Ander Swartz NLP HW # 2 Report 2/17/22

Unigram:

Accuracy on development set: 0.1533

Unigram only predicts “ “ because that is the most frequent character in the training data.

English:

V1 (5-gram) Accuracy on development set: 0.5283

V2 (5-gram with backoff) Accuracy on development set: 0.5338

My first modification was to implement the Katz backoff algorithm. In my code, I store a “scores” vector that hold the probability of each potential character that the model might predict. To introduce backoff, I inserted this section of code into a while loop and continually reduced n (the number of grams) as long as the sum of scores == 0, meaning that all of the probabilities were 0 because the current ngram model had never seen any characters following the current state.

V3 (5-gram with backoff properly trained) Accuracy on development set: 0.5454

V3 (10-gram with backoff properly trained) Accuracy on test set: 0.5868

My second modification involved noticing that I had not changed the training function when implementing backoff. If I had started the model at 5-gram, for example, it would only learn the probabilities for each 5-gram token, but not the 4-gram, 3-gram, etc. Somehow, the model was still learning some of the probabilities for the smaller ngrams, but forcing it to learn the probabilities for all sizes from n - 1 drastically improved the model. Lastly, I experimented with the different n values and found diminishing returns with n>5, so the highest I kept it was 10, even though a larger n value might add a very small performance increase. This is because the largest the ngram model, the lower probability there is for what is essentially an entire sentence to appear in the data.

Chinese:

V1 (bigram) Accuracy on development set: 0.8881

V2 (trigram with fixed spaces and backoff) Accuracy on development set: 0.9102

My first modifications came after noticing that the model was predicting “ ” far too frequently, which was accounting for most of the error. This had to do with the model being confused when reading the <space> tokens, which were not present in the training data. I fixed this by checking each individual individual token to see if it was <space>, and handling those accordingly, while discounting the model’s probability of predicted “ ” otherwise. I also made implementing a smaller version of backoff than in the English section, which is hard coded to backoff from a trigram to a bigram or unigram if necessary. Unlike the English ngram which can backoff from any starting ngram, this requires n = 3. This restriction did not seem problematic because my Chinese ngram had not been affected by increase in n>3 before making this change.

I also changed the requires for backing off, so that instead of only backing off if the current ngram model had 0 probability for all characters following the current state, this version would backoff if there were more than one “winner” (if the max probability belonged to multiple characters), or if the difference between the character with the highest probability and the second highest probability was small. I believed that this would be a better way to decide when to use backoff, since there were times when the model was choosing whichever probability came first if there multiple characters with the same probability. Additionally, a character might have a slightly higher probability than another at a certain ngram, but backing off might find a larger difference that is more meaningful.

V3 (trigram with interpolation and smoothing) Accuracy on development set: 0.8693

My final modification to the ngram for the Chinese data set was to implement interpolation and smoothing. This was a lot easier than my other modifications, since I was simply able to add each of the probabilities from the unigram, bigram, and trigram model for each character when predicting each token. This code could be improved to be more modular, since it currently only works for interpolating when set to a trigram model. The smoothing made a very small difference but still helped the model.