PharmaSUG 2025 - Paper AI-362

AI-Powered Data Issue Tracker for Efficient Data Issue Tracking and Resolution

Bharath Donthi, Statistics & Data Corporation

ABSTRACT

Traditional methods for documenting and resolving clinical data issues, often reliant on spreadsheets shared among statistical programmers, statisticians, clinical data managers, and sponsors, present significant inefficiencies. These methods can lead to redundant reporting of similar issues, difficulty in tracking resolution status, inadequate detection of duplicates, and a lack of comprehensive audit trails. To address these limitations, a system incorporating Artificial Intelligence (AI) has been developed. This system facilitates structured issue entry, assignment, real-time status tracking, and maintains a robust audit history. Key AI components include semantic analysis for duplicate issue detection, natural language processing for user interaction, and automated generation of SQL queries for data investigation. The objective is to reduce manual effort, improve the consistency and timeliness of responses, and ultimately enhance the quality and integrity of clinical research data through a more structured and intelligent issue management process.

INTRODUCTION

Maintaining high-quality data is fundamental in the pharmaceutical and clinical research industries for regulatory compliance and informed decision-making. Current practices for identifying, documenting, and resolving data discrepancies frequently involve manual processes centered around shared documents like Excel spreadsheets. While familiar, these methods introduce several challenges:

- **Redundancy:** Different reviewers may independently identify and report the same or semantically similar issues, creating unnecessary workload and complicating tracking.
- **Lack of Centralized Tracking:** Dispersed documentation makes it difficult to maintain a real-time, consolidated view of issue status and resolution progress.
- **Extended report turnaround time:** Requiring programmer intervention to generate reports.
- **Insufficient Audit Trails:** Manual tracking often lacks the granularity and immutability required for rigorous audit trails, potentially compromising compliance and oversight.
- **Delayed Resolution:** Communication overhead, manual handoffs between teams (e.g., requesting data pulls from programmers), and difficulty identifying systemic patterns can delay issue resolution.
- **Compromised Data Integrity:** Inefficiencies in issue management can lead to unresolved data problems, impacting the overall quality and reliability of the clinical dataset.

To mitigate these challenges, we propose an AI-powered system designed to centralize issue management, automate repetitive tasks, and provide intelligent assistance throughout the issue lifecycle. By integrating techniques such as natural language processing (NLP), semantic similarity analysis, and automated query generation, the system aims to improve efficiency, collaboration, and data integrity in clinical data management workflows.

SYSTEM ARCHITECTURE AND FUNCTIONALITY

The system is designed with several core components to address the limitations of traditional processes:

CENTRALIZED ISSUE REPOSITORY AND WORKFLOW MANAGEMENT

- Provides a single, unified platform for logging, viewing, and managing all data issues.
- Enables assigning issues to specific individuals or teams, tracking status changes (e.g., Open, In Progress, Resolved, Closed), and maintaining a detailed, immutable audit trail for each issue.
- Facilitates structured communication and handoffs between different roles (programmers, data managers, medical monitors, sponsors).

AI-DRIVEN DUPLICATE ISSUE DETECTION

- Upon entry of a new issue description, the system utilizes an AI model to compare it against the repository of existing open and recently closed issues.
- Identifies potential duplicates based on semantic meaning rather than just keyword matching, marking the issue as duplicate, and prompting the user to review similar existing issues. This directly addresses redundancy.

NATURAL LANGUAGE QUERYING INTERFACE

- Allows users, including those without SQL expertise (e.g., data managers, clinicians), to request data checks or explore data patterns using plain English queries.
- The system interprets these natural language requests and translates them into appropriate database queries (in SQL).

AUTOMATED REPORT GENERATION AND PATTERN IDENTIFICATION

- Leverages the natural language interface and direct integration with study databases to generate reports summarizing issues or exploring data related to an issue (e.g., listing all subjects with a specific type of data discrepancy).
- AI components can assist in identifying potential systemic patterns or trends across subjects or visits that might be missed in manual reviews (e.g., consistent data anomalies under specific conditions).
- Assist in exploring potential correlations, such as identifying concomitant medications (CMs) frequently associated with specific adverse events (AEs) or finding related data points (e.g., lab values like HbA1c) when investigating an issue related to a specific condition or treatment.

