Benchmarking Large Language Models for Clinical Data Retrieval via FHIR: A Prompt-and-Feedback Baseline for Tool-Augmented Agentic Systems

Johannes Schmidt
Nordakademie
Elmshorn, Germany
mail@johannesschmidt.net

Arne Ewald
Nordakademie
Elmshorn, Germany
mail@aewald.net

***Abstract*—This paper evaluates the ability of Large Language Models (LLMs) to generate syntactically and semantically correct FHIR REST queries from natural language for retrieving medical data from Clinical Data Repositories (CDRs). The goal is to explore natural language interfaces that can improve clinical data access and interoperability across healthcare systems. Six experiments were conducted with nine LLMs, comparing baseline prompting against structured prompts, few-shot examples, and feedback-loops using HTTP error codes or messages. Results show that even without external tools, several models achieve high syntactic validity, with accuracy further improved by prompt-engineering and simple feedback mechanisms. However, semantic correctness remains challenging, in particular for medical codes, date logic, and site-specific conventions. Error analyses demonstrate where Retrieval-Augmented Generation (RAG), terminology services, and agentic repair could provide immediate gains, making this work a valuable prompt-centric baseline for the next generation of tool-augmented clinical query systems.**

Keywords—Large Language Models (LLMs), Fast Healthcare Interoperability Resources (FHIR), Natural Language Processing (NLP), Healthcare Interoperability, Retrieval-Augmented Generation (RAG), Agentic Workflows

I. INTRODUCTION

The digitalization of healthcare has been driven by various new technologies, government initiatives, and economic interests of healthcare providers in recent years. Diagnosis and treatment decisions are no longer solely based on a single examination but on a holistic picture from various data sources. This leads to a continuous increase in the amount of data that is available and processed in specialized systems from different manufacturers in different formats. For optimal patient care, it is therefore crucial that access to and use of this data is as seamless as possible to relieve and support clinical staff in the best possible way.

To use data between specialized systems, interoperability standards such as Health Level 7 (HL7) or Fast Healthcare Interoperability Resources (FHIR) are already used in many clinical systems. Usage is further strengthened by state-driven initiatives, such as the Electronic Health Record (EHR) or the Hospital Future Act (KHZG) in Germany [1]. At a technical level, the use of such standards allows to merge data and exchange it between systems. However, the success of interoperability depends on the respective version and implementation of the standards. Therefore, to evaluate and use the standardized data, expert knowledge about the system and its environment or additional software solutions, that can access and interpret the data across systems, are still required.

When using FHIR, data from connected systems can be stored in structured form in a Clinical Data Repository (CDR). The data is available as FHIR resources, which represent persons, objects, and processes in the healthcare system. The resources can be linked to each other to establish context. Various interfaces can be used with a CDR to exchange and retrieve data. In particular, the REST API is recommended for this purpose [2]. By using REST queries, complex health data can be systematically queried and evaluated.

Artificial intelligence, and especially Large Language Models (LLMs), are increasingly being integrated in various products and services. LLMs are specialized in interpreting and generating natural human language. They can be trained for specific use cases or utilized in various areas by using a comprehensive and diverse amount of training data. With these generalist LLMs, it is possible to generate domain-specific database queries in SQL or REST using natural language [3][4].

The combination of FHIR and LLMs could support medical staff in retrieving clinical data from a database or from a CDR using natural language. By using LLMs to generate REST queries, it is not necessary for the querying person to know the exact specifications of the standard (FHIR) and its corresponding syntax. For example, codes used to identify diseases or diagnoses could be recognized by an LLM based on their trivial name. Furthermore, the ability to interpret natural language makes it possible to formulate queries verbally. Thus, inputs and queries can also be made when it is not possible to use hands or to ensure better infection protection [5].

This paper investigates to what extent it is possible to generate syntactically correct REST queries for a FHIR CDR from natural language using different LLMs. In addition, it is examined if the content of the syntactically correct REST call still complies with the semantics of the original query. Furthermore, the influence of various modifications, such as prompt engineering, feedback, and iterations, on the results is examined. The required test data and experiments are designed in a way that further LLMs and prompting techniques can be tested with the developed framework and compared with the results in the future.

