FHIRViz: Multi-Agent Platform for FHIR Visualization to Advance Healthcare Analytics

Mariam ALMutairi
Virginia Tech University
Falls Church, VA, USA
malmutairi@vt.edu

Lulwah AlKulaib
Kuwait University
Kuwait City, Kuwait
lalkulaib@cs.ku.edu.kw

Shengkun Wang
Virginia Tech University
Falls Church, VA, USA
shengkun@vt.edu

Zhiqian Chen
Mississippi State University
Mississippi State, MS, USA
zchen@cse.msstate.edu

Youssif ALMutairi
Oakton High School
Oakton, VA, USA
Youssif.algrabah@gmail.com

Thamer M. Alenazi
King Faisal Specialist Hospital &
Research Centre
Riyadh, Saudi Arabia
talenazi@kfshrc.edu.sam

Kurt Luther
Virginia Tech University
Falls Church, VA, USA
kluther@vt.edu

Chang-Tien Lu
Virginia Tech University
Falls Church, VA, USA
clu@vt.edu

Abstract

The shift to electronic health records (EHRs) has enhanced patient care and research, but data sharing and complex clinical terminology remain challenges. The Fast Healthcare Interoperability Resource (FHIR) addresses interoperability issues, though extracting insights from FHIR data is still difficult. Traditional analytics often miss critical clinical context, and managing FHIR data requires advanced skills that are in short supply. This study presents FHIRViz, a novel analytics tool that integrates FHIR data with a semantic layer via a knowledge graph. It employs a large language model (LLM) system to extract insights and visualize them effectively. A retrieval vector store improves performance by saving successful generations for fine-tuning. FHIRViz translates clinical queries into actionable insights with high accuracy. Results show FHIRViz with GPT-4 achieving 92.62% accuracy, while Gemini 1.5 Pro reaches 89.34%, demonstrating the tool's potential in overcoming healthcare data analytics challenges.

CCS Concepts

• **Applied computing → Health informatics**; • **Computing methodologies → Information extraction**.

Keywords

LLMs, FHIR, Clinical Analytics, Multi-Agent, Knowledge Graph, visualization, Health Informatics

ACM Reference Format:
Mariam ALMutairi, Lulwah AlKulaib, Shengkun Wang, Zhiqian Chen, Youssif ALMutairi, Thamer M. Alenazi, Kurt Luther, and Chang-Tien Lu. 2024. FHIRViz: Multi-Agent Platform for FHIR Visualization to Advance Healthcare Analytics. In *15th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '24), November 22–25, 2024, Shenzhen, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3698587.3701392

BCB '24, November 22–25, 2024, Shenzhen, China

ACM ISBN 979-8-4007-1302-6/24/11
https://doi.org/10.1145/3698587.3701392

1 Introduction

The digital transformation of healthcare through Electronic Health Records (EHRs) has improved patient care and enabled large-scale medical research [10]. However, fully leveraging EHR data is hindered by challenges like data sharing issues, variations in standards, and regulatory constraints [4]. Inconsistent data formats and terminologies across systems exacerbate this, complicating the integration of patient records [25].

The Fast Healthcare Interoperability Resources (FHIR) standard, developed by Health Level 7 (HL7), provides structured formats for common healthcare concepts like patients and observations [15]. FHIR's modular structure, along with its use of modern web technologies, facilitates data sharing between different systems [20]. Despite this, the complex and heterogeneous nature of clinical data across systems remains a major challenge [5]. FHIR data is deeply nested, making it difficult to query within traditional databases. Terminologies like SNOMED CT further complicate queries, and the shortage of professionals skilled in FHIR data analytics worsens this issue [3, 13]. These factors lead to inefficiencies in analyzing clinical data and making informed decisions.

To address these challenges, we introduce FHIRViz, a healthcare analytics platform that integrates a multi-agent system with large language models (LLMs) to interpret complex EHR data. At the core of FHIRViz is a knowledge graph that enhances the LLM's ability to understand medical concepts within FHIR data, enabling accurate and relevant analysis. By providing advanced visualization tools, FHIRViz offers a solution to FHIR's data analytics complexity challenges.

The key contributions of this work are:

- **Developed a Semantic Layer using a Knowledge Graph:** That bridges LLM Multi-Agents and FHIR data. This graph encodes medical concepts and relationships, helping LLMs understand query context and retrieve relevant data more accurately.
- **Proposed a novel FHIR-Analytics using Multi-Agents Graph LLM:** These agents collaboratively interpret clinician queries and extract insights from varied data types, providing comprehensive and precise analysis.
- **Implemented System Improvement through Human-in-the-loop:** Using clinician feedback and storing query resolutions. This allows the system to learn over time, improving its query-handling capabilities and enhancing future performance by fine-tuning LLM models.

