CMPSCI 585: Natural Language Processing
Programming Assignment 3: HMM

Out: October 14, 2004
Due: October 26, 2004

Part-of-speech Tagging with HMMs

Goal

The goal of this assignment is to implement a Hidden Markov Model (HMM) to label words with their part-of-speech. Here, the observation sequence is a sentence, and the hidden states represent the part-of-speech for each word. A description of HMMs can be found in the lecture slides as well as in the Manning and Schutze text. For the main assignment, you can assume that you have completely labeled training data, which means you do not have to use Baum-Welch to estimate the parameters of the model.

Data

The data was collected from `http://cnts.uia.ac.be/conll2000/chunking/`. For this assignment, use the slightly modified version found at `http://canberra.cs.umass.edu/~culotta/cs585/ass3-data.tgz` (280K). The data has already been split into train.txt (~4K sentences) and test.txt (~1K sentences). Each sentence in the file is separated by a blank line, and each word is placed on a separate line with its true part-of-speech.

Implementation Notes

To deal with *out-of-vocabulary* words (words in the test set but not in the training set), you should first read in *all* sentences (testing and training) to create a dictionary of all words V. If S is the set of states, then you must estimate and store the $|S|$x$|S|$ transition probability matrix A and the $|S|$ x $|V|$ emission probability matrix B. Remember to include initial and final "dummy" states. Also, use Laplacian smoothing (a.k.a. +1 smoothing) as you did for previous assignments.

When implementing the Viterbi algorithm, use assertions liberally to avoid any "off-by-one" errors.

Tasks

Train your HMM using the sentences in train.txt, and then test the HMM on the sentences in test.txt. Report the precision and recall for each of the

8 parts of speech, as well as the overall precision.

Additional Experiments

Perform **1 of the following 3** experiments:

- **Out-of-Vocabulary Words**: An alternate method of dealing with unknown words (discussed in class) replaces each token that occurs only once in the training data with a special token UNK, representing an unknown word. Estimate the paramters for the HMM in the same way as before. When an unknown word is seen in the testing set, treat it as an UNK token. Compare these results with your previous experiment.
- **Baum-Welch:** Implement Baum-Welch (a specific instance of EM). Create an additional noun states and use Baum-Welch to estimate the parameters for these two noun states. Estimate the parameters for the remaining states as before. This provides more granularity for labeling noun tokens.
- **2nd Order Model:** It's possible that knowing information about tokens two time steps ago can help predict the part-of-speech of the current token. Implement one of the following changes:
  - Condition state s_i on s_{i-1} and observation o_{i-1}
  - Condition state s_i on s_{i-1} and s_{i-2}

  Note that each of these models requires increasing the dimension of the probability tables, since there is one more variable to consider.

What to turn in

Code: Print out all source code written for the project.
Report: Write a 2 page report that includes the following:

1. Implementation experience - What questions and issues arose during your implementation. If there were difficulties, how did you resolve them?

2. Experimental results - Report the per part-of-speech precision and recall, as well as the overall accuracy for all parts-of-speech. Report results for the original HMM and any variations you tried.

3. Discussion - Discuss your experiments and explain your results. What additional experiments do you think would improve performance?