While this study was originally conducted in the context of prompt-based and feedback-enhanced LLM usage, the field

has rapidly evolved in the past year. Retrieval-Augmented Generation (RAG) [6], agentic workflows [7], and structured tool-use protocols like Model Context Protocol (MCP) [8][9] have significantly enhanced the ability of LLMs to reason over structured medical data with external knowledge and iterative refinement. However, this work presents a prompt-based baseline that can inform and complement newer, tool-augmented approaches to accessing and querying FHIR-based data.

In the following section, the related work of this study area is summarized. Based on this, the methodology is described, followed by the presentation of the results. In the discussion, the results are further elaborated and set in context for current and future research. Finally, the findings are summarized in the conclusion.

II. Related work

Research at the intersection of LLMs and healthcare data interoperability with structured data, such as FHIR is still emerging. However, there is a lack of comprehensive studies combining the two areas for query generation to improve not only syntactic but also semantic interoperability between healthcare systems. This gap presents an opportunity for the current study to contribute novel insights to the field. Due to the limited research available in this field, the literature review was extended to related areas as well.

Interoperability in healthcare, particularly through standards like FHIR, has been widely discussed. Torab-Miandoab et al. [10] summarize these findings, highlighting benefits including improved patient care through seamless data transfer, optimized processes, cost reduction, and enhanced understanding of medical data concepts. The authors emphasize the importance of semantic interoperability to ensure consistent interpretation of data across systems. The adoption and use cases of FHIR as one of the most used interoperability standards in healthcare have been studied by Ayaz et al. [11], who found frequent use in mobile applications, clinical research, and hospital information systems. Government initiatives like Germany's KHZG further promote FHIR adoption for interoperability.

Bedi et al. [12] reviewed existing use cases for LLMs in healthcare. Common use cases include diagnosis support, clinical note generation, patient education, decision management, and medical research. The authors note promising results in diagnosis support and automated documentation but highlight challenges in fairness, bias, robustness, and the need for studies using real patient data. However, indications for using LLMs and REST to improve interoperability were not mentioned. Other studies have explored using LLMs to generate REST queries, though not specifically in healthcare. Song et al. [4] investigated LLMs' ability to generate REST calls for Spotify and TMDB (The Movie Database) APIs, achieving up to 72.70% accuracy with GPT-3.5. Another study by Kim et al. [3] used a few-shot learning approach with GPT-3.5-Turbo to generate REST calls for various APIs, achieving an average of 72.68% syntactic and semantic correctness across different interfaces.

The direct combination of LLMs and FHIR has been minimally explored. Li et al. [13] used GPT-4, Llama-2-70B, and Falcon-180B to convert treatment notes into FHIR MedicationStatement resources, achieving up to 90% accuracy with GPT-4. The authors noted challenges with code systems and the lack of standardized test data in this domain. Furthermore, HealthSage AI [14] developed an adapter for Llama-2 to translate treatment notes into FHIR resources, claiming better performance than GPT-4 but without detailed evaluation.

Recently, the field has evolved toward more sophisticated architectures, combining LLMs with external tools and knowledge retrieval mechanisms. RAG and agentic workflows are increasingly used in clinical contexts to mitigate hallucination risks and improve semantic robustness. Rezaei et al. proposed Agentic Medical Knowledge Graphs (AMG-RAG), a domain-aware system that combines graph-based retrieval with RAG and agent-style decision making, resulting in a higher factual consistency in clinical data tasks [15]. In a recent preprint, Jiang et al. [16] introduce MedAgentBench, a comprehensive benchmark for evaluating medical LLM agents in interactive FHIR-based environments [16]. Their framework assesses multi-step clinical tasks such as test ordering or EHR querying, demonstrating the potential of agentic systems to outperform static prompt-based approaches.

