Accuracy: ~96% on validation set with an 80:20 split

Execution time: ~10 seconds

Steps followed to implement the Viterbi decoder:

1. Extract the required counts from the training data to generate the required probability estimates for the model

   The training data was read from the file and the required data was extracted into lists. The counts of unique words, tags and (word, tag) tuples were calculated. The probabilities were then calculated using the counts and stored in dictionary.

2. Sentence markers were added

   A sentence marker <s> was added to the beginning of each sentence in order to include bigram counts for the first word in a sentence.

3. Deal with unknown words in some sensible way

   For the Viterbi decoder, words that occurred only once in the training set were replaced by the word "UNK". Thus, the words in the test data that never occurred in the training data would use the probability estimates of "UNK".

4. Do some form of smoothing for the bigram tag model

   Smoothing of emission and transition probabilities were done using the Laplace method. The accuracy dropped to 90% when a smoothed emission probability was used. Hence smoothing was performed only for transition probabilities.

5. Implement a Viterbi decoder

   Implementation of the Viterbi method used dictionaries as a data structure for fast execution. The dynamic programming algorithm yielded fast execution time.

6. Evaluate your system's performance on unseen data

   The training data was subject to a K-fold cross validation with k=10. This was done with and without shuffling the input data. On an average, the accuracy was a constant approximation of 96%.