Computational Linguistics 1 - Fall 2011
CMSC/LING723, LBSC 744
Homework 2 - Due 29 September 2011

Submission Guidelines

1. Print out any written portion of the assignment and submit in class on September 29, *as well as* emailing your writeup, in .pdf format, to `compling723.fall2011@gmail.com`.
2. Any code you write for this homework should also be submitted to `compling723.fall2011@gmail.com`.
3. Be sure include your full name and e-mail in your submission, and clearly indicate each problem number in your solution set.

Problem 1 (20 points)

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 2 (40 points)

For this and the next problem, we'll be building smoothed language models from our WSJ corpus, f2-21.txt. In order to have a training set and a testing set, we'll divide our corpus into quarters, using the first 3/4 as training and the last 1/4 as testing. These divided corpora will be made available on the class mailing list. Note that we have added sentence-boundary markers (<s> and </s>) to our text; use <s> and </s> both in probability computation, but only count </s> (not <s>) in N.

Write code to compute the expected frequency estimates for bigrams of rank[†] 0 through 10, according to the following discounting techniques:

1. Laplace's Law
2. Good-Turing Discounting.

In order to calculate these numbers, you'll need to count n-gram occurrences[*] from our training corpus, f2-21.train.txt. You'll use these counts again for Problem 3.

[†]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.

[*]Note: If you are using NLTK for this problem, then you may find NLTK's built-in `FreqDist` class to be useful. `grmcount` from the AT&T GRM Library may be helpful to those not using NLTK.

r	C_{LAP}	C_{GT}
0		
1		
2		
⋮		
9		
10		

Table 1: Expected Frequency Estimates

Your program should output a table with the rank as the first column and the two expected frequency estimates as the subsequent columns (see Table 1).

Problem 3 (40 points)

Now we're going to actually apply our language model. Build a Laplacian-smoothed bigram language model.

In order to handle out-of-vocabulary words (OOVs) in our LM, use the low-frequency strategy discussed in class, replacing every word that occurrs fewer than 5 times in the training set with the unknown word token, <unk>.

Problem 3(a) For each of the following bigrams, what is the smoothed probability of the second word given the first?

1. New York
2. of the
3. President Bush
4. stocks rose
5. the of

Do you find any of these either too high or too low?

Problem 3(b) Now read in our test data set, f2-21.test.txt, and assign a probability to each string. Remember to replace the rare and unknown words with the <unk> token. What is the perplexity of f2-21.test.txt?