Jiang, B. et al. (Eds.) (2025). Proceedings of the 33rd International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

QUAiL: A Web-Based Qualitative Analysis Tool for Textual Data

Saket SONTAKKE[a*], Vishwas BADHE[b], Amit PAIKRAO[b] & Ramkumar RAJENDRAN[b]
[a]*Dept. of Computer Engineering and Technology, Dr. Vishwanath Karad MIT-WPU, India*
[b]*Centre for Educational Technology, IIT Bombay, India*
*sontakkesaket9@gmail.com

Abstract: Qualitative Data Analysis is an intensive, time-consuming, and painstaking process. While existing Computer-Assisted Qualitative Data Analysis Software (CAQDAS) programs are available to enhance this process, they often divide features across multiple tiers and require a significant learning curve for effective use. This paper introduces "QUAiL", a web-based application designed to streamline the qualitative data analysis workflow. Developed using the MERN (MongoDB [1], Express.js [2], React [3], Node.js [4]) stack and a Python microservice for statistical calculations. QUAiL integrates features such as text and audio file import with transcription services, a user-friendly interface for flexible coding, highlighting, and adding memos. Additionally, it provides comprehensive code management utilities like code merging, code splitting, data visualization, and quantitative validation through statistical tests. The aim is to remove the burden of complexity and a fragmented workflow, offering students, educators, and researchers an accessible tool designed to be intuitive and to support entire qualitative workflow.

Keywords: Qualitative Data Analysis, CAQDAS, MERN Stack, Educational Technology, Mixed-Methods Research.

1. Introduction

Verbal analysis provides profound insights into human behaviour, however this process is challenging. Managing coding, and analyzing qualitative data, such as interview transcripts or classroom observations, is a laborious, time-consuming, and difficult task that is hard to scale for larger studies (Brailas, A., et. al. 2023; Kraiwanit, T., et.al, 2023). Computer-Assisted Qualitative Data Analysis Software (CAQDAS) tools (Nason, E. E., et. al., 2023; Rampin, R., & Rampin, V. 2021) help researchers to systematically organize, code, and analyze qualitative data, but they are proprietary and have high licensing fees. A more significant issue is they often fall short of providing meaningful bridges between qualitative insights and quantitative verification. This compels researchers into a multi-step process of manually exporting data to separate programs for statistical verification. This disjointed process is inefficient, error-prone, and disrupts the iterative research cycle. To overcome these issues, we created a web-based application at the Educational Technology Lab at IIT Bombay, which is designed to be an accessible, comprehensible, and dynamic tool for the entire qualitative workflow. This makes QUAiL ideal for students, teacher-educators, and academic researchers. This paper will review the system architecture and main features of QUAiL, provide a practical use case in educational research, and discuss its broader effects on the field of qualitative research.

[1] MongoDB is a leading NoSQL ("Not Only SQL") database that stores data in flexible, JavaScript Object Notation (JSON) like documents, rather than the rigid row-and-column structure of traditional SQL databases.

[2] Express.js is a Node.js web framework that simplifies building APIs and websites through easy HTTP request handling and URL routing.

[3] React is an open-source JavaScript library used for building user interfaces.

[4] Node.js is a back-end runtime environment that runs JavaScript outside the browser, enabling server-side development and APIs.

2. System Architecture and Core Technologies

QUAiL adopts a sophisticated, decoupled architecture, which was intentionally designed to use optimal tools for each specific task. In the system MERN-stack application provides the foundational infrastructure, and the calculations are performed via an independent Python microservice. Although performing these analyses using a JavaScript library is possible, this approach allows us to utilize SciPy, a rigorous and reputable academic library in Python, and leverage a language specifically designed for scientific computing. Additionally, delegating these computation-centric tasks to a separate server improves performance and allows for the easy integration of additional data science tools in the future. Figure 1 shows the architecture of QUAiL, following subsections discusses each block of architecture in details.

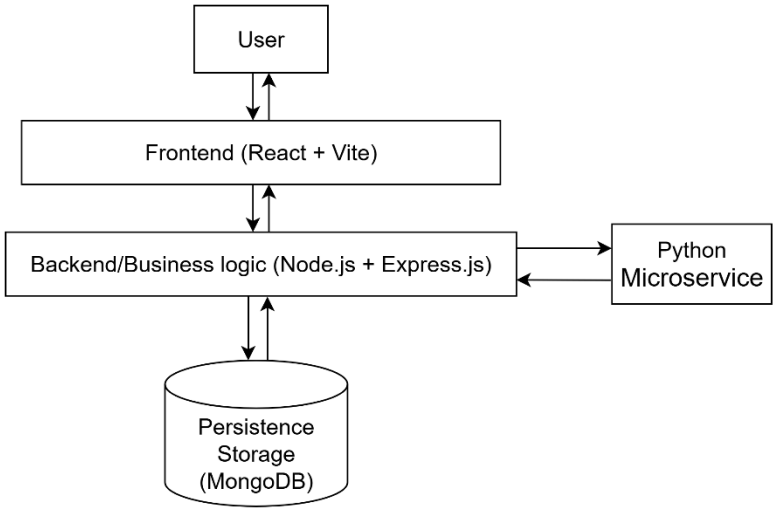

Figure 1. The Architecture diagram of QUAiL.

