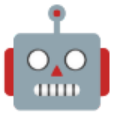

INLP Assignment 2 Report

Jainit Bafna

2021114003

There are some hyperparameters for which the model gave good results. I have used bidirectional LSTMS. They are

```
Enter the embedding dimension :100
Enter the hidden dimension : 100
Enter the learning rate : 0.003
Do you want to train the model with jugaad ? (y/n) : n
Enter the number of epochs :10
Device :  cpu
Loading data...
Printing only when accuracy increases
Epoch 1/10.. Train loss: 0.360.. Test accuracy: 0.980
Epoch 1/10.. Train loss: 0.035.. Test accuracy: 0.994
Epoch 2/10.. Train loss: 0.012.. Test accuracy: 0.994
Epoch 4/10.. Train loss: 0.006.. Test accuracy: 0.995
Epoch 9/10.. Train loss: 0.003.. Test accuracy: 0.995
Model trained
Testing model...
Test accuracy : 0.9924038052558899
Model tested
              precision    recall  f1-score   support

       <PAD>       1.00      1.00      1.00     25516
        PRON       0.98      0.99      0.99       784
         AUX       0.99      0.99      0.99       512
         DET       0.99      0.99      0.99      1024
        NOUN       0.96      0.99      0.97      2332
         ADP       0.98      0.99      0.99      2868
       PROPN       0.99      0.99      0.99      3134
        VERB       0.99      0.94      0.97      1258
         NUM       0.97      0.82      0.89       254
         ADJ       0.90      0.95      0.93       440
       CCONJ       1.00      0.99      1.00       218
         ADV       0.88      0.79      0.83       152
```

```
         PART       0.98      0.88      0.92       112
         INTJ       1.00      1.00      1.00        72

     accuracy                           0.99     38676
    macro avg       0.97      0.95      0.96     38676
 weighted avg       0.99      0.99      0.99     38676
```

When the test data was analyzed, it was found that only a few words were unknown. Also, when in training, there are no unknowns, so we needed to introduce some unknowns. So I applied a method that I named "Jugaad."

In this method, I took random sentences with a probability of 0.6 and replaced a few words with an unknown tag. This allowed the model to learn <UNK> tags (of words) and thereby increased robustness of the model. This also allowed me to increase my train dataset size.

The words were also decided on random basis.

Then after playing with some hyperparameters, i found a result which was the most accuracy.

I also introduced unknowingness in the validation data.

You can find my code on https://github.com/JainitBITW/POS_Tagger

```
Enter the embedding dimension :256
Enter the hidden dimension : 256
Enter the learning rate : 0.003
Do you want to train the model with jugaad ? (y/n) : y
Enter the number of epochs :10
Device :  cpu
Loading data...

Printing only when accuracy increases
Epoch 1/10.. Train loss: 0.351.. Test accuracy: 0.960
Epoch 1/10.. Train loss: 0.089.. Test accuracy: 0.973
Epoch 2/10.. Train loss: 0.062.. Test accuracy: 0.975
Epoch 2/10.. Train loss: 0.049.. Test accuracy: 0.976
Epoch 2/10.. Train loss: 0.045.. Test accuracy: 0.978
Epoch 4/10.. Train loss: 0.023.. Test accuracy: 0.979
Model trained
Testing model...
Test accuracy : 0.9934685230255127
Model tested
precision    recall  f1-score   support
   <PAD>       1.00      1.00      1.00     25516
```

```
     PRON       0.98      0.99      0.98       784
      AUX       0.97      0.99      0.98       512
      DET       0.99      0.98      0.98      1024
     NOUN       0.99      0.99      0.99      2332
      ADP       0.98      1.00      0.99      2868
    PROPN       0.99      1.00      0.99      3134
     VERB       0.98      0.96      0.97      1258
      NUM       0.97      0.91      0.93       254
      ADJ       0.93      0.94      0.93       440
    CCONJ       0.98      0.98      0.98       218
      ADV       0.83      0.78      0.80       152
     PART       0.98      0.88      0.92       112
     INTJ       1.00      0.89      0.94        72

accuracy                           0.99     38676
macro avg       0.97      0.95      0.96     38676
weighted avg       0.99      0.99      0.99     38676
```

High Support is mere mirage because of the <PAD> as this is the most learnt tag and overfitted.

Different Configuration:

```
Enter the embedding dimension :256
Enter the hidden dimension : 256
Enter the learning rate : 0.003
Do you want to train the model with jugaad ? (y/n) : y
Enter the number of epochs :10
Device :  cpu
Loading data...
Printing only when accuracy increases
Epoch 1/10.. Train loss: 0.234.. Test accuracy: 0.971
Epoch 1/10.. Train loss: 0.067.. Test accuracy: 0.975
Epoch 2/10.. Train loss: 0.052.. Test accuracy: 0.979
Epoch 2/10.. Train loss: 0.040.. Test accuracy: 0.979
Epoch 3/10.. Train loss: 0.023.. Test accuracy: 0.980
Epoch 5/10.. Train loss: 0.010.. Test accuracy: 0.980
Model trained
Testing model...
Test accuracy : 0.9932074546813965
Model tested
precision    recall  f1-score   support

       <PAD>       1.00      1.00      1.00     25516
        PRON       0.97      0.99      0.98       784
         AUX       0.97      0.98      0.98       512
         DET       0.99      0.98      0.99      1024
        NOUN       0.99      0.99      0.99      2332
```

```
          ADP       0.98      0.99      0.99      2868
        PROPN       0.99      1.00      0.99      3134
         VERB       0.97      0.96      0.97      1258
          NUM       0.96      0.86      0.90       254
          ADJ       0.91      0.95      0.93       440
        CCONJ       1.00      0.99      1.00       218
          ADV       0.86      0.78      0.81       152
         PART       0.98      0.82      0.89       112
         INTJ       1.00      0.94      0.97        72

    accuracy                           0.99     38676
   macro avg       0.97      0.95      0.96     38676
weighted avg       0.99      0.99      0.99     38676
```

here are some sentences of tags are predicted using the latest model \

```
Enter the sentence : My name is jainit and I am good boy

My       DET
name     NOUN
is       AUX
jainit   PROPN
and      CCONJ
I        PRON
am       AUX
good     ADJ
boy      ADP

Enter the sentence : I am going to eat

I        PRON
am       AUX
going    VERB
to       ADP
eat      VERB
```