Documentation

System for analyzing words connections statistics in given text

System for preparing punctuation marks statistics for given text

Authors: Everest Alonso, Patrik Homola, Kamil Pecela

Project developed for the course “Development of Information Systems”

Faculty of Mathematics, Physics and Informatics

Comenius University Bratislava

2023-2024

Table of Contents

Requirements catalog

Introduction

The purpose of this requirements catalog

This document presents a summary of all requirements for two information systems. First one, named “Aphasia detection”, produces words connection statistics for given text and makes graphs based on user’s queries. Second one, named “Punctuation mark analysis”, produces statistics of punctuation marks for given text and makes graphs illustrating the outcomes. It was created as a project within the subject “Development of Information Systems” at the Faculty of Mathematics, Physics and Informatics of the Comenius University in Bratislava. The document is created based on the requirements that were written down at the meeting with the client and in the subsequent communication. It is intended for all persons involved in the development of the system, its administration and use. It also serves as a binding agreement on the functionality of the system between the client and the creators.

Scope of system use

The goal is to develop two applications. First one will be used as a tool in a search for signs of aphasia in texts based on statistics of the closest connections between words. The system will analyze given text and produce statistics of every word connection in it. Statistics will be stored in application folder on user’s computer. Based on this statistics system will produce graphs. Second system will prepare statistics of punctuations marks in texts. The system will analyze given text for the lengths (measured in the number of words) between any two punctuation marks: longest, shortest and average. Based on this statistics system can produce graphs. Application will run independently on personal computers.

Glossary

- **Text** – text in traditional meaning, saved in *.txt format.
- **Array of words** – text preprocessed by application.
- **Statistics file** – output of analysis stored in application directory on user’s computer in form of data file – format: *.txt. There is one file for each text. Statistic file contains description of text if the user introduced it while uploading text.

- **Record in log file** – after deleting statistic file proper record is added to log file, with information about deleted file and time of this operation.

Overview of the following chapters

The following chapters will describe the functionality of the system, general limitations, functional and quality requirements and interface requirements.

General description

System perspective

First system, “Aphasia detection”, will serve for academic scholars searching for possible symptoms of aphasia in texts to analyze any given text, prepare statistics of words connection and make graphs on this basis. Aim of the system is to store statistics of analyzed texts and enable easy way of querying these statistics to produce graphs. Statistics of connections between words may help to find or define symptoms of aphasia in texts.

Second system, “Punctuation mark analysis”, is aimed at analyzing any given text, preparing statistics of punctuation marks and making graphs on this basis. Statistics of punctuation marks may help to study language from different points of view.

Functions of the systems

First system “Aphasia detection” starts with initial graphical user interface displaying three options. First option (A) is to upload text for analysis, second one (B) is to delete file, third one (C) is to view statistics and produce graphs [bar charts].

Ad. A: After clicking button “Upload text”, user is redirected to another GUI. He is uploading the text in form of text document (*.txt) in utf8. There is additional field for a user to input optional description of uploaded text with disclaimer suggesting options of description (genre and information whether text is transcription of a spoken language and whether author suffers from aphasia). Disclaimer will be visible all the time: “All text within the given file will be analyzed. If you don’t want editorial notes to be analyzed as well, you need to delete it.”. Then user clicks “Go” button and analysis starts. Analysis will firstly preprocess text: it will convert text into simple array of words without punctuation marks. Analysis will count how often any two words appear as a pair regardless of order. This information will be stored in a text file called statistics file and saved in a application directory. User is returned to the initial screen with three choices. He can add another text, run query (see B), view statistics (see C) or end application.

Ad. B: After clicking button “Delete file” user is redirected to interface with list of saved statistics file. He chooses file to delete and execute his decision by clicking proper button. After deleting file new record is added to log file. User can click button “Go to start” (which will redirect him to initial interface) or delete another file.

Ad. C: After clicking button “View statistics and produce graph”, user is redirected to new GUI. There will be a list of statistics files displayed, from which user can choose one to query. Three buttons will allow him to choose searching by word (he must input it in proper bar), by a pair of words (he must input both of them) or search through entire text. In another input bar he can choose the number of outputs. It will be inactive if user chooses to search for a specific pair of words.

There will be a check box to produce graphs. User also can input logarithmic scale of displaying graph. At the bottom there will be a button: “See results”. By clicking “See results” user is redirected to interface that displays results: statistics and graphs. By clicking button “Save results”, user can save the results of query and the graph in form of image file in chosen directory. There is a button “Go to start”, which redirects user to initial page.