2 Related Work

Several efforts have focused on **mapping FHIR into relational databases** to enhance data storage and query handling. Gruendner et al. [14] introduced KETOS, a Jupyter Notebook platform that integrates FHIR with PostgreSQL for efficient storage and preprocessing. Cerner's SQL on FHIR [11] aligns the FHIR model with relational queries, improving handling capabilities. Additionally, Ayaz et al. [3] proposed a comprehensive FHIR Data Analytics Framework, which includes data storage, retrieval, and visualization, organized into six layers to optimize healthcare data analysis. In a related effort, Grimes et al. [13] developed Pathling to **extend FHIR APIs**, addressing limitations in data aggregation and search functionalities in FHIR REST API. Pathling provides additional operations for exploratory data analysis (EDA), patient cohort selection, and data preparation, easing analytics tasks for developers, though requiring some technical expertise. Moreover, **Big Data Tools** used to manage large FHIR datasets. for example, Liu et al. [18] explored Apache Parquet and Apache Spark for efficient storage and query handling at scale. Furthermore, Cerner's Bunsen library and Google's FhirProto also provide specialized tools for encoding and processing FHIR data, facilitating type safety, validation, and cross-language compatibility [1, 6]. These tools streamline the integration of FHIR data with cloud services and machine learning workflows.

Recent advancements in Large Language Models (LLMs) offer potential solutions to this gap. Yao et al. [27] highlighted LLMs' ability to interpret user intentions and perform tasks like generating visualizations from natural language prompts [16, 19]. TaskWeaver introduces autonomous LLM agents that translate requests into executable code [24], while an agent-based graph method enhances collaborative data analysis with minimal human input [9]. Gao's framework [12] categorizes LLM interactions into four modes: Standard Prompting, User Interface, Context-based, and Agent Facilitator. Leveraging LLMs and this framework, our study presents FHIRViz (figure 1), integrating a multi-agent system and knowledge graph to overcome FHIR data challenges, enabling stakeholders to perform diverse analytical tasks with patient data.

3 Methods

The development of FHIRViz utilized the Synthea dataset, a synthetic but realistic collection of FHIR-standard electronic health records. GPT-4 was integrated for its advanced language comprehension, enabling insights from complex healthcare data. A Neo4j knowledge graph served as a semantic layer, capturing relationships and temporal data, while a Qdrant vector store archived successful query resolutions for efficient retrieval and system improvement. The multi-agent system, managed via a finite state machine, ensured smooth operations across retrieval, coding, execution, and debugging. Evaluation through accuracy assessments and user feedback on visualizations guided further improvements.

3.1 Model and Dataset

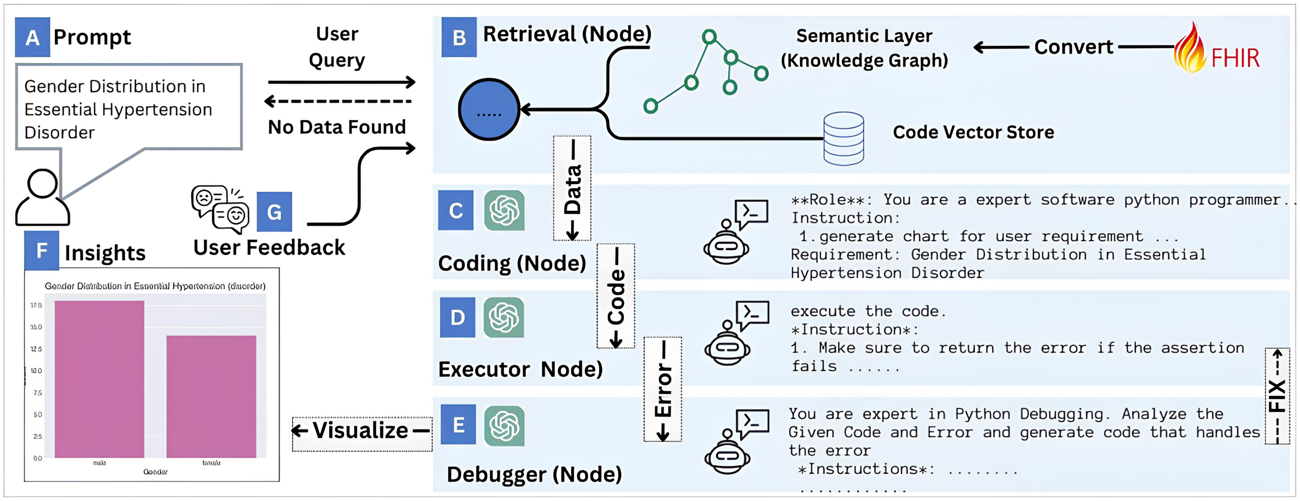
We employed GPT-4 for this scenario because of its superior capability in comprehending intricate queries and nuanced language. Its advanced understanding, facilitated by its deep learning architecture and extensive training on diverse datasets, enables it to produce contextually accurate and relevant responses, crucial for interpreting healthcare data and queries effectively. Additionally, GPT-4's capacity to generate code further enhances its suitability for this task. We leveraged GPT-4's strengths in conjunction with Synthea's synthetic health record data to thoroughly test and demonstrate its effectiveness. Synthea is an open-source, completely synthetic collection of electronic health record data developed by Walonoski et al. [26], which is a synthetic compilation of patient records crafted to mimic real health data while upholding privacy measures. Synthea consists of 150 patient records that include 4,195 conditions, 7,074 encounters, 1,968 medications, 6,535 medication requests, 85,434 observations, and 15,328 procedures. These records adhere to the FHIR standard, facilitating seamless interoperability within healthcare applications. While the dataset encompasses many FHIR resources, We deliberately excludes some of the FHIR resources for proof of concept purposes, focusing on the most commonly utilized ones as shown in Table 1.

