CS114 (Spring 2017) Homework 4
Part-of-speech tagging with Hidden Markov Model

Due March 25, 2017

You should use Ipython Notebook to do this programming assignment.

Supervised HMM

In this assignment, you will use the brown corpus (57340 sentences of POS-tagged sentences) to train a simple POS tagger. Please split the data into train/dev/test (80/20/20) by simply using the first 80% of sentences as training, next 10% as dev, and last 10% as test. Here is how you load and use the Brown corpus (If you haven't downloaded Brown corpus, it will throw an error asking you to download it by nltk.download(), just follow the installation step and look for Brown corpus in the list of packages for installation)

```
>>> from nltk.corpus import brown

>>> brown.tagged_sents()[0]

[(u'The', u'AT'), (u'Fulton', u'NP-TL'), (u'County', u'NN-TL'), (u'Grand', u'JJ-TL'),
(u'Jury', u'NN-TL'), (u'said', u'VBD'), (u'Friday', u'NR'), (u'an', u'AT'), (u'investigation', u'NN'),
...
```

In this assignment, you might want to reuse your bigram code (PA2) to train the HMM model. In specific, you will need to calculate the following probabilities:

- Transition probabilities for a pair of POSs (p, s):

$$P(s|p) = \frac{Count(POS_{prevword} = p, POS_{currentword} = s)}{Count(POS_{prevword} = p)} \tag{1}$$

- Emission probabilities for each pair of (Part-of-speech = s, Word type = w):

$$P(w|s) = \frac{Count(POS_{word} = s, word = w)}{Count(POS_{word} = s)} \tag{2}$$

Feel free to use any smoothing method that you have learnt. If you have trigram or n-gram language models working, feel free to bring them in to test, but I would not expect any significant improvement, because the training data is small. After you have fitted your POS tagger on

training data, use dev set for any hyperparameter tuning (for example if you use add α (Lidstone smoothing)). Report your accuracy on the test set.

Unsupervised HMM

This assignment is not mandatory but you will be awarded extra credit if you finish it.

You are given a list of sentences without any Part-of-speech tag. Here is the list of sentences:

```
He saw a cat.
A cat saw him.
He saw a dog.
A dog saw him.
He chased the cat.
The cat chased him.
He chased the dog.
The dog chased him.
```

Assuming that you have 4 hidden POSs, run your Baum-Welch implementation over this training dataset. Report the resulted model, and resulted tags for each sentence (You can number your tags 1,2,3,4). Do your model converge to a state that you expected?

Submission and grading

Submit your notebook file only (if you write other helper python files, compress your file to .zip or .gz before submission).

You will be awarded 90% of the credit for bigram implementation, 10% will be awarded for how you handle smoothing and effort for improving performance of supervised HMM. Up to 20% will be awarded as extra credit for unsupervised HMM.