Assignment 2

Introduction to Natural Language Processing
Fall 2016
Total points: 70
Issued: 10/06/2016 Due: 10/14/2016

All the code has to be your own (exceptions to this rule are specifically noted below). The code must run on the CAEN environment without additional installation or additional files (except for the data files specified in the assignment).

You can discuss the assignment with others, but the code is to be written individually. You are to abide by the University of Michigan/Engineering honor code; violations will be reported to the Honor Council.

1. [60 points] Viterbi Part-of-speech Tagger

Write a Python program *Viterbi.py* that implements the Viterbi algorithm for part-of-speech tagging, as discussed in class. Specifically, your program will have to assign words with their Penn Treebank tag. You will train and test your program on subsets of the Treebank dataset, consisting of documents drawn from various sources, which have been manually annotated with part-of-speech tags. The datasets (*POS.train and POS.test*) are available from the Files section of the Canvas class webpage; each line in these files corresponds to a sentence.

Programming guidelines:
Your program should perform the following steps:

- Starting with the training file, collect and store all the raw counts required by the Viterbi algorithm. Please make sure to also cover the "beginning of a sentence" in your raw counts.
- Implement the Viterbi algorithm and apply it on the test data. Make sure to strip off the part-of-speech tags in the test data before you make your tag predictions.
- Compare the tags predicted by your implementation of the Viterbi algorithm against the provided (gold-standard) tags and calculate the accuracy of your system.

The *Viterbi.py* program should be run using a command like this:

```
% python Viterbi.py POS.train POS.test
```

The program should produce at the standard output the accuracy of the system, as a percentage. It should also generate a file called POS.test.out, which includes the words in the test file along with the part-of-speech tags predicted by system.

Write-up guidelines:
Create a text file called Viterbi.answers, and include the following information:

- The accuracy of your system on the test data
- The accuracy of a simple baseline that assigns to each word its most frequent tag (according to the training data)
- Identify five errors in the automatically tagged data, and analyse them (i.e., for each error, write one brief sentence describing the possible reason for the error and how it could be fixed)

2. [10 points] Training on Large Data

Train your Viterbi tagger on the large training file (POS.train.large), which is also available from the Files section of the Canvas class webpage. Test the tagger on the same test file as before (POS.test).

Write-up guidelines:
Create a text file called Viterbi.large.answers, and include the following information:

- The accuracy of your system on the test data
- The accuracy of a simple baseline that assigns to each word its most frequent tag (according to the large training data)

General submission instructions:

- Include all the files for this assignment in a folder called *[your-uniqname].Assignment2/*
  Do not include the data files.
  For instance, lahiri.Assignment2/ will contain Viterbi.py, Viterbi.answers, Viterbi.large.answers
- Archive the folder using zip and submit on Canvas by the due date.
- Include your name and uniqname in each program and in each *.txt file
- Make sure all your programs run correctly on the CAEN machines.