Jason Lee jjl625
Professor Perlin
NLP Homework 3

To produce an output file:

Run hw3.py

Here is how I handled OOV items:

- *The tag here means the tag we're currently testing for that particular OOV word to see if it's the best tag to be used. Same with the previous POS (previous state).*

1. Tags that do not consist of English letters are unlikely to be OOV so I set the probability of these tags to 0.
2. If the tag is NNP, the current previous POS (the one we just transitioned from) is not the beginning of a sentence, and the first letter of the word is uppercase, the word is an NNP and set the probability of the tag to be 1.
3. If the tag is NNP but the first letter of the word is not uppercase, set the probability to be 0 as it is not an NNP.
4. If the tag is NNS but the word does not end with an s, then it is likely not an NNS and set the probability to be low. If the word does end with an s, set the probability to be high.

I tried two other strategies. However, they did not improve the output so I removed them:

1. I made an OOV counts dictionary that keeps track of the number of vocabs that appeared once for each POS, revealing which POS will tend to have more OOVs to determine what class an OOV belongs to more efficiently. However, I could not figure out a good way to factor them in. It decreased the accuracy of my program so I removed it.
2. I made a POS distribution dictionary to count the number of each POS tags, essentially counting the number of times a certain POS is found in the corpus. I then attempted to factor these probabilities in when I process OOVs, but the accuracy of my output did not improve so I removed it.