2.1 Database Design (Persistence Storage)

The persistence layer uses MongoDB, a NoSQL document database, along with Mongoose[5], which provides schema enforcement via Object Data Modelling. An important architectural decision was to use a document-embedded data model in which each project is saved as one document with its relevant data, including import files, code definitions, coded segments, memos, and highlights, all arranged as nested sub-arrays. The use of a document-embedded data model provides three significant advantages. First, it enables faster retrieval through data locality because all the relevant data is grouped together in a single Binary JSON (BSON) document. Second, it ensures transactional integrity when modifying multiple interconnected data elements, as all the changes to the project are done within a single document operation. Third, the use of a document-embedded data model supports schema flexibility, allowing structural changes without the cost of migrations. While the database remains schemaless, Mongoose's schema is enforced for validation purposes during development.

2.2 The Backend Core (Node.js & Express.js)

The backend is a modular API built with Node.js and Express.js and is built to organize the backend with formal separation of concerns across three functional domains. The Authentication module is responsible for user identity and access management, and session management utilizing JSON Web Token (JWT) to persist user sessions on the client-side, and route-protection employing middleware. The Project Management module is the heart of the application, providing comprehensive create, read, update, and delete (CRUD) operations. It also handles the complexities of data integrity with good design patterns like cascading deletes.

[5] Mongoose is a library that provides a structured way to interact with MongoDB, a database that uses a flexible, schema-less data model.

Furthermore, the module oversees complex asynchronous workflows, such as file uploads and coordination of third-party transcription, etc. The project management module additionally supports a reporting form for exporting structured JSON data into spreadsheets with custom styles. The Statistical Analysis module serves as a preprocessor and is responsible for transforming the data into a suitable format and sending the required data collected by the platform to a dedicated Python microservice, thereby offloading computation to an independent server while retaining responsiveness and enabling scalability for the entire architecture.

2.3 The Frontend Experience (React & Vite)

The frontend of the application was developed as a modern Single Page Application (SPA), using React and Vite[6] accomplishing high build speeds and rapid development speed. The User Interface (UI) follows a component-based design, creating a modular and reusable design that allows for scalability and maintainability. The application uses React hooks to manage local state as well as the “React Context API” for global state management for user authentication, as well as theme variables (light/dark mode). Ultimately, the UI focuses on providing a good user experience while minimizing disruption, allowing for a contextual and interactive workspace. It leverages libraries like “Framer Motion” to smoothly animate UI transitions and “Recharts” for data visualization. The major user actions (e.g., coding, adding memos, or navigating documents) are facilitated through contextually relevant modals and toolbars (not full-page transitions). Additional application functionality (e.g., code manipulation, statistical tools, etc.) is built into the application in an easily accessible way that is presented as modals, confirmation prompts, and guided user workflows.

2.4 The Decoupled Statistics Microservice

A significant architectural choice was to separate the statistical processing from the core application logic by creating a standalone Python microservice. This is consistent with the API Gateway Design Pattern. In this approach, the Node.js controller will first retrieve and validate any data from MongoDB, apply any required transformations (like creating contingency tables), and then send a clean payload to the microservice. This takes advantage of Python's strong data science ecosystem (including libraries like SciPy and NumPy) in order to conduct statistical computations in the most efficient and flexible manner. The microservice is highly modular and scalable, as any number of different statistical test types can be requested through a single generic endpoint, with room to add enhancements in the future. The microservice will return results in structured JSON format, which includes both the raw statistics as well as plain language explanations, facilitating access to statistical findings for users who are not specialists in this area, and allowing for better dissemination of the findings.

3. Core Features

3.1 Multi-Modal Data Import and Transcription Service

The application supports the import of both audio and text files (see Figure 3). The audio files are processed using a built-in transcription system, which automatically transcribes the audio to text. Furthermore, the import module provides segmentation options (see Figure 4) that include turn-wise (speaker) segments and sentence-wise (grammatical) segments.

3.2 Document Edit Mode

[6] Vite is a modern build tool that provides a fast development environment for JavaScript libraries like React.

