MSAI-337: Natural Language Processing
Group Project #1

Winter 2020

Description: For this project, your group will construct a corpus from scratch that will be used by your group for subsequent projects. Raw text should be sourced from either Wikipedia or a News Wire, and must be filtered to include at least two distinct but related topics of interest (*e.g.*, "biographies of scientists born after 1500AD" and "biographies of rock stars", or "AI companies acquired since 1990" and "STEM research groups", etc.). Wikipedia provides a tool to filter articles (see https://query.wikidata.org/). Similar capabilities should be available through News Wire APIs (see https://en.wikipedia.org/wiki/List_of_news_media_APIs for a partial list). Certain aspects of this assignment are deliberately under-specified so that you can experience some of the real-world challenges of corpus preparation.

Tasks: Please perform the following tasks to construct your corpus:

1. Use an appropriate tool/API to screen for content on two distinct but related topics of interest. Download and/or scrape related articles and save the resulting raw text to 'group0.raw.txt'. Please remember to retrieve content responsibly by following *terms of service* and including random pauses in you retrieval code. (1.0 pts)
2. Prepare the corpus in accordance with lecture Slide 2-8. You should strip HTML content, use a frequency threshold of 3, inserting <s> and </s> boundaries, and replace any Unicode characters and/or other tokens that may impact display in a standard text editor/spreadsheet. The corpus should be tokenized with the `NLTK` package. You should identify and maintain meta-data. Divide your tokenized corpus into training, validation and test sets or approximately 5 million, 250K and 250K tokens, respectively. The resulting corpora should be saved to files 'group0.train.txt, group0.valid.txt' and 'group0.test.txt', respectively. (1.0 pts)
3. Write Python code to read your corpora files. Construct a vocabulary from the tokens in the training set. Out-of-vocabulary words in the test and training sets should be replace with <unk>. Save the vocabulary to a Python list, and construct a Python dictionary keyed with a word to get the corresponding index in the vocabulary list. Construct integer representations of the training, validation and test corpora and save these to separate Python lists. (1.0 pts)
4. Identify and insert tags for (a) years, (b) real numbers (e.g., with a decimal) and (c,d) two other word classes of your choosing (e.g., city names, stop words, adjectives, etc.). You can identify tokens in these word classes using the `regex` package, or by writing your own Python code. Add these tags to your vocabulary and construct integer representations of the training, validation and test corpora and save these to separate Python lists, which parallel the other integer representations constructed in Step 3. (1.0 pts)
5. Prepare summary statistics for each corpus file that includes the number of tokens and vocabulary size for (i) the untagged corpus, (ii) the tagged corpus, and (iii) each of the four word classes. (0.5 pts)
6. In addition, please prepare a short write-up (about one page) describing how each of the above steps were preformed, where you encountered ambiguity, what decisions you made and how you arrived at these decisions. (0.5 pts).

Note: It is common to perform tasks #3-5 at runtime before launching a language model, and you may want to take care such that your Python code is reusable. Also, the NetIDs for each group member ***must*** be included in the write-up and correspond to the groups in Canvas

What to Turn In: Your submission should include (i) the raw text file, (ii) three corpora text files, (iii) the Python list of the vocabulary, (iv) the Python dictionary cross-reference words and indices, (v) a total of six Python lists of the integer representations of the corpora. The Python objects should be saved to a `pickle` file. The pickle file and text files should then be compressed to a zip file.