Home work 3

Steps to Find Part of Speech Tags

Part 1

Calculating and storing different counts in hashmaps. This will be helpful in calculating different conditional probabilities. For this we will use the file entrain.txt.

1. **Calculation of individual word counts and storing it in a hashmap “words”.**
2. **Calculation of individual tag counts and storing it in a hashmap named “tags”. This will be used in computing P(Current Tag | Previous tag).**
3. **Storing the bigram counts (tag sequence {tag1, tag2}) in a hashmap State_Trans. This will store the count of tag sequences.**
4. **Storing the observation-tag counts in a hashmap Obs_Prob. This will have the counts of Observations and their corresponding tag counts.**
5. **After Storing all these counts in different hashmaps we proceed to apply the viterbi algothim.**

Part 2

Calculating the Tag sequences.

For this we will use test.txt file

1. **After calculating the counts now we find out the tag Sequence for each sentence. Our job is find out the tag with highest probability at each step.**
2. **We find the maximum value of P(word|tag)P(tag| previous tag) at each level. Using the formula.**

   $$a_{ij} = \frac{C(q_t = s_i, \mathrm{q}_{\mathrm{t+1}} = s_j)}{C(q_t = s_i)}$$

   $$b_j(k) = \frac{C(q_i = s_j, o_i = v_k)}{C(q_i = s_j)}$$

3. **We store this tag in a hashmap along with the word.**
4. **We do this till the sentence ends.**
5. **After finding the tags for one sentence we move on to the next sentence and repeat the same procedure till we reach the end of file.**

Part 3

Calculating the error rate

$$b_j(k) = \frac{C(q_i = s_j, o_i = v_k)}{C(q_i = s_j)}$$

1. **Now we count compare the word-tag sequence with our output file and find the error rate.**

Smoothing

For Smoothing I used this formula. In our case the size of V is 12464.

Bigram $P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}w_n) + 1}{C(w_{n-1}) + V}$

Similarly for the Emission probability

$$b_j(k) = \frac{C(q_i = s_j, o_i = v_k) + 1}{C(q_i = s_j) + V}$$

Result

After running the Program the error rate was **0.1002** after smoothing