Michael Royster

CSCI 544 – Homework #2

Task 1: Vocabulary Creation

I utilized a pandas data frame to group occurrences by word, then sort the data frame by occurrences. I then merged the occurrence data into the original data frame and used this value with a threshold to turn words into <unk> before adding these totals back into the vocabulary. After some experimentation from 3 to 10 I settled on 5 being the optimal threshold.

Threshhold = 5

Before replacement: 43,193 words

After replacement: 11,688 words

<unk> occurrences: 50,296 times

Task 2: Model Learning

I utilized pandas data frames for the initial, transition and emission tables. I found it much easier to split the initial transitions out into its own table, rather than including it in the transition table itself.

Transition parameters: 1,378

Emission parameters: 17,116

Task 3: Greedy Decoding with HMM

I used matrix multiplication and np.argmax() to find the most likely index after the calculation. I converted my data frames into numpy arrays and used a dictionary to keep track of the string values for each row and column. I took a column vector from the emission array and multiplied it with a row vector from the transition array. This was much faster than traditional loops.

Greedy Dev Data Accuracy: 0.9216

Task 4: Viterbi Decoding with HMM

For the Viterbi algorithm, I created an optimum matrix (as in from dynamic programming) to store the best value for each combination of transitions given an emission and the values from the previous column. Additionally, I kept track of a backtrack array that placed the label index of the row that generated the max probability with np.argmax(). This allowed me to select the max probability in the last column and ‘backtrack’ using the index to identify the row. All of this was done using matrix multiplication similar to my greedy decoding implementation.

Viterbi Dev Data Accuracy: 0.9371