

Hey there

To run my program, pass in the training file as the first argument, and the working file as the second.

```
python viterbi.py wordFiles/WSJ_02-21.pos wordFiles/WSJ_24.words
```

After running this program, there will be a file named output.pos in the directory you ran the program in. You can use this file to run the accuracy script.

After I ran the accuracy script with WSJ_24, received 95.54 percent accuracy.

```
172-16-24-247:wordFiles j$ python score.py WSJ_24.pos ../output.pos
31391 out of 32853 tags correct
accuracy: 95.549874
```

I implemented a more naive viterbi algorithm and was very happy with the results, the program ran in under 30 seconds opposed to the complete viterbi algorithm which would of taken about 40 minutes to an hour. I think the accuracy gained versus the time it takes for the program to run makes it worth using a naive algorithm.

Basically the way my algorithm worked is by picking the max value in the column of the viterbi table, instead of traversing all combinations and paths from beginning to end to get the pos value.

For out of vocabulary words, I picked the pos that was most likely to come after the previous pos. The word may still be out of vocabulary if it is the first word in the sentence and nothing comes previously before it. For those words, I assign them as a 'NNP'. Before I was assigning those words as 'NN' but changed it to 'NNP' because that gave me 1 percent increase in accuracy.

I also included output.pos, this is the result from WSJ_23.

Please let me know if you have any questions by emailing jhishan@nyu.edu