CSCI 404 Assignment #3

Due: Wed, Feb 7 midnight (11:59pm), online submission

Submission

1. All files should be submitted to Canvas. Only one of your team members needs to submit on behalf of the team. Indicate clearly in your submission the team members. You can submit multiple times, but please have the same team member resubmit all required files each time.
2. Implementation in Python.
3. Submit the **assn3.zip** file. For each question, submit your programs (if any) and a report (**report.** txt, .doc, or .pdf) of what you did and all the various assumptions you made, including the result of running your code, and discuss the results if necessary.
4. **DO NOT** include the given data files in your submission!!

N-gram language models and smoothing (60 pts)

Training data and test data: For this assignment, you are provided a training dataset and a test data set. You will notice they have one sentence per line, and have been tokenized with punctuations removed. You need to surround each sentence by a start of sentence and end of sentence marker (e.g., <s>…</s>). Do not further process the corpus. Using the training data you are to build N-gram models**.**

Dealing with unknown/unseen words: In order to deal with unknown words, you need to replace all words that appear only once in the training corpus with a special token for unknown words, e.g. <UNK>

Smoothing: Use add-1/Laplace smoothing to create a bi-gram and a tri-gram language model.

Task 1 (10 pts): Extract the vocabulary V and include the start and end symbol as well as <UNK> in your vocabulary. Report the size of your vocabulary. Generate Unigram, Bigram, and trigram models and save them? Next smoothed models and sentence generation

Task 2 (25pts): Generate sentences: Use your unigram, smoothed bigram, and smoothed trigram models to generate 10 sentences, respectively, and attach the probability (in log-space) to each sentence. Note that if you train a trigram model, you must append two tokens of <S> to each sentence before training. For example,

P(I like running) = P(I|<S> <S>)*P(like|<S> I) * P(running| I like)*P(</S>|like running)

Task 3 (25pts): Computing the perplexity of the test data according to the unigram model, the smoothed bigram and the smoothed trigram models.

As an example, for bigram, the perplexity of a corpus W with N words is calculated as follows:

$$Perplexity(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}}$$

Since this is prone to underflow issues, you may be better off computing it as:

$$Perplexity(W) = 2^{-\frac{1}{N}\sum_{i=1}^{N} \log_2 P(w_i|w_{i-1})}$$

or (using natural logarithms):

$$Perplexity(W) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} \log P(w_i|w_{i-1})\right)$$

Report the perplexities for the 3 language models on the test data set.