2022 Spring -- CSCI 1300L
Project 02: Word Frequency Counter

Introduction

This project is used to conclude what we have learned in Python programming, mostly in the second half of the semester. After having covered the basic programming topics such as variables, data types, computations, input/output processing, decision structures and control structures, we have and will have learned basic file processing, lists, dictionaries, and string operations. By completing this project, it is expected that you can use these newly learned advanced concepts and techniques to make another practical computer program.

Project description

A word frequency counting program works as follows: the program opens a plain text file. Then, it will read the content from that file by processing each word: each time it processes one word, it will update the counter for that word by +1. The program will eventually generate a list of all words that are available in the document, and the frequency of each word. Symbols or special characters are not considered in the word counting process.

Project in detail:

Your Word Frequency Counter should generally follow the sequential process: open a file → process the content → output the result.

(1) File Processing.

Write the Python code to open a file named "document.txt". For simplicity of this project, assume that this file is a plain text file and it is in the same directory as the Python program. Note that this text file may contain several paragraphs. That is, there are multiple lines in a document to process. Also note that such a file "document.txt" that the user tries to open may not exist. Therefore, your program should be able to handle the situation where such a file cannot be found in the designated location.

If the file can be successfully opened, go to step-(2). If the file cannot be found or be successfully opened due to any reason, the program outputs an error and allows the user to re-try inputting a valid file name.

(2) Word Counting.

Your program will parse the content by reading each word. Pay attention that the program is to treat each word as a string. The space between two words is treated as the separator. Since this project does not need you or the program to "understand" the word, there is no exception to treat words in different ways. For example, "Coca Cola" is treated as two words, "Coca" and "Cola", not as one word.

Your program will ignore all non-word expressions, such as special characters, numbers, in the frequency counting process. Words with only case differences are treated as the same word. For example, "A" and "a" are considered as the same word.

Every time the program reads a new word, add this word into an appropriate data structure (think: which data structure to use?), and store the frequency value to 1. If the program reads a word that has already been added into the data structure, update the frequency value for that word by adding 1 (i.e. +1). Finally, the program reads the last word in the document (not the last word in a paragraph) and finishes counting of the words.

(3) Result displaying.

From the data structure that the program has used and all stored data, including the words and frequency counts, your program will output the word frequency and its associated word, in a sorted way: the No.1 frequent word is displayed first, then No.2 frequent word. The frequency is displayed first, followed by the word. Look at the example:

```
Frequency - Word
100 - "the"
86 - "a"
41 - "of"
…
```

Hints:

(1) You may need to write decision and control structures in the program. Moreover, you may consider the use one or more of the following functions or methods: open(), split(), sorted(), lower(), upper(), isalpha().
(2) When you sort the data in a descending order, consider to include a parameter when you use a built-in function. You can find the information on how to embed such a parameter from the function’s official "HOW TO" manual.
(3) How to process the "block" that contains both words and characters is a challenge. For example, In the sentence:

I like eating vegetables. What about you?

Your program should consider "vegetables" and "you" as the words, instead of "vegetables," and "you?". Note that the need to process the "block" is not limited to commas and question marks. A possible way to deal with the problem is to make a replacement of all punctuation marks, symbols, and digits.

You absolutely CANNOT use or modify any existing code of "Word Frequency Counter" or any similar program, which can be found from any textbook or online, to complete this project assignment. You must write the program line by line by yourself.

Your program's input or output does not need to be exactly the same as what is specified in the program design. However, your program must:
(1) handle the user input of a valid or invalid file name successfully,
(2) correctly count each word's frequency and store all values, and
(3) give formatted output according to the requirement.

Submission instruction

After you have completed the project, upload and submit the Python source code file *Project02.py* to eLC. Always double check that your submission was successful on eLC.

Grading

A score between 0 and 5 will be assigned.

1. The program can successfully process a valid file input (0.5) and an invalid file input (0.5). (1 point)
2. The program is able to calculate the frequency of each word correctly. (1 point)
3. The program is able to store all words and their frequency in the memory using a reasonable data structure. (1 point)
4. The program outputs the "word frequency – word" in a sorted format. (1 point)
5. Only a single source code file is submitted and no other file is submitted (0.5) and the entire Python program can be executed without any additional error (0.5). (1 point)
6. A bonus point will be given for correctly separating actual words and special characters by keeping the word part only for counting. (+1 point)

Special notice regarding the submission:

Late submission penalty. Points will be deducted from the original grade. If your submission is after the posted deadline…

(1) within 24 hours: -2
(2) between 24 hours and 48 hours: -3
(3) between 48 hours and 72 hours: -4
(4) after 72 hours: project will not be accepted.