# PROBLEM SET #2

MARIA MOYA

## 1. Bigram LM: MLE estimates, Perplexity, Laplace smoothing,SLI and Deleted Interpolation

### Text Preprocessing:

We will consider a Bigram LM and partition the data into three sets: the training corpra, the held-out set(or dev corpus) and the test corpra. Thus we need to account for words that occur in our test set but not our training data. Thus we convert any word in the training/test set that is not in our vocabulary V to $< UNK >$.From lines 21 through 34 I coded a way to look through the vocabulary in my training set.Then from lines 36 through 50 I defined a function that implemented the start and end tokens as well as impute $< UNK >$ tokens.

### Bigram Language Model Class:

Next we consider a Bigram LM class. For the *EstimateBigram* function, the estimate that we will use to compute our predictions is Maximum Likelihood Estimator(MLE) which maximizes the likelihood of the training set T given the model M. The bigram model is a product of conditional probabilities where each given probability is a MLE estimate defined by

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

Thus, in order to calculate these probabilities I first needed to specify my counts. I defined a function called *getCounts* to acquire the bigram and unigram counts. Next I assert a validity function labeled *CheckDistribution* in line 146. This verifies that the probabilities for these n-grams are valid. I specify that if the sum equals 1(isvalid=1) then we have a valid probability. Next we consider the perplexity where perplexity is a metric that is used to measure accuracy of our predictions/estimate. It is the probability of the test set normalized by the number of words:

$$PP(W) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_i|w_{i-1})}}$$

I defined a function labeled *Perplexity* which references the regular MLE, smoothing, the SLI and deleted interpolation. Moreover, $self.bigram\_prob[(data[i], data[i+1])$ allows us to reference each bigram probability where we are taking the next word $data[i+1]$ given the previous word $data[i]$. Note that given the equation above, we would expect the perplexity for MLE to approach infinity since some of the MLE will be zero, thus dividing by zero makes our definition of perplexity undefined. Therefore, next I consider Laplace smoothing and interpolation methods to account for the zero probabilities.

### Laplace Smoothing, Simple Linear Interpolation(SLI) and Deleted Interpolation

For efficiency sake, I only store the counts rather than the probabilities themselves. All other probabilities are zero(see lines 94-96). Moreover I will only add 1 to the zero entries(or unseen bigrams). For laplace smoothing I use the following equation:

$$P(w_n|w_{n-1}) = \frac{c(w_n w_{n-1})+1}{c(w_{n-1})+V}$$

which can be shown in lines 98 through 101(the numerator is a bigram count plus while the denominator is a unigram count plus the size of the vocabulary).

Next I constructed a simple linear interpolation which is a linear combination of a bigram and unigram which is an alternative method of converting a zero probability value into a nonzero entry:

$$\hat{P}(w_n|w_{n-1}) = \lambda_1 P(wn|w_{n-1}) + \lambda_2 P(w_n)$$
$$\text{where, } \lambda_1{+}_2 = 0.5 + 0.5 = 1$$

I implemented the SLI in lines 107-108.

Next I consider the deleted interpolation method. Here I derive $_1$ and $\lambda 2$ from the held-out corpus. We choose $\lambda$ in order to maximize our likelihood of the held-out data. We take our training data and train some n-grams then we figure out which lambdas would we use to interpolate those n-grams such that it gives me the highest probability of this held out data then I calculate the perplexity. My deleted interpolation can be found in lines 110-141.

**Results:Interpreting the perplexties**

| | |
|---|---|
| Perplexity without smoothing | inf |
| Perplexity with Laplace smoothing | 1385.802071 |
| Perplexity with linear interpolation | 232.0219868 |
| lambda 1 | 0.354212632 |
| lambda 2 | 0.645787368 |
| Perplexity with deleted interpolation | 220.8338368 |

Again, perplexity is a metric that is used to measure accuracy of our predictions/estimate. Moreover, minimizing perplexity is the same as maximizing probability. Therefore, to no surprise, the perplexity of Laplace smoothing is the has the highest (therefore it has the lowest probability). Laplace smoothing is only useful if you don't have a lot of outcomes.In order to perform Laplace smoothing, we have to redistributes counts at the expense from other bigram counts in order to add 1 to all zero bigram counts. Therefore a better metric for imputing values is the SLI which calculates an estimated value by the linear combinations of bigrams and unigrams which is weighted by lamdas. The deleted interpolation goes a step further as far as accuracy since it figures out the weights for lambda(rather than assuming they hold the same weight). For this assignment the linear combination is weighed more towards the unigram($\lambda 2 = 0.65 > 0.35 = \lambda 2$). We then take those new lamdas, plug them in and recompute the perplexity.