Text Analysis Tool

A. Objective

The objective of this project is to extract textual data articles from given URLs, perform text analysis, and compute various variables as specified in the "Text Analysis.docx" file.

B. Data Extraction

Input

The input data is provided in the "input.xlsx" file, containing a list of URLs. The article text is extracted from each URL using Python programming, utilizing the requests library to fetch data and BeautifulSoup for HTML parsing. The extracted articles are then saved in text files, named after their respective URL_IDs.

Data Analysis

Textual analysis is performed on each extracted article, calculating variables outlined in the "Text Analysis.docx" file. Python programming is used for data analysis, including NLTK for text processing and custom functions for calculating complex words, syllables, and personal pronouns. Sentences are tokenized using NLTK's sentence tokenizer.

C. Variables

The following variables are computed for each article:

1. POSITIVE SCORE
2. NEGATIVE SCORE
3. POLARITY SCORE
4. SUBJECTIVITY SCORE
5. AVG SENTENCE LENGTH
6. PERCENTAGE OF COMPLEX WORDS
7. FOG INDEX
8. AVG NUMBER OF WORDS PER SENTENCE
9. COMPLEX WORD COUNT
10. WORD COUNT
11. SYLLABLE PER WORD
12. PERSONAL PRONOUNS
13. AVG WORD LENGTH

D. Output Data Structure

The output is saved in the exact order as specified in the "Output Data Structure.xlsx" file. The output variables include all input variables, POSITIVE SCORE, NEGATIVE SCORE, POLARITY SCORE, SUBJECTIVITY SCORE, AVG SENTENCE LENGTH, PERCENTAGE OF COMPLEX WORDS, FOG INDEX, AVG NUMBER OF WORDS PER SENTENCE, COMPLEX WORD COUNT, WORD COUNT, SYLLABLE PER WORD, PERSONAL PRONOUNS, and AVG WORD LENGTH.

E. Usage

To generate the output, follow these steps:

1. Clone the repository.
2. Install dependencies using the following command:
   $$pip\ install\ -r\ requirements.txt$$
3. Run the Python script:
   $$python\ text\_analysis.py$$

F. Dependencies

- requests
- beautifulsoup4
- nltk

Make sure to have these dependencies installed before running the script.

G. Approach

Stopword Set:

Created a stopword set from both the NLTK stopword corpus and the provided stopword files. Added string.punctuation to the stopword corpus.

Tokenization and Cleaning:

Tokenized the data using NLTK's word_tokenizer with the 'punkt' model. Cleaned the text by removing stopwords.

Positive and Negative Word Sets:

Created sets of positive and negative words from the given files.

Custom Functions:

1. *Count_Complex_Words*: Defined a function (count_complex_words) to find the number of complex words. Takes a list of words or a string as input and returns the count of complex words.

2. *Count_Syllables*: Implemented a function (count_syllable) to find the number of syllables in each text. Takes a list of words or a string as input and returns the count of total syllables.

3. *Count_Pronouns*: Developed a function (count_pronouns) to count personal pronouns. Takes a string or a list of words and returns the count of pronouns.

Sentence Tokenization:

Used NLTK's sent_tokenizer to obtain sentences for counting the total number of sentences in the text files.

Data Cleaning:

Removed URLs (36, 49, 14, 20, 29, 43, 83, 84, 92, 99, 100) due to errors during connection, status code issues, or inability to parse titles and content.

DataFrame and Output:

Read the given output structure and entered the data variables into a dataframe.
Created a new "Output.csv" file to store the processed data.

The final output is saved in "**Output.csv**."

Feel free to reach out for any questions or improvements!

Contact :- adityasharma0100@gmail.com