Language Modeling

Deadline: 11:59 PM, 27th September
Submission format:

- Upload in your previous Github repo in a separate directory
- Submit the link on Google Classroom

Academic Honesty: Assignments should be completed independently by each student. Limit any discussion of assignments with other students regarding requirements or definitions of the problems, or to understanding the existing code or general course material. Do not use code already available on the internet.

Tasks:

- Download any of these textbooks from Project Gutenberg
    a. Jane Austen Novels: The Complete Works of Jane Austen
    b. Donald Trump Speeches: Speeches

        - Parse the dataset into sentences using sentence tokenizer and divide it into an 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.

Classical Approach: **(2 marks)**

- Compute MLE for unigram, bigram, trigrams, and quadgrams. How many n-grams are possible and how many actually exist? Use the training corpus and NLTK library.

- Develop a function named **Generator**(model_name) which generates sentences by utilizing MLEs from a specified n-gram model. Sampling from multinomial distribution can be done using a predefined function.

- Evaluate:
    - Compare perplexity of these models on the test dataset
    - Generate random text of 5 sentences, then comment on the readability

Neural Approach: **(2.5 marks)**

- Train the following RNN language models on the train-set. Use Keras/Tensorflow.
    - Vanilla RNN Based
    - LSTM Based

- Evaluate:
    - Compare perplexity of these models on the test dataset
    - Generate random text of 5 sentences, then comment on the readability

- Does Neural performs better than Classical, if so, why? If not, why not? **(0.5 marks)**