**NLP HW 3**

Task 1:

1. What is the selected threshold for unknown words replacement?

* I selected a threshold of 1 which means I replaced all the words with occurrences less then 1 to <unk>. The reason or selecting 1 was I tried various thresholds but the accuracy decreased.

1. What is the total size of your vocabulary and what is the total occurrences of the special token ‘< unk >’ after replacement?

* I also replaced numbers with <num> tag since it helped increase the accuracy, so the total size of my vocabulary is 19492 with <unk> being 14775 and <num> being 4293. I have also replaced the numbers in the vocabulary.txt file.

Task 2:

1. How many transition and emission parameters in your HMM?

* The number of emission parameters in 27998 and the length of transition parameters is 1392. I have also outputted the parameters in json format.

Task 3:

1. What is the accuracy on the dev data?

* The accuracy of dev data is 92.2%. I also created a greedy.out file.

Task 4:

I tried to implement Viterbi Algorithm and got stuck with errors. I tried to create a state array which would have the most likely tags.

So my thought process was lets say we start a sentence so we will first try all the <s>,tag transition probability and find the state probability by doing emission\_prb \* transition\_prb and append it to state list.

Also, is this was the first word of sentence their won’t be any best path yet.

Next we will select all the non zero probability tags.

Repeat the above process and if we get more then 1 tag for a word with some state probability their will be 2 branches and we will select the one with higher probability and discard the one with lower chances of tag sequence.

Let’s say we start with <s>, tag and get <s>, DT with some state probability and no other tag had non zero probability so we will have state with [ [ [DT], 0.16 ] ] along with prob.

Next we will repeat the process and suppose we see NN and VB with some probability.

So we will keep both of this tags in our state list and cannot discard anyone based on max probability as their might be case that a particular tag sequence might increase later.

So our state will be [ [ DT], 0.16 ] , [ [DT,NN], 0.0023 ], [ [DT,VB], 0.0014 ] ]

The next step is important as we will search both the branches of DT,NN and DT,VB. Then we will select the one with high prob and discard the latter sequence.

So, if we get DT,NN,NNP with higher prob then DT,VB,IN then we will remove the path DT,VB,IN from our state. And add DT,NN,NNP.

So our state will be [ [ DT], 0.16 ] , [ [DT,NN,NNP], 0.0018] ]

We will repeat this process until the EOS.

I hope I can get some partial credits for my understanding and partial implementatins.