Table 1: FHIR Resources Used in the Synthea Dataset

Resource	Description	Count
Patient	Represents an individual receiving care	115
Encounter	Represents a healthcare interaction	7074
Condition	Represents a clinical condition	4194
Observation	Represents clinical information	85434
Procedure	Represents an action performed on a patient	15328
Medication	Represents a substance used for medical treatment	1968
MedicationRequest	Represents a request for medication	6535
Practitioner	Represents an individual involved in healthcare	271
Organization	Represents an entity responsible for providing healthcare services	271

3.2 Semantic Layer Generation

To establish the semantic layer, we transformed the FHIR resources into a knowledge graph for several critical reasons. Firstly, this conversion ensures that complex data relationships and temporal aspects are accurately captured and maintained [8]. Secondly, it enables more efficient and advanced querying capabilities, allowing for sophisticated semantic searches and analytics that are challenging to achieve with traditional relational databases. Thirdly, it improves data visualization and understanding of the interconnected nature of healthcare information, aiding clinicians and researchers in uncovering insights. Finally, it enhances data quality and consistency by enforcing standardized relationships and attributes,



```
**Role**: You are a expert software python programmer..
Instruction:
 1.generate chart for user requirement ...
Requirement: Gender Distribution in Essential
Hypertension Disorder
```

```
execute the code.
*Instruction*:
1. Make sure to return the error if the assertion
fails ......
```

```
You are expert in Python Debugging. Analyze the
Given Code and Error and generate code that handles
the error
 *Instructions*: ........
 .............
```

Figure 1: Multi-Agents Graph LLM for FHIR Analytics

reducing redundancy, and preserving the temporal aspects of the data.

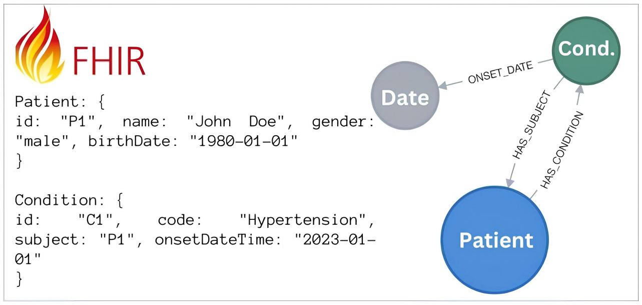
We utilized Neo4j for creating the semantic layer [21]. Neo4j was chosen for its robust handling of highly interconnected data, which is typical in healthcare contexts. Each FHIR resource $R \in \{\text{Patient, Encounter, Condition}, \ldots\}$ was converted into a node N_R within the knowledge graph. The attributes of these resources, such as a Patient's name, gender, and birth date, were mapped as properties $P(N_R)$ of the corresponding nodes. References within FHIR resources were transformed into relationships E between nodes; for instance, a reference from a Condition resource $N_{\text{Condition}}$ to a Patient resource N_{Patient} was represented as a relationship between both $E(N_{\text{Condition}}, N_{\text{Patient}})$. Temporal aspects were maintained by representing dates and times as nodes N_T, allowing for efficient querying of temporal relationships, such as retrieving all conditions diagnosed within a specific time frame. This transformation ensures that the complex and nested structure of FHIR data is flattened and organized in a way that facilitates semantic queries and analytics.

$$\begin{aligned}
&\text{Nodes:}\\
&\forall R \in \{\text{Patient, Encounter, Condition}, \ldots\}, \exists N_R\\
&P(N_R) = \{\text{attributes of } R\}\\
&\text{Edges:}\\
&E(N_{\text{Condition}}, N_{\text{Patient}}) = \{\text{references from Condition to Patient}\}\\
&\text{Temporal Nodes:}\\
&\exists N_T = \{\text{dates and times associated with events}\}
\end{aligned} \tag{1}$$

example is in figure 2. This transformation ensures that the complex and nested structure of FHIR data is flattened and organized in a way that facilitates semantic queries and analytics.

