**Steps to Approach the Solution**

1. **Extracting Article Content**:
   * **Libraries used**: requests, BeautifulSoup.
   * A function extract\_article\_content is created to fetch the article content from the URL.
   * Unwanted sections like headers and footers are removed from the content using BeautifulSoup’s decompose() method.
   * The main article content is extracted using a specific div class (td-post-content tagdiv-type), which is common for the articles in the provided URLs.
2. **Text Analysis**:
   * **Libraries used**: nltk (Natural Language Toolkit), pandas.
   * Various NLP techniques are used to analyze the article text:
     + Sentiment analysis using SentimentIntensityAnalyzer.
     + Complex word and sentence length analysis using word\_tokenize and sent\_tokenize.
     + Fog index calculation, which is a measure of readability.
   * Personal pronoun counts are calculated by checking the article text against a predefined list of pronouns.
   * The script uses a custom dictionary of positive and negative words to calculate sentiment-related scores.
3. **Saving Results**:
   * The script creates a directory for storing the text files and stores each article in a .txt format.
   * After processing all articles, the script saves the calculated metrics in an Excel file using the pandas library.

**How to Run the Python Script**

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

**1. Install Dependencies**

Make sure the following dependencies are installed in your Python environment:

*pip install pandas requests beautifulsoup4 nltk openpyxl*

The script requires these external libraries:

* **pandas**: Used for data handling and saving results into Excel.
* **requests**: For making HTTP requests to fetch articles.
* **beautifulsoup4**: For parsing and extracting article content from HTML.
* **nltk**: For Natural Language Processing, including tokenization, sentiment analysis, and stopwords handling.
* **openpyxl**: For saving data into an Excel file.

Additionally, nltk requires some additional datasets. You can download them by running the following commands inside a Python script or interpreter:

*import nltk*

*nltk.download('punkt')*

*nltk.download('vader\_lexicon')*

*nltk.download('stopwords')*

**2. Prepare the Files**

Before running the script, make sure the following files and directories are in place:

* **Input Data (Excel)**:
  + The script reads article URLs from an Excel file located at E:\\Blackcoffer\_assignment\\Input.xlsx. Ensure that the Excel file has at least two columns: URL\_ID and URL.
* **Dictionary Files**:
  + The dictionary files positive-words.txt and negative-words.txt must be present in the directory: E:\\Blackcoffer\_assignment\\MasterDictionary\\. These files should contain a list of words, one per line, classified as positive and negative respectively.
* **Text Files Directory**:
  + The script saves the extracted articles as text files in E:\\Blackcoffer\_assignment\\TextFiles. If the directory doesn't exist, the script will create it.

**3. Run the Script**

Once the required files are in place, navigate to the script’s directory and run the Python file:

***nlp\_assignment.py***

This will:

1. Extract articles from the provided URLs in the Excel file.
2. Process the article content and calculate various metrics like sentiment scores, sentence complexity, and readability.
3. Save the processed data into an Excel file (Output.xlsx) at E:\\Blackcoffer\_assignment\\Output.xlsx.

**Expected Output**

* **Text Files**: The script will save each article’s content in a .txt file under E:\\Blackcoffer\_assignment\\TextFiles.
  + Each text file will be named by its URL\_ID (e.g., 1.txt, 2.txt, etc.), and will contain the article's title and text.
* **Excel File**: An Excel file Output.xlsx will be generated in the specified directory, containing the calculated metrics for each article:
  + Columns will include URL\_ID, URL, and various text analysis metrics like:
    - **Positive Score**: Count of positive words in the article.
    - **Negative Score**: Count of negative words in the article.
    - **Polarity Score**: Difference between positive and negative words normalized.
    - **Subjectivity Score**: A measure of subjectivity based on positive and negative word occurrences.
    - **Average Sentence Length**: The average number of words per sentence.
    - **Percentage of Complex Words**: The proportion of words with more than two syllables.
    - **Fog Index**: A readability score indicating the complexity of the article.
    - **Average Words Per Sentence**: The average number of words in a sentence.
    - **Personal Pronoun Count**: The count of personal pronouns used in the article.
    - **Average Word Length**: The average length of words in the article.

**Troubleshooting**

* **Missing Dependencies**: If you get errors related to missing libraries, ensure you have installed all required dependencies using pip install.
* **File Paths**: Verify that the input Excel file and dictionary files are correctly placed in the specified paths.
* **Connection Issues**: If the script fails to fetch articles due to connection problems, check the internet connection or verify the URLs.