Data Extraction and NLP

Variables :

1. POSITIVE SCORE
2. NEGATIVE SCORE
3. POLARITY SCORE
4. SUBJECTIVITY SCORE
5. AVG SENTENCE LENGTH
6. PERCENTAGE OF COMPLEX WORDS
7. FOG INDEX
8. AVG NUMBER OF WORDS PER SENTENCE
9. COMPLEX WORD COUNT
10. WORD COUNT
11. SYLLABLE PER WORD
12. PERSONAL PRONOUNS
13. AVG WORD LENGTH

Explaining how you approached the solution

1. Gathering Data:

- Reads a list of URLs from an Excel file, where each URL has a unique ID.
- Loads two sets of words, positive and negative, to help with sentiment analysis later.
- Also loads a collection of common words called "stop words" that are usually ignored for analysis.

2. Fetching Content:

- Goes through each URL in the list:
- Fetches the website's HTML content using a library called requests.
- Extracts the specific text it needs (like the title and main article text) using a library called BeautifulSoup. This part may need adjustments depending on the website's structure.

3. Analyzing Text:

- **Sentiment Analysis:**
  - Counts the occurrences of positive and negative words in the text to gauge its overall sentiment (positive, negative, or neutral).
- **Text Complexity:**
  - Calculates factors like average sentence length, percentage of complex words (words with three or more syllables), and the Fog Index to measure how difficult the text is to read.

- **Other Text Features:**
    - Counts the average number of words per sentence, the number of complex words, the total number of words (excluding stop words), the number of syllables per word, the number of personal pronouns used, and the average word length.

4. Organizing Results:

- For each URL, creates a collection of information:
- The URL ID
- The URL itself
- The calculated sentiment scores
- The text complexity scores
- The other text features
- Then combines all those collections into a big table using a library called pandas.

5. Saving Results:

- At last saves that table of results as an Excel file for further analysis or reference.

Steps to run the .py file to generate output :

1. Install Python:

- Make sure the other computer has Python installed.

2. Install the Libraries:

- Open Jupyter Notebook or any other python platform on the other computer.
- Use the pip command to install the necessary libraries :
  pip install requests
  pip install beautifulsoup4
  pip install pandas
  pip install textblob
  pip install nltk

3. Transfer the Code and Files:

- Copy the Python code (the entire script) and any necessary files to the other computer.
- Contents inside the Folder -> MasterDictionary and StopWods

4. Run the Code:

- Open Jupyter Notebook or any other python platform and navigate to the directory where you saved the Python code file (the one with the .py extension).
- Click and press enter to open the python file.
- Now Execute the code.

5. Note:

- Make sure the file paths in the code (like the locations of Excel files) are adjusted according to the new location of the files on the other computer. If necessary, update the paths in the code before running it.

All dependencies that are required :

1. **requests**: This library helps fetch web content from URLs using HTTP requests.
2. **BeautifulSoup**: This library parses HTML content, making it easier to extract specific information from websites.
3. **pandas**: This library provides powerful tools for working with data in tables (DataFrames) and performing various calculations and manipulations.
4. **TextBlob**: This library offers functionalities for text processing, including sentiment analysis, tokenization, and cleaning text.
5. **nltk**: This library, along with its sub-modules like stopwords and word_tokenize, provides various functionalities for natural language processing tasks, including working with stop words, tokenizing text into words, and calculating syllable count.
6. **re**: This built-in Python library provides functions for regular expression matching, used for tasks like finding personal pronouns in the code.