The system provides a controlled editing environment (see Figure 5) for data preparation before the analysis can begin. Inside edit mode, users can perform tasks such as text correcting, changing sentence structure, and changing text formatting etc. For audio files, an integrated audio player is provided to allow users to reference the original audio while editing the transcript. Upon completion of edits, the documents are locked using a "Lock and Save" function, which prevents any further editing and preserves data integrity.

3.3 Main Workspace

The main workspace (see Figure 6) includes a multi-panel workspace comprising all analytic elements. The central document viewer displays text documents and, in the case of audio transcribed files, is accompanied by an integrated audio player. The resizable sidebar contains the navigation elements for the project (list of files, code definitions, an organized list of coded segments, and memos). The project elements update in real-time in response to any user action, reflecting the latest state of the project.

3.4 Dynamic Codebook

Code definitions can be defined and applied on-the-fly with attributes such as names, descriptions, and visual identifiers. The platform also supports advanced code management features like code splitting (see Figure 7) and merging (see Figure 8). This is supported by guided wizards and confirmation prompts to prevent data loss.

3.5 Project Overview and Statistical Analysis

This provides multiple analytical perspectives as tables, visualizations, and statistical analysis. The “Table View" consists of organized code segments (see Figure 9). The “Visualization” tab (see Figure 10) renders various charts that visually convey distributions of themes and their relationships. Finally, the “Statistical Analysis” (see Figure 11) module provides quantitative validation of qualitative results. The system automatically validates the underlying assumptions to ensure the credibility of the results. The statistics report includes statistical values, significance indicators, and narration on how to interpret the results (see Figure 13).

3.6 Multi-Format Data Export System

The export functionality offers multiple output formats to support varying user needs and to help ensure that the results and findings are transportable for use across multiple academic settings and research environments. Project data can be exported in excel format for external analysis as a table (see Figure 14), one-hot encoded matrix, or overlapped code segments, depending on user requirements. Visualization components can be exported as high-resolution PNG images, and complete statistical analysis reports can be exported as formatted PDF documents.

4. QUAiL’s Competitive Positioning

In the current CAQDAS environment, barriers to entry are large for academic users, with annual pricing around $250-$740 for tools like NVivo (Kraiwanit, T., et al., 2023), ATLAS.ti (Muhr, T., 1991), and MAXQDA (Kuckartz U., et al., 2019). While these commercial products have advanced features, they typically separate functionality into premium tiers or charge additional costs. For example, transcription services often require a separate subscription, collaboration is limited, and advanced statistical analysis is restricted to premium versions. In addition to these options, there are free and open-source tools available, but they usually do not offer transcription, advanced code management, and/or statistical verification as part of the same platform. This creates fragments in workflow where users can only use the system for certain

steps of qualitative analysis and then need to switch to alternative applications to complete the leftover steps. QUAiL provides a solution to these barriers with a completely web-based, open-source application that includes transcription, advanced code management, and statistical tools. By combining all of these features within a single application, QUAiL will improve research tool availability and reduce fragmentation in the research workflows.

5. Use Case Demonstration in Educational Research

Educational researchers usually work with qualitative data such as interviews and classroom observations to develop a rich understanding of both teachers' and students' experiences. For demonstration purposes, 3 sample interview transcripts and 1 audio interview were generated using GenAI to illustrate the end-to-end workflow of the application. The goal is to answer the research question: *"How do different stakeholders (administrators, teachers, students, and experts) perceive the role of technology in education?"*. Figure 2 illustrates an abstract workflow that a user might follow while interacting with QUAiL.

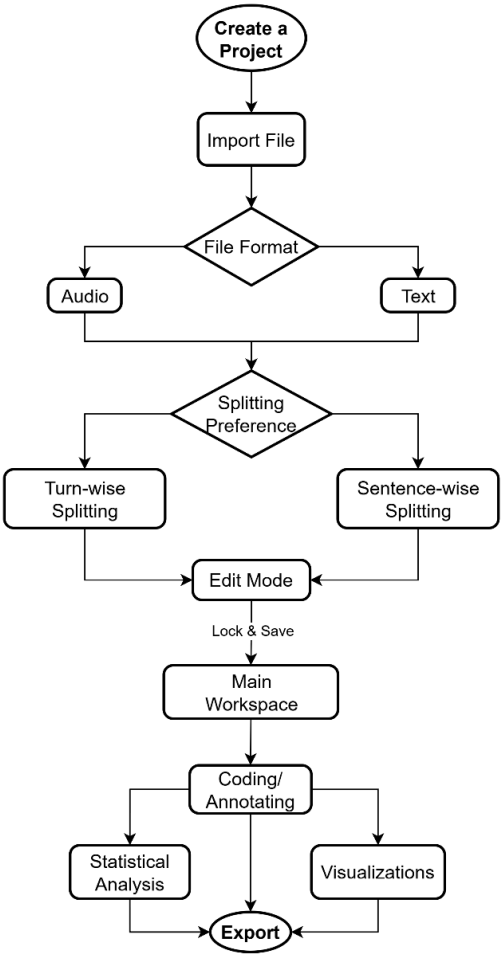

Figure 2. An Abstract User Workflow.

