LLD Report: Word Frequency Project

Team Members: Sohail Ahmad, Shirwan Kanj, Arthur Fong

1. The functions in the projects are to import text files into Python, remove punctuation from the text and store the result into a new text file, determine word frequency from the text file, and then to organize the words based on how much they are repeated and output the results as a text file.

Necessary Functions:

1. Read a .txt file
2. Function to remove punctuations from file (!_()@#$%^&*+-.;:’”,). Use the .lower function to convert every character to lowercase.
3. Function to split the text file into individual words using a blank space as a separator. Then store these words into a data structure where each individual word is one element.
4. Traverse through the list, one word at a time with a function that counts word frequency.

Detailed Breakdown of Functions

1. Import text file from specified .txt file such as from a book to start things off. Input is a **.txt** file from the book, output is **.txt** file into python.
2. Remove punctuation from inserted text file to have text file with only words left. The input would be a **.txt** in the original format containing all the punctuation marks. The output would be a **.txt** file that is stripped down and has all the punctuations (!_()@#$%^&*+-.;:’”,) removed from it. The function would traverse through each individual letter in the text, we would use a loop to make comparison of the current letter with the predefined punctuation marks such that when the current letter is equal to one of

the punctuations, we remove it from the file. As a last step, we would use the built in **.lower** function to convert the text to all lowercase.

3. Make a list that counts each unique word in the text file without punctuation, so they can be assigned a number variable later. Input is a text file without punctuation, output is a list of unique words. Then we assign a unique number variable to the list of unique words, so they can be incremented later. Input is a list of words, output is a list with assigned unique variables to each word.
4. Increment a number next to each assigned unique variable for each of that specified word there is, so there is a list with differing increments/'frequencies' for each word. Input unique variables list, output is incrementing number to each of those variables. Utilize the incremented number next to each of the unique variables and put them into an organized list so they can be outputted into a final text file. Input is the incremented numbers for each variable, output is an organized list.

2. To improve speeds for our programs run time, we can utilize multithreading or multiprocessing, in our case we will have tasks related to I/O of text so according to the reference material implementing multithreading would be ideal for this case. We will be making use of **concurrent.futures** library while importing **ThreadPoolExecutor** which we can utilize by passing our function as an argument and specifying how many workers to use. In order to measure the performance, we can make use of the time library and simply have variables with beginning time and ending time in the start and end of the function respectively. This would provide us two time values where the difference in the time would be our time to execute our program.

3. Our desired milestones for this project will be to have a decisive plan to start working on the actual coding by the week of November 7th, this would allow us enough time to have roughly one week to implement one major function of the program at a time and still providing us two weeks in the end to have time for troubleshooting and debugging. In the week of November 14th we intend to have our initial step of reading and reformatting the text file and get a resulting file with no punctuations. Ideally in the week of November 21st we plan to have figured out the implementation for splitting the text into unique words and storing them in a data structure. In the following week of November 28th, we would like to have the code for traversing the list of unique words and making increments for each repetition in the text. The final step would be to simply output the results in an organized manner, and figure out any issues we might have in the code before the December 20th, 2022 deadline.

Contributions of Each Member:

Sohail Ahmad: I will be working on planning out the implementation of the program and working on the functions to read the text file then making alterations such as removing punctuations, and making everything lowercase.

Shirwan Kanj: I will be working on implementing the necessary code for adding function four for the project. I will also be working on helping with troubleshooting for any issues or bugs that may arise.

Arthur Fong: I will do some testing with the project, to confirm that it works as intended, especially with different texts imported. I will also implement the function such as the organization of the texts.

4. We have created a GitHub repository for our project, an invitation has been sent to “chunyuyuan” as a collaborator to the private repository.