COMS 4705 Natural Language Processing: **Homework 02 - SVW2112**
Due: 11:59pm (New York EDT), Monday 28 March 2022

Task 1: For this problem we are asked to implement the Viterbi algorithm with our bigram and trigram models, to test on the development sets ptb.22.txt and ptb.23.txt. The results for bigram, naive trigram, and interpolated trigram are included below.

Bigram Performance:

```
error rate by word:       0.05414163571553207  (2172 errors out of 40117)
error rate by sentence:   0.6558823529411765  (1115 errors out of 1700)
```

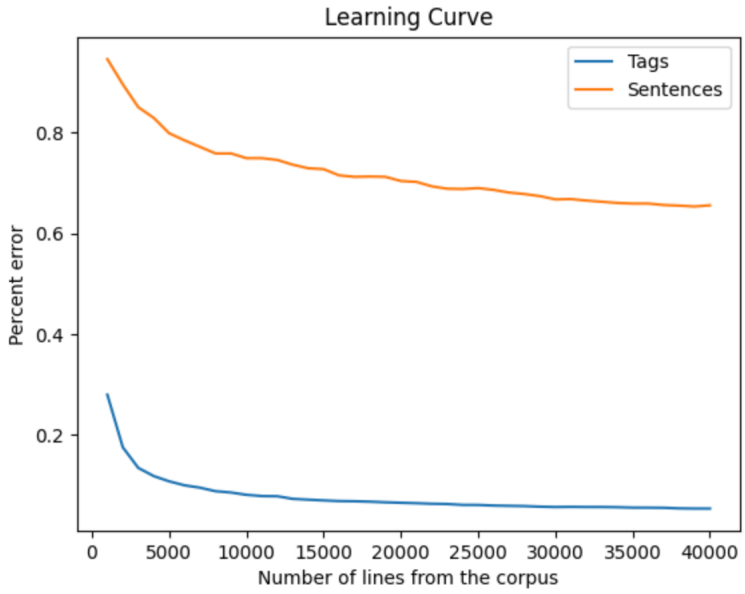
Naive Trigram Performance:

```
error rate by word:       0.08492658972505422  (3407 errors out of 40117)
error rate by sentence:   0.6305882352941177  (1072 errors out of 1700)
```

Interpolated Trigram Performance:

```
error rate by word:       0.08515093351945559  (3416 errors out of 40117)
error rate by sentence:   0.6241176470588236  (1061 errors out of 1700)
```

Task 2: For this problem we are asked to generate a learning curve for the bigram HMM, using the development data found in ptb.22.txt. The learning curve plots a performance measure evaluated on a fixed test set along the y-axis against the size of the training data along the x-axis. There are 39832 lines in our development set, the learning curve samples at 1000, 2000, 5000, 10000, 20000, and 40000 lines. The percentage errors against the training size are plotted in the learning curve below. As expected, the results demonstrate that percentage errors decrease as the size of the corpus grows, following a pattern of diminishing returns as more lines are added.



Task 3.1: For this problem we are asked to train our bigram and trigram models against the Bulgarian and Japanese training datasets, to compare the performance differences.

ENGLISH

Bigram Performance:

```
error rate by word:       0.05414163571553207  (2172 errors out of 40117)
error rate by sentence:   0.6558823529411765  (1115 errors out of 1700)
```

Naive Trigram Performance:

```
error rate by word:       0.08492658972505422  (3407 errors out of 40117)
error rate by sentence:   0.6305882352941177  (1072 errors out of 1700)
```

Interpolated Trigram Performance:

```
error rate by word:       0.08515093351945559  (3416 errors out of 40117)
error rate by sentence:   0.6241176470588236  (1061 errors out of 1700)
```

BULGARIAN

Bigram Performance:

```
error rate by word:       0.11830131445904954  (702 errors out of 5934)
error rate by sentence:   0.7688442211055276  (306 errors out of 398)
```

Naive Trigram Performance:

```
error rate by word:       0.21486349848331648  (1275 errors out of 5934)
error rate by sentence:   0.7864321608040201  (313 errors out of 398)
```

Inerpolated Trigram Performance:

```
error rate by word:       0.11830131445904954  (702 errors out of 5934)
error rate by sentence:   0.7688442211055276  (306 errors out of 398)
```

JAPANESE

Bigram Performance:

```
error rate by word:       0.06286114515846612  (359 errors out of 5711)
error rate by sentence:   0.1368124118476728  (97 errors out of 709)
```

Naive Trigram Performance:

```
error rate by word:       0.3141306251094379  (1794 errors out of 5711)
error rate by sentence:   0.22002820874471085  (156 errors out of 709)
```

Interpolated Trigram Performance:

```
error rate by word:       0.06286114515846612  (359 errors out of 5711)
error rate by sentence:   0.1368124118476728  (97 errors out of 709)
```

Task 3.2: Different languages are context dependent, since we are only running models that go up to trigram we will not catch other dependencies specific to certain languages that improve accuracy.

Task 3.3: Unsurprisingly, interpolated trigram performs better against the training datasets of each of the three languages, while bigram typically outperforms naive trigram for each language by word and sentence. The lone case where bigram did not outperform naive trigram was error rate by sentence for the English dataset. Given these results, adjacencies in the Bulgarian and Japanese languages play important roles in POS tagging.

Bonus Task: For this problem we are asked to implement a neural network model to train and test against the development sets of the three languages, once again producing tagging predictions similar to previous tasks. For this exercise, I chose to implement a version of the Transformer model because it is generally expected to produce better performance results than LSTM with CRF models. When running the model against the datasets, I found the error by word and sentence rates to be low at around 2-3% if you run on an expanded range of epochs. The model begins to overfit around 5 epochs before a sharp and steady drop in accuracy around 10 epochs.

ENGLISH

```
1245/1245 [==============================] - 672s 537ms/step - loss: 7924045.5000 - accuracy: 0.0509
```

BULGARIAN

```
401/401 [==============================] - 221s 541ms/step - loss: 398996.7188 - accuracy: 0.0399
```

JAPANESE

```
533/533 [==============================] - 296s 547ms/step - loss: 1882762.5000 - accuracy: 0.3060
```