CSE 476/598 Introduction to NLP
Assignment 2

Professor Heather Pon-Barry

Due Monday September 30th *5:00 PM*

General Instructions:

- The data for this assignment is available on Piazza, in the file `hw2_data.zip`.
- You may talk with others about the problems, however your code and your write-up must be your own work.
- You will use the Natural Language Toolkit (NLTK), a Python toolkit for NLP, in this assignment. Installation instructions are on Piazza.

Getting Started

We provide you with two python files: `hmm.py` and `hmmtrain.py`. In `hmm.py` we define a class called `hmm`. As soon as you instantiate this class, the various parameters of the HMM (transition probabilities, emission probabilities etc.) are automatically computed using the first 3000 sentences of the `nltk.corpus.treebank` corpus as the training data (using functions defined in `hmmtrain.py`). The following five parameters are available to each instance of the `hmm` class:

- **transitions:** The probability of transitioning from one state to another. To get the probability of going to state `s2` from state `s1`, use `self.transitions[s1].prob(s2)`.
- **emissions:** The probability of emitting a particular output symbol from a particular state. To get the probability of emitting output symbol `sym` in state `s`, use `self.emissions[s].prob(sym)`.
- **priors:** The probability of starting in a particular state. To get the probability that the HMM starts in state `s`, use `self.priors.prob(s)`.[1]
- **states:** The states (tags) in the hidden state layer of the trained HMM.[2]
- **symbols:** The output symbols (words) in the trained HMM.

[1] This initial probability distribution is equivalent to using a special state q_0, i.e., $prior(q_1) = P(q_1|q_0)$

[2] `-LRB-` and `-RRB-` are the tags used for the left parenthesis tag '`(`' and the right parenthesis tag, '`)`'.

The probability values that are calculated by `hmmtrain` are going to be extremely small in scale. Multiplying two very small numbers can lead to loss of precision. We strongly recommend that you use the log of the probabilities (logprobs) instead of raw probabilities. For example, use `self.transitions[s1].logprob(s2)`. See `http://nltk.org/_modules/nltk/probability.html` for details of the nltk probability distribution interfaces.

This assignment requires a two-dimensional chart. Rather than using Python lists to implement such a chart, we recommend using the `numpy` package's `array` object and the `float32` datatype.

Problem 1

[15 Points]

(a) Add a method called `exhaustive()` to the `hmm` class that takes a sentence O and exhaustively computes the most likely tag sequence. This method should compute the probability,

$$P(o_1, o_2, \ldots, o_T, q_1, q_2, \ldots, q_T|\lambda) \approx \prod_{i=1}^{T} P(q_i|q_{i-1})P(o_i|q_i)$$

for each possible tag sequence $Q = q_1, q_2, \ldots, q_T$. It should return the most likely tag sequence and its associated probability.

(b) According to your `exhaustive()` method, what is the most likely tag sequence for the sentence below?

> **You look around at professional ballplayers and nobody blinks an eye .**

Problem 2

[34 Points]

(a) Add a `decode()` method to the `hmm` class that performs Viterbi decoding to find the most likely tag sequence for a given word sequence.

(b) Add a `tagViterbi()` method that takes a file with one (tokenized) sentence per line as input and tags the words in the sentence using Viterbi decoding. It should output tagged sentences of the form: *This/DT is/VBZ a/DT sentence/NN.*

(c) According to your `tagViterbi()` method, what is the most likely tag sequence for the sentence below? Verify that it is the same as your answer from (1b).

> **You look around at professional ballplayers and nobody blinks an eye .**

(d) Tag each of the five sentences in the provided file `sentences.txt`. Include your tagged sentences in a text file (or directly in your write-up). Do you observe any errors in the tags assigned? If so, mention them. Is every word in every sentence assigned some tag? If not, speculate about why this is the case.

Problem 3

[**1 Point**] How long did you spend on this assignment?

Deliverables

- You must save your write-up as a PDF. If you have any hand drawn figures, you will need to scan them. At the top of the first page, include your full name, email address, and list any classmates you worked with. Save this file as `<lastname>_hw2_writeup.pdf`.
- Put your python code in a directory called `code`. If you have any additional scripts, include a `README` explaining the contents of each file. We will not be grading your code, but we will refer to it. Be sure to comment your code so we can read it.
- Create a zip file containing your write-up and your `code` directory. Save this file as `<lastname>_hw2.zip`.
- Upload this zip archive to the course site on Blackboard. Go to *Content*, then *Homework 2*.