Jihoon Oh

Natural Language Processing

Homework 1

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

   Answer: Vocabulary size is 15031 (see method vocabularySize)

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

   Answer: 498474 word tokens, including start and stop symbols (see method totalCountOf)

3. What percentage of word tokens and word types in each of the test corpora did not occur in training (before you mapped the unknown words to <unk> in training and test data)?

   Answer:

   brown-test.txt = 8.826%

   learner-test.txt = 6.818%

4. What percentage of bigrams (bigram types and bigram tokens) in each of the test corpora that did not occur in training (treat <unk> as a token that has been observed)?

   Answer:

   brown-test.txt = 47.993%

   learner-test.txt = 48.357%

5. Compute the log probabilities of the following sentences 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?

   Answer:

   - “He was laughed off the screen .”
     i. Unigram: -64.8659
     ii. Bigram: undefined
     iii. Bigram Add One: -72.9303
   - “There was no compulsion behind them .”
     i. Unigram: -59.0070
     ii. Bigram: -36.4261
     iii. Bigram Add One: -58.9312
   - “I look forward to hearing your reply .”
     i. Unigram: -84.6301
     ii. Bigram: undefined
     iii. Bigram Add One: -93.4932

   Note: “undefined” means the bigram token does not exist. Because it doesn’t exist, the probability comes out to 0, thus the log function becomes undefined. Also note that I took the base 2 of the probability, as suggested in Michael Collin’s notes.

   Furthermore, the word “compulsion” from the second sentence needed to be replaced by <unk> because that word was never seen in the training data.

6. Compute the perplexities of each of the sentences above under each of the models.

   Answer:

   - "He was laughed off the screen ."
     1. Unigram Perplexity: 147.7820
     2. Bigram Perplexity: 1000
     3. Bigram Add One Perplexity: 275.0141
   - "There was no compulsion behind them ."
     1. Unigram Perplexity: 94.1138
     2. Bigram Perplexity: 16.5338
     3. Bigram Add One Perplexity: 93.5661
   - "I look forward to hearing your reply ."
     1. Unigram Perplexity: 352.8747
     2. Bigram Perplexity: 1000
     3. Bigram Add One Perplexity: 652.2683

   Note: Perplexity is set to 1000 for those that are supposed to be undefined.

7. Compute the perplexities of the entire test corpora, separately for the brown-test.txt and learner-test.txt under each of the models. Discuss the differences in the results you obtained:

   Answer:

   - brown-test.txt
     i. Unigram for brown test: 320.1785
     ii. Bigram for brown test: 1000
     iii. Bigram Add One for brown test: 668.7095
   - learner-test.txt
     i. Unigram for learner test: 1000
     ii. Bigram for learner test: 1000
     iii. Bigram Add One for learner test: 1000

   Notes: Perplexity is set to 1000 for those that are supposed to be undefined. For learner-test, all of them are set to 1000 because one of the probabilities for one of the sentences came out to be 0. For unigram, one of the sentences was so long that the probability came out to be 0. More accurately, the probability stopped around a number that had an exponent of -320 and because python couldn't handle a number any less, it turned the probability into 0.

   The results came out to be as expected. I was expecting the perplexity of brown-test to be smaller than that of learner-test because the corpus of the former is similar to brown-train's corpus.