REPORT PART 3

1) Common model parameters :

- Optimizer : ADAM
- lr : 0.001
- Epochs : 5
- Evaluation frequency : 500

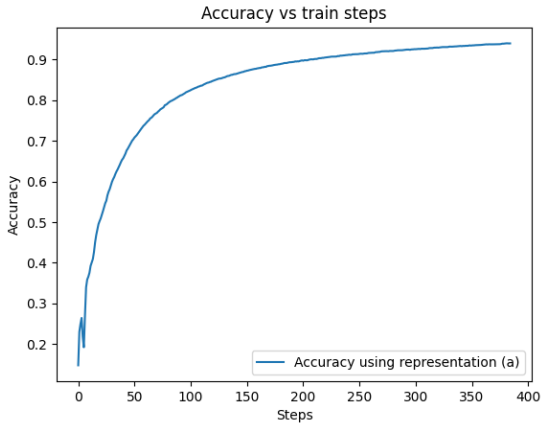
2) POS Tagging :

a) Representation a, word embedding

Model Architecture:

```
TaggerBiLSTM(
  (embedding): Embedding(112981, 32, padding_idx=100232)
  (bilstm): BiLSTM(
    (lstm_cell_forward): LSTM(
      (lstm_cell): LSTMCell(32, 32)
    )
    (lstm_cell_backward): LSTM(
      (lstm_cell): LSTMCell(32, 32)
    )
  )
  (classifier): Linear(in_features=64, out_features=46, bias=True)
)
```

Results:



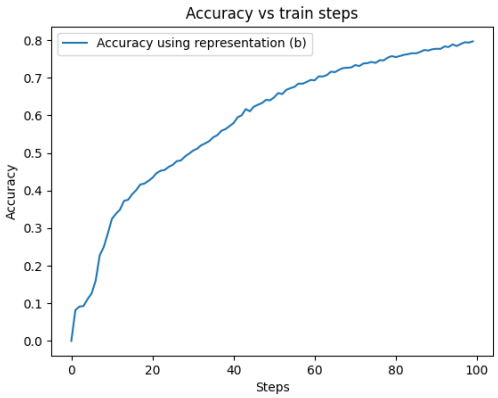
Accuracy on dev: 0.9340

b) Representation b, character level embedding

Model Architecture:

```
CharTaggerBiLSTM(
  (embedding): CharLSTM(
    (embedding): Embedding(112981, 32, padding_idx=94)
    (lstm): LSTM(
      (lstm_cell): LSTMCell(32, 4)
    )
  )
  (lstm): BiLSTM(
    (lstm_cell_forward): LSTM(
      (lstm_cell): LSTMCell(4, 32)
    )
    (lstm_cell_backward): LSTM(
      (lstm_cell): LSTMCell(4, 32)
    )
  )
  (classifier_2): Sequential(
    (0): Linear(in_features=64, out_features=256, bias=True)
    (1): ReLU()
    (2): Linear(in_features=256, out_features=256, bias=True)
    (3): ReLU()
    (4): Linear(in_features=256, out_features=46, bias=True)
  )
)
```

Results:



Accuracy on dev: 0.8011

Remark:

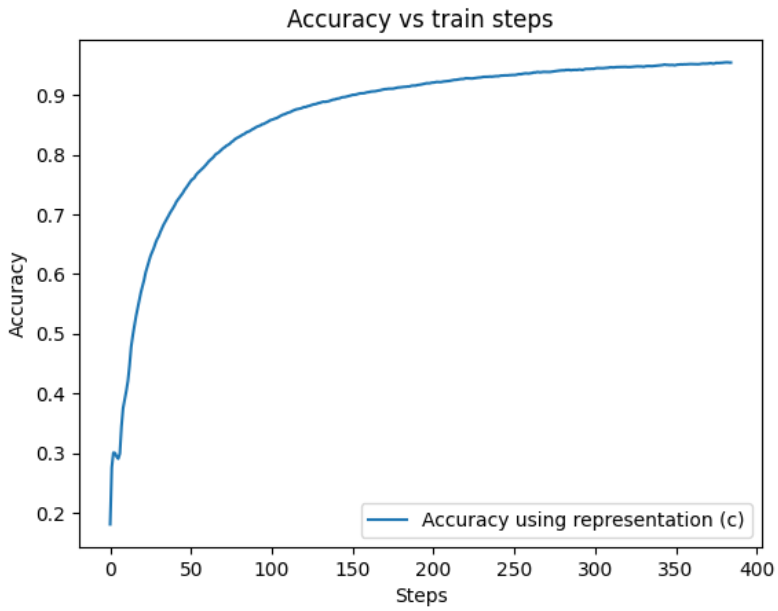
I order to reduce computation for this task, we removed sentences that were longer than 70 words, which made the number of hidden layer half reduced (from 141 to 70)