Second system “Punctuation mark analysis” starts with initial graphical user interface displaying button “Upload text” and button “Start”. Disclaimer will be visible all the time: “All text within the given file will be analyzed. If you don’t want editorial notes to be analyzed as well, you need to delete it.”. User will input author and title of the text. User can also analyze text for chosen punctuation mark or for any punctuation mark. First option will be executed by clicking one of the buttons with punctuation marks. When no such button is clicked, punctuation marks in general will be analyzed. After clicking “Start” analysis will begin. Analysis will firstly preprocess text: it will convert text into simple array of words. Analysis will count the number of words between occurrences of chosen punctuation mark or between occurrences of any punctuation marks. User can also choose the logarithmic scale for display. After the end of analysis results (in the form of a bar graph) will be shown on new page. User can save results in form of image file to chosen directory by clicking button “Save results”. At the bottom of result there will be button “Analyze new text”, which will redirect user to the initial page.

Users characteristics

User of both systems is an academic researcher presumably without experience in IT. There is only one role and profile of user. Users have rights to use all the functionalities of the systems.

General restrictions

Both systems analyze only written texts. Other restrictions are of copyright nature and are responsibility of the user, who should respect the rights of the analyzed texts authors.

Prerequisites and dependencies

Application will run in Windows OS. Besides installation it will need free disk space for statistics.

Specific requirements

First system: “Aphasia detection”

1. User opens the system by clicking the executable app.
2. User interacts with the system using a graphical user interface.
3. Initial page of application contains three buttons that represents three main functionalities of the system: 1) uploading and analyzing text; 2) deleting statistics file; 3) displaying statistics and graphs.
4. User can choose to upload and analyze text by clicking button “Upload text”. After doing that he is redirected to new page of application for uploading.
5. During the process of uploading the text disclaimer will be shown: “All text within the given file will be analyzed. If you don’t want editorial notes to be analyzed as well, you need to delete it.”
6. Process of uploading the text by the user consists of choosing text file to upload, adding name of the author and the title and optionally adding description of the text. Text should be in *.txt file with utf-8 encoding. Description may contain whatever information user considers useful. Process of uploading ends when user clicks the button “Upload”.
7. During the process of uploading the text disclaimer will be shown: “You may describe the text with additional information about genre of the text, whether it is transcription of a spoken language and whether author suffers from aphasia as well as any other useful information”.
8. System starts analysis of the text when the user ends the upload process. At first, the system transforms the text into an array of words, then the system counts how often any two words appear as a pair of consecutive words regardless of their order.
9. System saves results of the analysis in a special directory in the form of *.txt file with utf-8 encoding. This file will be named after the author and the title of the text. If the user added description in the process of uploading text, this description will be added as a header to output statistics file. Statistics file will contain a list of a pairs of words with number representing the frequency of occurrence for each pair.
10. System will redirect the user to the initial page after the analysis is finished and the statistics file is saved.
11. User can choose to delete the statistics file from the application directory by clicking a “Delete file” button on the initial page. He or she will be then redirected to a new page of application for file removal.

12. Process of deleting the file by the user consists of choosing the file from the list, pressing a button to delete and confirming this decision. After deleting the file, the user will be automatically redirected to the initial page.
13. After deleting the statistics file by the user, system will automatically add a proper record to log file in the system directory. This record will contain information about deleted file and time of operation.
14. User can choose to view statistics of previously analyzed text by clicking “View statistics and produce graph” on the initial page. He or she will then be redirected to a new page of application for viewing statistics.
15. Process of viewing statistics and producing graph consists of several steps. User must choose a statistics file from the list. Then he or she chooses one of three options: searching by a singular word, a pair of words or through entire text. If the user wants to produce a graph, he or she must separately tick a proper checkbox. User can also choose a logarithmic scale for display. The process starts after the user clicks the button “See results”.
16. Results of viewing statistics file and producing bar graph are displayed on a new page of application. At x axis of the graph there will be items representing pairs of words ore one word, depending on the choice made by user. Y axis will show frequency in numbers. User can save outputs of query in form of image format to chosen directory by clicking button “Save results”. At the bottom of this page there is a button “Go to start”, which redirects the user to the initial page.