5.1 Data Import and Edit Mode

The researcher creates a demo project and clicks on the "Import File" button in the left panel. Figure 3 shows import options, then s/he clicks "Import Audio" to import the interview audio file and in the next step, s/he chooses "Turn-wise Splitting" preference as portrayed in Figure 4. Once the transcript is imported, the system now enters "Edit Mode" which can be seen in Figure 5. S/he makes all the necessary changes within the edit mode itself and then locks and saves the document. Similarly, s/he imports the remaining text transcript files.

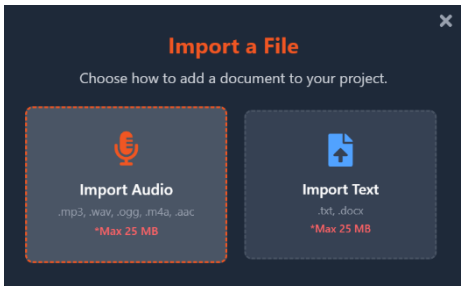

Figure 3. File import options.

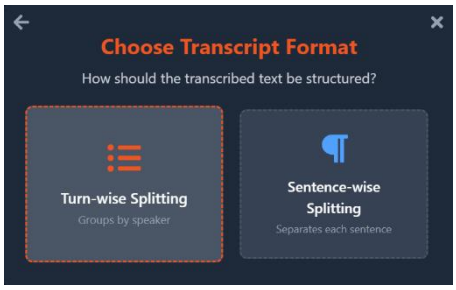

Figure 4. Splitting Preferences.

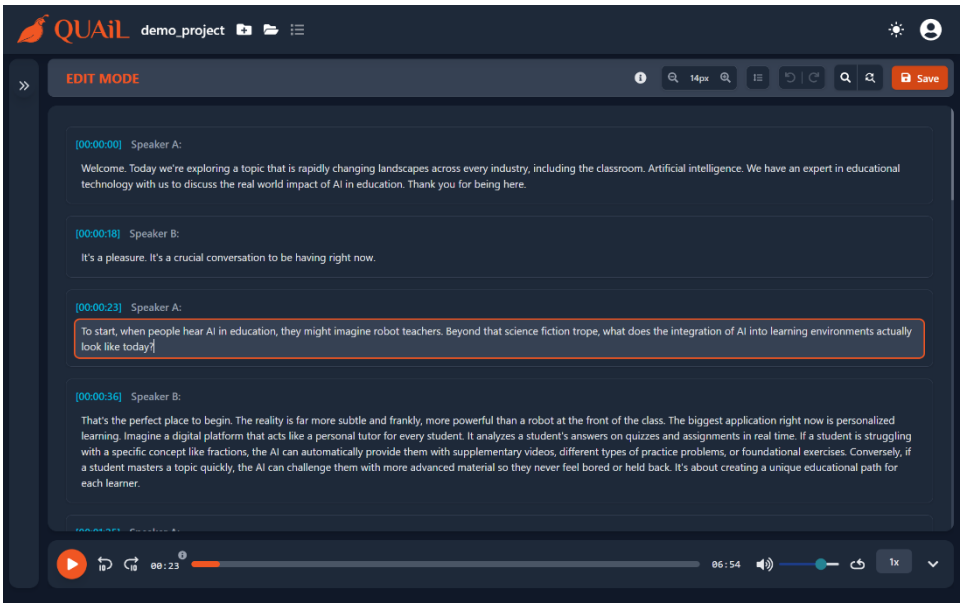

Figure 5. Edit Mode.

5.2 Main Workspace

Figure 6 presents the main workspace of QUAiL. This is where the researcher will spend the majority of his/her time. S/he clicks on “Define New Code” and adds a new code, “Technical & Logistical Issues” and its respective definition. Then s/he selects a text segment and assigns a code from the list. S/he defines new codes as and when required.