These developments underscore the importance of benchmarking prompt- and feedback-based methods – such as those presented in this study – against increasingly capable agent-based and tool-integrated pipelines. To the best of our knowledge, previous benchmarks either focus on synthetic data across different database types [17], conversion into FHIR bundles [18], or multi-step agentic workflows using FHIR [16]. None evaluate the standalone capability from natural language using only prompting and feedback. This paper addresses this specific gap.

III. Methods

Six experiments were conducted to examine the capabilities of different LLMs to create FHIR REST queries from natural language to improve interoperability in healthcare. Selection criteria for the LLM set included model size, different developers, open-source status, availability, and cost. Nine LLMs were selected: Gemini 1 Pro (Google DeepMind), Gemini 1.5 Pro (Google DeepMind), GPT-4o (OpenAI), Llama 3 8B (Meta), Llama 3 70B (Meta), Mistral 7B (Mistral AI), Mixtral 8x7B (Mistral AI), Phi-3 3.8B (Microsoft), and Medllama2 (Adapter for Llama 2). This selection covers a range of sizes and developers, ensuring diversity in training approaches and capabilities. The inclusion of both general-purpose (e.g., GPT-4o, Gemini) and domain-adapted models (MedLlama2) allows comparison between general LLM capabilities and those fine-tuned or adapted for medical contexts.

Due to the lack of standardized datasets for this specific task, custom test data was created. Five FHIR resources were selected for testing: Observation, Patient, Condition, Encounter, and Specimen. These resources were selected based on their prevalence in clinical workflows and their representation in the literature and FHIR case studies [11]. For each resource, 15 test cases were manually defined, resulting in a total of 75 test queries in natural language. The test data covers a variety of query types in English language. Among others, different formulations (statements and questions), code systems (SNOMED CT and LOINC), date ranges and relative time queries, combinations of multiple parameters, and edge cases to test model understanding (e.g., ambiguous name queries) were used.

The 75 natural language queries were processed by each LLM with different prompts to generate REST requests for the FHIR-CDR, which responded with the requested patient data. Six experiments were designed to evaluate the LLMs' performance and the impact of different prompt engineering techniques and feedback mechanisms: (1) Baseline: Simple prompt with minimal information, (2) Improved Prompt: More detailed prompt with guidelines, (3) One-Shot Learning: Improved prompt with one example, (4) Few-Shot Learning: Improved prompt with three examples, (5) Feedback (error code): Iterative improvement using HTTP error codes, (6) Feedback (error message): Iterative improvement using detailed error messages.

These experimental conditions were selected to reflect a progression from zero-shot to more guided prompt-based generation, mirroring standard practices in prompt engineering research. Baseline and improved prompts establish reference points for minimal and structured input. The one-shot and few-shot conditions test the model's ability to generalize from examples, a known strength of transformer based LLMs [19]. Finally, the feedback conditions simulate real-world interaction patterns, where LLMs receive signals from system outputs (e.g., error codes or messages) and iteratively refine their responses. This progression allows us to assess how different levels of guidance and feedback influence the accuracy of FHIR REST query generation, especially in the absence of fine-tuning or tool augmentation.

The performance of each LLM in each experiment was assessed based on syntactic and semantic accuracy. Syntactic accuracy measures whether the generated output string was a syntactically valid FHIR REST query URL. Firstly, the correctness of the base URL, valid resource types and special characters were checked. This was assessed through automated parsing, validation rules and attempting to execute the query against the FHIR server. Afterward, a binary score (correct/incorrect) was assigned per query. The semantic accuracy measures whether the generated query, if syntactically correct, accurately reflected the meaning and intent of the natural language question and would retrieve the correct data from the CDR. Retrieved results from syntactically correct queries were automatically compared against the expected results based on the natural language question and the ground-truth query. For this measure, a binary score (correct/incorrect) was assigned as well.

Each experiment was repeated ten times to assess consistency and account for potential variations in model outputs. A Python-based framework was developed to automate the testing process, ensuring reproducibility, and minimizing human error. The framework was designed to be flexible, allowing for easy addition or removal of models and experiments. All experiments were conducted in a Python environment using the Ollama toolbox [20].

