How To Compile:

Compile and run using Python 2. Include the file paths to the probabilities and sentences files as command line arguments in that order.

Results and Analysis:

The viterbi algorithm correctly identified the parts of speech for each of the sentences in the test file, as well as additional sentences I tested the algorithm on, including “bears bear fish”, “bear bears fish”, “fish fish fish”, and “john grows flowers”. A large part of this is probably due to the high probability assigned to the first word of a sentence being a noun. When there is some ambiguity, such as “fish fish”, the algorithm finds noun, verb to be the most likely part of speech order, although verb, noun is valid as well.

The forward algorithm did not produce significantly different results from the viterbi algorithm, with higher overall probabilities, but mostly the same ordering of likelihoods. My implementation should be faster than my implementation of the viterbi algorithm due to not needing to find the max value, but I’d need to do more testing to see if that speed comes as the cost of some accurracy.

Known Bugs, Problems, and Limitations:

There are no bugs, problems or limitations that I’m aware of. I had trouble verifying the accuracy of the values the forward algorithm printed, but the printed values corresponded to the correct tag sequence.