MSAI-337, Spring 2024
Homework #1A: Tokenization and N-Gram Models
Due Date: Monday, April 15th @ 11:59PM
Total Points: 5.0 (plus optional 1.0 bonus point)

In this assignment, you will work with your group to tokenize several small corpora, build your own n-gram model and compare your results with those of a large language model (LLM). You will use several prominent NLP tools to familiarize yourself with their usage. You must implement your n-gram model and smoothing from scratch **without using tools like SRILM.** You may discuss the homework with other groups but do not take any written record from the discussions. Also, do not copy any source code from the Web.

Steps to complete the homework

1. (1.0 points) Tokenize the Wikitext-2 train, validation and test corpora using an NLTK tokenizer of your choice (see www.nltk.org/api/nltk.tokenize.html) and the pre-trained GPT2TokenizerFast tokenizer (see huggingface.co/docs/transformers/en/model_doc/gpt2). Your objective is to generate tokenized corpora appropriate for training n-gram models. Please discuss your choice of NLTK tokenizer. Examine the tokenized test sets. Show the first 200 tokens from the untokenized test set and the corresponding results from the NLTK and GPT2 tokenizers. Please comment on the differences between the NLTK and GPT2 tokenized corpora.

2. (1.0 points) Implement your own uni-gram, bi-gram, tri-gram and 7-gram models and train them on the NLTK and GPT2 tokenized training sets. Hint: Python dictionaries may be helpful when building these models. Calculate and report perplexities for each model on the NLTK and GPT2 tokenized test sets. Please comment on your results.

3. (1.0 points) Modify the models implemented in Step #2 to generate perplexity values when they encounter unseen tokens as the target for uni-grams, bi-grams, tri-grams or 7-grams -or- unseen contexts for bi-grams, tri-grams or 7-grams. You can simply skip these instances in your perplexity calculation. Alternatively (for 0.5 bonus points), you can assign a uniform probability equal to `1.0/|V|` to targets in these instances. Please comment on your results.

3a. (1.0 bonus points) Add LaPlace Smoothing to models implemented in Step #2. Calculate and report perplexities for each model on the NLTK and GPT2 tokenized test sets. Please comment on your results.

4. (1.0 points) Calculate perplexity on the Wikitext-2 test set using a pre-trained GPT2LMHeadModel. You should install Hugging Face Transformers from source (see //huggingface.co/docs/transformers/en/installation). Please comment on your results relative to those obtained using n-gram models. Hint: You may need to change directory to `transformers/examples/pytorch/language-modeling`.

5. (1.0 points) Calculate perplexity for the text sequences in the `examples.txt` file using your smoothed uni-gram, bi-gram, tri-gram and 7-gram models and compare against results from the pre-trained GPT2LMHeadModel. **You should submit 11 perplexities for each model.** Please interpret and comment on your results.

Submission Instructions

Turn in your homework as a single zip file in Canvas. This should include the code for your n-gram models, scripts used to run the GPT2LMHeadModel, and a PDF file of your results and write-up.

Good luck, and have fun!