Data Extraction

In [2]:
```
!pip install textstat
```

```
Collecting textstat
  Downloading textstat-0.7.3-py3-none-any.whl.metadata (14 kB)
Collecting pyphen (from textstat)
  Downloading pyphen-0.15.0-py3-none-any.whl.metadata (3.3 kB)
Downloading textstat-0.7.3-py3-none-any.whl (105 kB)
   ---------------------------------------- 0.0/105.1 kB ? eta -:--:--
   ------- -------------------------------- 20.5/105.1 kB 640.0 kB/s eta 0:0
0:01
   ---------------- ----------------------- 51.2/105.1 kB 525.1 kB/s eta 0:0
0:01
   ---------------------------------------- 105.1/105.1 kB 759.0 kB/s eta 0:0
0:00
Downloading pyphen-0.15.0-py3-none-any.whl (2.1 MB)
   ---------------------------------------- 0.0/2.1 MB ? eta -:--:--
   - -------------------------------------- 0.1/2.1 MB 2.2 MB/s eta 0:00:01
   --- ------------------------------------ 0.2/2.1 MB 2.3 MB/s eta 0:00:01
   ------ --------------------------------- 0.3/2.1 MB 2.5 MB/s eta 0:00:01
   -------- ------------------------------- 0.5/2.1 MB 2.6 MB/s eta 0:00:01
   ----------- ---------------------------- 0.6/2.1 MB 2.6 MB/s eta 0:00:01
   ------------ --------------------------- 0.7/2.1 MB 2.6 MB/s eta 0:00:01
   ------------ --------------------------- 0.7/2.1 MB 2.6 MB/s eta 0:00:01
   ------------ --------------------------- 0.7/2.1 MB 2.6 MB/s eta 0:00:01
   ---------------- ----------------------- 0.8/2.1 MB 2.1 MB/s eta 0:00:01
   ------------------ --------------------- 1.0/2.1 MB 2.2 MB/s eta 0:00:01
   ------------------- -------------------- 1.0/2.1 MB 2.1 MB/s eta 0:00:01
   ---------------------- ----------------- 1.2/2.1 MB 2.2 MB/s eta 0:00:01
   ------------------------ --------------- 1.3/2.1 MB 2.2 MB/s eta 0:00:01
   --------------------------- ------------ 1.4/2.1 MB 2.3 MB/s eta 0:00:01
   ------------------------------ --------- 1.6/2.1 MB 2.4 MB/s eta 0:00:01
   -------------------------------- ------- 1.7/2.1 MB 2.4 MB/s eta 0:00:01
   ---------------------------------- ----- 1.7/2.1 MB 2.3 MB/s eta 0:00:01
   ------------------------------------ --- 2.0/2.1 MB 2.5 MB/s eta 0:00:01
   ---------------------------------------- 2.1/2.1 MB 2.4 MB/s eta 0:00:00
Installing collected packages: pyphen, textstat
Successfully installed pyphen-0.15.0 textstat-0.7.3
```

In [3]:
```
import pandas as pd
import requests
from bs4 import BeautifulSoup
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
import textstat
import re
```

In [7]:
```
# Ensure necessary NLTK data files are downloaded
nltk.download('punkt')
nltk.download('stopwords')
```

```
# Load the input data
df = pd.read_excel(r"C:\Users\samru\Downloads\Input.xlsx")
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\samru\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\samru\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [8]:
```
df
```

Out[8]:

	URL_ID	URL
0	blackassign0001	https://insights.blackcoffer.com/rising-it-cit...
1	blackassign0002	https://insights.blackcoffer.com/rising-it-cit...
2	blackassign0003	https://insights.blackcoffer.com/internet-dema...
3	blackassign0004	https://insights.blackcoffer.com/rise-of-cyber...
4	blackassign0005	https://insights.blackcoffer.com/ott-platform-...
...	...	...
95	blackassign0096	https://insights.blackcoffer.com/what-is-the-r...
96	blackassign0097	https://insights.blackcoffer.com/impact-of-cov...
97	blackassign0098	https://insights.blackcoffer.com/contribution-...
98	blackassign0099	https://insights.blackcoffer.com/how-covid-19-...
99	blackassign0100	https://insights.blackcoffer.com/how-will-covi...

100 rows × 2 columns

In [9]:
```
# Function to extract text from a URL
def extract_text(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Extract the title and main content
    title = soup.find('h1').get_text()
    paragraphs = soup.find_all('p')
    content = '\n'.join([para.get_text() for para in paragraphs])

    return title, content
```

In [10]:
```
# Function to save text to a file
def save_text(url_id, title, content):
    with open(f'{url_id}.txt', 'w', encoding='utf-8') as file:
        file.write(title + '\n' + content)
```

In [16]:
```
# Extract texts and save them
for index, row in df.iterrows():
```

```
    url_id = row['URL_ID']
    url = row['URL']
    save_text(url_id, title, content)
```

In [17]:
```
def compute_positive_score(text):
    # Example: Implement logic to compute positive score
    positive_words = [...]  # list of positive words
    words = word_tokenize(text)
    positive_score = sum(1 for word in words if word in positive_words)
    return positive_score

def compute_negative_score(text):
    # Example: Implement logic to compute negative score
    negative_words = [...]  # list of negative words
    words = word_tokenize(text)
    negative_score = sum(1 for word in words if word in negative_words)
    return negative_score

def compute_polarity_score(positive_score, negative_score):
    # Example: Implement logic to compute polarity score
    polarity_score = (positive_score - negative_score) / ((positive_score +
    return polarity_score

def compute_subjectivity_score(text):
    # Example: Implement logic to compute subjectivity score
    words = word_tokenize(text)
    subjective_words = [...]  # list of subjective words
    subjectivity_score = sum(1 for word in words if word in subjective_words
    return subjectivity_score
```

In [18]:
```
# Other required functions to compute metrics...

# Perform text analysis
output_data = []
for index, row in df.iterrows():
    url_id = row['URL_ID']
    with open(f'{url_id}.txt', 'r', encoding='utf-8') as file:
        text = file.read()

    positive_score = compute_positive_score(text)
    negative_score = compute_negative_score(text)
    polarity_score = compute_polarity_score(positive_score, negative_score)
    subjectivity_score = compute_subjectivity_score(text)

    # Compute other metrics...

    output_data.append([url_id, positive_score, negative_score, polarity_scc
```

In [19]:
```
# Save the results
output_df = pd.DataFrame(output_data, columns=[
    'URL_ID', 'POSITIVE SCORE', 'NEGATIVE SCORE', 'POLARITY SCORE', 'SUBJECT
    # Add other columns...
])
output_df.to_excel('Output Data Structure.xlsx', index=False)
```

In [20]:
```
output_df
```

Out[20]:

	URL_ID	POSITIVE SCORE	NEGATIVE SCORE	POLARITY SCORE	SUBJECTIVITY SCORE
0	blackassign0001	0	0	0.0	0.0
1	blackassign0002	0	0	0.0	0.0
2	blackassign0003	0	0	0.0	0.0
3	blackassign0004	0	0	0.0	0.0
4	blackassign0005	0	0	0.0	0.0
...	...	...	...	...	...
95	blackassign0096	0	0	0.0	0.0
96	blackassign0097	0	0	0.0	0.0
97	blackassign0098	0	0	0.0	0.0
98	blackassign0099	0	0	0.0	0.0
99	blackassign0100	0	0	0.0	0.0

100 rows × 5 columns

In [ ]:
```
```