CONFIGURABLE DATA MONITORING AND ALERTING

- Allows authorized users to configure automated data checks to run at specified intervals (e.g., daily, weekly).
- Upon detection of anomalies or data patterns meeting pre-defined criteria, the system can automatically generate issues or trigger notifications to designated personnel, enabling proactive issue identification.

CONTROLLED ACCESS AND DATA SECURITY

Implements role-based access control (RBAC) to ensure users can only access data and functionalities relevant to their assigned studies and roles, maintaining confidentiality and data security in line with regulatory standards.

AUTOMATED ARCHIVAL

Provides functionality for systematic archival of resolved issues and associated reports, ensuring long-term data retention and retrievability for compliance and historical analysis.

TECHNICAL IMPLEMENTATION WITH AI

The system leverages several AI techniques to deliver its features. The implementation focuses on integrating these components into a cohesive workflow:

DUPLICATE ISSUE DETECTION

- **Embeddings and Semantic Similarity**: Issue descriptions (both new and existing) are converted into high-dimensional numerical vectors, embeddings, using pre-trained language models ModernBERT. These embeddings are stored in a vector database. The cosine similarity or other distance metrics between the vector of the new issue and vectors of existing issues is calculated. Issues with similarity scores above a configurable threshold are flagged as potential duplicates.
- **Retrieval-Augmented Generation (RAG)**: For enhanced analysis, a RAG approach is employed. When a user enters a new issue, its embedding is used to retrieve the most semantically similar existing issues from the vector database. This retrieved context (details of similar past issues and their resolution) is then provided, along with the new issue description, to a Large Language Model (LLM). The LLM synthesizes this information to provide a concise summary to the user, explaining why the retrieved issues might be duplicates and highlighting key similarities or differences, aiding the user's decision on whether to accept that the issue raised is in fact a duplicate, or continue with the new issue.

NATURAL LANGUAGE QUERYING (NLQ)

- **LLM for Text-to-SQL:** An LLM, specifically fine-tuned for Text-to-SQL generation, is used. The model is provided with the user's natural language query and, crucially, fine-tuned with the relevant database schema information (table names, column names, data types, and potentially relationships/foreign keys) for the specific study database.
- **Prompt Engineering:** Carefully crafted prompts guide the LLM to generate syntactically correct and semantically appropriate SQL queries that reflect the user's intent and adhere to the database structure. Prompts include examples (few-shot learning) and constraints.
- **Validation:** Generated SQL queries undergo automated checks using a sandboxed SQL execution environment:
- **Syntax Validation:** Parsing the SQL to ensure it is valid.
- **Schema Adherence:** Verifying that tables and columns referenced exist in the provided schema.

REPORT GENERATION

- **Natural Language Driven Reporting:** Leveraging the NLQ interface described in Section 3.2, users can request specific reports using plain English. For instance, a user might ask for "a list of all subjects with adverse events related to drug X" or "a summary of unresolved data discrepancies assigned to the data management team." The system interprets these requests and utilizes the Text-to-SQL generation capability to formulate the necessary database queries to retrieve the required data.
- **Standardized and Scheduled Reports:** The system supports the creation of standardized report templates for common oversight tasks (e.g., summary of open issues, issue resolution timelines, data validation check outputs). Furthermore, based on user configuration (as mentioned in Section 2.5), these reports or specific data checks can be scheduled to run automatically at defined intervals (e.g., daily, weekly). The output reports are then generated and made available or distributed to designated personnel, ensuring timely data monitoring.
- **Consistent Formatting:** The system aims to deliver reports in a consistent, predefined format. While it can adapt to specific ad-hoc formatting requests, it defaults to established layouts for subsequent

standard reports, ensuring predictability and ease of use for reviewers.