IV. Results

The findings are structured into quantitative and qualitative results, providing a comprehensive overview of the data.

A. Quantitative results

Prompt engineering strategies as applied in experiments #1 to #4 led to improvements in both syntactic and semantic results. When comparing results from experiment #4 to the baseline established in experiment #1, all scores were improved except the syntactic correctness of the Mixtral 8x7B model (see Table I). The highest syntactic results up to the fourth experiment were achieved with Gemini 1.5 Pro (89.20 % accuracy), GPT-4o (88.00 %), and Llama 3 70B (73.60 %). The highest semantic values were also achieved with the large models, GPT-4o (69.47 %), Gemini 1.5 Pro (68.40 %) and Llama 3 70B (51.60 %). Although the other small- and medium-sized models achieved lower scores, the increase in value is highest for these models. However, this increase is always less pronounced in terms of semantic correctness, and the smallest models (Phi-3 3.8B and Medllama2) lag in semantic correctness with 12.53 % and 6.67 %.

TABLE I. SYNTACTIC AND SEMANTIC RESULTS OF EXPERIMENT #4

	Syntactic correctness % (compared to baseline experiment #1)	**Semantic correctness % (compared to baseline experiment #1)**
Gemini 1 Pro	70.00 (+16.53)	42.80 (+12.67)
Gemini 1.5 Pro	**89.20** (+6.80)	68.40 (+9.73)
Mistral 7B	52.40 (+20.53)	19.73 (+6.53)
Mixtral 8x7B	57.60 (-4.93)	43.60 (+8.27)
Llama 3 8B	61.87 (+26.27)	30.27 (+17.07)
Llama 3 70B	73.60 (+4.93)	51.60 (+3.20)
GPT-4o	88.00 (+2.53)	**69.47** (+2.94)
Phi-3 3.8B	33.73 (+21.20)	12.53 (+9.20)
Medllama2	26.80 (+21.47)	6.67 (+5.34)

The use of feedback-loops led to further improvements in both syntactic and semantic accuracy. Since the same initial prompt is used in these experiments (#5 and #6) as in the fourth experiment, the results of experiment #6 are compared with the results of the fourth experiment in Table II. In terms of syntactic correctness, the highest results were achieved with the large LLMs Gemini 1.5 Pro (100 %), GPT-4o (99.87 %) and Llama 3 70B (99.87 %). However, the results of the medium-sized models were also significantly improved. Although with small models such as Phi-3 3.8B improved results were achieved, they still lag behind. The highest semantically correct results were achieved with GPT-4o (78.13 %), Gemini 1.5 Pro (73.47 %) and Llama 3 70B (62.13 %). The large- and medium-sized Llama 3 and Mistral models (large and small) benefited particularly strongly from the feedback-loops, with an increase of over ten percentage points.

TABLE II. SYNTACTIC AND SEMANTIC RESULTS OF EXPERIMENT #6

	Syntactic correctness % (compared to experiment #4)	**Semantic correctness % (compared to experiment #4)**
Gemini 1 Pro	92.53 (+22.53)	51.20 (+8.40)
Gemini 1.5 Pro	**100.00** (+10.80)	73.47 (+5.07)
Mistral 7B	88.13 (+35.73)	29.73 (+10.00)
Mixtral 8x7B	95.73 (+38.13)	54.53 (+10.93)
Llama 3 8B	98.93 (+37.06)	42.53 (+12.26)
Llama 3 70B	99.87 (+26.27)	62.13 (+10.53)
GPT-4o	99.87 (+11.87)	**78.13** (+8.66)
Phi-3 3.8B	67.20 (+33.47)	14.67 (+2.14)
Medllama2	42.67 (+15.87)	8.13 (+1,46)

Prompt engineering techniques (improved prompt, one-shot, and few-shot learning) consistently improved both syntactic and semantic accuracy, with larger relative gains observed for smaller models. Feedback mechanisms in experiments #5 and #6 led to additional performance gains in both syntactic and semantic correctness by enabling iterative query refinement across multiple attempts (see Fig. 1).

