LING 570: Hw6
Due date: 11pm on Nov 11
The total point is 100.

All the examples are under **~/dropbox/21-22/570/hw6/examples/.** Also, see the slides for Hw6, which will be discussed in class on Nov 4.

As mentioned in class, for Hw6 and other assignments, when we compare your output file with the "gold standard", we will sort your file before comparison. Thus, do not worry if your sort function produces slightly different results from sort functions used by others.

Q1 (10 points): write a script, **ngram_count.sh**, that collects unigrams, bigrams, and trigrams.

- The command line format is: ngram_count.sh training_data ngram_count_file

- The format of the training data: w1 w2 … w_n; that is, one sentence per line (e.g., **examples/training_data_ex**)

- The format of ngram_count_file is: count word1 … word_k (e.g., **examples/ngram_count_ex**).

    - The delimiter for the line "count word1 … word_k" can be either a single whitespace or a tab, and our auto-grader code will treat them the same way.
    - In the file, list lines for unigrams first, bigrams next, and then trigrams.
    - For each n-gram "chunk", sort the lines by frequency in descending order.
    - For ngrams with the same frequency, it is up to you whether you want to sort the lines by the ngrams alphabetically.
    - The example file **examples/ngram_count_ex** is sorted by frequency only, not by ngrams to break the ties.

- When counting ngrams, do not forget to "insert" ONE (not two) BOS marker and ONE (not two) EOS marker in each sentence. Let's represent BOS as "<s>" and EOS as "</s>" in the ngram_count_file. For instance, if the input sentence is
  John call Mary

  You will count the ngrams as if the sentence were written as
  <s> John call Mary </s>

Q2 (15 points): write a script, **build_lm.sh**, that builds an LM using ngram counts WITHOUT smoothing:

- The format is: build_lm.sh ngram_count_file lm_file

- ngram_count_file is an input file produced by Q1.

- lm_file is the output file, and it follows the modified ARPA format, as discussed in class and on hw6 slides (e.g., **examples/lm_ex**).

- When printing out the prob and lgprob numbers on each line, you do NOT need to truncate or round the numbers. For instance, it is fine if the number is displayed as something like 0.000251234 or 2.51234e-4. As long as the numbers in your system output are close to the ones in the gold standard, we will regard them as the same.

- **examples/lm_ex** shows correct formats, but lines are not sorted alphabetically. It is provided just to show the format. Your lm_file should be sorted by the frequency (and then by ngrams to break the ties).

- Do not use smoothing for the probability distributions.

Q3 (35 points): Write a script, **ppl.sh**, that calculates the perplexity of a test data given an LM. For smoothing, use interpolation.

- The format is: ppl.sh lm_file l1 l2 l3 test_data output_file

- lm_file is an input file created in Q2.

- Use interpolation to calculate probability: l1, l2, and l3 are lambda_1, lambda_2, and lambda_3 in the interpolation formula, respectively. They are non-negative real numbers and the sum of them should be equal to 1.

- test_data has the same format as the training data (e.g., **examples/test_data_ex**)

- The format of output_file has been discussed in class (e.g., **examples/ppl_ex**)
  - Like in Q2, you do NOT need to truncate the real numbers in the output_file (e.g., the prob, logprob, ppl).
  - The number of OOVs is just the number of unknown words in the sentence. To determine whether a word is OOV or not, just check whether the word has a line in the unigram model in the lm_file.
  - The file **examples/ppl_ex** shows the correct format.

Q4 (15 points) Use examples/wsj_sec0_19.word as training data and calculate the perplexity of examples/wsj_sec22.word by running the following commands. Fill out the table below and submit all the output files (see submit-file-list):

```
./ngram_count.sh   570/hw6/examples/wsj_sec0_19.word    wsj_sec0_19.ngram_count

./build_lm.sh        wsj_sec0_19.ngram_count   wsj_sec0_19.lm

./ppl.sh  wsj_sec0_19.lm   0.05  0.15  0.8  570/hw6/examples/wsj_sec22.word  ppl_0.05_0.15_0.8

./ppl.sh  wsj_sec0_19.lm    0.1    0.1    0.8  570/hw6/examples/wsj_sec22.word  ppl_0.1_0.1_0.8

…

./ppl.sh  wsj_sec0_19.lm  1.0  0   0   570/hw6/examples/wsj_sec22.word   ppl_1.0_0_0
```

lambda_1	lambda_2	lambda_3	Perplexity
0.05	0.15	0.8	
0.1	0.1	0.8	
0.2	0.3	0.5	
0.2	0.5	0.3	
0.2	0.7	0.1	
0.2	0.8	0	
1.0	0	0	

The submission should include:

- The readme.[txt | pdf] file that includes the table in Q4.

- hw.tar.gz includes all the files specified in submit-file-list.