

AI Project - 02

Final Project Report

QUALITATIVE DATA ANALYSIS

horizontal line

**Team Members:**

Ashok Chaudhari (200525339)

Harshdeep singh (200592613)

Birva Chudasama (200596766)

Sukhjeet Kaur (200575307)

# 

# 

# Introduction

This research represents an innovative approach to addressing the complex challenges of qualitative data analysis in institutional research. By harnessing the power of artificial intelligence and natural language processing, we aim to provide IR departments with a cutting-edge tool that transforms how qualitative insights are generated and utilized.

The Qualitative Data Analysis (QDA) Tool is an advanced Python-based application designed to transform unstructured textual data into meaningful insights. By integrating multiple natural language processing (NLP) techniques, the tool offers researchers, analysts, and professionals a powerful platform for extracting deep semantic understanding from various document formats.

**Problem Statement**

### Background

In the era of big data, organizations and researchers face significant challenges in analyzing large volumes of textual information. Traditional manual analysis methods are:

* Time-consuming
* Prone to human bias
* Limited in scalability
* Inconsistent in interpretation

The problem to be addressed through this study is the inefficiency and inconsistency in analyzing qualitative data for Institutional Research (IR) in higher education. Manual analysis of open-ended survey responses, interview transcripts, and focus group data is time-consuming and prone to human error, leading to potential biases in research outcomes. Despite the increasing volume of qualitative data collected by IR departments, there is a lack of standardized, automated tools specifically designed for higher education contexts to streamline this process.

This issue affects IR analysts, administrators, and ultimately, the decision-making processes within educational institutions. Without efficient analysis of qualitative data, valuable insights may be overlooked, leading to missed opportunities for improving student experiences, curriculum development, and institutional policies.

To address this problem, research is needed to develop an AI-powered qualitative data analysis tool tailored for IR departments. This study aims to investigate the application of natural language processing and machine learning techniques to automate theme extraction, sentiment analysis, and report generation from diverse qualitative data sources in higher education settings. By bridging this gap, the research will contribute to more timely, accurate, and comprehensive institutional research, ultimately enhancing data-driven decision-making in higher education.

**General Objectives:**

* To develop a software prototype that can analyze qualitative data, such as focus groups, interviews, and open-ended survey responses.
* To reduce the time and effort required by analysts to manually review, analyze, and report qualitative data.
* To provide automatic theming of data, performing context analysis and sentiment analysis to extract key insights and themes from qualitative data.

**Specific Objectives:**

* To design and implement a solution capable of processing qualitative data from Word and Excel documents.
* To create an automatic theming system that identifies and categorizes key themes from the data.
* To integrate context analysis that interprets the data in relation to the questions and objectives of the survey or interview.
* To build sentiment analysis features that detect the tone and emotional sentiment of the responses.
* To provide a user-friendly interface for analysts to interact with the software and review the extracted themes and insights.
* To ensure that the software can output results in formats that support reporting, including summary reports and visual representations of themes.
* To validate the accuracy and usefulness of the software through testing with real-world qualitative data samples.

**Project Planning**

### Project Scope

1. Development of a Data Import Module:
   * Design and implement a module capable of reading and importing qualitative data from Word and Excel documents.
   * Ensure compatibility with common document formats used by the Institutional Research (IR) team.
2. Implementation of NLP Algorithms for Thematic and Sentiment Analysis:
   * Develop Natural Language Processing (NLP) algorithms to analyze qualitative data.
   * Implement functionality for thematic analysis to identify key themes and trends from the data.
   * Integrate sentiment analysis to assess the emotional tone of responses, enabling sentiment categorization (positive, negative, neutral).
3. Creation of Customizable Reports :
   * Develop features for generating customizable reports summarizing the key themes, insights, and sentiment analysis results.
4. Design and Development of a User-Friendly Interface:
   * Create an intuitive, easy-to-navigate user interface for IR analysts to interact with the software.
   * Include features for easy data input, theme extraction, sentiment analysis, and output customization.
   * Ensure accessibility and clarity, minimizing the learning curve for end-users.
5. Comprehensive Testing:
   * Perform thorough testing to ensure the system meets its objectives:
     + Unit Testing: Validate individual components of the system (data import, analysis, etc.).
     + Integration Testing: Ensure seamless interaction between modules (data processing, report generation).
     + User Acceptance Testing (UAT): Involve IR team members in testing to ensure the software meets their requirements and expectations.
6. Documentation of All Features, Functionalities, and User Guides:
   * Prepare comprehensive technical documentation detailing the system architecture, functionalities, and algorithms used.
   * Create user guides and manuals to assist IR analysts in effectively using the software.

