Report on the Text Analysis Project

Introduction:

This project aims to perform a comprehensive text analysis on a collection of articles. The analysis involves extracting article titles and text, performing sentiment and readability analyses, and saving the results in a structured format. Below, I will explain the approach taken, the steps involved, and how to run the provided Python script.

Approach:

1. Data Extraction:

- Extracting Article Content: The `extract_article` function takes a URL, fetches the webpage content, and extracts the article title and text.
- Saving Articles: The `save_article` function saves the extracted article content to a local file for future reference.

2. Downloading Necessary Files:

- Downloading Stopwords and Master Dictionary: The `download_from_drive` function downloads necessary stopword lists and master dictionary files from Google Drive links.

3. Loading Stop Words and Master Dictionary:

- Stop Words: The `load_stop_words` function loads stopwords from the downloaded files.
- Master Dictionary: The `load_master_dictionary` function loads positive and negative words from the downloaded files.

4. Text Analysis:

- Cleaning Text: The `clean_text` function tokenizes the text, converts it to lowercase, and removes stopwords.
- Counting Syllables: The `count_syllables` function counts syllables in a word, which is crucial for readability metrics.
- Analyzing Text: The `analyze_text` function calculates sentiment scores, readability metrics, and other textual features.

5. Executing the Functions:

- The script processes each article URL provided in an Excel file (`Input.xlsx`), extracts and analyzes the text, and saves the results to an output Excel file (`Output Data Structure.xlsx`).

How to Run the Script:

To run the script and generate the output, follow these steps:

1. Ensure Dependencies are Installed:

   Make sure you have all necessary Python libraries installed. You can install them using pip:

   “pip install os gdown nltk pandas requests beautifulsoup4 openpyxl”.

2. Prepare the Input Data:

   Ensure you have the `Input.xlsx` file with the columns `URL_ID` and `URL`. This file should list the URLs of the articles you want to analyze.

3. Run the Script:

   Save the provided Python script to a file, for example, `text_analysis.py`. Then, execute the script:

   “python text_analysis.py”.

Dependencies Required:

- `os`: For file operations.

- `gdown`: To download files from Google Drive.

- `nltk`: For natural language processing tasks such as tokenization.

- `pandas`: For handling data frames and Excel files.

- `requests`: For making HTTP requests.

- `beautifulsoup4`: For parsing HTML content.

- `openpyxl`: For reading and writing Excel files.

Conclusion:

By following the steps outlined in this report, you should be able to perform a comprehensive text analysis on a collection of articles and generate a structured output with various metrics. This process involves downloading necessary resources, extracting and analyzing text, and saving the results for further use.