3.3 Vector Store

The Vector Store serves two primary purposes: storing successful generations with the associated user queries and providing an initial check for similar past queries when a new user query is submitted. This dual functionality ensures efficient query handling by referencing a historical record of interactions. We have implemented this using the Qdrant dataset [23], a high-performance, open-source vector database optimized for real-time, scalable operations. To complement this setup, we utilized the text-embedding-ada-002 model [22] for embedding user queries. The model provides high-quality, versatile embeddings for various NLP tasks, enhancing efficiency and understanding of text. It supports multilingual applications, scales well with text length, and integrates easily with OpenAI's ecosystem.



```
Patient: {
id:  "P1",  name:  "John  Doe",  gender:
"male", birthDate: "1980-01-01"
}

Condition: {
id:    "C1",    code:    "Hypertension",
subject: "P1", onsetDateTime: "2023-01-
01"
}
```

Figure 2: Example of Converting FHIR Resource to Knowledge Graph

Data storage in the Vector Store involves embedding the user queries while storing the successful generations without embedding them. We employed cosine similarity with a score threshold of 0.7 for ranking. The cosine similarity between two vectors A and B is given by: $\text{cosine\_similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$. Where $A \cdot B$ represents the dot product of vectors A and B, and $\|A\|$ and $\|B\|$ are the magnitudes of vectors A and B respectively. By setting a score threshold of 0.7, we ensure that only the most relevant and contextually similar results are prioritized: $\text{cosine\_similarity}(A, B) \geq 0.7$. This improves the overall efficiency of query resolution by filtering out less pertinent matches. Moreover, the accumulated data in the Vector Store presents an opportunity for fine-tuning LLMs in the future, continually improving their performance and adaptability to user needs.

3.4 Multi-Agents Graph LLM

We have integrated the Multi-Agent Graph Architecture into our methodology, centered on an agent-based graph where multiple agents collaborate to address data analytics challenges. The agent workflow utilizes a Finite State Machine (FSM) [2] developed using LangGraph [17], a tool for building stateful, multi-actor LLM applications (Figure 1).

The **Retrieval Node** consists of a Knowledge Graph and a Vector Store. A semantic search is performed on user queries by checking the Vector Store for similar past queries. If a match is found, the corresponding stored response is used, enhancing efficiency. If no match exists, the Knowledge Graph is queried for relevant data or returns a "no data found" message to mitigate hallucinations. The retrieved data is passed to the **Coding Node**, which transforms the data into visual formats using Chain-of-Thought (CoT) prompting to generate accurate, well-commented Python code. The system ensures data availability and manages errors by returning specific messages when issues arise. Visualizations are generated following a Matplotlib stylesheet and saved as PNG files to ensure consistency and reliability. The generated code is transferred to the **Executor Node**, which handles code execution and creates visual outputs. This node verifies the code's integrity to ensure reliable results and prevent runtime errors before returning the outputs to the user. If the Executor Node encounters errors, the **Debug Node** resolves them by leveraging LLM-generated insights to produce revised code. Once the errors are corrected, the process returns to the Executor Node for re-execution, ensuring robust error handling throughout the process.

The multi-agent graph methodology involves four key phases: retrieval, coding, execution, and debugging. These phases are modeled using a Finite State Machine (FSM) with states $S = \{s_1, s_2, s_3, s_4, s_5\}$, representing retrieval, coding, execution, debugging, and the end of the process, respectively. The transition function δ dictates the movement between states based on specific conditions. The process starts in the retrieval state s_1, moves to coding (s_2, and then to execution s_3. If execution is error-free, it transitions to the end state s_5. If errors are detected, the process moves to debugging s_4, and loops back to coding until successful execution is achieved.

The FSM is represented with a transition matrix P, where each element P_{ij} indicates the probability of transitioning from state i to state j. These transitions are deterministic, with probabilities either 0 or 1, except during execution, where $P_{34} = p_1$ (probability of error) and $P_{35} = p_2$ (probability of success), ensuring $p_1 + p_2 = 1$. The transitions can be formally described as follows:

$$\begin{aligned}\delta(s_1, \text{complete}) &= s_2,\\ \delta(s_2, \text{complete}) &= s_3,\\ \delta(s_3, \text{no error}) &= s_5,\\ \delta(s_3, \text{error}) &= s_4,\\ \delta(s_4, \text{complete}) &= s_2.\end{aligned}$$

The transition matrix P can be defined as:

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & p_1 & p_2 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