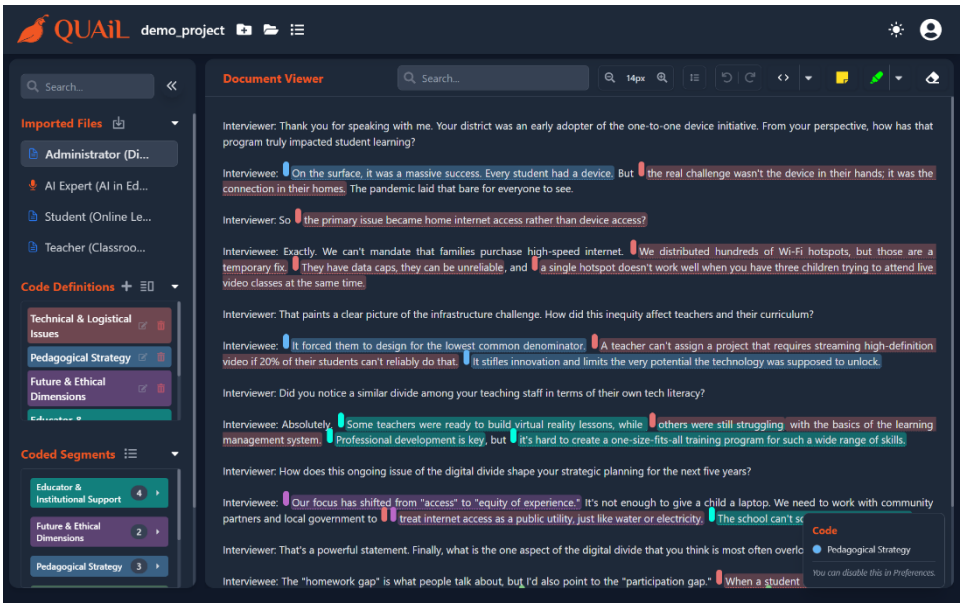

Figure 6. The Workspace of QUAiL

5.3 Iterative Analysis: Splitting and Merging Codes

The researcher expands or refines the codebook as the analysis progresses. As demonstrated in Figure 7, s/he can merge codes with overlapping context (“Human & Social Factors”, “Learner Experience”) together into a single code (“Social & Emotional Engagement”). Additionally, as shown in Figure 8, s/he can split a broader code (“Stakeholder Experience”) into two finer-grained sub-codes (“Learner Experience” & “Educator & Institutional Support”). Using a wizard, QUAiL guides him/her through a review of the original re-coding of each segment of data.

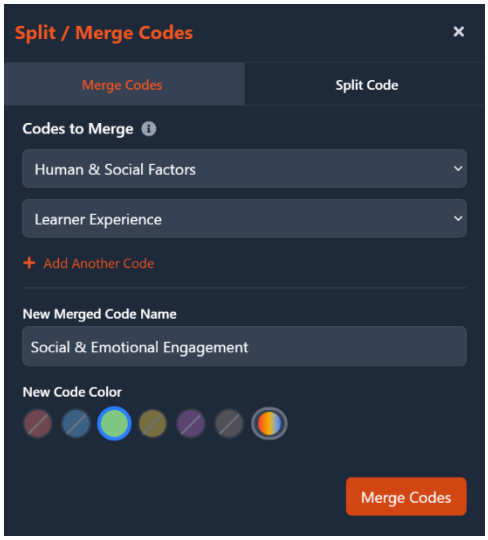

Figure 7. Merge Codes Window.

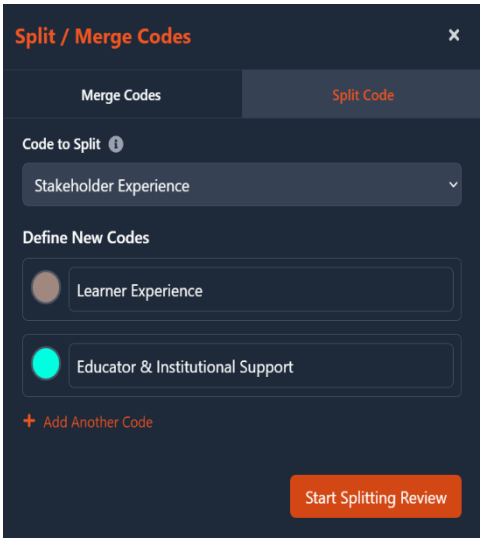

Figure 8. Split Code Window.

5.4 Project Overview

Figure 9 and Figure 10 illustrate subsections of the “Project Overview” window where the researcher can get quick summaries of the entire project. S/he uses the “Table View” and the “Visualizations” tab separately or together as per his/her requirements, which helps him/her analyze the project data from a detailed or a broader perspective.