**Project Timeline**

| Phase | Task | Start Date | End Date | Duration | Notes |
| --- | --- | --- | --- | --- | --- |
| 1. Planning | Define project scope, roles, and milestones | Sep 18, 2024 | Sep 22, 2024 | 1 week | Discuss requirements with the team and sponsor. |
|  | Gather requirements from stakeholders (IR team) | Sep 22, 2024 | Sep 27, 2024 | 5 days | Collect input from IR on key themes and reporting needs. |
|  | Create system requirements specification (SRS) | Sep 27, 2024 | Sep 29, 2024 | 2 days | Document all functional and non-functional requirements. |
| 2. Analysis | Analyze data flow and module requirements | Sep 30, 2024 | Oct 2, 2024 | 3 days | Identify core modules and how they'll interact (input, processing, output). |
|  | Finalize technology stack and libraries | Oct 2, 2024 | Oct 3, 2024 | 1 day | Choose libraries (pandas, python-docx, nltk, gensim, VADER, etc.). |
| 3. Design | Design system architecture (flow of data) | Oct 4, 2024 | Oct 8, 2024 | 5 days | Outline system components (front-end, back-end). |
|  | UI/UX design for front-end (if applicable) | Oct 8, 2024 | Oct 10, 2024 | 3 days | Design a simple UI for file uploads and displaying results (optional). |
|  | Create database and file storage design | Oct 10, 2024 | Oct 11, 2024 | 2 days | Plan how data will be stored and structured (Excel, CSV export). |
| 4. Development | Set up development environment | Oct 12, 2024 | Oct 13, 2024 | 2 days | Set up virtual environment, install dependencies, and version control. |
|  | Develop file input module (Word & Excel) | Oct 14, 2024 | Oct 17, 2024 | 4 days | Develop functionality to read Word and Excel files using pandas & docx. |
|  | Develop text preprocessing module | Oct 18, 2024 | Oct 20, 2024 | 3 days | Implement tokenization, stopword removal, etc. |
|  | Develop theme extraction (LDA) module | Oct 21, 2024 | Oct 26, 2024 | 1 week | Implement theme extraction using LDA. |
|  | Develop sentiment analysis module (VADER) | Oct 27, 2024 | Oct 31, 2024 | 5 days | Implement sentiment analysis using VADER or another tool. |
|  | Develop result export module (Excel/CSV) | Nov 1, 2024 | Nov 4, 2024 | 4 days | Code the ability to export analysis results in a user-friendly format. |
|  | Front-end development (UI) | Nov 5, 2024 | Nov 9, 2024 | 5 days | Build a simple UI for file uploads and displaying results (optional). |
|  | Integration of all modules | Nov 10, 2024 | Nov 12, 2024 | 3 days | Ensure all modules work together seamlessly. |
| 5. Testing | Unit testing for individual components | Nov 13, 2024 | Nov 15, 2024 | 3 days | Test each module (input, analysis, export) independently. |
|  | Integration testing for end-to-end process | Nov 16, 2024 | Nov 18, 2024 | 3 days | Ensure the entire system works smoothly from start to finish. |
|  | User acceptance testing (UAT) | Nov 19, 2024 | Nov 22, 2024 | 4 days | Test with real data. |
| 6. Deployment | Prepare deployment environment | Nov 23, 2024 | Nov 24, 2024 | 2 days | Set up deployment platform (local or cloud-based). |
|  | Deploy application | Nov 25, 2024 | Nov 26, 2024 | 2 days | Final deployment of the software to the target environment. |
| 7. Maintenance | Bug fixing and adjustments based on feedback | Nov 27, 2024 | Dec 2, 2024 | 1 week | Address issues from UAT and ensure everything works smoothly. |
|  | Final system checks and documentation | Dec 3, 2024 | Dec 4, 2024 | 2 days | Final quality assurance and project documentation before submission. |

### System Analysis

The system analysis phase focused on understanding the Institutional Research (IR) team's needs and developing software to automate qualitative data analysis. The software was designed to handle data from focus groups, interviews, and surveys, providing essential features such as thematic and sentiment analysis and generating reports compatible with Word and Excel.

Client’s Needs and Requirements:

* Automating Data Analysis: The software automates the analysis of qualitative data, reducing manual work and using Natural Language Processing (NLP) algorithms for accurate and efficient analysis.
* Thematic and Sentiment Analysis: The software categorizes data into themes and performs sentiment analysis to classify responses as positive, negative, or neutral.
* Generating Reports and Visualizations: The system generates customizable reports and visualizations, helping analysts interpret and share insights effectively.
* Word and Excel Compatibility: The software imports and exports data in Word and Excel formats to integrate with the IR team’s workflow.

System Design and Architecture:

* Data Import Module: Handles Word and Excel documents for seamless integration.
* NLP Engine: Processes data for thematic and sentiment analysis.
* User Interface (UI): Simple and intuitive UI for easy navigation.
* Report Generation Tools: Customizable reports to highlight key findings.

Challenges Addressed:

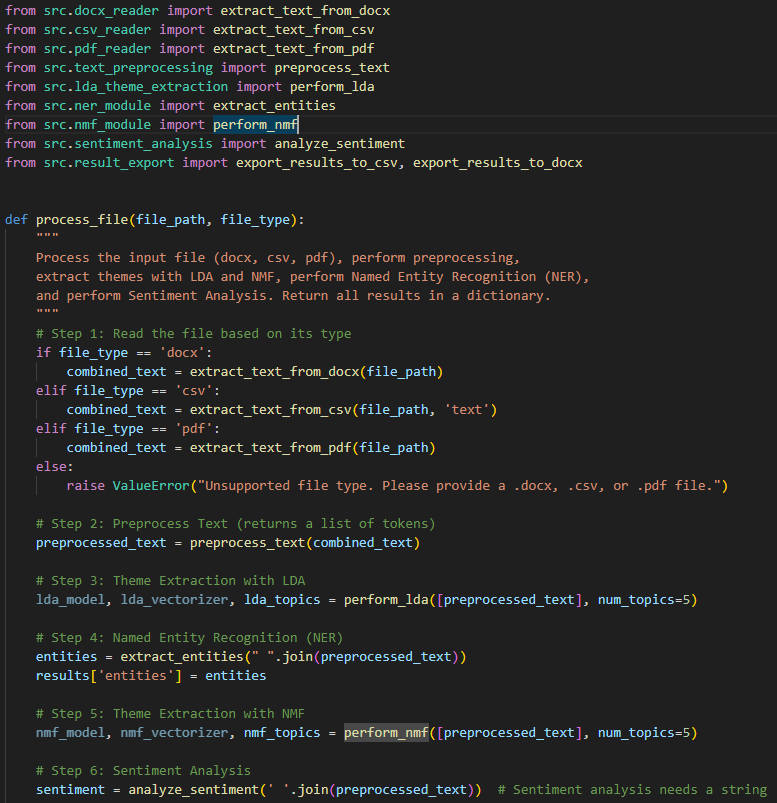
* Data Complexity: NLP algorithms process unstructured data with variations in language and tone.
* Time Efficiency: Automating analysis reduces the time spent on manual tasks, enabling analysts to focus on higher-level work.

