

CMSC723: Computational Linguistics I

Assignment 4: N-grams and Smoothing

Bonnie Dorr (professor), Nitin Madnani (co-instructor)
Hamid Shahri, Alexandre Tzannes (TAs)

Out: **October 31, 2007**
Due: **November 14, 2007**

Submission Guidelines

1. Print out any written portion of the assignment and submit in class on November 14, 2007.
2. If you wish to submit anything electronically, use the subject *CMSC723: Assignment 4*.
3. Please attach a zip file or tarball to your email, not each individual file.

Problem 1

Show that using ELE (Expected Likelihood Estimator) for unigrams yields a well-formed probability distribution, i.e.,

$$\sum_{w_i} P_{\text{ELE}}(w_i) = 1$$

Problem 2

Assume that we have the following scenario: 100 samples have been seen from a potential vocabulary of 1000 items, and in that sample, 9 items were seen 10 times, 2 items were seen 5 times and the remaining 989 items were unseen. Work out the expected frequency estimates for each of the three kinds of items according to Laplace's Law. Also calculate the probability mass that will be assigned to the unseen items by this law.

Problem 3

(a) Write a Python program using NLTK that computes the expected frequency estimates for bigrams of rank[†] 0 through 10, according to the following discounting techniques:

1. Laplace's Law
2. Lidstone's Law of Succession ($\gamma = 0.5$)
3. Good-Turing Discounting

Use Jane Austen's *Persuasion*, *Sense and Sensibility* and *Emma* together as the training corpora. These corpora come bundled with NLTK as sections of the Gutenberg corpus collection[‡]. Your program should output a table with the rank as the first column and the three expected frequency estimates as the subsequent columns.

(b) For each of the discounting techniques, calculate:

- The total probability mass that is assigned to unseen bigrams.
- The probability assigned to an unseen bigram assuming that the mass is uniformly distributed.

Note: You may find NLTK's built-in `FreqDist` class to be useful for this problem.

[†]A bigram is of rank n if it occurs exactly n times in the training data. Bigrams with rank 0 are obviously the unseen bigrams.

[‡]For example, to access a list of sentences in the *Persuasion* section of the Gutenberg corpus, use `nltk.corpus.gutenberg.sents('austen-persuasion')`