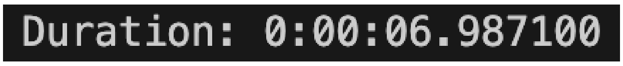
Layug, Mikaella Louise D.
Trani, Giancarlo Gabriel T.

CMSC 135 Lecture

Multiprocessing challenge

In order to implement multiprocessing for this project, the **multiprocessing** module was imported. The file containing the dataset will be read using the **readFile**() function and then preprocessed using the **preprocess_text**() function provided by the lecturer. The multiprocessing **Pool** will be utilized for the top one hundred tokens to be determined based on their frequency in the dataset. Specifically, the **Pool.map**() function will be used to apply a function for every item in an iterable. Every item in the variable **preprocessed**, which contains the list of tokens, will be passed as an argument to the **count_occurences** function in order to facilitate counting of word occurrences. The **count_occurences** function returns the **word_count** dictionary, which consists of a word as a key and its frequency in the dataset as the value. Since every time a **count_occurences**() function call is made returns a dictionary, it is only appropriate to insert the key-value pairs in one dictionary in preparation for sorting the tokens from most frequently occurring to least frequently occurring. Once the tokens have been sorted in descending order, the top one hundred tokens will be obtained and stored in the variable **top_100_tokens**. On the other hand, obtaining the unique tokens involves using the **set**() function. The Python **set**() function does not allow for duplicates, so converting the list containing the preprocessed tokens to a set would be optimal for determining unique tokens. Finally, both required outputs, obtaining unique tokens and the top one hundred tokens based on the frequency of appearance in the dataset, will be written in two separate text files. The calculated time execution of the program is 6.9871 seconds.



```
Duration: 0:00:06.987100
```