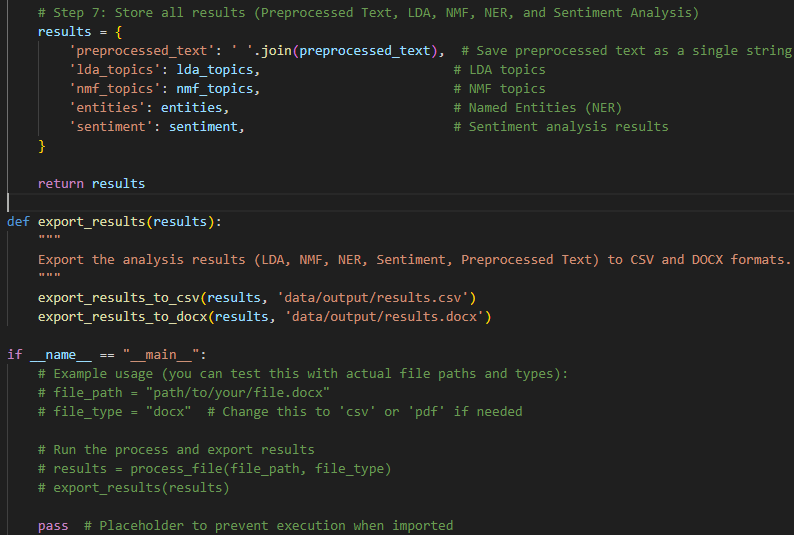
Conclusion: The system was designed to enhance efficiency by automating data analysis, improving accuracy, and streamlining workflows, meeting the client’s needs for thematic and sentiment analysis, report generation, and document compatibility.

**System Design:**

* Objective: The system design phase focused on creating the software’s architecture, user interface, and technical components that would enable the system to meet the requirements identified during the analysis phase.
* Activities:
  + Architecture Design: Designed a scalable and modular system architecture that included a data import module, NLP engine for thematic and sentiment analysis, and a reporting and visualization module.
  + UI/UX Design: Created wireframes and prototypes for the user interface, ensuring the software was intuitive and user-friendly for the IR team. Key features included easy navigation for data uploading, analysis selection, and report generation.
  + Algorithm Design: Developed the design for the NLP algorithms to automate the thematic and sentiment analysis processes. Ensured that the system could process varied qualitative data effectively and generate actionable insights.
  + Integration Plan: Designed the system to be compatible with Word and Excel for importing and exporting qualitative data, ensuring seamless integration with the IR team’s existing tools.

