Homework 1: Programming Prose

1. Introduction

Now that you have learned the fundamentals of clean code and the importance of writing to an engineer, lets combine the two! In this assignment you will learn about a new book, programmatically analyze the text, and make a brief presentation.

2. Setup

Create a new **private** git repository inside a **hw1/** directory. Initialize the repository and add your instructor as a collaborator.

3. Background

Your computer science background allows you to approach many academic problems uniquely. This assignment combines uses computer science to explore literature in a refreshing way.

4. Your Assignment

This homework consists of three parts. First, you will select one book from Project Gutenberg's Top 100 List. Secondly, you will write a python program to collect data about the text that would otherwise be monotonous by hand. Third, you will prepare a presentation on your book and your findings.

4.1 Select and Understand a Text (10 pts)

Project Gutenberg[1] offers over 50,000 free ebooks. The Top 100 List can be found here.[2] You are free to choose any book on the Top 100 List, but I recommend one that is a single novel as opposed to a collection of short stories. Additionally, your selection must be unique; no other students in the class should select the same book as you.

[1] https://www.gutenberg.org/

[2] https://www.gutenberg.org/browse/scores/top

Write a summary of the text you have selected. Feel free to use online sources to learn more about the plot of the book you've selected, **but you must list your sources** at the bottom of your summary. This is a writing exercise. Plagiarism will not be tolerated.

4.2 Write your text parser (10pts)

In order to analyze and draw conclusions from the text you selected more efficiently, you'll need to programmatically read and parse the ebook. Download a **.txt** version of your selected text and include it in your **hw1/** directory.

Each ebook begins and ends with various information regarding Project Gutenberg. The book officially begins with a line that starts with the following:

> **** START OF THIS PROJECT GUTENBERG EBOOK …*

And ends with a line that starts with the following:

> **** END OF THIS PROJECT GUTENBERG EBOOK …*

Create a method **get_ebook_content(file)** that opens the text file **file** and returns everything (including whitespace and special characters) between these two boundaries.

4.3 Analyze word frequency (20pts)

Write a script which returns the top 20 words (**a**nd their frequencies) for each chapter in the book and for the book as a whole. More specifically, create a method **count_word_freq(text)** that takes a string parameter **text** and returns a dictionary of words and their respective occurrences in **text.** For example:

```
>>> count_word_freq(“Luke, I am your father! Luke!”)
{‘Luke’:2, ‘I’:1, ‘am’:1, ‘your’: 1, ‘father’:1}
```

Pick at least 10 words specifically relevant to your chosen text and track their frequency over each chapter. Record this data for Part 5.

Please do not count any portion of the Project Gutenberg file besides the original text of the ebook, and don't differentiate words based on casing or an beginning/ending punctuation (for example: “Scrooge”, “scrooge”, “Scrooge!”, “Scrooge,”, and “Scrooge?” should all be treated the same.)

4.4 Incorporating URL Requests (10pts)

Each book available through Project Gutenberg is available in plaintext an endpoint. For example, observe the appropriate endpoint for Pride and Prejudice found here.[3] Add functionality to your program to analyze word frequency using a url endpoint parameter. More specifically, create a method **fetch_ebook(url)** which takes in a URL as a string and returns the plaintext for your selected text. This method should not return the entire payload (HTTP response) but rather a subset.

HINT: I recommend looking into two Python modules. The first of which is *urllib2*, which will help you make your HTTP request. The second is *BeautifulSoup*, which will allow you to easily parse the returned HTML. It is unlikely that you will have these modules preinstalled on your machine, in which case you'll need to install them yourself. Here[4] is a good place to start.

Do not replace or remove the functionality you used in 4.2 and 4.3. Ideally, your code is modularized enough that implementing 4.4 has minimal effects on the rest of your script.

5 Prepare to Present (10pts)

Prepare a brief presentation about your selected text, your findings about popular words and the frequencies of specific words throughout the text (4.3). Because you are familiar with the plot of the text you selected, your presentation should include conjecture about why certain words are popular and any noticeable trends in the words you chose to track over multiple chapters.

6 Feedback (5pts)

Respond to each of the following questions in a file called **feedback.txt:**

1. What was the most frustrating part of this assignment?
2. What was the most boring part of this assignment?
3. What was your favorite part of the assignment?
4. How long did it take for you to complete this assignment take?
5. How would you improve this assignment?

---

[3] http://www.gutenberg.org/cache/epub/1342/pg1342.txt
[4] https://docs.python.org/3/installing/

7 Extra Credit: Count Words in Place (5pts)

Modify your program to count word frequencies as your read lines incrementally from the filesystem. In other words, when opening **your_ebook.txt**, do not read the entire book into a string buffer before operating on it.