3.11 Programming problems

Problem 3.1: Part-of-speech tagging using HMMs

The data directory contains files `wsj2-21.txt` and `wsj22.txt`. Each file contains one sentence per line, where each line is a sequence of pairs consisting of a word and its part of speech. Take a look at the files so you know what the precise format is. `wsj22.txt` has been pre-filtered so that words that don't appear in `wsj2-21.txt` have been replaced with the unknown word symbol ∗U∗.

The assignment directory contains the script templates `tag` and `score`. Follow the directions in the templates carefully.

1. Find maximum-likelihood estimates for the parameters $\hat{\boldsymbol{\sigma}}$ and $\hat{\boldsymbol{\tau}}$ from the file `wsj2-21.txt`. Note that you'll have to smooth the parameter estimates for $\tau_{y,*\mathrm{U}*}$; at this stage you can just give these a pseudo-count of 1.

2. Implement the Viterbi decoding algorithm and find the most likely tag sequence $\hat{\boldsymbol{y}}$ for each sentence $\boldsymbol{x}$ in `wsj22.txt`. Compute the percentage of words for which their Viterbi tag is in fact the correct tag.

3. Now we'll try to find a better estimate for $\tau_{y,\cdot\texttt{UNK}\cdot}$. We note that words that appear once in our training data `wsj2-21.txt` are only one occurrence away from not occurring at all. So suppose we go through our training data and change all words that only appear once to ∗U∗. We can now compute $\tau_{y,*\mathrm{U}*}$ just like everything else and there is no need to smooth. Also note that, say, $\tau_{NN,*\mathrm{U}*}$ can (and will) differ from $\tau_{DT,*\mathrm{U}*}$. In general this should increase accuracy because it correctly models the fact that an ∗U∗ is more likely to be a noun than a determiner. Implement this and report your new accuracy. (One problem with this is that you lose the part-of-speech tags for all these words. Fortunately they are rare words, so the odds are that few of them will appear in your test data. However, if this bothers you, you can count each word with one token twice, once as itself, once as ∗U∗.)

3.12 Further reading

HMMs are used not just in computational linguistics, but in signal processing, computational biology, analysis of music, hand-writing recognition, land-mine detection, you name it.