Text Analysis

Web Scrapping & Text Analysis

Approach Report

Submitted By:

Suzal Kachhadiya

Submitted To:

Blackcoffer

Date:

14 November 24

Project Description:

This project implements a comprehensive text analysis pipeline that automates the processing of web content from provided URLs. The goal is to extract valuable insights through various natural language processing (NLP) techniques, including sentiment evaluation, readability metrics, and linguistic feature extraction.

The pipeline follows a modular architecture, enabling scalable processing of multiple URLs, independent analysis modules that can be maintained and updated separately, and an efficient workflow from data extraction to analysis. The key features of the pipeline include automated web scraping and text preprocessing, a sentiment analysis module with custom dictionaries, readability assessment, and an in-depth text statistics module covering measures like word count, sentence length, syllables, and personal pronoun usage.

The project is built using a modular Python codebase. This means the different components of the pipeline are organized and maintained separately, making the system more efficient and easier to work with. The code uses effective data structures and can handle large amounts of data quickly and scalably.

Table of Contents:

Project Workflow 1

Tools and technologies 5

Project Workflow

The text analysis pipeline follows a structured workflow to systematically process the input data and generate the desired insights. The core components of this workflow are:

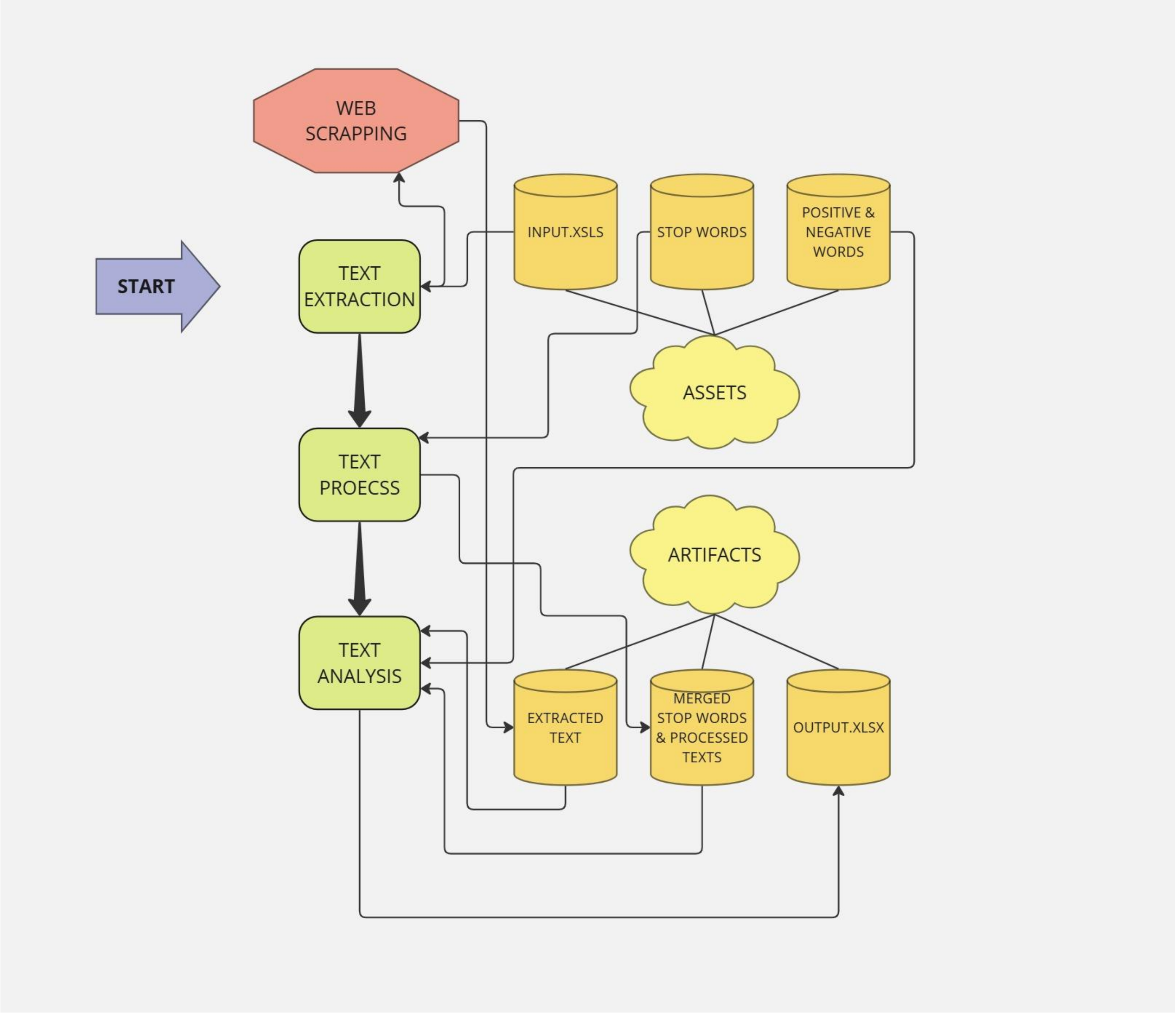

Figure 1.1 Work flow architecture of the project

1. Text Extraction

The first step involves extracting the raw text content from the URLs provided in the input Excel files. This is accomplished through automated web scraping techniques, which retrieve the HTML content of each webpage and parse out the relevant text data. The web scraping module ensures efficient and scalable extraction of text from multiple sources.

2. Text Processing

I Once the raw text has been collected, the pipeline moves on to the preprocessing stage. This includes various text cleaning and normalization tasks, such as:

- Removal of stop words: Common words (e.g., "the", "a", "and") that do not carry significant meaning are identified and removed from the text.
- Text formatting: The extracted text is formatted to ensure consistency, such as converting to lowercase.
- Merge Stop Words Files: Merging all files containing different different types of stop words.

3. Output Generation

3.1 Text Analysis

- Calculate different different scores for sentiment analysis
- Calculate different different variables for checking Readability & words and sentence characteristics

3.2 Deliverables

- Excel file containing all variables along with URL_ID and URL

Tools & Technologies

1. Programming Languages

1.1 Python

- Primary development language
- Version: 3.8

2. Dependencies

- nltk
- ipykernel
- pandas
- python-box==6.0.2
- pyYAML
- ensure==1.0.2
- types-pyYAML