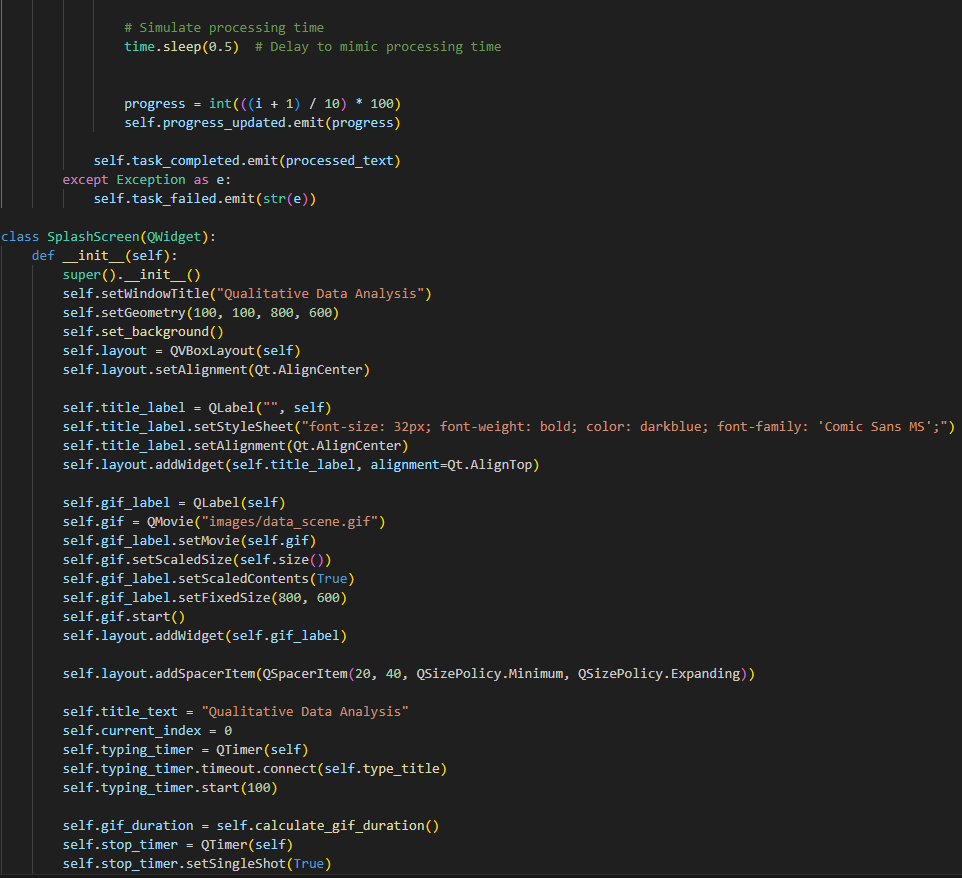
**main.py**

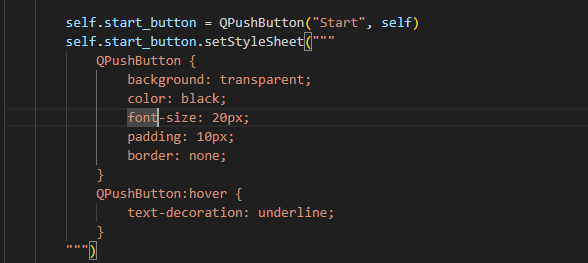
****

****

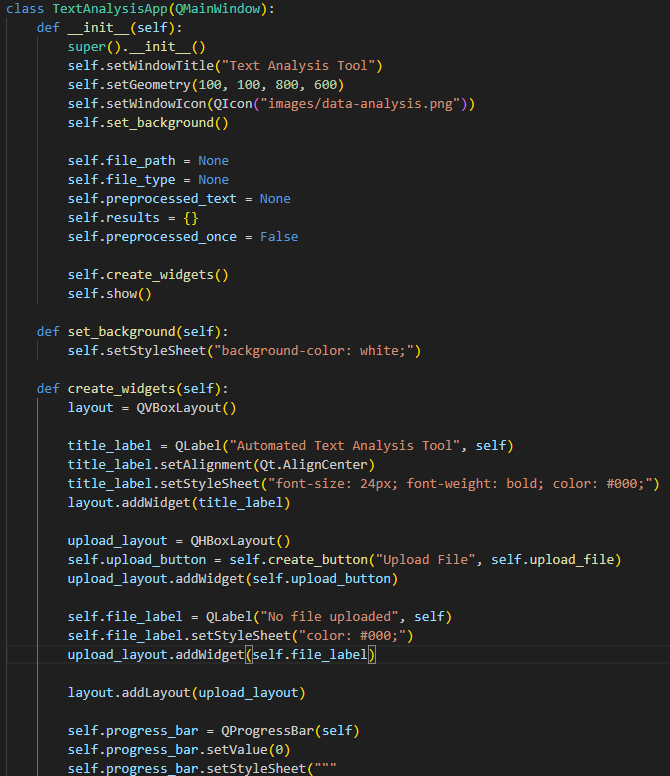
**gui.py**

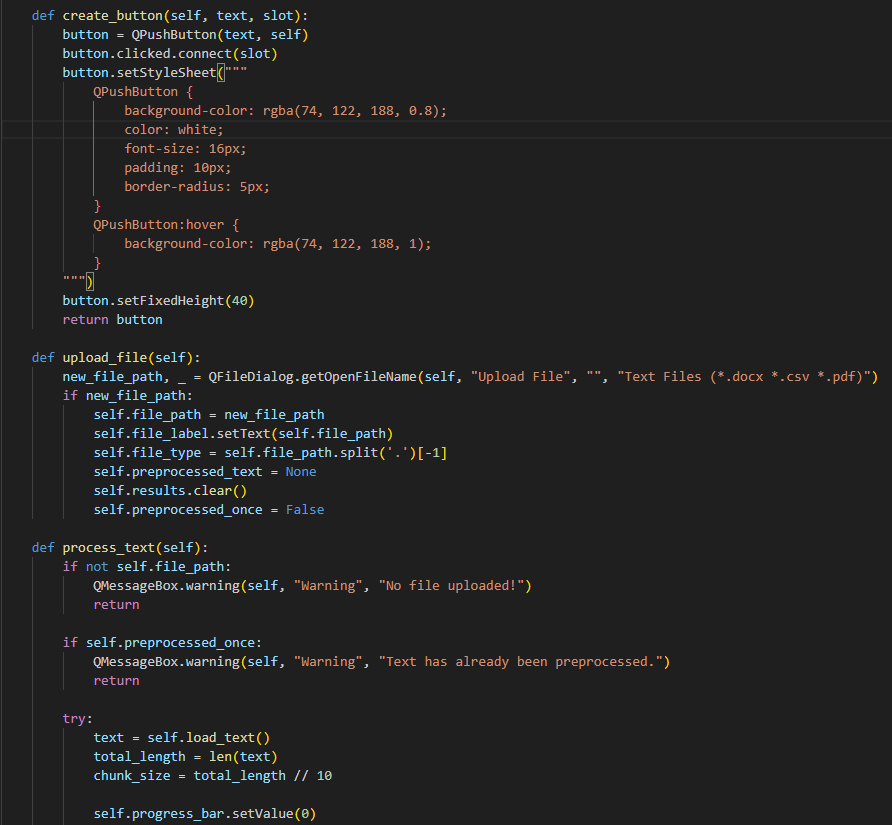
****

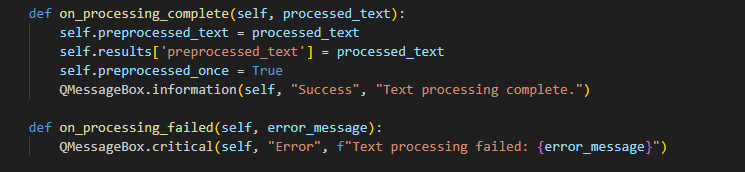
****

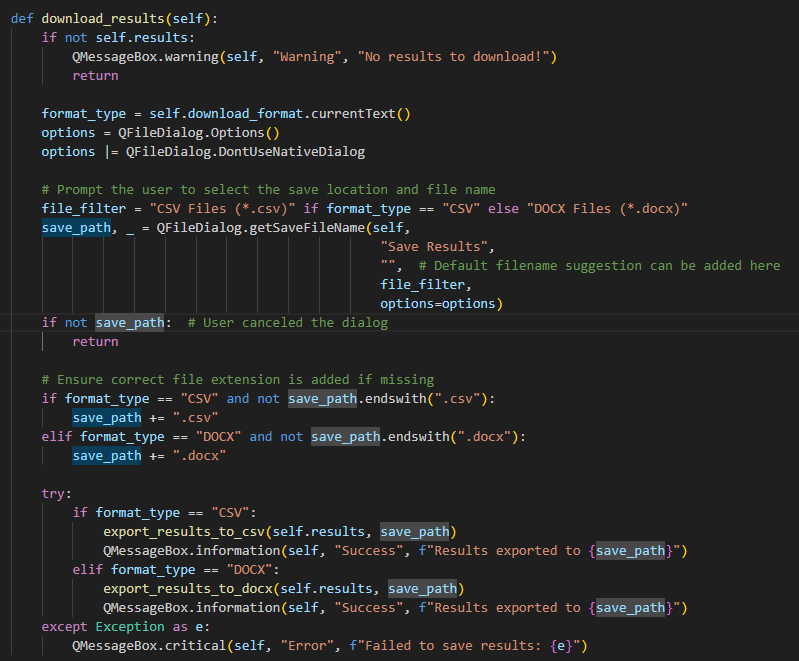
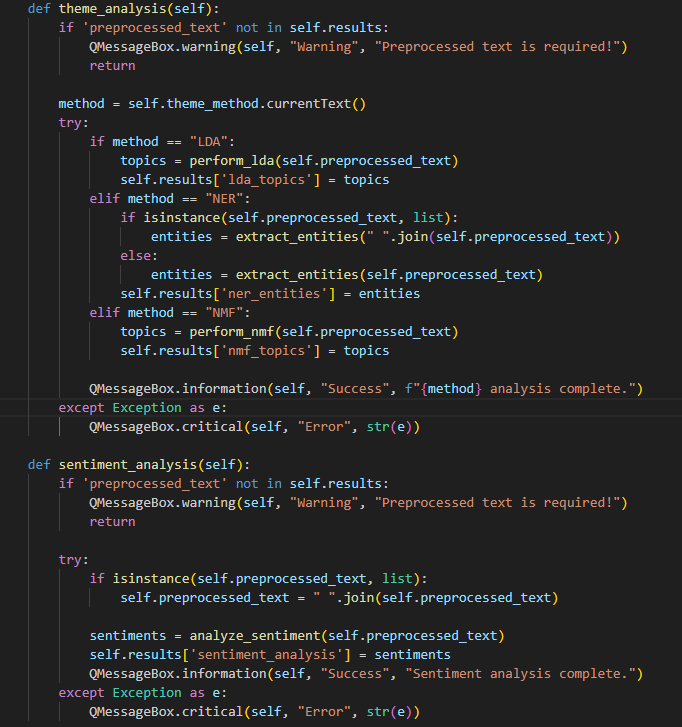
****

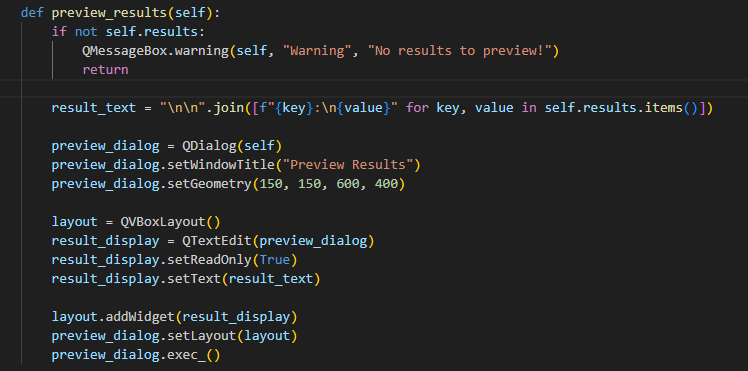
****

****

****

****

****

****

**System Implementation:**

* Objective: The system implementation phase involved the actual coding and development of the software based on the designs created in the previous phase. The focus was on building the data import module, NLP algorithms, and the user interface.
* Activities:
  + Data Import Module: Developed a module that could read and import data from Word and Excel documents, ensuring compatibility with the data formats used by the IR team.
  + NLP Algorithms: Implemented the thematic and sentiment analysis algorithms using advanced Natural Language Processing techniques to categorize and analyze qualitative data automatically.
  + User Interface Development: Built the user interface based on the design mockups, ensuring all key features were included, such as easy data upload, analysis options, and report generation.
  + System Integration: Integrated all system components to ensure smooth interaction between the data import module, analysis engine, and report generation module.
  + Version Control & Documentation: Used version control to manage the development process and documented the code and system features to ensure maintainability.

**System Testing:**

* Objective: The system testing phase focused on identifying and fixing any bugs or issues in the software to ensure it met the specified requirements and was fully functional.
