# Text Analysis

This code performs text analysis on a collection of web pages to compute various linguistic features. The analysis includes sentiment analysis, readability metrics, and other derived variables. The code reads web page URLs from an Input.xlsx file, fetches the content of each URL, and saves the title and descriptions of each page in separate text files. Then, it performs the text analysis on these text files and saves the results in the Output.xlsx file.

## Prerequisites

Before running the code, ensure we have the following libraries installed:

- pandas: For handling data in tabular format.
- requests: For making HTTP requests to fetch web page content.
- beautifulsoup4: For parsing HTML content.
- nltk: The Natural Language Toolkit library for natural language processing.
- pyphen: For counting syllables in words.

We can install these libraries using pip: **pip install pandas**
**pip install requests**
**pip install beautifulsoup4**
**pip install nltk**
**pip install pyphen**

## Data Files

Ensure that the following data files are present in the same directory as the code:

- **Input.xlsx:** it contains the URL id and URLs.
- **Output Data Structure.xlsx:** this file specifies the output file format.
- **MasterDictionary/positive-words.txt**: A text file containing a list of positive words, one word per line.
- **MasterDictionary/negative-words.txt:** A text file containing a list of negative words, one word per line.
- **StopWords:** A collection of text files containing stop words. Each filename should start with “StopWords” and ends with ‘.txt’.the folder contains these files :
  StopWords_Auditor.txt
  StopWords_Currencies.txt
  StopWords_DatesandNumbers.txt
  StopWords_Generic.txt
  StopWords_GenericLong.txt
  StopWords_Geographic.txt
  StopWords_Names.txt

# Code Execution

- **Loading and Creating Master Dictionary:** The code starts by loading the positive and negative word lists from the MasterDictionary directory and creating a master dictionary with these words, along with their corresponding polarity values.

- **Loading Stop Words:** Next, the code loads stop words from various files in the StopWords directory and creates a tuple containing all the stop words.

- **Reading Input and Output Files**: The code reads two Excel files, Input.xlsx and Output Data Structure.xlsx, using the pandas library. The Input.xlsx file contains two columns: URL_ID and URL, where URL_ID is an integer identifier for each URL, and URL contains the web page URLs to analyze. The Output Data Structure.xlsx file contains the same URL_ID column and additional columns to store the computed text analysis variables.

- **Fetching Web Page Content:** The code fetches the content of each web page using the requests library and saves the title and descriptions of the page in separate text files within the **"URL_text_files"** directory.

- **Text Analysis Functions:**
  Two functions are defined for text analysis:
  - **clean_text:** This function tokenizes the input text, converts it to lowercase, and removes stop words and non-alphabetic tokens. The resulting cleaned tokens are returned.
  - **analyze_text:** This function takes the cleaned text and computes various text analysis variables, such as positive score, negative score, polarity score, subjectivity score, average sentence length, percentage of complex words, fog index, the average number of words per sentence, complex word count, word count, syllables per word, count of personal pronouns, and average word length.

- **Text Analysis and Data Saving:** The code iterates over each URL in the Output Data Structure.xlsx file, reads the corresponding text file created earlier, performs text analysis using the analyze_text function, and stores the computed values in the respective columns of the df_output DataFrame.

- **Saving Output:** Finally, the code saves the updated df_output DataFrame to an Excel file named **'output.xlsx'**.

# Usage Instructions

- Place the code file in a directory along with the required data files (***MasterDictionary/positive-words.txt, MasterDictionary/negative-words.tx***t, and stop word files in the ***StopWords*** directory).

- Create an ***Input.xlsx*** file with two columns:
    - **URL_ID**: An integer identifier for each URL.
    - **URL**: The web page URLs to analyze.
- Create an ***Output Data Structure.xlsx*** file with the following columns:
    - URL_ID: The same integer identifier for each URL as in the Input.xlsx file.
    - Additional columns for storing the computed text analysis variables.
- Execute the code using a Python interpreter or IDE.
- After execution, the computed text analysis results will be saved in the **output.xlsx** file.