Approach –

* Copied one URL from the file input.xlsx to write the base code.
* Saved the web page of that URL to a html file for data crawling and then used beautiful soup to get title and content of the article and saved it as a .txt file.
* Opened the saved .txt file and used the NLTK library to tokenize words and sentences.
* Computed the variables as told.
* Once this process was completed and I got the desired variables for one text file, I defined functions for purposes of converting to html and text files and computing variables.
* Then looped through each URL from the input.xlsx file and called the defined functions on each URL to compute variables for each file and append the variables in each loop to an initially empty dataframe.
* Once the loop is executed, the dataframe is exported to a excel file.

How to run main.py –

* Install and import the required dependencies.
* In line 13, change the value of the variable ‘folder\_path’ to that of the folder which contains the file containing stop words.
* In line 34 and line 37, change the file path to that of the file containing positive words and negative words respectively.
* In line 185, change the file path to that of the excel file containing all the URLs.
* In line 192, change the file path to that of the folder where you want to generate html files of all URLs and add ‘{url\_id}.html’ at its end as in the original code.
* In line 193, change the first file path to that of the folder where you generated the html files of all URLs and change the second file path to that of the folder where you want to generate all the text files, as in the original code.
* In line 195, change the file path to that of the folder where you generated all the text files and add ‘{url\_id}.txt’ at its end as in the original code.
* In line 211, change the path to that of the excel file in which you want your output to be generated.
* Now, run the code and open the file in the path given in the previous step.

Dependencies required –

* beautifulsoup4==4.12.3
* bs4==0.0.2
* certifi==2024.2.2
* charset-normalizer==3.3.2
* click==8.1.7
* colorama==0.4.6
* et-xmlfile==1.1.0
* idna==3.6
* joblib==1.3.2
* nltk==3.8.1
* numpy==1.26.4
* openpyxl==3.1.2
* pandas==2.2.1
* python-dateutil==2.9.0.post0
* pytz==2024.1
* regex==2023.12.25
* requests==2.31.0
* six==1.16.0
* soupsieve==2.5
* syllapy==0.7.2
* tqdm==4.66.2
* tzdata==2024.1
* urllib3==2.2.1