Section 1: Q&A

1. Q: What is the selected threshold for unknown word replacement?

A: I set the threshold to 2 (replace if less than) as it gives the best accuracy. I think the reason for unknown word replacement is to prevent these low-occurrence words from diluting other relatively meaningful words. As HMM is based on probability, unifying these un-common and sparse (POS tag) words into a larger set (the <unk>) can give a better result.

2. Q: What is the total size of your vocabulary and what are the total occurrences of the special token "<unk>" after replacement?

A: Vocabulary total size: 23183, Total occurrences of <unk> after replacement: 20011

3. Q: How many transition and emission parameters are in your HMM?

A: Total transition parameters: 1392, Total emission parameters: 30303

4. Q: What is the accuracy on the dev data?

A: Greedy Decoding Accuracy = 0.9317740270778945

Viterbi Decoding Accuracy = 0.9304687025681501

Section 2: Reasoning on why Greedy Decoding gives better accuracy given training.txt

The Viterbi examines the whole search space and thus must be at least as good as Greedy. What causes this is that the given training data isn't sufficient enough to determine a decent number of instances (the available transition and emission products all result in 0). I use two different methods to guess the POS tag in these instances, and it turns out the method Greedy decoding is using not only gives better guesses but even surpasses Viterbi's accuracy in the end.

Greedy's method: Guess the most possible POS tag using the transmission probabilities in HMM dictionary. If it's still insufficient to guess the POS tag, then simply guess the POS tag that has the highest occurrences in the training data (After hyper-tuning it, "CC" gives the best accuracy).

Viterbi's method: I view the Viterbi matrix as a Flow network. When the training data isn't sufficient to determine an instance (when an entire column is 0), the Flow network is split in half. In these instances, I set the column that has all 0 to all 1. By connecting the two halves manually, I let the flows determine (guess) the most possible POS tag for these instances.

Che Wei Wu