This FSM model ensures efficient workflow management and error handling

3.5 Human in the Loop

The methodology integrates a robust feedback mechanism involving healthcare professionals, who contribute insights regarding the efficacy of the generated analytics. This iterative human feedback loop is instrumental in continuously refining the model, thereby enhancing its performance incrementally. Specifically, after the successful generation of a requested chart, clinicians are prompted to assess and provide feedback on the generation process's effectiveness. Positive feedback is systematically archived in a vector store, serving as a valuable reference for future iterations. Moreover, these successful generations can be leveraged to fine-tune an LLM, thereby improving its capabilities in health analytics. This dynamic interplay between human expertise and machine learning models ensures the ongoing optimization of analytical outputs, aligning them more closely with clinical needs and improving their utility in real-world healthcare settings.

4 Evaluation

For the FHIRViz to achieve success and effectively assist clinicians, it must 1. enhance data analysis by accurately interpreting EHR data, 2. simplify FHIR data queries for easy visualization without technical skills, and 3. continuously improve through feedback to better understand user needs.

4.1 Evaluation Dataset and Baseline

In the evaluation phase, we curated an evaluation dataset from the Synthea FHIR dataset, extracting 122 analytical charts to represent various medical administration scenarios. These charts were categorized into three complexity levels based on the number of queried FHIR resources. This categorization structured the assessment of data processing tasks, with complexity increasing as more resources were queried. The complexity levels and their details are summarized in Table 2. The charts include histograms, bar charts, pie charts, network graphs, line charts, heatmaps, stacked bar charts, tree maps, scatter plots, and box plots. Manual comparison with FHIR Synthea data provided a baseline for benchmarking system performance and identifying areas for improvement.

4.2 Evaluation Metrics

To assess the accuracy of generated charts using this formula: Accuracy of Generated Charts = $\frac{\text{Number of Correct Charts}}{\text{Total Charts Generated}} \times 100$. This formula quantifies FHIRViz's accuracy by comparing the number of correct charts to the total charts produced, thereby evaluating its performance in accurately representing medical data. This metric is

Table 2: Evaluation Dataset by Complexity

Complexity	Description	# Reports
1-Resource	Only one FHIR resource is queried	49
2-Resources	Aggregated two FHIR resources	57
3-Resources	Aggregated three FHIR resources	11
4-Resources	Aggregated four FHIR resources	5

crucial for facilitating optimization and refinement decisions. The generated output is compared with manually curated charts in the evaluation dataset described in Section 4.1, ensuring a thorough assessment of FHIRViz's capability to produce reliable and accurate visualizations.

4.3 Comparative Analysis

After evaluating the accuracy of the FHIRViz with GPT-4, we conducted a thorough comparison to assess the performance of three distinct LLMs: GPT-4, ChatGPT 3.5, and Gemini Pro 1.5. Utilizing the dataset described in Table 2 , we prompted each LLM using various prompting styles, including Zero shot, Chain of Thought (CoT), and ReAct, to generate Python code to create the chart. We then executed the generated code and examined the outputs to assess chart accuracy and alignment with the knowledge graph data. This comparative analysis provided valuable insights into the nuances of each LLM's performance with different prompting styles, allowing us to identify potential strengths and weaknesses and inform future optimization strategies and decision-making processes.

4.4 Human Evaluation

The quality of chart presentations was evaluated based on readability, visual appeal, and accuracy, focusing on chart components like axis labels and titles. Six participants from medical and health information fields assessed 10 use cases, comparing system-generated charts with manually created ones. Chart ratings ranged from "Excellent" (high clarity, accuracy, and usability) to "Very Poor" (unclear and unusable). Overall satisfaction was categorized from "Very Satisfied" to "Not Satisfied at All," providing insights into user contentment with chart quality.

5 Results

5.1 Performance Results

The performance of FHIRViz, as compared to other prompting styles across various models, reveals significant differences in effectiveness. According to (Table 3), GP4-FHIRViz leads with 113 successful generations out of 122, achieving an accuracy rate of 92.62%. Gemini 1.5 Pro - FHIRViz follows with 109 successful responses (89.34%). In contrast, GPT 3.5 - Zero Shot achieved only 23 successful generations with an accuracy rate of 18.85%. The Chain of Thought (CoT) and ReAct prompting styles also demonstrated notable improvements, with GPT-4 - CoT achieving 84 successful responses (68.65%) and Gemini 1.5 Pro - CoT generating 82 (67.21%). The ReAct style performed even better, with GPT-4 - ReAct generating 94 successful responses (77.04%) and Gemini 1.5 Pro - ReAct achieving 91 (74.59%). These results show that advanced prompting methods like CoT and ReAct significantly outperform simpler zero-shot methods.

