NLP Monsoon 2018 P1A : Spelling and Grammatical Error Detection/Correction

Task Description

- Task 1 : Tokenisation
    - Given text, tokenise into sentences and words. Take into consideration various delimiters and formats refer to (https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization?lang=en)
    - Data: Tokenise the Gutenberg corpus

- Task 2 : Unigrams & Spelling Detection/Correction
    a. From data, what are the most frequent words ?. Plot them according to their frequencies.
    b. Using LM, build a model that can recognise and suggest the correct spelling of a given word.
    c. Data : Gutenberg Corpus - Tokenised at Task 1

- Task 3 : Grammaticality Test
    a. Build a model from the data which can give a score of grammaticality for a given sentence. For example : 'I have a red apple' should have a higher score than 'apple a have I red'.
    b. Data :Gutenberg Corpus - Tokenised at Task 1

Guidelines for the tasks

1. For any language model you build (any N-Gram), Include a sample of the top 10 and the bottom 10 frequently occuring N-grams.
2. For any language model you build (any N-Gram), Include plots of N-Gram vs Freq sorted by frequency (descending) and log-log plots of the same
    a. For N-Grams of order 2 and Above, your code should ask the N-1th Gram and plot the remaining possibilities

        Example : Given N-1th gram (2) - 'the blue' plot, all N-Grams (3) that follow say - 'the blue pot', 'the blue eyes' and so on
3. For any language model you build (any N-Gram), Implement the following smoothing algorithms
    a. Add-One (Laplace)
    b. Any one from Witten-Bell, Good-Turing and Kneser-Ney
    c. Any one from Backoff, Deleted Interpolation.

Deadline: 18 August 2018

Submission Format

- A scaffold is provided in the ipython notebook given to you. Please put all the code in there and submit the .ipynb file.
- Include all graphs, tables, observations (in Markdown) in the notebook. Please structure the notebook task wise.