* Activities:
  + Unit Testing: Performed unit testing on individual components, such as the data import module, NLP algorithms, and reporting functions, to ensure they worked as intended.
  + Integration Testing: Tested the integration of all components, ensuring the data flows correctly from the import process through analysis and into the report generation module.
  + Bug Fixing and Refinement: Addressed any bugs or issues identified during testing and refined the software for better performance and reliability.

**Acceptance, Installation, and Deployment:**

* Objective: The acceptance, installation, and deployment phase focused on preparing the software for the client’s use, ensuring proper installation, and providing the necessary documentation and training.
* Activities:
  + Acceptance Testing: Ensured that the software met all the client's requirements and expectations before final acceptance. This included testing the automation of data analysis, the accuracy of thematic and sentiment analysis, and the report generation capabilities.
  + Deployment: Deployed the software ensuring it was properly installed and configured for use with Word and Excel documents.
  + Documentation: Delivered technical documentation, covering the software’s features, installation process, and troubleshooting tips.
  + Post-Deployment Support: Provided post-deployment support to address any issues that arose after the software was in use, ensuring smooth operation.

**Tool Requirements and Technical Specifications**

#### 1. Programming Language: Python

Python is chosen for its versatility, extensive library support, and suitability for text analysis. It is widely used for machine learning and natural language processing (NLP) projects.

