

**Academic Year 2023-24**

**Subject: Natural Language Processing**

**Project Report**

**Topic: Text Analysis**

|  |  |
| --- | --- |
| **Submitted By :**  **Mridul Singh**  **SAP ID: 500094181**  **Batch: Devops B1 H** | **Submitted To:**  **Dr. Touseef Iqbal**  **School of computer Science** |

**Abstract**

This project focuses on automating text analysis through Natural Language Processing (NLP) techniques. The primary objective is to extract textual data from provided URLs and perform a comprehensive analysis to derive various metrics such as positive and negative sentiment scores, average sentence length, Fog Index, average word count per sentence, word count, syllables per word, complex word count, and average word length. The input data, consisting of URL\_IDs and corresponding URLs, is processed using web scraping techniques. Text preprocessing is then applied to clean and tokenize the extracted text. Through the implementation of various NLP algorithms, including sentiment analysis and readability metrics calculation, the project aims to provide insightful analysis of the text data. The results are presented in a structured format, facilitating easy interpretation and further analysis. This automation streamlines the text analysis process, enabling efficient extraction of valuable insights from textual data.

**Introduction**

In an era dominated by vast amounts of textual data available online, the need for efficient analysis techniques has become paramount. Natural Language Processing (NLP) offers powerful tools to extract insights from text, aiding decision-making processes across various domains. This documentation presents a project focused on automating text analysis through NLP techniques. The project's objective is to extract textual data from URLs provided in an input .xlsx file and perform comprehensive text analysis to derive various metrics such as sentiment scores, readability indices, and word statistics. By leveraging web scraping and NLP algorithms, this project aims to streamline the process of extracting valuable insights from textual data. Through the implementation of robust methodologies and algorithms, the project aims to provide a systematic approach to text analysis, facilitating informed decision-making and deeper understanding of textual content. This documentation outlines the methodology, implementation details, and results of the project, providing a comprehensive guide to automated text analysis using NLP techniques.

**Python Library used in project**

* **NLTK** is an open-source library that provides tools for tasks such as classification, stemming, tagging, parsing, semantic reasoning, and tokenization.
* **Pandas** is an open-source Python library that provides high-performance, easy-to-use data structures and data analysis tools for working with structured data.
* **Beautiful Soup** library is used for parsing HTML and XML websites or documents. It is mainly used for web scrapping.
* **Pyphen** is a Python library that is used for hyphenating text. It is a pure Python module that does not require any external dependencies.
* **Requests** module allows you to send HTTP requests using Python. The HTTP request returns a Response Object with all the response data (content, encoding, status, etc).

**Parameters for Text Analysis**

**Data Collection:**

* Read the input .xlsx file to get the URL\_ID and URLs.
* Use a web scraping library like BeautifulSoup or Scrapy to extract textual data from the URLs.

**Text Preprocessing:**

* Remove HTML tags, punctuation, special characters, etc.
* Tokenize the text into words and sentences.
* Convert text to lowercase for consistency.

**Text Analysis:**

* **Positive Score**: This score is calculated by assigning the value of +1 for each word if found in the Positive Dictionary and then adding up all the values.
* **Negative Score**: This score is calculated by assigning the value of -1 for each word if found in the Negative Dictionary and then adding up all the values. We multiply the score with -1 so that the score is a positive number.
* **Average Sentence Length** = the number of words / the number of sentences
* **Complex words**are words in the text that contain more than two syllables (is a part of a word that contains a single vowel sound and that is pronounced as a unit).
* **Percentage of Complex words** = the number of complex words / the number of words
* Analysis of Readability is calculated using the Gunning Fox index formula
* **Fog Index** = 0.4 \* (Average Sentence Length + Percentage of Complex words)
* (Fog index describes the readability of the document.)
* **Average Number of Words Per Sentence =**the total number of words / the total number of sentences.
* **Syllable Count Per Word**: A syllable is a part of a word that contains a single vowel sound and that is pronounced as a unit. So, for example, exclamation has 4 syllables which are ex, cla, ma, tion. And another example "book" has only one syllable. It is calculated with the help of NLTk and pyphen library.
* **Average Word Length** = Sum of the total number of characters in each word/Total number of words

**Output:**

* Write the results to a .csv file. Each row could represent a URL\_ID with columns for the extracted metrics.

**Conclusion**

In conclusion, this project presents a robust framework for automated text analysis utilizing Natural Language Processing (NLP) techniques. The primary aim was to extract textual data from provided URLs and conduct comprehensive text analysis to derive various metrics such as sentiment scores, readability indices, and word statistics. Overall, this project underscores the power of NLP in unlocking insights from textual data and highlights its potential for driving impactful analysis and decision-making processes.