

**Elementary Systems Programming Project**

**01286120 Elementary Systems Programming**

**Software Engineering Program,**

**Department of Computer Engineering,**

**School of Engineering, KMITL**

**By**

**66011533 Eaint Kay Khaing Kyaw**

**Text Data Statics Collector**

**Introduction**

Text Data Statics Collector is a command line utility that allows users to analyze and rank text documents found in a specified folder. It provides various insights about the documents such as word count, character count, line count, avg word length and most common words. Users can choose to sort the documents by different criteria, including word count, line count and character count. Additionally, the tool can search for a specified word within the documents and provide the locations where the word is found.

**Motivation**

This project was motivated by the need for a simple and flexible tool to quickly access and compare text files in a directory. Sometimes, we may want to find some words easily and may sort the files by different criteria. Moreover, this project mat be beneficial in various contexts such as content management, research or data exploration.

**Features:**

Document Statistics: Collect and display the following statistics for each text file:

* Word Count
* Character Count
* Line Count
* Average Word length
* Most Common words

Folder Statistics:

* Provide overall statistics for the entire folder, including most common words and their frequencies across all files.

Document Ranking:

* Ranking by Word Count: Ranks documents based on their word count in descending order.
* Ranking by Line Count: Ranks documents based on their line count in descending order.
* Ranking by Character Count: Ranks documents based on their cha count in descending order.

Word Search:

* Searches for a specified word within all documents in a folder and provides locations where the word is found.
* Generates a csv file including those locations

HTML Report:

* Generates an HTML report summarizing the analysis and rankings of the documents. The report includes a table with document statistics and a histogram of the most common words.

**Usage**

Command Line Interface

User can generate a html report and histogram including word count, line count, character count, average word length, most common words in each file and in specific folder.

* To sort by line count

cargo run <input folder> line

**Example Usage**

cargo run "D:\KMITL\rust programming\Rust Project\data\_collecting\Test For Project" line

* To sort by word count

cargo run <input folder> word

* To sort by character count

cargo run <input folder> cha

* To find specific word

cargo run <input folder> find\_word <word>

**Example Usage**

cargo run "D:\KMITL\rust programming\Rust Project\data\_collecting\Test For Project" find\_word HAHA

**Implementation**

This project utilizes several libraries, including:

* std::env: For command-line argument handling
* std::path: For working with file paths
* std::collections::HashMap: For storing and analyzing word frequency
* std::fs For file I/O operations
* csv: For writing search results to a csv file
* std::error::Error: For error handling

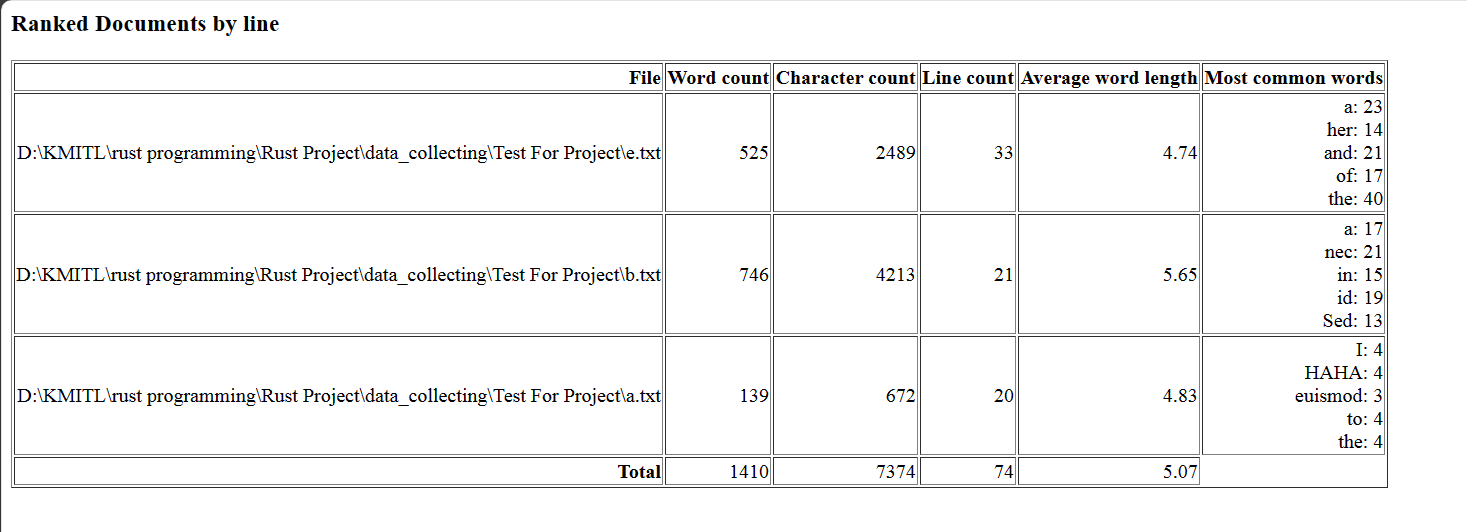
The core function of the Text Statistics Collector is encapsulated in the “Document” struct , which represents individual documents and provides methods for document analysis, ranking and HTML report generation.