c) Representation c, Bag Of Words

Model Architecture:

```
CBOWTagger(
  (embedding): Embedding(112981, 32, padding_idx=100232)
  (bilstm): BiLSTM(
    (lstm_cell_forward): LSTM(
      (lstm_cell): LSTMCell(32, 32)
    )
    (lstm_cell_backward): LSTM(
      (lstm_cell): LSTMCell(32, 32)
    )
  )
  (classifier): Linear(in_features=64, out_features=46, bias=Tr
  (pre_embedding): Embedding(112981, 32, padding_idx=112980)
  (suf_embedding): Embedding(112981, 32, padding_idx=112980)
)
```

Results:



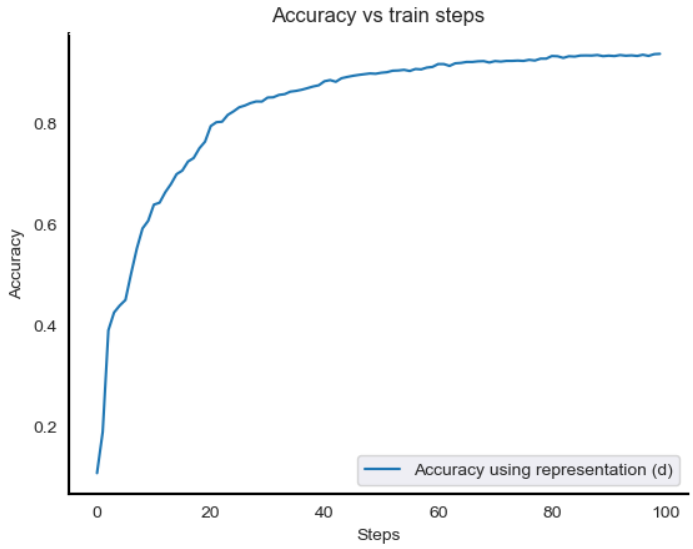
Accuracy on dev: 0.9541

d) Representation d, word + character level embedding

Model Architecture:

```
MaxLSTM(
  (char_embedding): CharLSTM(
    (embedding): Embedding(112981, 4, padding_idx=94)
    (lstm): LSTM(
      (lstm_cell): LSTMCell(4, 32)
    )
  )
  (word_embedding): Embedding(112981, 32, padding_idx=100232)
  (linear): Linear(in_features=64, out_features=32, bias=True)
  (bilstm): BiLSTM(
    (lstm_cell_forward): LSTM(
      (lstm_cell): LSTMCell(32, 32)
    )
    (lstm_cell_backward): LSTM(
      (lstm_cell): LSTMCell(32, 32)
    )
  )
  (classifier): Linear(in_features=64, out_features=46, bias=True)
)
```

Results:



