Language Technology I, Winter 2016-2017

Exercise 4: N-grams

You can earn up to 20 points on this exercise.
You may work as a group of up to 3 people, but please submit your own version.

Please email your solution to `langtech1saarlandws1617@gmail.com` by **15:00, November 30, 2016**. *Use the subject line "Exercise 4" (there was no Exercise 3), and attach an answer PDF file as well as the model file requested in Task 2. (Some of the following tasks* ***may*** *have been blatantly copied from Statistical Machine Translation book available here:* ***http://www.statmt.org/book/*** *We didn't check.)*

Task 1

Consider the following Mother Goose nursery rhyme, without case or punctuation (it is a string of words). This is our corpus.

<s> peter piper picked a peck of pickled peppers </s>
<s> a peck of pickled peppers peter piper picked </s>
<s> if peter piper picked a peck of pickled peppers </s>
<s> where s the peck of pickled peppers peter piper picked </s>

1. Construct and show a table of probabilities for observed bigrams, including the start and end symbols, given the above corpus. (2 points)
2. Construct and show a count-of-counts table for the bigrams. (2 points)
3. Adjust and show the bigram table using Good-Turing discounting. Use add-one smoothing on the count-of-counts table if there are missing counts. (2 points)
4. Give the probability of a missing bigram. (1 point)
5. Briefly explain in your own words the purpose of smoothing the count-of-counts table. (1 point)

Task 2

This exercise is to get you to used to using NLTK and Python for doing common text processing tasks.

1. Download the sample training and test texts posted on the web site.
2. Write a Python script that uses NLTK's Punkt sentence tokenizer and the WordPunctTokenizer to turn the posted sample texts into files with the following characteristics: one sentence per line, each line lowercased and tokenized. Use default values for the tokenizers.
3. Submit the script you used to do this (3 points).
4. Report the total numbers of lines and tokens you get from processing the training and test corpora, each file separately (2 points).

Task 3

This exercise is to get you familiar with using a popular LM toolkit. Download and install the SRILM[1] toolkit. SRILM is a great free language modelling toolkit for doing various n-gram language models. It includes several other smoothing techniques, but today we will use Good-Turing and Laplace.

1. Train the following models on the training set:
   a) Order-2 SRILM Good-Turing discounted model.
   b) Order-3 SRILM Good-Turing discounted model.
   c) Order-4 SRILM Good-Turing discounted model.
   d) Order-3 SRILM Laplace smoothing model adding 1.
   e) Order-3 SRILM Laplace smoothing model adding 0.001.

   Attach the order-2 model with your submission. (1 point)

2. Report the perplexity for each model on the test corpus with and without end-of-sentence tags. (2 points)

3. Explain in your own words the differences between the perplexities of the order 2, 3, and 4 Good-Turing discounted models. What happens as you increase the order, and why? (2 points)

4. Explain in your own words the differences between the perplexities of the default Good-Turing order 3 model, the model with add-1, and the model with add 0.001. What happens as you change the added constant? (2 points)

[1] `http://www.speech.sri.com/projects/srilm/download.html`