Blackcoffer Data Extraction and NLP Test Assignment

Approach

1) When working with data extraction from an external webpage, the most important part of the project is the scraping approach. So, I first went through the source HTML pages that I had to extract to perform NLP upon. Overall, I found two main sources that I could use to scrape the data from the webpage:

- All of the webpages contained an `<article>` tag that contained the contents of the webpage.
- Inside the `<article>` tag specifically was the `<div class="td-page-content">` section, that actually contained the article text.

Upon prototyping with both approaches, I chose to extract from the `td-page-content` div, as I found that when scraping the entire `<article>` section, the scraper would extract the footer sections as well, pulling in extraneous information that would skew the NLP results.

2) As for the choice of scraper, I went with `BeautifulSoup`, as that was the scraper I was most familiar with. I had never worked with `Scrapy` before, and I felt that `Selenium` was not particularly well-suited for this particular task, as `Selenium` is mainly aimed towards browser automation, browser interaction and testing. `BeautifulSoup` along with `requests` I felt was perfect for this usecase.

3) Upon scraping the webpages using `BeautifulSoup`, I still found that the scraper would pull in extraneous information due to inconsistencies in the article structure itself from article to article, so I restricted the scraper to get text only from the tags: `<p>, <h1>, <h2>, <h3>` which are the tags for paragraphs, and headings. I also restricted the scraper to stop extracting text once it found the stop phrase `"Contact Details"` as everything after that phrase was extraneous details.

4) I also loaded the custom stopwords and sentiment analysis words from the directories `StopWords/` and `MasterDictionary/`. I had to perform stripping upon the stopwords, as some of them had extraneous information separated by a `|` symbol. As an example, the `StopWords_Currencies.txt` list has entries like `BAHT | THAILAND`, where `BAHT` is the actual stopword, and `| THAILAND` is extraneous information. I also converted the stopwords and sentiment specifying words to lowercase, as some of them were in uppercase. Finally, I accounted for errors in their encoding, and set in logic to ignore as many errors as possible, and convert the files to `latin-1` encoding, if the errors were not ignorable and resulted in an exception.

5) I then conducted the Natural Language Processing (NLP) analysis with

the help of the formulae given in the `Text Analysis.docx` as part of the assignment. I utilised the `NLTK` library to tokenise the input text into words and senteneces, and applied the given formulae. I also used the `textstat` library, as it contained methods to calculate the Gunning Fog Index and to calculate the syllable count.

6) Finally, I exported the results to an Excel spreadsheet the same way I imported the `Input.xlsx` spreadsheet, with the help of the `Pandas` library and the `openpyxl` extension to work specifically with Excel files.

Instructions

I assume that the environment that the program is going to be run on is a Linux environment, as that is the platform that I use, and the instructions are therefore tailored to being run on Linux. For alternate platforms like Windows and MacOS, the analogues of the commands mentioned will need to be searched online.

- To run the `analysis.py` file, first you must ensure that the following directory structure is maintained:

```
MasterDictionary/
    positive-words.txt
    negative-words.txt
StopWords/
    StopWords_Auditor.txt
    .
    .
    .
    StopWords_Names.txt
Input.xlsx
requirements.txt
analysis.py
```

- Next, assuming you are working from the above directory and its structure, you must run the command `python -m virtualenv venv` and `source venv/bin/activate`, to create a virtual environment to make managing the dependencies easier.
- Next, you must run the command `pip install -r requirements.txt` in order to install the dependencies.
- Finally, you must run the program itself like so: `python analysis.py`

Dependencies

The dependencies for running the program are: `pandas`, `requests`, `bs4`, `textstat`, `nltk`, `openpyxl`.

Submitted By:


```
# Name: Aseem Aniket Athale
# Email: athaleaseem@gmail.com
```