Approach to the Solution:

1. Understanding the Objective:
    - The objective is to analyze text from a set of URLs and extract various metrics. This involves web scraping, text cleaning, and analysis.

2. Modular Function Approach:
    - The script is divided into modular functions for better readability and maintainability.
    - Functions are created for extracting text from URLs, cleaning the text, and performing text analysis.

3. Data Storage:
    - The results are stored in a pandas DataFrame (output_data) for ease of handling structured data.

4. Error Handling:
    - The script includes error handling to capture and log any issues during text analysis.
    - It continues processing other URLs even if one URL analysis fails.

Running the .py File:

Dependencies:

- requests: To fetch web content.
- BeautifulSoup: For HTML parsing.
- pandas: To handle data in tabular form.
- nltk: Natural Language Toolkit for text processing.

Steps:

1. Install Dependencies:
    - Open a terminal and run the following command to install required packages:
      pip install requests beautifulsoup4 pandas nltk

2. Download NLTK Resources:
    - Uncomment and run the following line in the script to download NLTK resources:
      nltk.download('punkt')

3. Prepare Input Data:
    - Create an Excel file (Input.xlsx) with two columns: URL_ID and URL.
    - Populate it with the URLs you want to analyze.

4. Run the Script:
    - Save the script in a file, e.g., analyze_text.py.
    - Open a terminal, navigate to the script's directory, and run:

        python analyze_text.py

5. Review Output:
    - Check for text files named with url_id.txt for cleaned text.
    - Review the final analysis results in Output Data Structure.xlsx.

Notes:

- Ensure an active internet connection for URL fetching.
- Verify that you have necessary permissions to write files in the script's directory.
- If any issues occur, check the terminal/console for error messages.