COMP 150-04 Natural Language Processing
Fall 2016

Problem Set 2: Language Modeling

1. Build a bigram language model for the Brown corpus. This is a 57K-sentence data set. Your training set will be the first 50K sentences of the corpus `brown.sents()[:50000]` , your test set the last 3K `brown.sents()[-3000:]` , and your held out set the second-to-last 3K `brown.sents()[-6000:-3000]` .
    - 1.1. Text preprocessing: The NLTK version of the corpus is already tokenized. You will need to do minimal preprocessing involving introducing unknown tokens `<UNK>` and adding sentence boundary symbols `<S></S>` at the beginning and end of every sentence. Treat every word <u>occurring not more than once in the training set</u> as an unknown token. Transform all three of the data sets and print out the first sentence of each.
    - 1.2. Build a bigram language model class called `BigramLM`. The interface of this class is given to you in `pset2_template.py` . You will define the following functions
        - 1.2.1. `EstimateBigrams` for estimating bigram MLE probabilities given the preprocessed training data.
        - 1.2.2. `CheckDistribution` for checking the validity of your bigram estimates from `EstimateBigrams` . This should `assert` for every valid unigram context that the bigram probabilities sum to one, that is, Σ_j P(v_j | v_k) = 1.0.
        - 1.2.3. `Perplexity` for computing the perplexity given a test corpus. Use natural `log` and `exp` functions throughout the problem set.
    - 1.3. Using your class object from above, estimate a `BigramLM`. Check that it defines a valid probability distribution. Compute its perplexity on the test corpus. Explain your result.
    - 1.4. Introduce smoothing in your language model. First, implement Laplace smoothing. Compute the perplexity of the smoothed model on the test corpus. Explain your findings.
    - 1.5. Implement simple linear interpolation (interpolating bigrams with unigrams). Use interpolation weights of (0.5, 0.5) and compute the perplexity of the model on the test corpus. Explain.
    - 1.6. Implement the deleted interpolation algorithm to estimate the interpolation weights (SLP Figure 5.19) using the held out corpus. What are the interpolation weights that correspond to the unigram and bigram components? Re-compute the perplexity of the interpolated model with the estimated interpolation weights. Explain your findings.
2. **Extra credit:** Extend your model and smoothing methods to trigrams. Explain your methodology. Compare trigram perplexities to bigram perplexities and explain your findings.

Deliver the following by Friday 10/21/2016 after midnight 02:59 AM.

1. A PDF write-up. All questions 1-2 attempted should have at least a paragraph describing your reasoning and final solution, explaining any decisions you had to make in detail (include code blurbs if necessary). The write-up is where your outputs (answers) should go, since we cannot guarantee to search the output of your submitted code to find your answers.
2. Your code. This is a .py file that compiles and runs. It should demo the answers that you produced in your work. But again, include these answers in your write-up, just printing on the console output will not get you the right grade.

How to submit your homework:

1. Log onto the Tufts server:
   ```
   ssh your_username@homework.cs.tufts.edu
   ```
2. Navigate to the directory of your submission files and type the following command to submit all of your files together:
   ```
   provide comp150nlp pset2 [file1 file2 ...]
   ```