Special Course in Software Engineering – Autumn 2023

Python programming exercise 3 (MAX: 7 points)

For this program, you will need Python's **NLTK (Natural Language Tool Kit)** suite of libraries, which helps you process and analyze raw text.

More importantly, you are **HIGHLY ENCOURAGED** to use ***CoPilot*** to help you work on this exercise. A quick demonstration has been provided. If you feel that is inadequate, then we recommend that you do your own research on the use of ***CoPilot*** and then use it in the way you find suitable.

Write a Python program that…

- Reads the ***Moby Dick*** file from the ***Gutenberg*** dataset
- Processes the text from the above ***Moby Dick*** file as follows:
    1. **Tokenization:** The program ***tokenizes*** the entire book. Meaning, it splits the entire book into tokens (words)
    2. **Stop-words filtering**: Filters out the ***stopwords*** from the above tokens. Stopwords are quite common English words that are typically removed from a text before analyzing them. These are the words that do not add much to the text. For example, *'the', 'a', 'an', 'is', 'in',* etc.
    3. **Parts-of-Speech (POS) tagging:** Using the tokens, the program tags the different parts of speech (*e.g., noun, verb, adjective, adverb, preposition, etc.*) for each word.
    4. **POS frequency**: The program then counts and displays the 5 most common parts of speech (e.g., noun, verb, adverb, adjective, etc.) and their total counts (frequency).
    5. **Lemmatization***:* Using the *pos-tagged* tokens, the program then lemmatizes the **top 20 tokens**. Lemmatization is the process of converting a given word to its root word. For example, the root of ***"running", "runs", "runner",*** and *"**ran**"* is **"run"**. Similarly, the root of ***"singing", "singer", "sings", "sang",*** and ***"sung"*** is **"sing".**
    6. **Plotting frequency distribution***:* At the end, plot a bar chart to visualize the frequency of POS. Meaning, all the parts of speech (e.g., noun, verb, adjective, adverb, etc.) and their total occurrences are plotted as a bar chart.
- Commit your exercise solution to your GitHub account
- Enter the URL of your exercise repository in the **"Exercise 3 Submission Box"**

Special Course in Software Engineering – Autumn 2023

The following is not a mandatory part of the exercise, but it's for those who want something more challenging. Doing this optional part of the exercise successfully can give you **5 bonus points**.

- For the ***Moby Dick*** text, perform **sentiment analysis**.
    - Sentiment analysis is basically a way to analyze how the overall tone of the text is. In simplest terms, it tells you if the text is positive or negative. Sentiment analysis is a very common analysis done on text, especially online reviews.
- At the end, the program should display an ***average sentiment score*** and state if the overall text sentiment is positive or negative.
    - If the ***average sentiment score*** is above 0.05, then the overall text sentiment is positive, or else it's negative.

NOTE: We are aware that solutions for the above exercise can be found quite easily online. It is okay to look for ideas and syntax and some help with pieces of your code. However, please try to put those pieces together to build your own code. Avoid using an already available solution.