#### 2. Libraries

##### a. PyQt5

* Purpose: Provides tools to create a graphical user interface (GUI) for user interaction.
* Role:
  + Build an intuitive interface for users to upload files, process text, and export results.
  + Add interactive widgets like buttons, dropdowns, progress bars, and file explorers.

##### b. NLTK (Natural Language Toolkit)

* Purpose: Facilitates text preprocessing and sentiment analysis.
* Role:
  + Tokenizes text into sentences and words.
  + Removes stopwords and performs stemming or lemmatization.
  + Provides pre-trained models like Vader for sentiment analysis.

##### c. Scikit-learn

* Purpose: Implements machine learning algorithms, including LDA (Latent Dirichlet Allocation) and NMF (Non-negative Matrix Factorization).
* Role:
  + LDA: Extracts topics from text documents by identifying word distributions.
  + NMF: Decomposes the text into components for latent feature discovery.

##### d. NumPy and Pandas

* Purpose: Handles data manipulation and numerical computations.
* Role:
  + Pandas: Reads input files (CSV, DOCX) and organizes data into dataframes.
  + NumPy: Facilitates numerical transformations and matrix computations for algorithms.

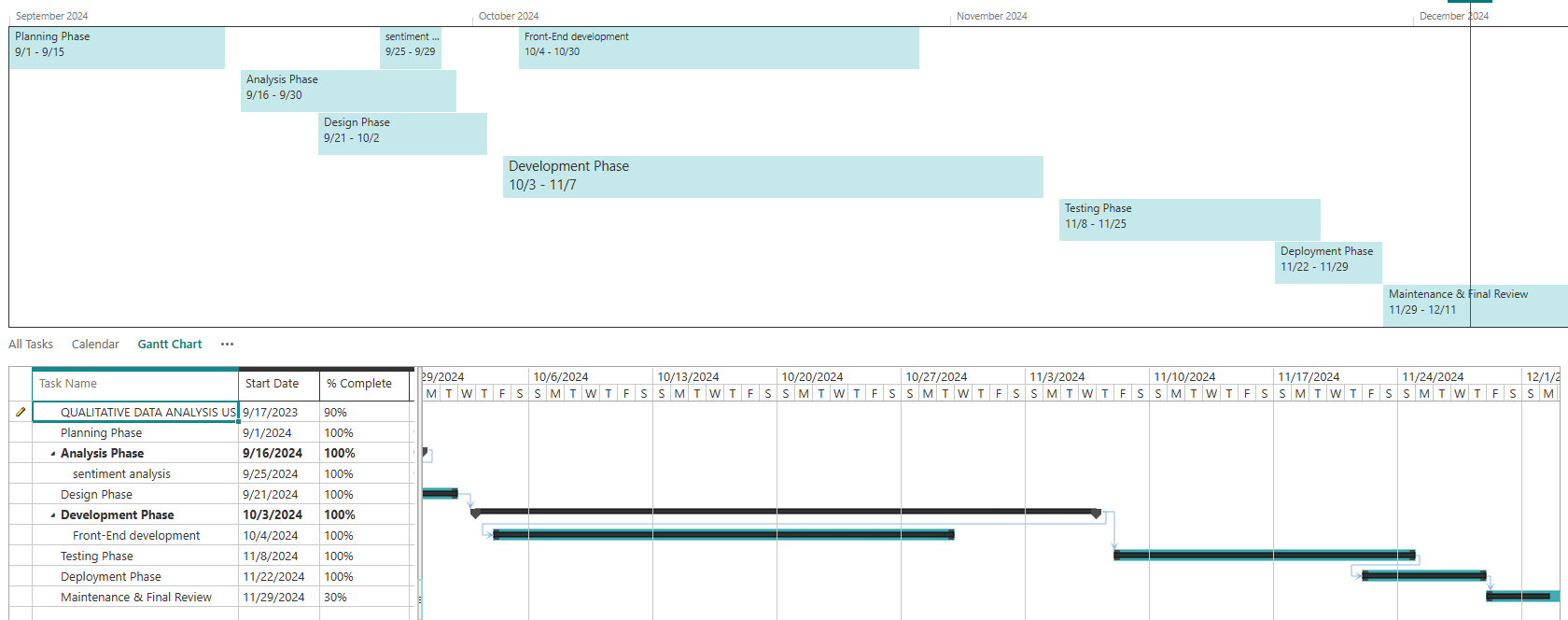
e. External Models: Pre-trained NER and sentiment analysis models.

f. File Support: Ability to process DOCX, CSV, and PDF.

g. Development Environment: IDE (VSCode, PyCharm) and Jupyter Notebook for testing.

**Gantt Chart**

**https://georgiancollege.sharepoint.com/sites/GC-AnalyzeQualitativeDatausingAI/Lists/MRP%20HighLevel%20Task%20List/gantt.aspx**

****

### Literature Review

The development of qualitative data analysis software addresses critical needs in modern data processing, such as efficiency, accuracy, and seamless integration into workflows. Advancements in technology and methods have informed the project design, ensuring it meets the challenges of analyzing qualitative data sets effectively.

**1. Automation in Qualitative Data Analysis**Manual analysis of qualitative data is time-consuming and resource-intensive, particularly when dealing with large datasets from sources such as interviews, surveys, and focus groups. Automated tools streamline this process by applying advanced algorithms to identify themes, trends, and insights. These tools not only save significant time but also reduce human error, enhancing the reliability of the results. Automation allows analysts to focus on interpreting data and drawing meaningful conclusions rather than spending time on repetitive tasks like coding and categorization.

**2. Natural Language Processing (NLP)**NLP is a cornerstone technology in qualitative data analysis, enabling systems to interpret and process unstructured text data efficiently. Techniques such as tokenization (breaking text into manageable units), sentiment scoring (quantifying emotional tones), and thematic analysis (identifying key themes and patterns) are essential for extracting insights. Modern NLP algorithms leverage machine learning to adapt and improve over time, providing more accurate results as the system processes diverse datasets. This capability ensures that the software is flexible enough to handle variations in language, phrasing, and context.

