Homework 2: POS Tagging

This assignment is about POS tagging. You should be aware that POS tagging is widely used, and that it's widely seen in the NLP community as a solved problem (or at least well enough solved that most people don't put much effort into improving POS tagging for its own sake). It's a good problem for learning about NLP, though, and that's what we are aiming for in this exercise. For this assignment, you may only use the datasets provided. We realize that you could probably track down the test data we will evaluate your models on, but that would be cheating – so don't.

Download and unpack homework files. There is a **`README`** file, one Perl script, some Python scripts, and data. Using what we've given you, you can build a baseline bigram HMM POS tagger on **the training set (sections 2-21, `ptb.2-21.*`)** from the Penn Treebank and evaluate it on **the development set (section 22, `ptb.22.*`)**. (Actually, we already did that, but you should take two minutes to follow the `README` and do it yourself to make sure you understand the tools.) Note that we have not given you the standard test set output.

Task 2 (60 pts):

In this task, you will implement a trigram HMM. We have already provided you with the bigram HMM (*train_hmm.py*) and a Viterbi algorithm in Perl (*viterbi.pl*). You can evaluate its performance with:

```
# train with provided training set
python train_hmm.py data/ptb.2-21.tgs data/ptb.2-21.txt > my.hmm

# predict results on development set
perl viterbi.pl my.hmm < data/ptb.22.txt > my.out

# evaluate accuracy
python tag_acc.py data/ptb.22.tgs my.out
```

Part 1 (10 pts):

The first part is to implement your Viterbi algorithm for bi-gram in Python. You can use the Perl script *viterbi.pl* as reference. Your tagging output should be the same as the Perl script *viterbi.pl*. Write your code in the file ***viterbi.py.***

Part 2 (50 pts):

Implement a trigram HMM and the corresponding Viterbi algorithm. Write your code in the file ***trigram_hmm.py.*** This script should output the final POS tagging predictions in the same format as *viterbi.pl*, and can be evaluated with *tag_acc.py*.

Report your model's performance on the development data (***ptb.22.txt***). Also, run your tagger on ***ptb.23.txt*** (the test data) and turn in the output so we can evaluate it using *tag_acc.py*. Name the output file as **ptb.23.out.**

Task 2 (20 pts):

A learning curve is a useful tool that lets you visualize how a model's performance depends on the amount of training data. You can vary the amount of training data by creating a smaller sub-corpora from the original full corpus.

The learning curve plots a performance measure evaluated on a fixed test set (y-axis) against the training dataset size (x-axis). You should choose the range from **1000 to 40000** (all data) lines. Generate a learning curve for the bigram HMM as we've provided it, using the development data (***ptb.22.txt***) to evaluate. What are your thoughts about getting more POS-tagged data and how that would affect your system?

Task 3 (20 pts):

1) Train the bigram model and your trigram model on the Japanese (`jv.*`) and Bulgarian (`btb.*`) training datasets, and compare them on the test sets. Report the performance and compare the performance differences between English, Japanese, and Bulgarian. (10 pts)

2) You sould look at the data (obviously!) and give some analysis – what factors lead to the differences among performance on English, Japanese, and Bulgarian? (Your answer should be concise.) (5 pts)

3) Also, what about the trigram HMM makes them relatively better or worse on the data in these two languages? Explain your result. (Your answer should be concise.) (5 pts)

Bonus Task (20 pts):

Implement a neural network-based model (RNN or Transformer) for this task. You can use any public library. Train and test your model on the development sets of the three languages. Include the tagging predictions in the format mentioned before.

Briefly discuss what model you choose and compare the performance of your model with HMM and explain the result. Again, you can only train on the training data (no external resources are allowed).

Submission

Please submit a zip archive containing the following items. Note that you are not submitting code for this assignment; we rely on you to describe your approach well.

- Your modified code (you can submit the entire folder, compressed)
- the output of your new model on `ptb.23.txt` (we will evaluate it against gold-standard POS tags),
- a text or pdf file containing your answers to all questions, including the plot for the learning curve. **Whether or not you collaborated with other individuals or employed outside resources, you must include a section in this file documenting your consultation (or non-consultation) of other persons and resources.**