Homework 4
Due Tuesday, 02/26/08

Please turn in a print-out of your homework solutions at the beginning of class. If a program is required for a problem then you should provide sample input and output. (If the input is data from an on-line source then you may just indicate where the file can be found.) Programming solutions to problems should be placed in separate files whose names indicate the problem number (e.g. prob1.py). These separate files should be put into a folder whose name includes the assignment number and your name (e.g., Hmwk1-yourname). This folder should be dragged into the class dropoff folder at

`/common/cs/cs181/dropbox`

While I expect that I will be able to do most of the grading based on the papers that you hand in at the beginning of class, I would like to have access to the programs so that I can test them if necessary. I should be able to load your file into python and have it compile and run without error. All programming solutions should be fully documented and use good variable names.

Note that the files on our system can be accessed using ssh via the Mac server xserv.cs.pomona.edu or the linux server linus.cs.pomona.edu. Files and/or folders can be deposited into the dropbox remotely by using a program utilizing the sftp protocol.

Warning: There are several different versions of Bird et al floating around. From now on all assignments will be relative to the pdf edition of the entire book dated January 24, 2008, and available via the course web page.

1. There is only one problem this week. Implement an HMM part-of-speech tagger using the Viterbi algorithm. Train your tagger using the `treebank.train` file available from the homework web page.

   Use the file in the `treebank.test` file to test your tagger. Note that this file is fully tagged, so you will need to strip off the tags before using it as input to your tagger.

   (a) Compare the results of your tagger with the original tagging of the test file. What is the accuracy of your tagger on the test data?

   (b) How does this compare to the accuracy obtained by always choosing the most frequent tag for each word?

   (c) Identify five errors in the automatically tagged data, and analyse them.

   (d) Run your tagger on the larger training file `treebank.train.large`. Run it on the same test data file as before. What is the accuracy now?