CSCI544: Homework Assignment – 2

Name: Ashish Bhagchandani
USC ID: 4690271015

Task 1: Vocabulary Creation

In this section train data is parsed, three lists are created for words, tags and (words, tags). With word count and threshold value 2, words are replaced with <unk> tag which are having count less than or equal to 2. Next with the final list vocab file is created.

1) What is selected threshold for unknown words replacement?
   - 2
2) What is the total size of your vocabulary?
   - 16920
3) What is the total occurrences of the special token ‘< unk >’ after replacement?
   - 32537

Task 2: Model Learning

For model learning, two functions are created for calculating transition probability and emission probability respectively. These functions only count transition and emission probabilities of (tag1,tag2) and (word, tag) respectively if the pair exists in data else its consider as zero by defaultdict. Next, created various dictionary files to store values and count or words and tags. After creating both transition probability and emission probability dictionary, I generated hmm.json file combining both dictionaries.

1) How many transition and emission parameters in your HMM?
   - Transition parameters: 1417
   - Emission parameters: 23373

Task 3: Greedy Decoding with HMM

For this task I created a function greedyDecoding. Next, I parsed each sentence of dev data to this function and stored the predicted POS tags of the words in the sentence. The function takes input a sentence and initialize the variables "iniTag" and "iniWord" with the first tag and first word of the sentence, respectively. For each Part-of-Speech tag "p" in "posTags", it calculates the product of the transition probability from the initial tag to the current tag and the emission probability of the current tag and the current word. Further, keeps track of the part of speech tag with the maximum result and add it to the "predTags" list. The function ran on each sentence of the dev data. After this I calculated the dev data accuracy by comparing the POS tags of words with the predicted POS tags. For test data I parsed each sentence to greedyDecoding function and stored the predicted tags and returned in greedy.out file**.**

1) What is the accuracy on the dev data?
    - `92.75985484913257`

Task 4: Viterbi Decoding with HMM

For this task I created a function vitebiDecoding. Next, I parsed each sentence of dev data to this function and stored the predicted POS tags of the words in the sentence. The function ran on each sentence of the dev data. The function takes input a sentence. First, it initializes a set of part of speech tag (posTags), which is created from the training data and with the "start" tag removed. The function is implemented using a dictionary viterbiDP, where the keys are pairs of a tag, and the values are the probabilities of the most likely sequence of tags ending with the given tag at the given position. Finally, the function traces back the most likely sequence of tags by starting from the end of the sentence and selecting, for each position, the tag with the highest probability as the previous tag. The resulting sequence of tags is returned by the function. After this I calculated the dev data accuracy by comparing the POS tags of words with the predicted POS tags. For test data I parsed each sentence to viterbiDecoding function and stored the predicted tags and returned in viterbi.out file**.**

1) What is the accuracy on the dev data?
    - `93.11883780768268`