Second system: “Punctuation mark analysis”

1. User opens the system by clicking the executable app.
2. User interacts with the system using a graphical user interface.
3. On the initial page of application user is uploading text to be analyzed and describes it with author and title.
4. Uploaded text should be a *.txt file with utf-8 encoding.
5. Disclaimer will be visible all the time on the initial page: “All text within the given file will be analyzed. If you don’t want editorial notes to be analyzed as well, you need to delete it.”
6. User can choose what kind of punctuation mark should be the object of analysis by clicking one of the buttons with punctuation marks. If none of the punctuation marks is chosen, analysis will be done for occurrences of any punctuation marks.
7. User can input the logarithmic scale of display.
8. Analysis starts after user clicks “Go” button at the bottom of initial page.

9. System starts the process of analysis by transforming text into array of words. Then system will count the number of words between occurrences of chosen punctuation marks or any punctuation marks.
10. After the analysis finishes, results will be displayed on another page. Results will consist of statistics and bar graph. X axis of the graph will represent frequency and y axis will represent distance in words.
11. User can save outputs of query in form of image format to chosen directory by clicking button “Save results”.
12. At the bottom of the result page there will be a button “Analyze new text”, which will redirect user to the initial page.

Documentation for the “Punctuation Analysis Tool” application.

Introduction

This document was created within the subject Development of Information Systems in 2023/2024 and serves as a complete design of the application “Punctuation Analysis Tool”. It contains information necessary for the explanation and understanding the functionality as well as the method of system implementation. This document refers to a requirements catalog. The following chapters will present the implementation plan, system architecture description, and test scenarios.

Implementation Plan

Project Setup

Create a new project directory.

Set up version control.

Initialize a virtual environment for the project.

Dependencies

- Install required dependencies (Tkinter, Matplotlib).

GUI Design and Layout

17. Design the initial page with widgets for file selection, author, title, and punctuation mark buttons.
18. Implement the "Go" button for initiating the analysis.
19. Implement the disclaimer display.

File Handling

13. Implement functionality to handle file uploads.
14. Ensure that only *.txt files with utf-8 encoding are accepted.
15. Extract author and title information from the uploaded text.

Punctuation Selection

- Create buttons for each punctuation mark.
- Allow users to select a specific punctuation mark for analysis.

Logarithmic Scale

- Implement checkboxes for logarithmic X and Y axes.
- Incorporate these choices in the plot generation.

Analysis Process

- Implement the analysis process.
- Transform the text into an array of words.
- Count the number of words between occurrences of chosen punctuation marks or any punctuation marks.

Results Display

- Create a new page for displaying analysis results.
- Show statistics and a bar graph.
- Allow users to save results in image format to a chosen directory.

"Analyze New Text" Button

- Implement a button on the result page to redirect users to the initial page for a new analysis.

Error Handling

- Implement error handling for file reading and invalid input.

Documentation

- Document the code using comments and docstrings.
- Create a README file explaining how to run the application.

Testing

- Conduct unit testing for individual functions.
- Perform integration testing for the entire application.

User Interface Polish

- Enhance the user interface for better user experience.
- Ensure consistency in design and responsiveness.

Deployment

- Package the application for distribution.

- Create an executable file for users to run the application.

User Manual

- Prepare a user manual explaining how to use the application.

Architecture and Design Overview

High-Level Architecture:

- The application follows a Model-View-Controller (MVC) architectural pattern.
- Model: Encapsulated in the count_words_between_punctuation function, responsible for text processing and analysis.
- View: Implemented using Tkinter for the graphical user interface (GUI) elements.
- Controller: Orchestrated through the submit_form function, handling user inputs and triggering the analysis process.

File Handling and Model:

- The count_words_between_punctuation function handles file reading, text processing, and word count analysis.
- It utilizes regular expressions to split the text into segments based on user-defined punctuation marks.
- The result is a dictionary (word_counts) containing word frequencies based on the distance before the next punctuation mark.

User Interface Design (View):

- The GUI is developed using Tkinter with two main frames: one for file handling and another for text input and analysis configuration.
- Widgets include labels, buttons, entry fields, and checkboxes for user interactions.
- The application features error message pop-ups using Tkinter's messagebox for a user-friendly experience.

Graphical Representation (View):

- Matplotlib is employed for graphical representation.
- The show_plot function generates a bar graph based on the analysis results, with the option for logarithmic scales on the X and Y axes.

User Interaction (Controller):

- Users interact with the system by uploading a text file, selecting punctuation marks, and configuring logarithmic scales.
- The submit_form function orchestrates the entire process, including error handling for file selection, punctuation input validation, and triggering the analysis.

