ReadMe:
Collaborators: Michael Pirrall, Sufian Mushtaq, Abdul Moid Munawar, Tianyi Ma
NetIds: mpirrall, smushtaq, amunawar, tma8

Changes in nlputil.py:

- We changed the nlputil functions so that we only train on sequences with less than 100 words. Also, we include the sample_size parameter so that we pick up sample_size/2 number of samples from pos dataset and the other half from neg dataset.

How to run hmm.py: (examples for each task you might wanna do are at the end of file)

- When training, include --train_path for the location of the training data. This should just be a path to the training data folder.
- Also when training, include --max_iters (it's an integer) for the maximum number of EM iterations this should run for,
- --hidden_states (an integer) for number of hidden states to use,
- --train_sample_size for the number of samples to be used for training.
- --mode should be 1 by default so you do not need to specify. --mode is 2 when you want to test, 1 when you want to train, and 0 when you want to predict.
- When you are testing and predicting trained models, use --model_path to specify the path to your model files, and --test_sample_size for the number of samples to be used for testing. Notice that you still need to specify --train_sample_size \ = the number of samples you used for training that specific model and --train_path when testing since you need to build the same vocab you used for training that specific model.

Testing Prediction Accuracy of trained models:

- For prediction, --dev_path is used to specify the location of the testing data. When doing predictions, you should set --mode to 0 to run the prediction code. The simple accuracy is very slow so if you want to test only the viterbi individually then comment out lines that have the comment “Comment out lines” next to them which you find using ctrl+f in hmm.py

- --model_path is used when you are loading a saved model for prediction. The path should be to the folder with the models in it, not the model itself. **If you use our code structure as is then our code will test prediction on the model trained on 15 hidden states.**

Command line example:
To train a model:

```
python hmm.py --mode 1 --train_path path_to_train_files --max_iters 1000 --hidden_states 10 --train_sample_size 100
```

(Note: file paths for training and testing are dependent on where you put your data relative to the starterCode folder)

Command line example:
To get prediction accuracy:

```
python hmm.py --train_path path_to_train_files --dev_path path_to_test_files --model_path path_to_model_files --train_sample_size 100 --test_sample_size 50 --mode 0
```

(**Note**: file paths for training and testing are dependent on where you put your data relative to the starterCode folder)
(**Note**: code is currently set to test [1,3,5] words into the future for test data. We changed the path manually during experimentation from test data to train data in our code to get prediction accuracy on train data. To do this yourself change test_paths to train_paths in the line with commented label “Change this line to switch between prediction accuracy on test and train”)

Command line example:
To test the log likelihood of a model on test data:

```
python3 hmm.py --train_path path_to_train_files --dev_path path_to_test_files --model_path path_to_model_files --train_sample_size 100 --test_sample_size 50 --mode 2
```

(Note: testing log likelihood using our code as is then it will test all the models in the model folder and plot graphs for them all)

Suggested parameters:
When training, a sample size of around 100 tends to be the highest you should go if you want reasonable training times. It’ll still likely take a few hours to converge with 100 samples though, so if you are on a time crunch, reduce the samples further. Varying hidden units is standard across tests. We tried 5, 10, and 15, with 15 being the best during training.

When an iteration of training does not change the values by more than our defined epsilon of 0.00001, the training will end automatically as the model has converged, so if you want to train until convergence, set max_iters very high. Otherwise set max_iters to however many iterations you think it will take to reach near enough to convergence.

During training, every 5 iterations a model file and a plot of the log likelihood over time is saved. This allows you to be able to cancel training before reaching max_iter, and you will still have the model you trained so far as well as a plot of the log likelihood over time.