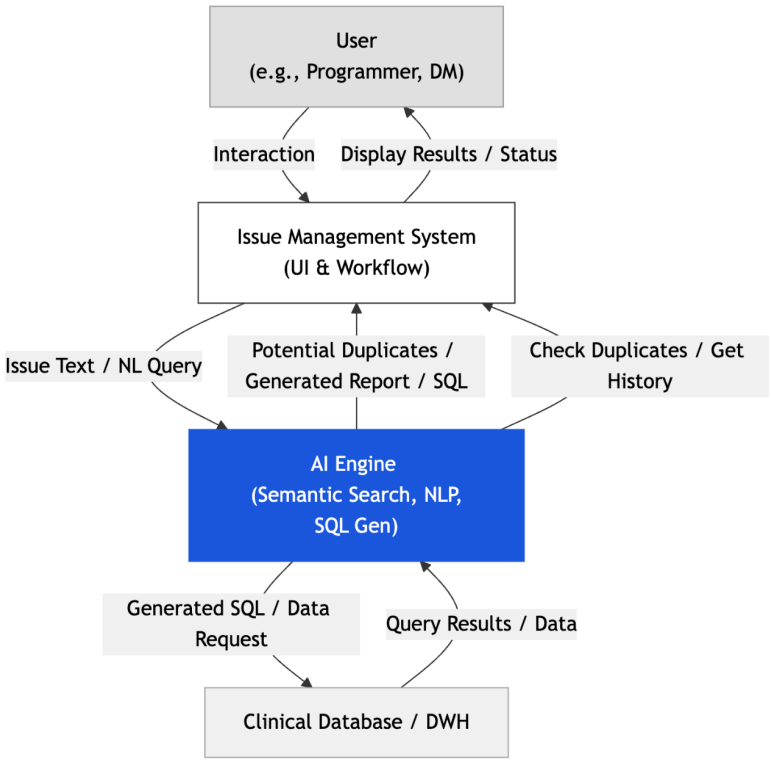

Figure 1. System Workflow

MODEL TRAINING, REFINEMENT, AND EVALUATION:

- **Baseline Establishment:** Performance of the AI components is measured against baselines where a base LLM is used without any sort of fine tuning. The inputs and outputs are reviewed and scored by domain experts.
- **Prompt Engineering:** Iterative refinement of prompts used to interact with LLMs is crucial for controlling output format, tone, and accuracy for tasks like summarization, SQL generation, and duplicate explanation.
- **Data for Fine-tuning:** The embedding model and the base LLM model are fine-tuned using domain-specific data. This includes anonymized historical issue descriptions, query logs, clinical trial protocols, and relevant biomedical ontologies/terminologies to improve understanding of clinical context.
- **RAG Optimization:** The retrieval mechanism within RAG is optimized by tuning the number of documents retrieved and potentially re-ranking retrieved issues based on relevance feedback.
- **Evaluation Metrics:**
  - **Duplicate Detection:** Precision, Recall, F1-score.
  - **Text-to-SQL:** Query execution success rate, result accuracy (requires comparison against gold-standard queries or results).
  - **User Feedback:** Incorporating feedback mechanisms (e.g., users confirming/rejecting duplicate suggestions, rating the usefulness of generated reports/SQL) provides valuable data for iterative improvement.
  - **Tracing and Logging:** All AI model inputs, outputs, and intermediate steps (like retrieved

documents in RAG) are logged. This tracing is essential for debugging, understanding model behavior, and identifying areas for improvement.

PRIVACY, SECURITY, COMPLIANCE, AND DETERMINISM

Integrating AI technologies within the regulated clinical research environment necessitates careful consideration of privacy, security, compliance, and the reliability of system outputs. The system design incorporates measures to address these critical aspects.

