**Overview of Approach**

**1.Data Extraction:**

* Read URLs from Input.xlsx.
* For each URL, fetch the article content using HTTP requests.
* Parse the article's HTML to extract the title and main content.
* Save the extracted text to individual text files named by URL\_ID.

**2.Text Analysis:**

* Tokenize the text into sentences and words.
* Remove stopwords and non-alphabetic tokens.
* Calculate various linguistic metrics like positive/negative scores, polarity, subjectivity, and more.
* Save the computed metrics to an Excel file named Output Data Structure.xlsx.

**Prerequisites**

**1.Dependencies:**

* pandas
* requests
* beautifulsoup4
* openpyxl
* nltk

**2.NLTK Data:**

* Download necessary NLTK data files using the NLTK downloader within the script.

**3.Files:**

* Input.xlsx: Contains the list of URLs and their corresponding URL\_IDs.
* positive-words.txt: Contains a list of positive words, one per line.
* negative-words.txt: Contains a list of negative words, one per line.

**4.Folder Structure:**

your-folder/

Input.xlsx

positive-words.txt

negative-words.txt

Script.py

**Steps to Run the Script**

**1.Download and Place Files:**

* Ensure Input.xlsx, positive-words.txt, and negative-words.txt are in the same directory as your\_script.py.

**2.Run the Script:**

* Open a terminal or command prompt.
* Navigate to the directory containing your script and input files.
* Execute the script.py:

**3.Output:**

* The script will create a directory named articles where it will save the extracted article text files.
* The script will generate an Excel file named Output Data Structure.xlsx containing the analysis results for each URL.