Language Modeling and Smoothing

Deadline: 17th September
Submission format: Upload in your previous github repo in a separate directory.

1. Download any of these text books from Project Gutenberg
    a. Alice in Wonderland: Alice's Adventures in Wonderland
    b. Sherlock Holmes: The Adventures of Sherlock Holmes

2. Parse the dataset into sentences using sentence tokenizer and divide it into 80/20 ratio. Keep 80% dataset for training N-grams and keep 20% for test. You can filter out unnecessary symbols, newlines, etc. You can add symbols <s> and </s> to mark sentence start and end.

3. Compute MLE for unigram, bigram, trigrams and quadgrams. How many n-grams are possible and how many actually exists?

4. Develop a system that has two functions:
    a. Generator(model_name): generates sentences by utilizing MLEs from specified n-gram model. Sampling from multinomial distribution can be done using a predefined function. Note, 5-10 sentences would suffice for this task.
    b. Probability(sentence,model_name): Compute the probability of a given sentence in log-space. Note you can provide any sentence, however, a random sentence will mostly lead to zero probability. The better idea is to take sentences from the corpus itself.

5. Implement add-1 smoothing for bigram model and give 2-3 examples where drastic change in the count occurs post-smoothing. Can you explain this drastic change in a sentence?

6. Do you observe the constant discounting value 'd' by implementing Good-turing smoothing technique? If yes, what is the value of 'd'?
Hint: You can check for n-grams having original counts between 1-10.

7. Compute the perplexity value for the test dataset for the bigram model using add-1 and Good-turing. Which performs better?