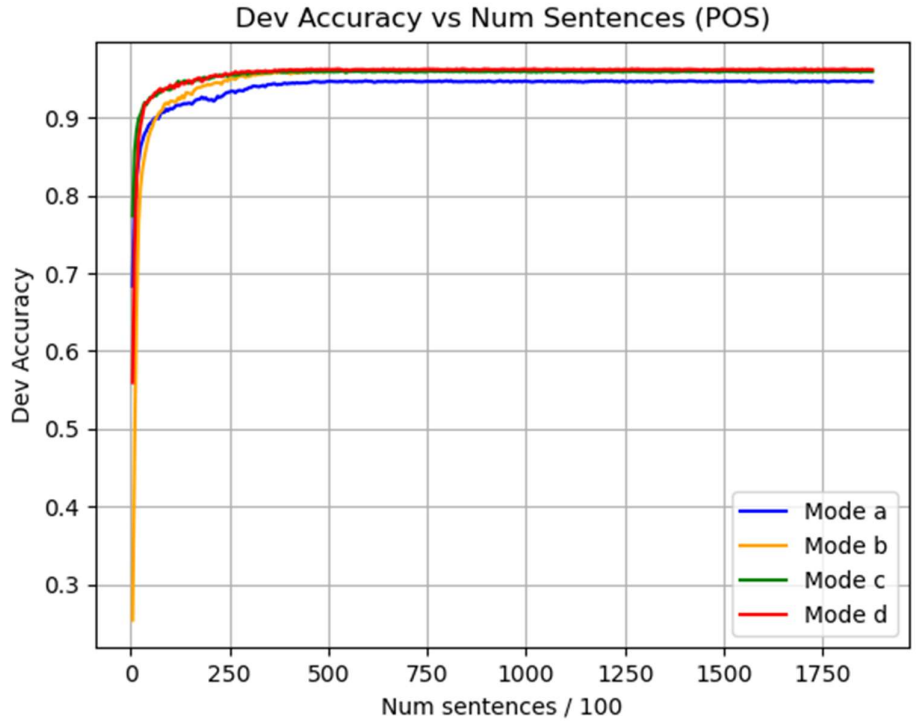
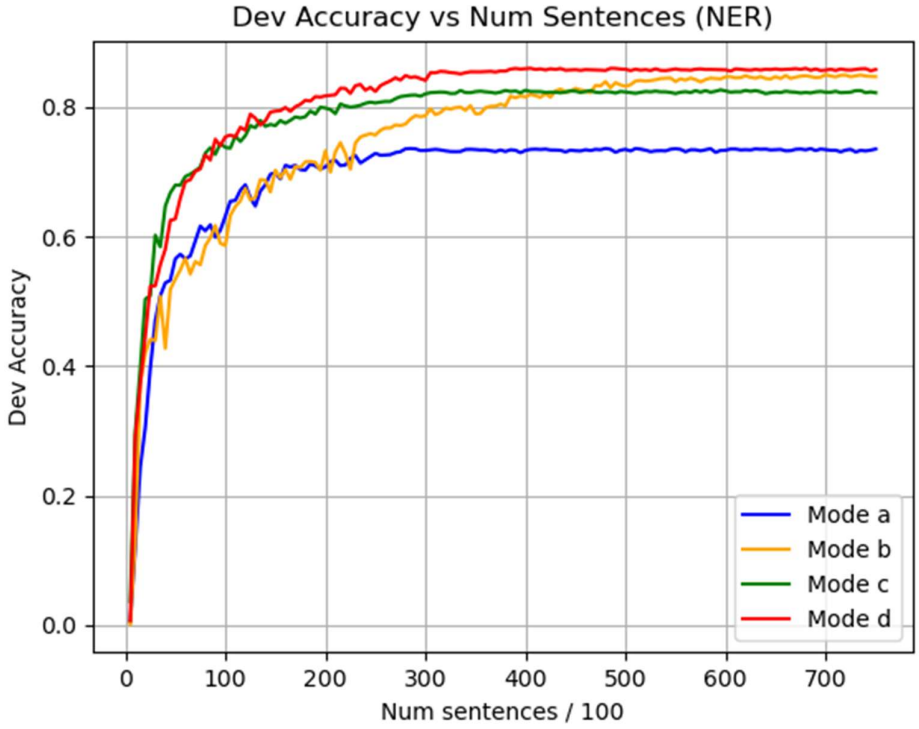
Bilstm:

- We experimented with various hyperparameter, but our final architecture was the following:
    - **Learning rate – 0.015**
    - **Embedding dimension - 200**
    - **Char embedding dimension – 100** (for modes 'b' and 'd')
    - **Hidden dimension - 128**
    - **Batch size – 25 sentences**
    - **Dropout – 0.25**

  The only exception to the above was when we trained our model in mode 'd' for the named entity recognition task. The NER task is known to be harder than the part of speech tagging task, and the increased number of parameters for word embeddings was hard for our model to express with these parameters. We had to increase them to be much bigger for the mode to work nicely, and we ended up with:
    - **Learning rate – 0.005**
    - **Embedding dimension - 400**
    - **Char embedding dimension – 200**
    - **Hidden dimension - 512**
    - **Batch size – 50 sentences**
    - **Dropout – 0.4**

- Another interesting point worth mentioning arose while we implemented mode 'b' of the tagger. We first implemented (using LSTM cells) a one-directional LSTM network to turn the word's characters into embeddings. That approach gave us terrible word representation and accuracy. Interestingly, switching to bi-directional LSTM network for embedding, gave us wonderful results, and improved the accuracy by a huge amount.
- The required graphs are below, the first one is for the POS task, and the second for the NER tasks. The different colors are assigned to the different modes according to the legend in each graph:





- One thing worth noting is the different accuracy curves of mode ‘b’ in the POS and NER tasks. In our opinion, it can be explained by the fact that mode ‘b’ mostly helps assign tags to unknown words, so because the NER dataset is smaller, it gained more from switching to mode ‘b’.