Deveshwar Singh March 18, 2020

Machine Learning for Natural Language Processing Professor: A. Rozovskaya

**HW1**

**Part1**

BigramAdd1: p(('am', 'sam')) (This is the Bigram)

BigramAdd1: p(0.03571428571428571) (This is the probability of it with the text given from the book)

**Part2**

1. How many word types (unique words) are there in the training corpus? Please include the padding symbols and the unknown token.

The number of unique words in the training corpus including the padding symbols and the unknown token is: 83046

1. How many word tokens are there in the training corpus?

The number of word tokens not including padding symbols is: 2409517

1. What percentage of word tokens and word types in the test corpus did not occur in training (before you mapped the unknown words to <unk> in training and test data)? Please include the padding symbols in your calculations.

The percentage of word tokens in the test corpus that did not occur in training is: 0.017234919445485202

The percentage of word types in the test corpus that did not occur in training is: 0.0005539098812706211

1. Now replace singletons in the training data with <unk> symbol and map words (in the test corpus) not observed in training to <unk>. What percentage of bigrams (bigram types and bigram tokens) in the test corpus did not occur in training (treat <unk> as a regular token that has been observed).

The percentage of bigram types in the test corpus that did not occur in training is: 0.27489539748953973

The percentage of bigram tokens in the test corpus that did not occur in training is: 0.23907547851209823

1. Compute the log probability of the following sentence under the three models (ignore capitalization and pad each sentence as described above). Please list all of the parameters required to compute the probabilities and show the complete calculation. Which of the parameters have zero values under each model? Use log base 2 in your calculations. Map words not observed in the training corpus to the <unk> token.

Unigram: p(<s>)+p(i)+p(look)+p(forward)+p(to)+p(hearing)+p(your)+p(reply)+p(.)+p(</s>)

Unigram: p(0.038321267882140644)+p(0.00010777465139603619)+p(2.5317275689627692e-08)+p(4.598701091766609e-12)+p(9.348545938425964e-14)+p(7.487385984191813e-18)+p(3.4918909295327205e-21)+p(1.7395779404359262e-26)+p(5.859262978423797e-28)+p(2.2453438618808756e-29)

Log Probability Unigram: -95.1689783491538

Bigram: p(('<s>', 'i'))+p(('i', 'look'))+p(('look', 'forward'))+p(('forward', 'to'))+p(('to', 'hearing'))+p(('hearing', 'your'))+p(('your', 'reply'))+p(('reply', '.'))+p(('.', '</s>'))

Bigram: p(0.02006)+p(4.100013625834583e-05)+p(2.2740695477712207e-06)+p(4.79761507968612e-07)+p(5.426347926051259e-11)+p(0)+p(0)+p(0)+p(0.0)

Log Probability Bigram: Log(0) is Undefined

Bigram parameters that have 0:

p(('hearing', 'your'))=0

p(('your', 'reply'))=0

p(('reply', '.'))=0

p(('.', '</s>'))=0

BigramAdd1: p(('<s>', 'i'))+p(('i', 'look'))+p(('look', 'forward'))+p(('forward', 'to'))+p(('to', 'hearing'))+p(('hearing', 'your'))+p(('your', 'reply'))+p(('reply', '.'))+p(('.', '</s>'))

BigramAdd1: p(0.010964457021732241)+p(1.9409339198729422e-06)+p(8.120188765769729e-10)+p(9.819672717226324e-13)+p(5.050752349154575e-17)+p(6.066605428087893e-22)+p(7.199607690312347e-27)+p(8.668064496697945e-32)+p(4.203154311845068e-32)

Log Probability Bigram with Add-One Smoothing: -104.23022661277582

1. Compute the perplexity of the sentence above under each of the models.

The Perplexity of Unigram Model is: (1/10) \* -95.1689783491538 =

-9.516897834915381

The Perplexity of Bigram Model is: Undefined

The Perplexity of Bigram with Add-One Smoothing Model is:

(1/9) \* -104.23022661277582 = -11.581136290308423

1. Compute the perplexity of the entire test corpus under each of the models. Discuss the differences in the results you obtained.

The Perplexity of Unigram Model is: Undefined

The Perplexity of Bigram Model is: Undefined

The Perplexity of Bigram with Add-One Smoothing Model is:

(1/2769) \* -32403.13181153221 = -11.702106107451142