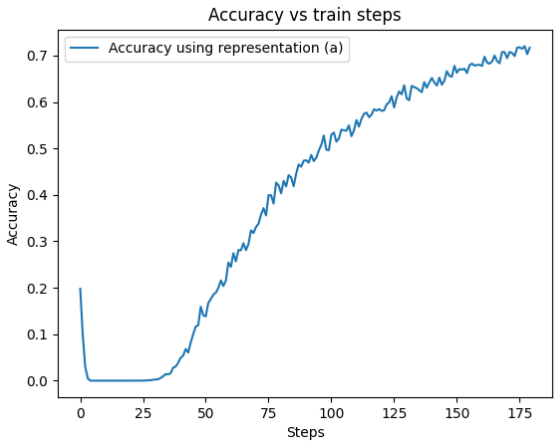
Accuracy on dev: 0.9021

3) NER Tagging

We used the same architecture for both pos and ner tagging models

a) Representation a, word embedding

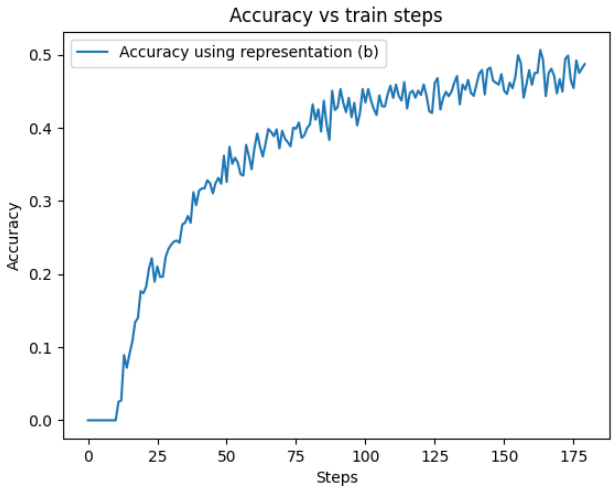
Results:



Accuracy on dev: 0.713

b) Representation b, character level embedding

Results:



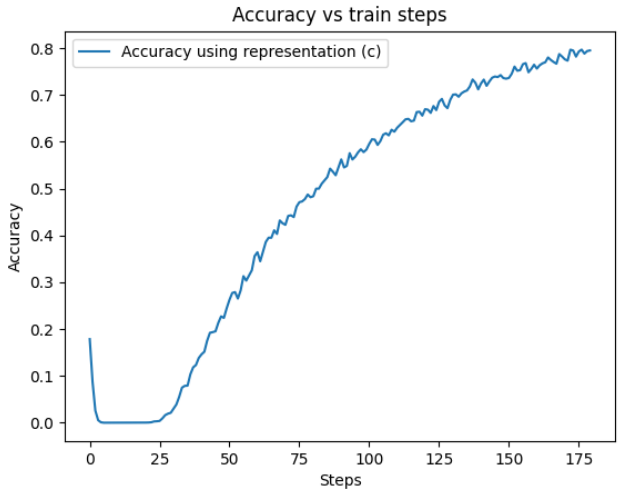
Accuracy on dev: 0.5174

Remark:

I order to reduce computation for this task, we removed sentences that were longer than 70 words, which made the number of hidden layer half reduced (from 141 to 70)

c) Representation c, Bag Of Words

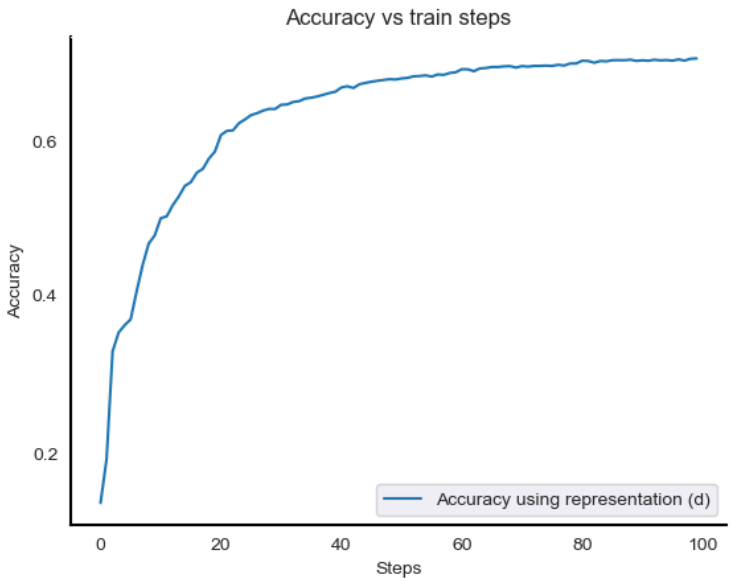
Results:



Accuracy on dev: 0.7952

d) Representation d, word + character level embedding

Results:



Accuracy on dev: 0.6833