- **Privacy and Security:** Base AI models selected for use are vetted, often originating from open-source communities with transparent development practices. Deployment occurs within controlled environments, specifically virtual private networks (VPNs), isolating the system from external threats. For any model fine-tuning activities, only appropriately anonymized data is utilized, ensuring patient confidentiality is maintained. Access to the system, its underlying models, and the data it processes is strictly controlled through role-based mechanisms and is subject to continuous monitoring and auditing to prevent unauthorized access and ensure accountability.
- **Compliance:** The system's architecture, including secure infrastructure, controlled access protocols, and comprehensive audit trails for issue tracking and data queries, is designed to support adherence to relevant regulatory requirements (e.g., GCP, 21 CFR Part 11). The immutable audit logs provide traceability for actions performed within the system, which is essential for inspection readiness.
- **Determinism and Traceability:** While some AI components, particularly LLMs used for interpretation or generation, can exhibit variability, the system employs strategies to ensure reliable and traceable outputs for critical functions.
- **Database Interaction:** The definitive results presented to the user (e.g., lists of subjects, data summaries) are generated by executing SQL queries against the validated study database or data warehouse. The AI assists in generating the query, but the database execution itself is a deterministic process based on the data state at the time of execution.
- **Traceability:** Every AI-assisted action, such as query generation or report creation, is logged. Outputs are tagged with metadata including timestamps and unique tracing IDs. This allows auditors or users to trace an output back to the specific input prompt, the data context provided, and the version of the AI model used, facilitating reproducibility analysis and debugging.
- **Validated Queries:** For frequently generated standard reports, SQL queries that have been reviewed, validated, and approved by relevant personnel can be saved within the system. Subsequent requests for these specific reports utilize the saved, validated query, bypassing the AI generation step entirely. This ensures absolute consistency and deterministic output for routine reporting tasks.

PRELIMINARY PERFORMANCE OBSERVATIONS

Initial evaluations of the system have yielded preliminary data suggesting potential improvements in efficiency and data quality oversight compared to traditional manual processes. While extensive validation is ongoing, early observations include:

Duplicate Issue Reduction: In pilot user groups comparing the AI-assisted workflow against manual tracking methods for the same datasets, the system's duplicate detection feature flagged potential redundancies prior to issue creation. User review confirmed these suggestions as valid duplicates in approximately **35%** of cases, leading to an estimated **20-25% reduction** in the logging of truly redundant issues compared to baseline manual methods where duplicates were often missed or identified later.

Report Generation Efficiency: For common data review tasks typically requiring programmer intervention (e.g., generating listings of subjects meeting specific discrepancy criteria), users utilizing the Natural Language Querying (NLQ) interface reported significant time savings. Compared to the typical turnaround

time involving request submission, queuing, programmer coding, and validation, generating similar outputs via the NLQ interface showed an estimated average **reduction of 40-50%** in end-to-end time for generating these specific report types.

NLQ-to-SQL Accuracy: The accuracy of the Text-to-SQL translation component was evaluated for a predefined set of common data cleaning and review queries (e.g., identifying missing dates, out-of-range values, inconsistent data across forms). For these well-defined query types, the system generated syntactically correct and semantically appropriate SQL queries requiring no or minimal user modification in approximately **60%** of test cases. Accuracy for more complex or ambiguously phrased queries was lower, highlighting the importance of clear user input and ongoing model refinement.

These preliminary results are encouraging, indicating the system's potential to streamline workflows and enhance data oversight. However, these figures represent early findings and require confirmation through broader deployment and more rigorous, long-term comparative studies to fully quantify the impact on clinical data management processes.

FUTURE DEVELOPMENTS

In future iterations, the system will integrate API functionality to automate query management. When a data manager identifies a need for a query, the system will automatically generate and submit it via API integration. Upon receiving the site's response, the system will review the resolution and either close the query or initiate additional steps, such as documenting protocol deviations, assigning training, or conducting other necessary follow-ups. This advancement aims to streamline the query lifecycle, improve efficiency, and ensure comprehensive issue resolution.

CONCLUSION

The proposed AI-powered data issue tracking system offers a structured approach to address common inefficiencies in clinical data management. By employing techniques like semantic similarity for duplicate detection, natural language processing for query generation, and background agents for continuous monitoring, the system aims to reduce manual effort, enhance collaboration, and improve the timeliness of issue resolution. While not a replacement for human oversight, this system has the potential to significantly augment the capabilities of clinical trial teams, contributing to higher data quality and more efficient research processes.

ACKNOWLEDGMENTS

The author greatly acknowledges Uday Chandra from singular for his technical guidance and editing on the AI aspects of this paper and Bigyani Samal from Statistics & Data Corporation for her review and editing.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Bharath Donthi
Enterprise: Statistics & Data Corporation
Address: 63 South Rockford Drive, Suite 240, Tempe, AZ 85281
bdonthi@sdcclinical.com
https://www.sdcclinical.com