TEXT ANALYSIS

The project is aimed at extracting and analyzing text content from various URLs provided in an Excel file. It involves web scraping using Scrapy, a Python web crawling framework, to extract text data from web pages. The extracted data includes article titles, paragraphs, and list items, which are then processed to derive various linguistic and statistical metrics such as word count, average sentence length, polarity score, and subjectivity score. Finally, the analyzed data is stored in an Excel file for further analysis or reporting.

Technologies Used:

Scrapy: Chosen for its powerful and swift web scraping capabilities, asynchronous processing, and built-in support for parsing HTML and XML. It simplifies the task of extracting structured data from websites.

Pandas: Utilized for data manipulation and analysis. Pandas is well-suited for handling tabular data structures such as Excel files, making it convenient for processing and storing the extracted information.

NLTK (Natural Language Toolkit): Employed for natural language processing tasks such as tokenization, stop word removal, and syllable counting. NLTK provides various tools and algorithms for text analysis, making it suitable for deriving linguistic features from the extracted text.

As an avid reader and someone deeply interested in natural language processing (NLP), I embarked on a personal project to analyze and extract insights from online articles. The goal was to develop a tool that could efficiently parse through articles, extract key information, and provide valuable insights.

In summary, this project was a fascinating journey into the realms of web scraping and natural language processing, fueled by my passion for literature and data analysis. By leveraging Python, Scrapy, Pandas, and NLTK, I was able to create a robust tool for extracting insights from online articles. While the chosen technologies proved effective, exploring alternative approaches could offer further enhancements and possibilities for future iterations of the project.

INSTRUCTIONS

1. First, you need to set up a Scrapy project. If you haven't already installed Scrapy, you can do so via pip:

   ~$ pip install scrapy

2. Once Scrapy is installed, you can create a new project using the following command:

   ~$ scrapy startproject <article_extractor>

   Note: You can use any name instead of **article_extractor (Be careful to follow the below steps accordingly)**

3. Paste the given python file(text_analysis.py) in article_extractor > article_extractor > spider.
4. Paste the given '**requirement.txt**' in article_extractor > article_extractor > spider.
5. Paste the **MasterDictionary** folder in article_extractor > article_extractor > spider.
6. Paste the **StopWords** folder in article_extractor > article_extractor > spider.
7. Open terminal in article_extractor > article_extractor > spider
8. Install the required packages

   ~$ pip install -r requirements.txt

9. Append these below lines in the file article_extractor > article_extractor > **settings.py** for sequential output (in same order of input)

   CONCURRENT_REQUESTS = 1

   CONCURRENT_REQUESTS_PER_DOMAIN = 1

   CONCURRENT_REQUESTS_PER_IP = 1

10. Run the Scrapy project

    ~$ scrapy crawl article

11. The Location article_extractor > article_extractor > spider > ouput.xlsx contains the output

Note: The time may vary according to the size of the data.

DEPENDENCIES

fdg attrs==23.2.0
Automat==22.10.0
certifi==2024.2.2
cffi==1.16.0
charset-normalizer==3.3.2
click==8.1.7
constantly==23.10.4
cryptography==42.0.5
cssselect==1.2.0
et-xmlfile==1.1.0
filelock==3.13.2
hyperlink==21.0.0
idna==3.6
regex==2023.12.25
requests==2.31.0
requests-file==2.0.0
Scrapy==2.11.1
service-identity==24.1.0
six==1.16.0
tldextract==5.1.2
tqdm==4.66.2
Twisted==24.3.0
typing_extensions==4.10.0
tzdata==2024.1
incremental==22.10.0
itemadapter==0.8.0
itemloaders==1.1.0
jmespath==1.0.1
joblib==1.3.2
lxml==5.1.0
nltk==3.8.1
numpy==1.26.4
openpyxl==3.1.2
packaging==24.0
pandas==2.2.1
parsel==1.9.0
progress==1.6
Protego==0.3.0
pyasn1==0.5.1
pyasn1-modules==0.3.0
pycparser==2.21
PyDispatcher==2.0.7
pyOpenSSL==24.1.0
python-dateutil==2.9.0.post0
pytz==2024.1
queuelib==1.6.2
urllib3==2.2.1
w3lib==2.1.2
zope.interface==6.2