Error Handling:

- Robust error handling is implemented throughout the application.
- Users receive informative error messages in case of issues, such as file reading errors or invalid input for punctuation marks.

Global Variables:

- Global variables (file_path, fig_canvas, canvas_frame, log_x_axis, and log_y_axis) are used to maintain state and share information across functions.

Testing scenarios

File Upload and Encoding

- **Scenario 1:** User uploads a valid *.txt file with utf-8 encoding.
    - Expected Result: File is accepted, and the system extracts author and title information.
- **Scenario 2:** User tries to upload a non-*.txt file.
    - Expected Result: System displays an error message indicating that only *.txt files are accepted.
- **Scenario 3:** User uploads a *.txt file with a different encoding.
    - Expected Result: System displays an error message indicating that utf-8 encoding is required.

Punctuation Selection

- **Scenario 1:** User chooses specific punctuation marks for analysis.
    - Expected Result: Analysis is conducted based on the chosen punctuation marks.
- **Scenario 2:** User doesn't select any punctuation mark.
    - Expected Result: Analysis is conducted for occurrences of any punctuation marks.

Logarithmic Scale

- **Scenario 1:** User checks both logarithmic X and Y axes.
    - Expected Result: Graph is displayed with logarithmic scales on both axes.
- **Scenario 2:** User checks only logarithmic X axis.
    - Expected Result: Graph is displayed with a logarithmic scale on the X axis.

Analysis Process

- **Scenario 1:** User initiates analysis with a valid file and punctuation marks.
    - Expected Result: The system successfully analyzes the text and generates statistics and a bar graph.
- **Scenario 2:** User initiates analysis without selecting a file.
    - Expected Result: System displays an error message prompting the user to select a file.

Results Display

- **Scenario 1:** User saves analysis results as an image.
    - Expected Result: Results are saved in the specified directory in image format.
- **Scenario 2:** User clicks "Analyze New Text" on the results page.
    - Expected Result: User is redirected to the initial page for a new analysis.

Error Handling

- **Scenario 1:** User encounters an error during file reading.
    - Expected Result: System displays an error message indicating the issue with file reading.
- **Scenario 2:** User enters invalid punctuation marks.
    - Expected Result: System displays an error message indicating the invalid input.

User Interface

- **Scenario 1:** User interacts with the graphical user interface.
    - Expected Result: All widgets and buttons respond correctly to user interactions.
- **Scenario 2:** User clicks the "Go" button without selecting a file.
    - Expected Result: System displays an error message prompting the user to select a file.

General Application Flow

- **Scenario 1:** User goes through the entire process from file upload to result display.
    - Expected Result: The application successfully conducts the analysis and presents the results.
- **Scenario 2:** User closes and reopens the application.
    - Expected Result: The application retains its state, and the user can resume or start a new analysis.

User's Guide for Punctuation Analysis Tool

Introduction

The Punctuation Analysis Tool is a simple graphical user interface (GUI) application designed to analyze the frequency of words between punctuation marks in a given text file. This guide provides an overview of the tool's features and instructions on how to use it effectively.

Starting application

The Punctuation Analysis Tool is implemented in Python and utilizes the Tkinter library for the graphical interface. Click on the icon of the file to run the program.

File Selection

Click the "Select File" button to choose a text file for analysis.
The selected file's path will be displayed on the GUI.

Text Input

- Enter punctuation marks separated by spaces in the text entry field.
- If no marks are provided, the tool will use a default set: `, . ? ! - ' : ; – ( ) / " [ ] { }`.

Logarithmic Axes

20. Check the "Logarithmic X-axis" box to enable logarithmic scaling on the X-axis.
21. Check the "Logarithmic Y-axis" box to enable logarithmic scaling on the Y-axis.

Running Analysis

16. Click the "Submit" button to initiate the analysis.
17. A bar chart will be displayed showing the frequency of words between punctuation marks.
18. The X-axis represents the distance before the next punctuation mark (in words), and the Y-axis represents the frequency of occurrence.

Error Handling

- If an error occurs while opening the file, an error message will be displayed.
- Invalid input for punctuation marks will trigger an error message.
- If the selected file is not a .txt file or is corrupted, an error message will be shown.

Saving results

- Click on the disk icon to save results
- Choose directory to which graph will be saved.

About the GUI

- The GUI is divided into two frames: one for file selection and another for text input and options.
- A label at the top displays the selected file path.
- The tool allows customization of punctuation marks and provides options for logarithmic scaling on the chart axes.

Note: Ensure that the file to be analyzed is a valid .txt file, and the text is appropriately formatted for accurate analysis.