**3. Integration with Institutional Workflows**The success of new software depends on how well it integrates with existing systems and workflows. Institutional Research teams often rely on tools like Word and Excel for data collection and reporting. Ensuring compatibility with these formats minimizes disruptions and enhances usability. For instance, the ability to import data directly from Word and Excel documents streamlines the transition from raw data to analysis, while export options maintain consistency in reporting formats. Seamless integration reduces the learning curve for users and encourages adoption of the software.

**4. Addressing Data Complexity**Qualitative data is often unstructured, diverse, and complex, presenting challenges in analysis. Responses can vary widely in length, tone, and phrasing, requiring sophisticated algorithms to interpret them effectively. Modern approaches incorporate machine learning and artificial intelligence to handle this complexity, ensuring that the analysis captures subtle nuances and patterns in the data. For example, thematic analysis algorithms can group similar ideas across multiple responses, while sentiment analysis can provide a holistic view of emotional tones within the dataset.

#### **Deliverables**

#### **1. GUI Application**

**Description**: A fully functional graphical user interface (GUI) designed for seamless text analysis, built using PyQt5.

* **Features**:
  + **File Upload**: Supports file formats including DOCX, CSV, and PDF for analysis.
  + **Text Processing**:
    - Preprocessing (tokenization, stemming, stopword removal).
    - Theme extraction using LDA and NMF.
    - Named Entity Recognition (NER).
    - Sentiment analysis.
  + **Interactive Design**:
    - Dropdown menus for selecting algorithms (e.g., LDA, NMF, NER).
    - Progress bars to show the status of the analysis.
    - Buttons for actions like "Upload", "Process Text", and "Export Results".
  + **Result Preview**:
    - Displays intermediate and final results in an easy-to-read format.
* **User Experience**:
  + Designed for both technical and non-technical users.
  + Minimal learning curve due to intuitive button labels and tooltips.
* **Development Tools**:
  + **PyQt5**: Framework for creating the GUI.
  + **Python**: Core language for backend text analysis.
* **Benefits**:
  + Eliminates the need for multiple tools.
  + Provides an integrated solution for qualitative text analysis.

#### **2. Documentation**

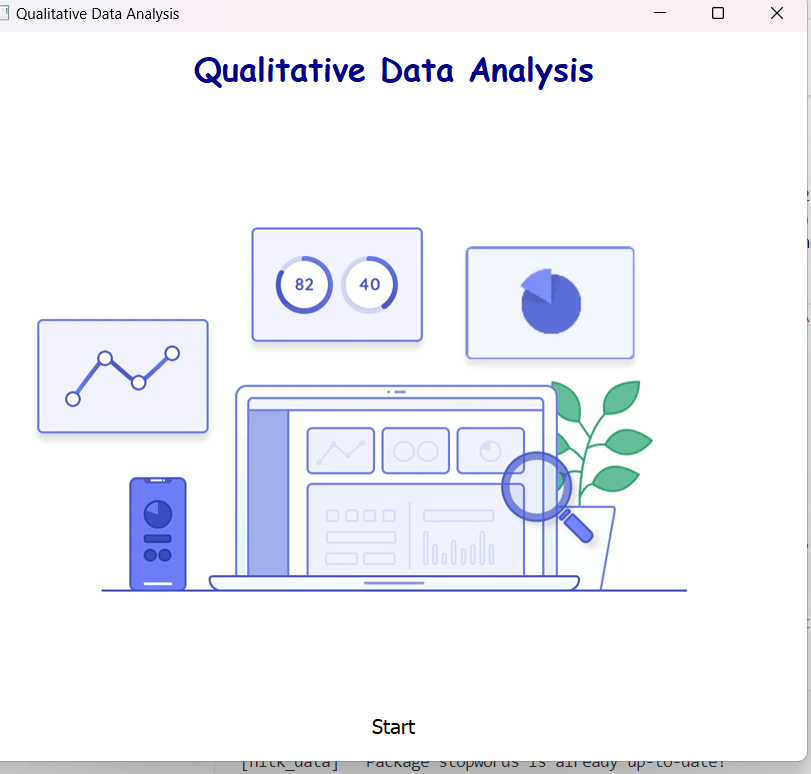
**Description**: Comprehensive documentation aimed at facilitating installation, usage, and troubleshooting.

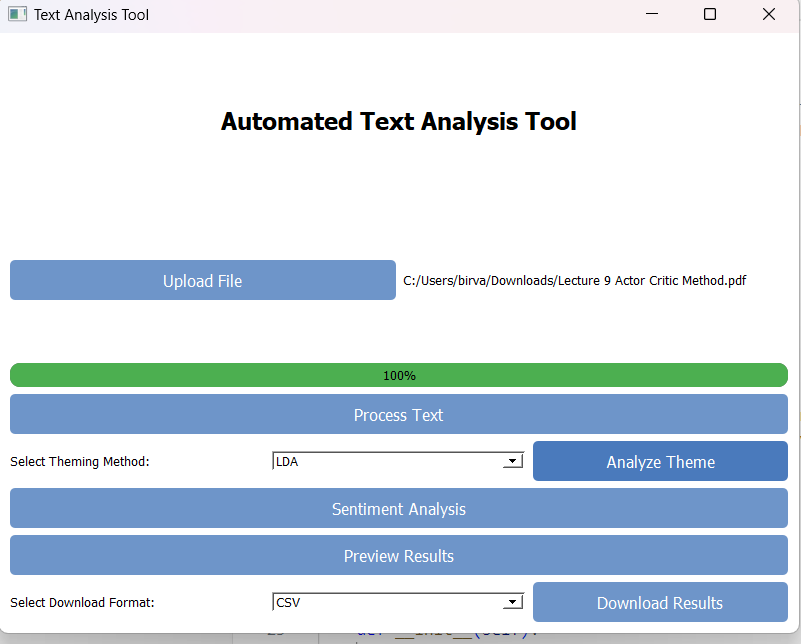
* **Components**:
  + **User Manual**:
    - Step-by-step guide on how to install the application, including dependencies.
    - Instructions on using key features like file upload, theme analysis, and sentiment extraction.
    - Screenshots to demonstrate GUI navigation.
    - Troubleshooting common issues (e.g., "File not supported", "Export failure").
  + **System Requirements**:
    - Hardware: Minimum specifications for running the application.
    - Software: Python version, required libraries, and supported operating systems.
* **Format**:
  + Written in a clear, step-by-step manner.
  + Delivered in both **PDF** formats for accessibility.