**Conclusion**

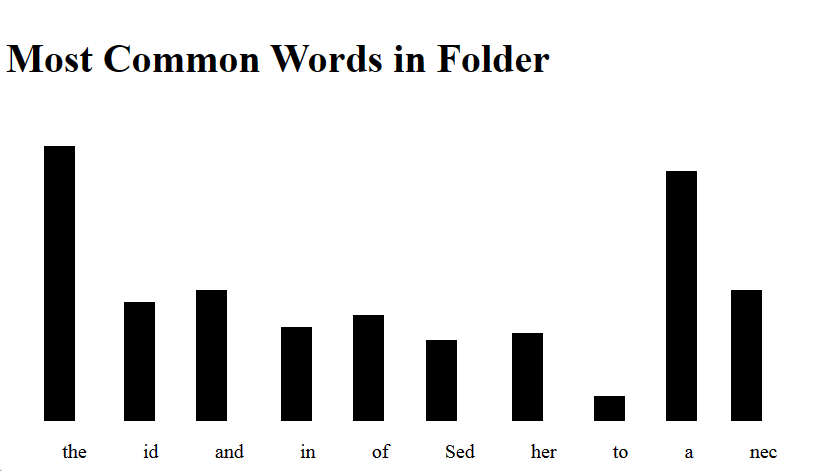
The Text Statistics Collector is a versatile and efficient tool for analyzing and searching within text documents. It serves as a valuable resource for various use cases, including content management, data analysis, and research. Users can easily tailor the tool to their specific needs and preferences by choosing the sorting methods and searching for specified words within documents.

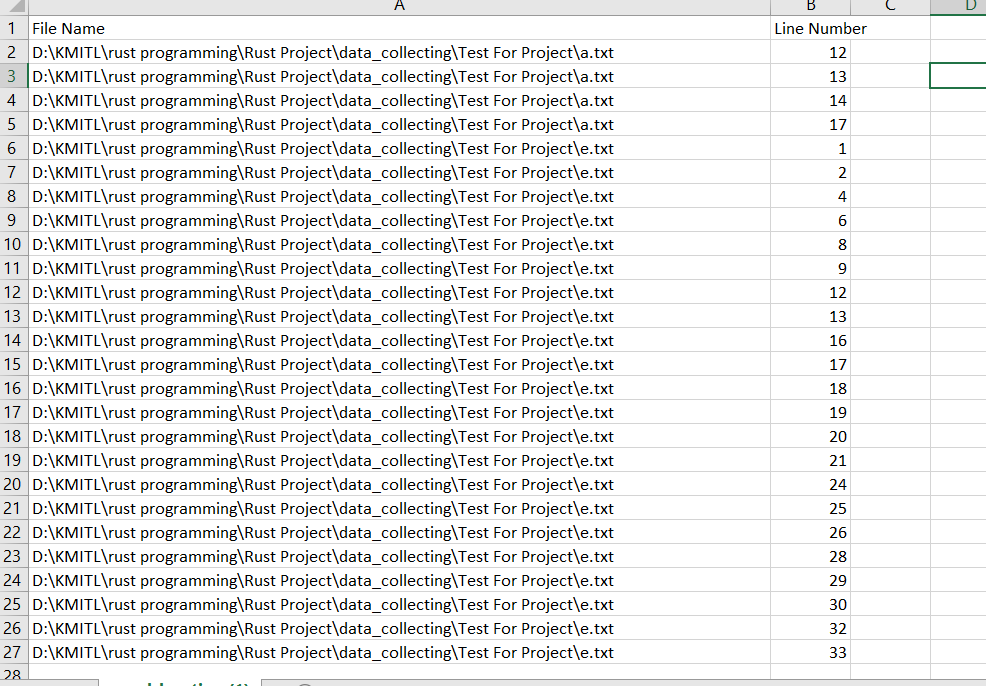
The project can be extended with additional features, such as more advanced search options, support for additional file formats.

Text Statistics Collector represents a powerful tool for text document analysis demonstrating the capabilities of the Rust Programming language and its ecosystem for building efficient and versatile command-line utilities.

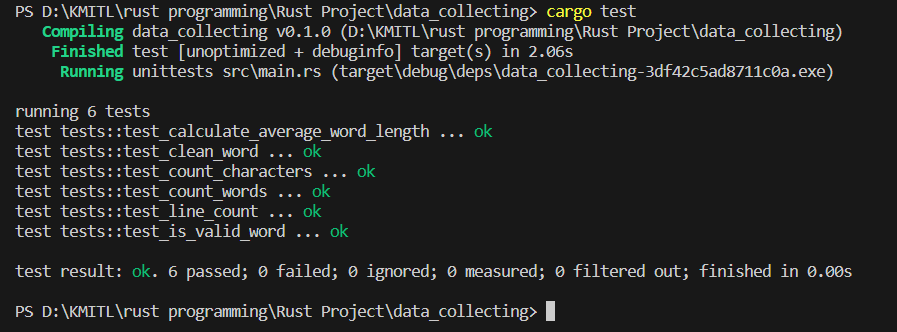


**Figure 1. HTML report for each file and folder showing word, character, line count and others**



****

**Figure 4. Showing Word Location in csv file**

****

**Figure 4. Test Files**

**▪**