Text Processing with Web Scraping

Text Processing with Web Scraping" involves utilizing web scraping techniques to extract and process textual information from web pages. Web scraping is a method of automatically gathering data from websites by parsing and extracting relevant content.

Web scraping typically involves using tools and libraries to navigate web pages, extract HTML content, and isolate relevant text. Once the data is obtained, it can be further processed using various text processing techniques and algorithms to derive meaningful insights or perform specific tasks.

Objective of the Assignment:

The objective of this is to Scrape a Wikipedia webpage about Alexander the Great and Summarize the content while preserving the original headings and sections.

Dataset Generation:

The data is taken from the Wikipedia webpage "https://en.wikipedia.org/wiki/Alexander_the_Great".

Important packages used:

I used packages like web scraping techniques to collect data from the specified Wikipedia page. The tool to be used is BeautifulSoup for Retrieve both the text and headings, including sub-headings from the Wikipedia page.

Extraction of Data:

Using BeautifulSoup the data can be extracted and used re package, which is used for text pattern matching and manipulation.

Text Cleaning:

Text Cleaning:

The text is first cleaned to remove numbers, extra spaces, symbols, and brackets. This ensures that the text is in a clean and standardized format for further processing.

Text Preparation:

The cleaned text is then passed through a PlaintextParser which prepares the text for analysis. The Tokenizer('english') is used to tokenize the text into sentences or words, making it suitable for summarization.

Extractive Summarization:

The LsaSummarizer is employed for extractive summarization, which is a technique to extract the most important sentences from the text to form a summary.

Stop Words and Stemming:

Stop words (common words like 'and', 'the', 'is', etc.) are removed to focus on the more significant words. Stemming is also applied to reduce words to their base or root form.

Generating the Summary:

The LsaSummarizer generates the summary by selecting the most important sentences from the prepared text, based on their semantic relevance and importance.

The number of sentences to include in the summary is specified (e.g., 2 in this case), which can be adjusted based on requirements.

Returning the Summary:

The function returns a string that represents the summary, which consists of the selected sentences extracted from the input text.

Text Summarization:

Generate concise summaries for the text under each section while keeping the original headings intact.

Aim for coherence and accuracy in the summaries.

Printing the Text:

It pairs the cleaned headings with summaries and displays each heading in bold followed by its summary for clear presentation and readability.