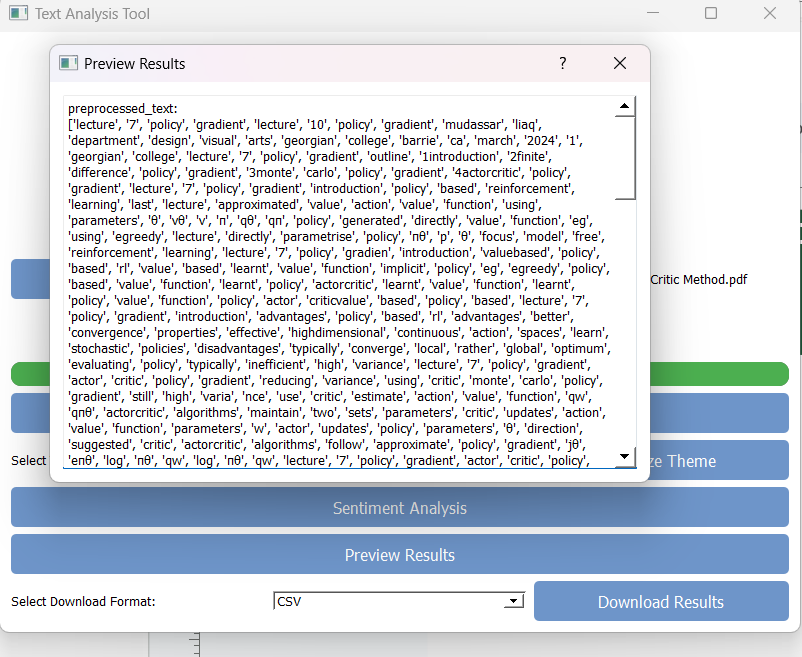
#### **3. Export Functionality**

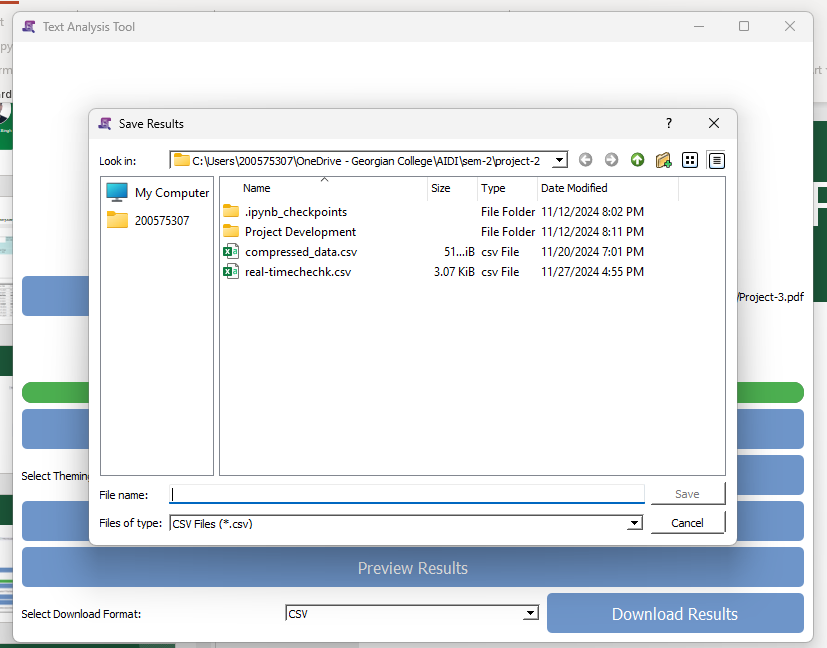
**Description**: Enables users to save analysis results in user-friendly formats like CSV and DOCX for further use.

* **Features**:
  + **CSV Export**:
    - Saves structured data, such as topics, entities, and sentiment scores.
    - Easily compatible with Excel or data visualization tools.
  + **Steps**:
    - Collect processed data.
    - Format data according to the selected export type.
    - Save the file in the user-selected location.
* **Benefits**:
  + Provides flexibility for users to use results in reports or further analysis.
  + Compatible with common software tools.
* **Snippets of Front end:**









### Conclusion

This project successfully automates the qualitative analysis of text data through an integrated application that combines theme extraction, sentiment analysis, and named entity recognition (NER). By utilizing advanced machine learning techniques such as LDA and NMF for topic modeling, along with pre-trained sentiment and NER models, the tool provides comprehensive insights into unstructured data.

The graphical user interface (GUI), built with PyQt5, offers an intuitive and user-friendly experience, enabling users to process multiple file formats (DOCX, CSV) effortlessly. The inclusion of export functionality in CSV and DOCX formats further ensures the tool's practical utility for researchers and professionals.

By streamlining the text analysis workflow, the project addresses limitations in existing manual and semi-automated approaches, offering scalability, efficiency, and ease of use. This solution bridges the gap between complex analytical methodologies and accessible, actionable insights, paving the way for future advancements in automated qualitative data analysis.