**Instructions Documentation**

**1.Approach to the Solution**

**Overview:**

The goal of this code is to analyse textual data from URLs and perform several linguistic calculations and sentiment analysis, such as word count, sentiment scores, syllable count, and more. The code reads input data from a list of URLs in an Excel file, processes the webpage text, and generates an output file with various calculated metrics.

**Steps Involved:**

1. **Reading Input Data:**
   * The URLs to be analysed are read from an Excel file (Input.xlsx), which contains a list of URLs.
2. **Stopwords Handling:**
   * The code reads a list of stopwords from text files inside the **Stopwords directory**. These stopwords will later be used to filter unnecessary words from the text.
3. **Loading Predefined Positive and Negative Words:**
   * Positive and negative words are preloaded from text files (positive-words.txt and negative-words.txt), which help in performing sentiment analysis.
4. **Fetching Web Data:**
   * The **support\_webdata** function fetches and processes the webpage content of each URL using **BeautifulSoup** to parse the HTML content. The function supports fetching only HTML pages (i.e., it checks the content type).
5. **Cleaning and Formatting Text:**
   * The content fetched from each URL is cleaned and formatted using regular expressions to remove unnecessary characters, symbols, and spacing.
6. **Word and Sentence Tokenization:**
   * The text is tokenized into words and sentences using NLTK's **word\_tokenize** and **sent\_tokenize** functions.
7. **Sentiment Analysis:**
   * Positive and negative words from the preloaded lists are used to compute sentiment scores by checking how many of these words appear in the text.
8. **Calculating Metrics:**
   * The code calculates various metrics for the text extracted from each URL in input, such as:
     + **Positive Score:** Assigned 1 for each word in word tokens if its present in pre defined positive words and added all values
     + **Negative Score:** Assigned 1 for each word in word tokens if its present in pre defined negative words and added all values
     + **Polarity Score:** Sentiment polarity based on positive and negative scores.
     + **Subjectivity Score:** Based on the number of positive and negative words.
     + **Average Sentence length:** Number of Words divided by Number of Sentences
     + **Percentage Complex Words:** Number of complex words divided by Number of words
     + **Fog Index:** Uses average sentence length and percentage of complex words to calculate readability.
     + **Complex Word Count:** Count of Complex words(words with more than two syllables)
     + **Word Count:** Count of cleaned words
     + **Syllables per Word:** Average syllables per word.
     + **Personal Pronouns:** Counts pronouns using Part-of-Speech (POS) tagging.
     + **Average Word Length:** Sum of total characters in each Word divided by Total number of words
9. **Saving Results:**
   * The results for each URL, including all calculated metrics, are saved to an output Excel file (**output.xlsx**).

**2.Steps to run .py file**

**1.Pre-requisites:**

* Install Python 3.x on your system.

**2. Install Dependencies:**

* Open a terminal or command prompt and install the necessary dependencies using the following command:
* ***pip install pandas nltk beautifulsoup4 lxml tqdm pyphen numpy openpyxl***

**3. Download NLTK Data:**

* Open a Python shell or run the following in the terminal to download the required NLTK data (**stopwords** and **punkt** for tokenization):
* ***import nltk***

***nltk.download('stopwords')***

***nltk.download('punkt')***

**4. Prepare the Input Files:**

* Ensure the following files are present in the same directory as **url\_analysis.py** script:
  + **Input.xlsx:** Contains the URLs to be processed.
  + **Master directory:** Contains the **positive-words.txt** and **negative words.txt** files for sentiment analysis.
  + **Stopwords directory:** Contains the stopword files to be used.

**5.** Run the **url\_analysis.py:**

* Navigate to the directory where your .py script and input files are stored.
* Run the Python script by executing the following command in your terminal:
* ***python url\_analysis.py***
* The script will process the URLs and generate an output file (output.xlsx) in the same directory.

**3.Required Dependencies**

**The following dependencies are required to run the script:**

* **pandas:** To handle dataframes and read/write Excel files.
* **nltk:** For tokenizing words and sentences, performing sentiment analysis, and handling stopwords.
* **glob:** To find the stopwords files in the directory.
* **BeautifulSoup (bs4):** For parsing HTML content fetched from URLs.
* **urllib:** For fetching webpage content.
* **SentimentIntensityAnalyzer:** For performing sentiment analysis.
* **tqdm:** For adding progress bars during dataframe operations.
* **pyphen:** For syllable counting in words.
* **numpy:** For efficient numerical operations.