Documentation for the “Words Statistics Analysis Tool” application.

Introduction

This document was created within the subject Development of Information Systems in 2023/2024 and serves as a complete design of the application “Words Statistics Analysis Tool”. It contains information necessary for the explanation and understanding the functionality as well as the method of system implementation. This document refers to a requirements catalog. The following chapters will present the implementation plan, system architecture description, and test scenarios.

Implementation Plan

User Interface Design

Design the graphical user interface (GUI) with an initial page containing three buttons for main functionalities. Create separate pages for uploading text, deleting statistics file, and viewing statistics and producing graphs.

System Launch

Develop the executable application. Implement the functionality for the user to open the system by clicking the executable app.

Text Upload Functionality

Design and implement the "Upload text" button on the initial page. Create a new page for uploading text with fields for choosing a text file, adding author name, title, and optional description. Show disclaimers during the uploading process. Implement the transformation of the text into an array of words. Count the frequency of consecutive word pairs. Save results in a special directory as a *.txt file with utf-8 encoding, named after the author and title.

File Deletion Functionality

Implement the "Delete file" button on the initial page. Create a new page for file removal, listing available statistics files. Allow the user to choose and delete a file, with confirmation. Automatically redirect the user to the initial page after deletion. Log the file deletion operation with proper details in a log file.

View Statistics and Produce Graph Functionality

Implement the "View statistics and produce graph" button on the initial page. Create a new page for viewing statistics with options to choose a file, search by a word or pair of words, and choose display options. Allow the user to produce a graph, tick checkboxes for specific options, and choose a logarithmic scale. Display the results on a new page with a bar graph. Provide the option to save the results in image format to a chosen directory. Include a "Go to start" button at the bottom of the page for redirection to the initial page.

Overall System Flow

Implement the overall flow to redirect the user to the initial page after each operation (upload, delete, view). Ensure proper error handling and user feedback throughout the application.

Testing

Conduct thorough testing for each functionality, including positive and negative scenarios. Debug and refine the application based on testing results.

Documentation

Document the implementation details, including code structure, functions, and modules. Create user documentation explaining how to use the application.

Deployment

Package the application for deployment on relevant platforms.

Maintenance

Plan for ongoing maintenance, updates, and support.

Architecture & Design Overview

Application Overview:

The application is designed as a text analysis tool implemented in Python using the Tkinter library for the graphical user interface. It allows users to upload text files, perform various analyses, and visualize the results through a user-friendly interface.

Modules:

Main Application Module (main.py):

Initializes the Tkinter application.
Manages different frames representing various sections of the application.
Handles the overall flow of the application.

Frames:

BaseFrame:

Serves as the base class for other frames.
Implements common functionality like switching frames and updating.

MainMenu:

Displays the main menu with buttons to navigate to different functionalities.

UploadText:

Allows users to upload text files.
Collects information like name and description for the uploaded text.
Performs analysis on the uploaded text.

DeleteText:

Enables users to delete previously uploaded texts.
Lists available texts for deletion.

SelectToAnaliseFrame:

Provides options to select a text for analysis.
Takes input for words and parameters for analysis.

StatsFrame:

Displays general information about the selected text.

SearchPairFrame:

Extends StatsFrame.
Displays results for a search pair analysis.

Analyses:

Base class for specific analyses frames.
Contains a scrollable table and a bar chart for visual representation.

FullAnalyses:

Displays results for a full analysis.

ForWordAnalyses:

Extends Analyses.
Displays results for a word-specific analysis.

Supporting Modules:

classQueryModule:

Implements the QueryModule class handling text file analysis queries.

Analysis:

Provides functions for analyzing text files and counting word connections.

file_check:

Checks if the uploaded file is compatible (UTF-8 encoded text).

SaveLoadSystem:

Handles saving and loading objects using pickle.

scrollableTable:

Implements a scrollable table and bar chart using Tkinter and Matplotlib.

Data Structures:

TextPath:

Represents a text file's path, title, and description.

Data Flow

- User uploads a text file through the UploadText frame.
- The application performs analysis on the uploaded text using the Analysis module.

Query Module

22. Loads data from the selected text file.
23. Runs queries based on user input (word searches, pair searches, full analysis).
24. Returns results to the respective frame for visualization.

Graphical Interface

19. The Tkinter library is used to create a responsive and intuitive GUI.
20. Frames are dynamically switched based on user interactions.