Our experiments suggest that response quality improves with more detailed and precise queries, leading us to enhance FHIRViz to request additional information when queries are ambiguous. The system's flexibility in generating various chart types, such as bar and pie charts, was also evident. However, the evaluation results indicated that complex queries involving two to four FHIR resources negatively impacted accuracy, highlighting a need for improvements in managing complex data requests. These findings emphasize the importance of continuous refinement, particularly in data visualization and user satisfaction. While FHIRViz demonstrates high accuracy and effectiveness, ongoing enhancements are essential to maintain and improve its performance, especially as the complexity of queries grows. Sample Generations are in Figure4.

Table 3: Performance of Different Models and Prompting Styles and FHIRViz

Model & Prompt Style	Successful Gen.	Accuracy (%)
GPT 3.5 - Zero shot	23	18.85
Gemini 1.5 Pro - Zero Shot	72	59.01
GPT 4 - Zero shot	72	59.01
GPT 3.5 - COT	42	34.42
Gemini 1.5 Pro - COT	82	67.21
GPT-4 - COT	84	68.65
GPT 3.5 - ReAct	53	43.44
Gemini 1.5 Pro - ReAct	91	74.59
GPT-4 - ReAct	94	77.04
GPT 3.5 - FHIRViz	75	61.47
Gemini 1.5 Pro - FHIRViz	109	89.34
GP4-FHIRViz	113	92.62

5.2 Human Evaluation Results

The human evaluation metrics for FHIRViz's chart generation revealed notable findings in chart quality and user satisfaction (Figure 3). The chart presentation was evaluated on a rating scale. Results showed 60% of charts rated as Excellent and 25% as Good, with 85% rated as either Excellent or Good, indicating a high overall quality. Similarly, user satisfaction was rated on five levels. Findings showed 62.5% of users were Very Satisfied, and 22.5% were Satisfied, with 85% overall satisfaction. Despite the positive feedback, users highlighted issues with charts containing too much information, making them difficult to interpret. Limiting data points in a single chart is recommended to improve clarity. While FHIRViz provides high-quality visualizations, attention to information density is essential for maintaining clarity and usability.

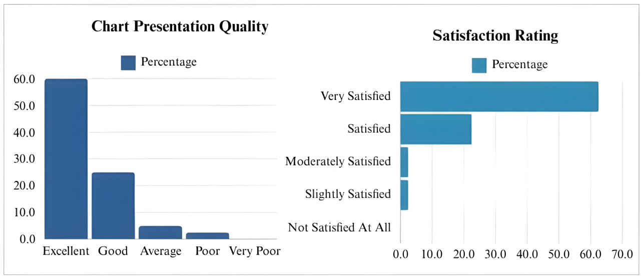


Figure 3: Summary of Human Evaluation Metrics. The left chart shows the quality of chart presentation, The right chart presents the satisfaction rating

6 Discussion

FHIRViz effectively supports FHIR analytics. Our evaluation of the multi-agent LLM for FHIR analytics demonstrates high accuracy and user satisfaction. The system successfully interprets user queries and generates visually appealing charts, addressing the challenges associated with FHIR data standard analytics. However, this study focuses on FHIR version R4, indicating a need for further validation across other versions. User feedback has been overwhelmingly positive, with users appreciating the system's ability to understand queries on the first attempt. High satisfaction and ease of use suggest that our multi-agent LLM solution effectively fills gaps in existing user-friendly analytics frameworks. Yet, there are calls for more dynamic chart customization and faster generation times, with limitations like resource constraints and dataset size highlighting the need for larger datasets and improved visuals in future studies.

Our findings align with previous studies. Our results align with Ayaz et al. [3], who pointed out the lack of comprehensive analytics tools. We build on this by enhancing user interaction with advanced natural language processing, aligning with the LLM advancements noted by Yao et al. [27], Maddigan and Susnjak [19], and John et al. [16]. Although the Synthea dataset was useful, it lacks real-world care variations, limiting chart evaluation [7]. The innovative FHIRViz approach incorporates a multi-agent system with a knowledge graph, improving the efficiency of handling complex queries while reducing human intervention, resonating with the findings of Chugh et al. [9]. Ethical considerations, such as securing patient data and ensuring transparency, are critical to the system's success, though further validation with real-world data and larger datasets is essential for confirming our findings.

7 Conclusion

The FHIRViz platform showcase the innovation integration of FHIR data with LLM Multi-Agents and Knowledge graph Semantic Layer, marking a significant advance in healthcare analytics by enhancing the accuracy and efficiency of data analysis and visualization. This approach opens new avenues for improving healthcare decision-making and patient care through sophisticated, data-driven insights. By expanding the system to include a broader range of FHIR resources and integrating real-world data, future developments promise to further leverage LLM Multi-Agents'potential in healthcare, aiming to revolutionize how healthcare professionals access and utilize data for better clinical outcomes.