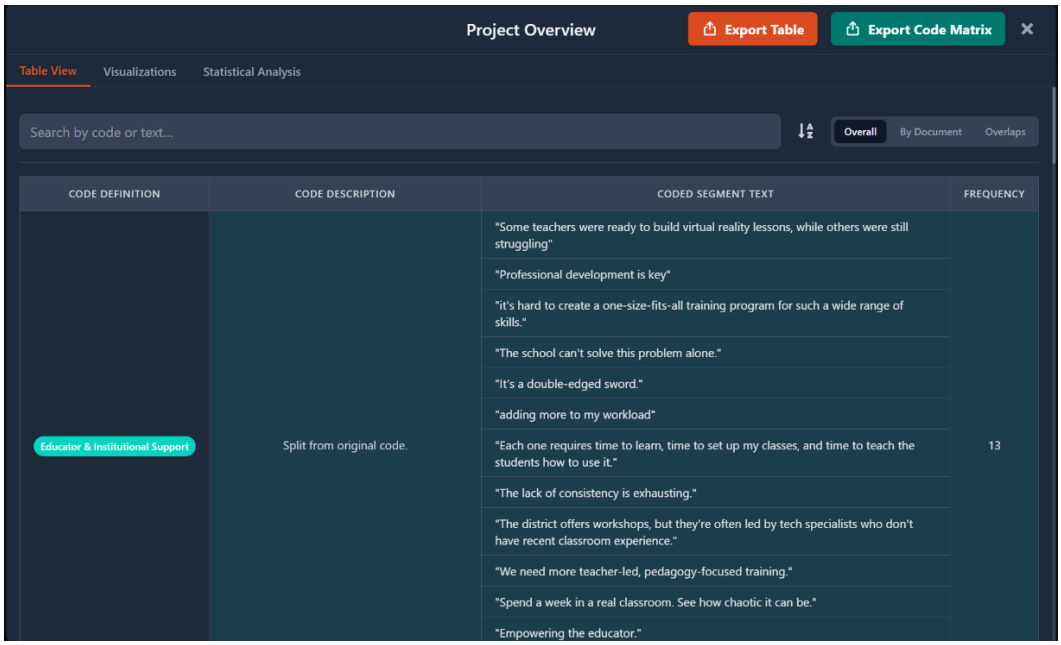

Figure 9. Project Overview - Table View tab.

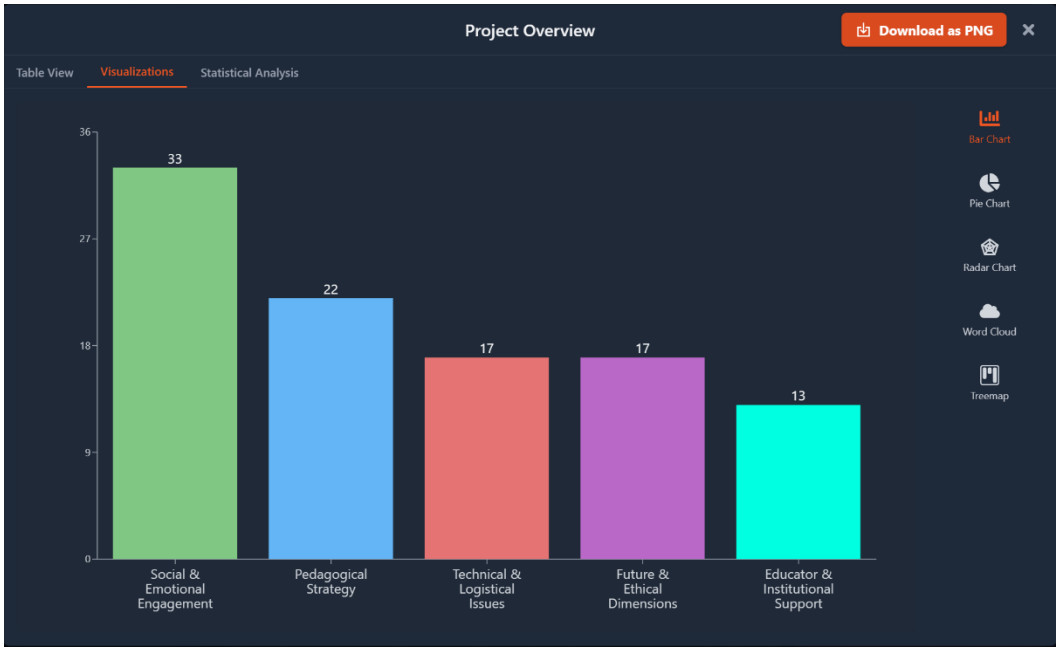

Figure 10. Project Overview - Visualizations tab.

5.5 Quantitative Validation with Chi-Square Test

Next, the researcher accesses the 'Statistical Analysis' module. S/he groups the four transcripts into two groups: Group A (Educator Perspective): Administrator and Teacher transcripts; Group B (External Perspective): Student and AI Expert transcripts. As an important methodological action, the tool begins by presenting a checklist of the assumptions that are prerequisites to the type of statistics being tested, as shown in Figure 11.

