Radar Rasmussen
CS256
Lab Assignment 4
2/29/2024

Lab Assignment 6 Design Document

Introduction *(1-5 sentences, starting with lab 2)*
The program, “Word Frequency,” analyzes any text set (The Phantom Tollbooth in this case) and outputs the fifty most common words. It will exclude all stop words such as conjunctions, pronouns, prepositions, and articles. The program will separate each word in the text, count the number of occurrences of each unique word, filter out the excluded words, and store them in a data set. The output will have each word and their frequency starting from most frequent to least.

Functional Requirements *(starting with lab 3)*
- The program will take input text from the ‘get_text()’ function in the ‘phantom_tollbooth.py’ file.
- It will split and store each unique word in the text.
- Stop words will be excluded from the data set.
- The program will find the fifty most common words based on the highest frequencies.
- Will generate and print out each word and its frequency.

Design Requirements *(starting with lab 4)*
Data:
- Input text from the ‘get_text()’ function
- Counts of words/their frequencies (dictionary)
Loops:
- Iterate over each word in the text
- Counting words and update in the dictionary
- Filters to exclude/remove stop words
Conditions:
- Checking if a word is meant to be excluded
- Checking for words similar to stop words (there, they’re, their, there’s, etc.)
- Checking for errors

Testing Predictions Results *(starting with Lab 6 and on)*
Tested inputs:
- Input text without excluded words.
- Input text with excluded words.
Predictions:
- Should count and report the fifty most common words in the text.
- Stop words should not appear in the output.
Actual results:

- Program filtered out excluded words correctly.
- Words and frequency were counted correctly.
- Frequencies sorted from largest to smallest.
- Outputted the 50 most common words in The Phantom Tollbooth.
- Has some extra UX design.

Reflection and Questions

When designing this program, I first tried deconstructing it by asking: how can I remove words from a string? I recalled when I used tagcloud.com and wondered how it worked. What code did they use, and how did they achieve it so well? I started with writing a function to remove punctuation. It worked great, so I used the same formula for the other functions with minor adjustments. For convenience, I also set the book text to all lowercase automatically, that way I didn't have to add lower() to every other line. I was able to successfully exclude unwanted words, turn all the unique ones (and their frequencies) into a dictionary, reverse order them, and print out the top 50 words with some extra formatting.

I believe that I achieved 100% completion of the program. The only thing I missed was the odd hyphens ('—') that seemed to attach to the words. They would prevent the removeal of some stop words, but I couldn't figure out how to get rid of them. When I copy pasted it to the punctuation function, it just broke. I spent 10 minutes on it before giving up and realizing they won't be shown anyway. Oh well. I think that I understood everything that I was given; I completed the bulk of the code within a few hours without much help (how? I don't really know). I only used AI to ask questions and alphabetize my word lists because I couldn't be bothered.