References

[1] [n. d.]. Google FHIR Protocol Buffers and Utilities - GitHub. https://github.com/google/fhir. Accessed: 2024-03-08.

[2] 2012. *Introduction to the Theory of Computation, 3d Edition.* Cengage Learning, Berlin.

[3] Muhammad Ayaz, Muhammad Fermi Pasha, Tahani Jaser Alahmadi, Nik Nailah Binti Abdullah, and Hend Khalid Alkahtani. 2023. Transforming Healthcare Analytics with FHIR: A Framework for Standardizing and Analyzing Clinical Data. In *Healthcare*, Vol. 11. MDPI, 1729.

[4] IMP Barbalho, F Fernandes, DMS Barros, JC Paiva, J Henriques, AHF Morais, KD Coutinho, GC Coelho Neto, A Chioro, and RAM Valentim. 2022. Electronic health records in Brazil: Prospects and technological challenges. *Frontiers in Public Health* 10 (2022), 963841. https://doi.org/10.3389/fpubh.2022.963841

[5] Duane Bender and Kamran Sartipi. 2013. HL7 FHIR: An Agile and RESTful approach to healthcare information exchange. *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems* (2013), 326–331. https://api.semanticscholar.org/CorpusID:11069145

[6] Cerner Engineering Blog. 2017. Announcing Bunsen: FHIR Data with Apache Spark. https://engineering.cerner.com/blog/announcing-bunsen-fhir-data-with-apache-spark/. Accessed: 2024-03-09.

[7] Junqiao Chen, David Chun, Milesh Patel, Epson Chiang, and Jesse James. 2019. The validity of synthetic clinical data: a validation study of a leading synthetic data generator (Synthea) using clinical quality measures. *BMC medical informatics and decision making* 19 (2019), 1–9.

[8] Weisi Chen, Zoran Milosevic, Fethi A. Rabhi, and Andrew Berry. 2023. Real-Time Analytics: Concepts, Architectures, and ML/AI Considerations. *IEEE Access* 11 (2023), 71634–71657. https://doi.org/10.1109/ACCESS.2023.3295694

[9] Tushar Chugh, Kanishka Tyagi, Rolly Seth, and Pranesh Srinivasan. 2023. Intelligent agents driven data analytics using Large Language Models. In *2023 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD).* IEEE, 152–157.

[10] Dana Edberg and Jeanne Wendel. 2018. *Healthcare Transformation: The Electronic Health Record.* Springer International Publishing, Cham, 121–145. https://doi.org/10.1007/978-3-319-93003-9_7

[11] FHIR. 2023. SQL on FHIR v2. https://github.com/FHIR/sql-on-fhir-v2?tab=readme-ov-file. Accessed: 2024-03-09.

[12] Jie Gao, Simret Araya Gebreegziabher, Kenny Tsu Wei Choo, Toby Jia-Jun Li, Simon Tangi Perrault, and Thomas W Malone. 2024. A Taxonomy for Human-LLM Interaction Modes: An Initial Exploration. *arXiv preprint arXiv:2404.00405* (2024).

[13] John Grimes, Piotr Szul, Alejandro Metke-Jimenez, Michael Lawley, and Kylynn Loi. 2022. Pathling: analytics on FHIR. *Journal of Biomedical Semantics* 13, 1 (2022), 23.

[14] Julian Gruendner, Christian Gulden, Marvin Kampf, Sebastian Mate, Hans-Ulrich Prokosch, Jakob Zierk, et al. 2021. A framework for criteria-based selection and processing of fast healthcare interoperability resources (FHIR) data for statistical analysis: design and implementation study. *JMIR medical informatics* 9, 4 (2021), e25645.

[15] Health Level Seven International (HL7). 2021. FHIR Overview. https://www.hl7.org/fhir/overview.html Accessed: 2024-07-14.

[16] Rogers Jeffrey Leo John, Dylan Bacon, Junda Chen, Ushmal Ramesh, Jiatong Li, Deepan Das, Robert Claus, Amos Kendall, and Jignesh M Patel. 2023. Datachat: An intuitive and collaborative data analytics platform. In *Companion of the 2023 International Conference on Management of Data.* 203–215.

[17] LangChain. 2024. LangGraph by LangChain. https://www.langchain.com/langgraph. Accessed: 2024-07-14.

[18] Dianbo Liu, Ricky Sahu, Vlad Ignatov, Dan Gottlieb, and Kenneth D Mandl. 2019. High performance computing on flat FHIR files created with the new SMART/HL7 bulk data access standard. In *AMIA Annual Symposium Proceedings*, Vol. 2019. American Medical Informatics Association, 592.