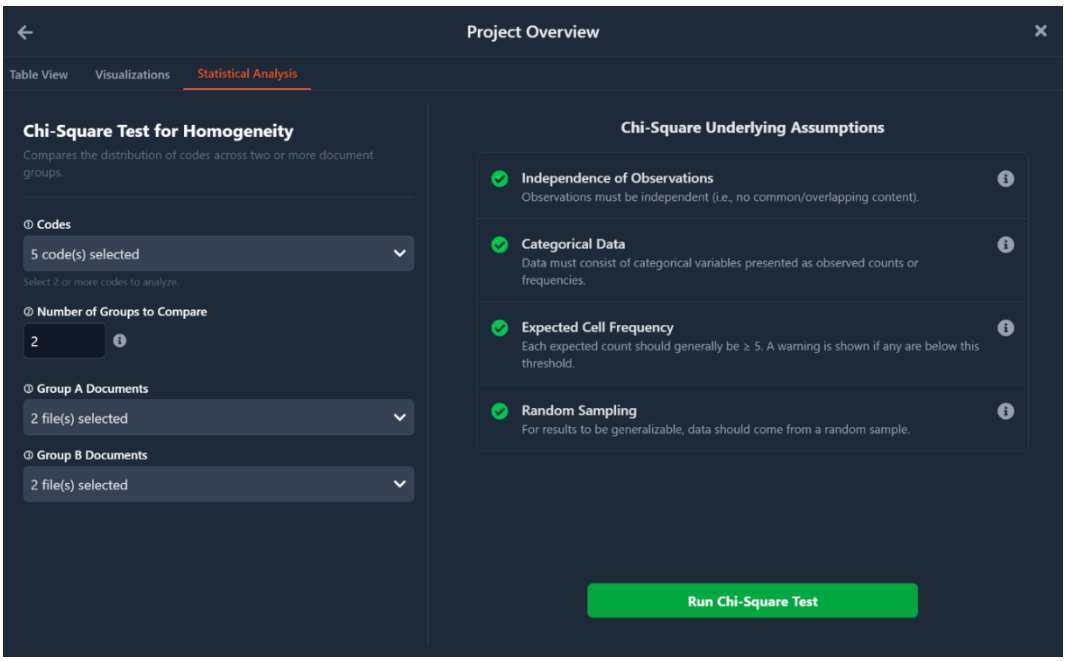

Figure 11. Underlying Assumptions Checklist.

Once the researcher verifies these assumptions, the application calculates the Chi-Square statistic and its p-value. The results, depicted can be seen in Figure 12 and Figure 13, are presented in clear, comprehensible explanations and interactive charts. S/he may export this test result as a PDF to report his/her findings.

Figure 12: Chi-Square Test of Homogeneity Result

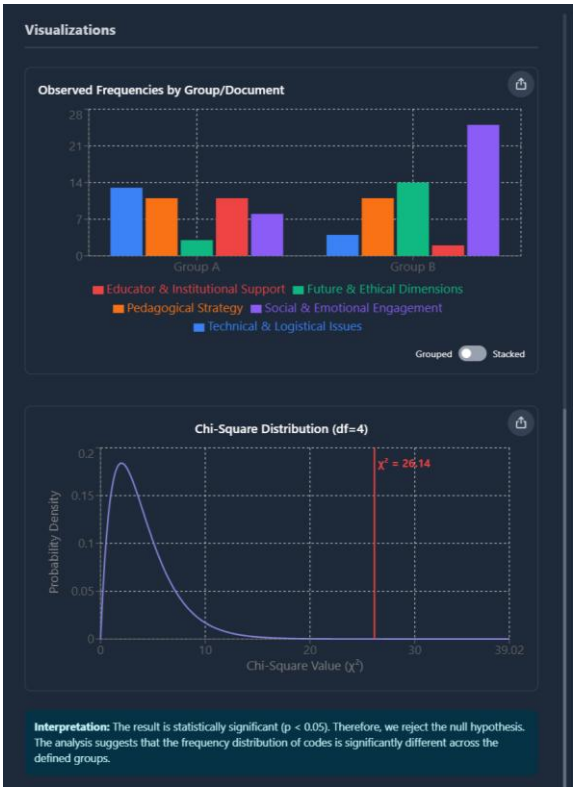

Figure 13: Chi-Square Test of Homogeneity Visualizations

5.6 Data Export

To export his/her findings, the researcher can export the overall coded segments table as demonstrated in Figure 14. If the need arises, s/he may export the project data in the form of a one-hot encoded code matrix. S/he may also export high-quality images of charts to support his/her findings.

