Ling 5801: Problem Set 5 (Project)

Due via Carmen dropbox at 11:59 PM 11/27.

PROGRAMMING: Download the `simplewiki.gcg15.linetrees` tree file from the course web site. This is a list of trees for the first 1000 sentences in the September 3, 2014 dump of Simple English Wikipedia, annotated according to the grammar described in the lecture notes on context-free grammars. You can use this file to test your code.

The assignment is to write python scripts to do all of the following using the syntax described in the lecture notes:

1. [10 pts.] Write a python program called `q1.py` to read a tree file from standard input and output a probability model $\mathsf{P}(X_t \mid Y_t)$ of words X_t given pre-terminal symbols Y_t (categories immediately dominating words). This model should be printed to standard output in the CondModel format as defined in the lecture notes on probability models:

   ```
   XgivY V-aN-bN : has = 0.12794613
   XgivY V-aN-bN : have = 0.05387205
   XgivY V-aN-bN : became = 0.04040404
   ⋮
   ```

2. [10 pts.] Write a python program called `q2.py` to read a tree file from standard input and output a probability model $\mathsf{P}(Y_t \mid Y_{t-1})$ of pre-terminal symbols Y_t given previous pre-terminal symbols Y_{t-1}. This model should be printed to standard output in the CondModel format as defined in the lecture notes on probability models:

   ```
   YgivY D : N-aD = 0.59183673
   YgivY D : A-aN = 0.25686137
   ⋮
   ```

3. [10 pts.] Write a python program called `q3.py` to read a tree file from standard input and output a probability model $\mathsf{P}(Y_0)$ of pre-terminal symbols Y_0 at the beginnings of sentences. This model should be printed to standard output in the Model format as defined in the lecture notes on probability models:

   ```
   Y : N = 0.34271357
   Y : D = 0.19396985
   ⋮
   ```

4. [10 pts.] Write a python program called `q4.py` to implement a hidden markov model recognizer, based on the algorithm in the lecture notes on sequence modeling. Your recognizer should read in the models you defined in the previous problems in this problem set from standard input. It should also read input sentences on lines beginning with the letter 'I' from standard input:

   ```
   I the country had a name
   ```

   Following Treebank convention, you should assume punctuation marks will be spaced apart as separate tokens. Evaluate your recognizer as a filter (report the joint probability distribution over the last hidden variable, and all the evidence: $\mathsf{P}(Y_T, x_1, ..., x_T)$) on the sentence 'the country had a name' For example, this should print:

   ```
   Y_fwd : N-aD = 5.86650762039e-11
   Y_fwd : N-aD-bO = 3.69969155486e-11
   ```

⋮

5. [extra credit – 5 pts.] Write a python program called `q5.py` that modifies your recognizer to output the most likely sequence of hidden states according to the model defined in problems 1 and 2. Evaluate your recognizer on the same sentence 'the country had a name'. For example, this should print something like:

```
preterminal 1 = D
preterminal 2 = N-aD
preterminal 3 = V-aN-bN
preterminal 4 = D
preterminal 5 = N-aD
⋮
```