[19] Paula Maddigan and Teo Susnjak. 2023. Chat2vis: Generating data visualisations via natural language using chatgpt, codex and gpt-3 large language models. *Ieee Access* (2023).

[20] Joshua C. Mandel, David A. Kreda, Kenneth D. Mandl, Isaac S. Kohane, and Rachel Badovinac Ramoni. 2016. SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. *Journal of the American Medical Informatics Association : JAMIA* 23 (2016), 899 – 908. https://api.semanticscholar.org/CorpusID:25510437

[21] Neo4j. 2024. Neo4j Knowledge Graph Dataset. https://neo4j.com/. Accessed: 2024-03-13.

[22] OpenAI. 2024. Embeddings. https://platform.openai.com/docs/guides/embeddings/ Accessed: 2024-07-17.

[23] Qdrant Technologies. 2024. Qdrant Website. https://qdrant.tech/. Accessed: 2024-03-13.

[24] Bo Qiao, Liqun Li, Xu Zhang, Shilin He, Yu Kang, Chaoyun Zhang, Fangkai Yang, Hang Dong, Jue Zhang, Lu Wang, et al. 2023. TaskWeaver: A Code-First Agent Framework. *arXiv e-prints* (2023), arXiv–2311.

[25] James Sorace, Hector H. Wong, Thomas DeLeire, Di Xu, Shira Handler, Benjamin Garcia, and Thomas MaCurdy. 2020. Quantifying the competitiveness of the electronic health record market and its implications for interoperability. *International Journal of Medical Informatics* 136 (2020), 104037. https://doi.org/10.1016/j.ijmedinf.2019.104037 Epub 2019 Nov 27.

[26] Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. 2018. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association* 25, 3 (March 2018), 230–238. https://doi.org/10.1093/jamia/ocx079

[27] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629* (2022).

A Generated Charts Samples

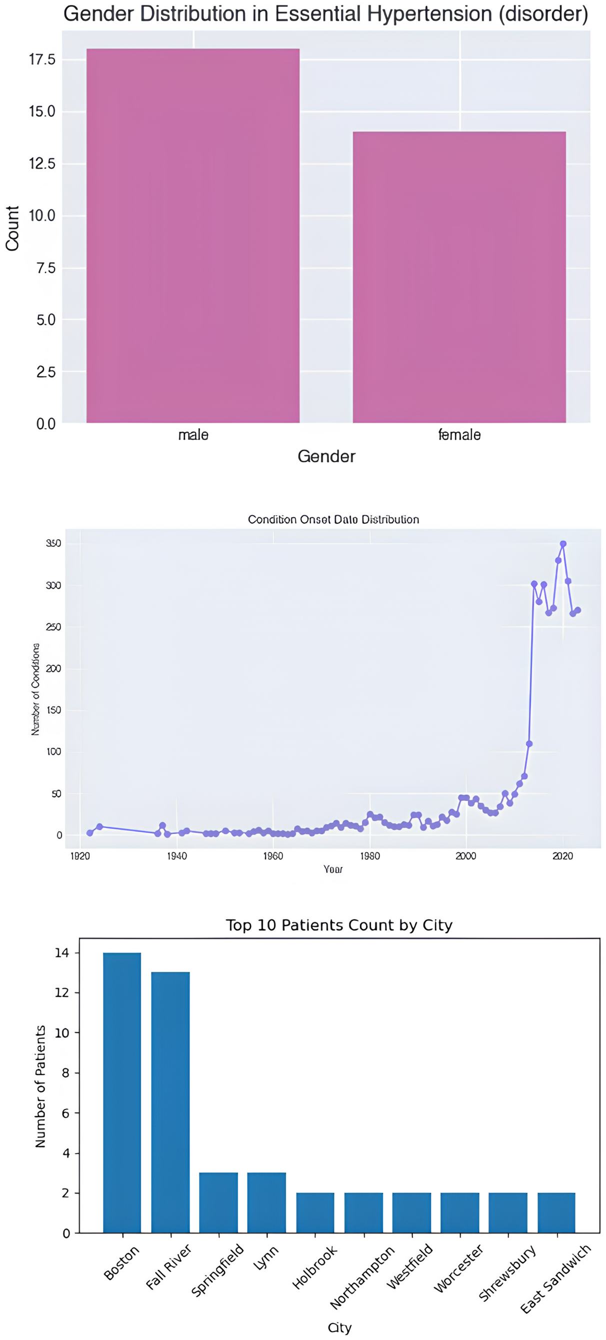


Figure 4: a. The top chart displays the number of condition onsets over time.
b. The middle chart displays the number of condition onsets over time.
c. the bottom chart displays the number of condition onsets over time.

B Code Availability

The source code for this project is available at https://github.com/Mars-2030/FHIRVIz.

Received 15 July 2024; accepted 22 September 2024