Extensibility

- The modular design allows for easy addition of new analysis functionalities or frames.
- QueryModule can be extended to include additional query types.

Conclusion: The architecture follows a modular and extensible design, providing a user-friendly interface for text analysis. The combination of Tkinter for GUI, analysis modules, and supporting functions creates a robust tool for exploring relationships within text data.

Testing Scenarios

Uploading Text

- **Scenario 1: Successful Upload**
    - User selects the "Upload Text" option from the main menu.
    - User selects a valid text file (UTF-8 encoded).
    - User provides a unique name and description for the text.
    - User clicks the "Upload" button.
    - Verify that the application processes the file without errors.
    - Verify that the new text appears in the list of available texts.
- **Scenario 2: Upload Failure**
    - User selects the "Upload Text" option from the main menu.
    - User selects an invalid file (non-UTF-8 encoded or unsupported format).
    - User provides a name and description for the text.
    - User clicks the "Upload" button.
    - Verify that the application displays an appropriate error message.
    - Verify that the text list remains unchanged.

Deleting Text

- **Scenario 1: Successful Deletion**
    - User selects the "Delete Text" option from the main menu.
    - User selects a text from the list for deletion.
    - User clicks the "Delete" button.
    - Verify that the application prompts for confirmation.
    - User confirms deletion.
    - Verify that the text is removed from the list.
- **Scenario 2: Cancel Deletion**
    - User selects the "Delete Text" option from the main menu.
    - User selects a text from the list for deletion.
    - User clicks the "Delete" button.
    - Verify that the application prompts for confirmation.
    - User cancels the deletion.
    - Verify that the text list remains unchanged.

Selecting Text for Analysis

- **Scenario 1: Successful Selection**
    - User selects the "Select Text to Analyze" option from the main menu.
    - User chooses a text from the list.
    - User enters valid words and parameters for analysis.

    - User clicks the "Analyze" button.
    - Verify that the analysis results frame is displayed.
    - Verify that the correct analysis results are shown.
- **Scenario 2: Invalid Selection**
    - User selects the "Select Text to Analyze" option from the main menu.
    - User does not choose any text or enters invalid parameters.
    - User clicks the "Analyze" button.
    - Verify that the application displays an error message.
    - Verify that the analysis results frame is not displayed.

General Interaction:

- **Scenario 1: Navigation**
    - User navigates through different frames using the provided buttons.
    - Verify that frames switch correctly based on user interactions.
- **Scenario 2: Closing the Application**
    - User clicks the close button.
    - Verify that the application closes without errors.

User’s Guide for Words Statistics Application

Introduction

Welcome to the Text Analysis Application! This application allows you to upload, delete, and analyze text files. You can perform various analyses on the uploaded texts, such as searching for word pairs, displaying statistics, and more.

Main Menu

To access the main menu, launch the application.
The main menu provides options to:
Upload Text: Navigate to the Upload Text section.
Delete Text: Navigate to the Delete Text section.
Show Stats: Navigate to the Statistics section.

Upload Text

- Click on the "Upload Text" button in the main menu.
- Select a text file using the "Select file" button.
- Enter a name and description for the text.
- Click the "Submit" button to upload the text.
- Click "Cancel" to go back to the main menu.

Delete Text

- Click on the "Delete Text" button in the main menu.
- Select the text file to be deleted from the dropdown.
- Click the "Delete" button to remove the text.
- Click "Go to Start" to go back to the main menu.

Show Stats

- Click on the "Show Stats" button in the main menu.
- Select a text file from the dropdown.
- Choose analysis options (e.g., statistics for one word, pair of words or for the entire text).
- Click the "See Result" button to view the analysis results.
- Navigate to different analysis frames based on the result.

Save results

- Click on the disk icon to save graph.

- Choose the place on your computer to save your file.
- View the results in specific frames (Word Analysis, Full Analysis, Search Pair).

Statistics Frame

- Displays author, title, and additional description of the selected text.
- Information is loaded from the selected text file.

Search for the Entire Text Frame

- Displays comprehensive analysis results for the entire text.
- Includes a scrollable table and a bar chart.

Search for the Pair of Words Frame

- Displays output of the search pair analysis.
- Shows word1, word2, and the query result.

Search for One Word Frame

- Displays word analysis results for a selected word.
- Shows the frequency of the word in the text.

Scrollable Table

- Displays analysis results in a scrollable table format.
- Allows easy navigation and viewing of the data.

Bar Chart

- Displays analysis results in a bar chart format.
- Provides a visual representation of word frequencies.