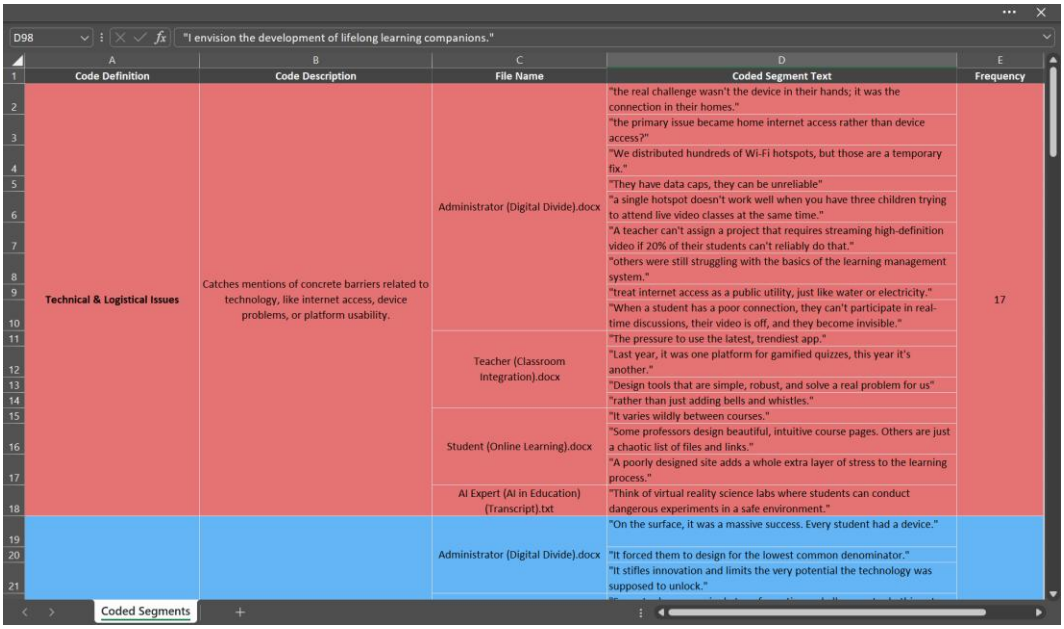

Figure 14. Overall Coded Segments Export Table.

This use case showcased a demonstration of QUAiL and its integrated workflow through the entire process from importing data, qualitative coding, code management, and finally statistical analysis. The application provides a single, unified platform for this entire process.

6. QUAiL's Broader Impact on Teachers

The broader significance of QUAiL on the field of qualitative research lies in its role as an open-source tool for examining classroom interactions ranging from student group discussions to active learning sessions and assessments. By providing an alternative to proprietary software, QUAiL enables educators to analyze discourse and interaction data, thereby supporting evidence-based improvements in teaching practices and curriculum design. This helps educational institutions to conduct research on teaching and learning practices without any constraints or hurdles. Moreover, the platform's accessibility encourages the use of qualitative methodologies across various fields, promoting multidisciplinary research. QUAiL's ease-of-use and guided workflows reduce the technical barrier, which allows users to focus on their data and bypass a steep learning curve. As a result, more teachers can perform their own analyses independently, leading to better, faster insights.

7. Conclusion and Future Work

QUAiL is a web-based platform developed to reduce barriers and fragmentation in qualitative research. It offers a comprehensive, open-source, and user-friendly web-based toolset, from transcription to integrated statistical validation. This enables students, educators, and researchers to perform thorough qualitative analysis in a time-effective manner. The tool is currently in the post-development stage, and we are actively seeking feedback to guide its evolution. Future work will prioritize real-time collaborative capabilities to support research teams. We will also be expanding the statistical module to include even more statistical tools, as we aim to position it as a prominent platform for mixed-methods research.

References

Brailas, A., Tragou, E., & Papachristopoulos, K. (2023). Introduction to qualitative data analysis and coding with QualCoder. *American Journal of Qualitative Research*, *7*(3), 19-31.

Kraiwanit, T., Limna, P., & Siripipatthanakul, S. (2023). NVivo for social sciences and management studies: A systematic review. Advance Knowledge for Executives, 2(3), 1-11.

Kuckartz U, Rädike S. Analyzing Qualitative Data with MAXQDA. Switzerland: Springer International Publishing (2019). doi: 10.1007/978-3-030-15671-8

Muhr, T. ATLAS/ti — A prototype for the support of text interpretation. *Qual Sociol* 14, 349–371 (1991). https://doi.org/10.1007/BF00989645

Nason, E. E., Wang, K., & Ausbrooks, A. R. (2023). A User-Friendly Introduction to RQDA for Qualitative Research: Recommendations for Social Work Students and Educators. *Journal of Social Work Education*, *59*(3), 831-847.

Rampin, R., & Rampin, V. (2021). Taguette: open-source qualitative data analysis. *Journal of Open Source Software*, *6*(68), 3522.