Sentiment Analysis and Text Feature Extraction of Online Content

Abstract

The vast amount of online content presents challenges in understanding its emotional tone (sentiment) and level of complexity (readability). This can hinder effective communication, information processing, and decision-making for users.

This project aims to develop a system for analyzing online content based on sentiment and text features, including readability metrics, complexity measures, and language usage patterns. By providing insights into the tone and complexity of online texts, this system can help users make informed decisions, improve content creation, and facilitate effective communication.

Approach

1. Import libraries and mount Google Drive: The script starts by importing necessary libraries like requests, BeautifulSoup, pandas, nltk, and os. Additionally, it mounts Google Drive to access data files.
2. Read input data: It reads the URL list and URL ID information from an Excel spreadsheet named "Input.xlsx".
3. Extract content from URLs: For each URL, it attempts to:
    - Fetch the content using the requests library and BeautifulSoup.
    - Extract the title (if available) and text content from paragraphs.
    - Write the extracted title and content to a separate text file.
4. Preprocess text: It performs the following text preprocessing steps:
    - Downloads NLTK resources for tokenization and stopword removal.
    - Loads stop words, positive words, and negative words from specified directories.
    - Loops through text files and performs the following for each file:
        - Reads the text content.
        - Removes punctuation and tokenizes the text.
        - Filters out stop words.
5. Perform sentiment analysis: It calculates the following for each document:
    - Number of positive and negative words.

    - Polarity score and subjectivity score.
6. Calculate text features: It calculates the following metrics for each document:
    - Average sentence length.
    - Percentage of complex words.
    - Fog index.
    - Complex word count.
    - Word count.
    - Average syllable count per word.
    - Number of personal pronouns.
    - Average word length.
7. Prepare output data: It reads the "Output Data Structure.xlsx" spreadsheet which defines the structure for the output data.
8. Write output data: It iterates through calculated variables and assigns them to respective columns in the output DataFrame.
9. Save output data: Finally, it saves the analyzed data to an output CSV file named "Output.xlsx".

How to Run the Script

1. Save the script as a .py file.
2. Ensure you have the dependencies installed (run `pip install requests beautifulsoup4 pandas nltk` if not installed).
3. Replace the directory paths defined in the script (`dir`) with the actual paths where your files are stored.
4. Ensure you have the following files in the specified directories:
    - `Input.xlsx`: containing the input URL list
    - `Output Data Structure.xlsx`: defining the output data structure
    - `/StopWords`: containing stopwords text files
    - `/Dictionary`: containing positive and negative words text files with the same file name as it was given
5. Open a terminal window and navigate to the directory containing the script.
6. Run the script using the command `python <script_name>.py` or Import the script to Colab and run it.
7. The script will generate an output CSV file named "Output.xlsx" containing the analyzed data.

Dependencies

- `requests`: for fetching webpages
- `beautifulsoup4`: for parsing HTML content
- `pandas`: for data manipulation and analysis
- `nltk`: for text processing and tokenization
- `os`: for file I/O operations
- `re`: for regular expressions and pattern matching

Note

If the below error occurs, please re-run the code. This error happens sometimes in Colab since we are creating a new folder `/TextFile` using code, it is taking some time to be created and it is already being accessed.

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-9-abf15a3c9351> in <cell line: 32>()
     53     # Write title and content to a text file
     54     file = dir + '/TextFile/' + str(url_id) + '.txt'
---> 55     with open(file, 'w') as f:
     56         f.write(title + '\n' + content)
     57

FileNotFoundError: [Errno 2] No such file or directory:
'/content/drive/MyDrive/BC/TextFile/blackassign0009.txt'
```