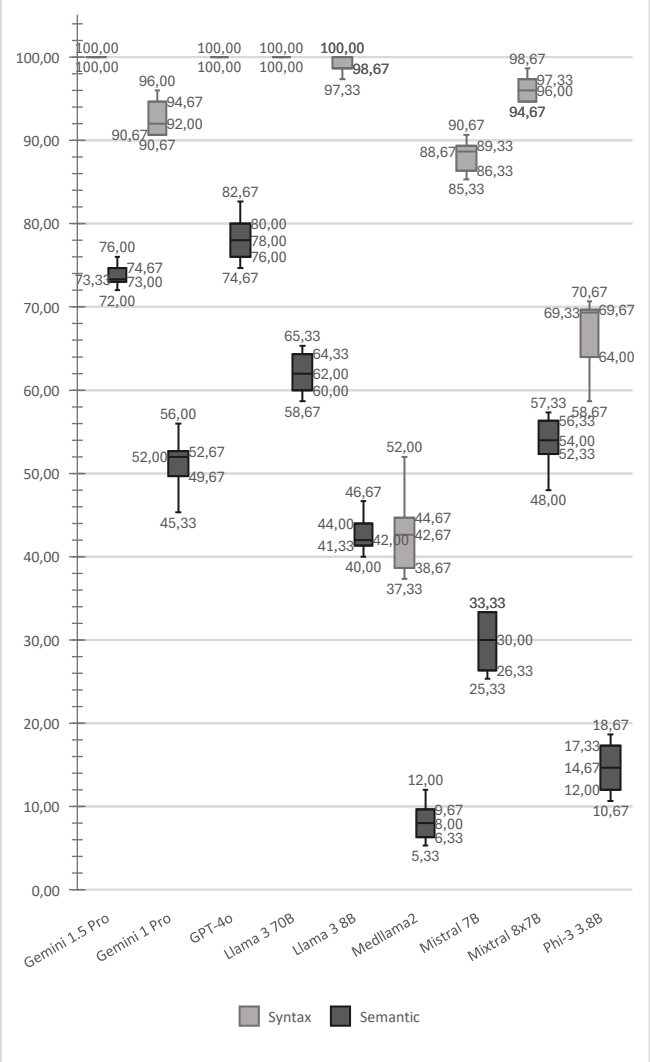

Fig. 1: Syntactic and semantic results experiment #6

B. *Qualitative results*

For the top three models (Gemini 1.5 Pro, Llama 3 70B, and GPT-4o), a detailed analysis of semantic errors has been conducted. To better understand the causes of semantic inaccuracies, errors were grouped into five categories:

(1) *Alternative interpretations*: Cases where the generated query could be correct under a different understanding of FHIR implementation (e.g., using technical IDs instead of clinical IDs or using different but correct code systems). (2) *Date and time handling*: Difficulties with relative date queries and occasional unit conversion errors. (3) *Code and code system errors*: Incorrect use of SNOMED CT or LOINC codes or using codes when codes were optional or irrelevant for the query. (4) *Parameter misuse*: Using incorrect parameters or misapplying correct ones (e.g., using name instead of name.family or diagnosis:code instead of diagnosis.code). (5) *Resource misidentification*: Occasionally using the wrong FHIR resource type, particularly confusing Condition with Patient or Observation.

Errors concerning alternative interpretations (1) were most frequent for GPT-4o (55 %) but they were also the second most common category with Llama 3 70B (19 %) and Gemini 1.5 Pro (28 %). Code and code system errors (3) were the most prominent category in the results of Llama 3 70B (51 %) and Gemini 1.5 Pro (32 %). Results from GPT-4o were less often prone to errors of that category (11 %) but were more susceptible to date and time handling errors (2) (18 %). Parameter misuse (4) and resource misidentification (5) were less frequently represented across all models (see Fig. 2).

These patterns suggest that providing explicit implementation context and code system references may mitigate many of these errors.

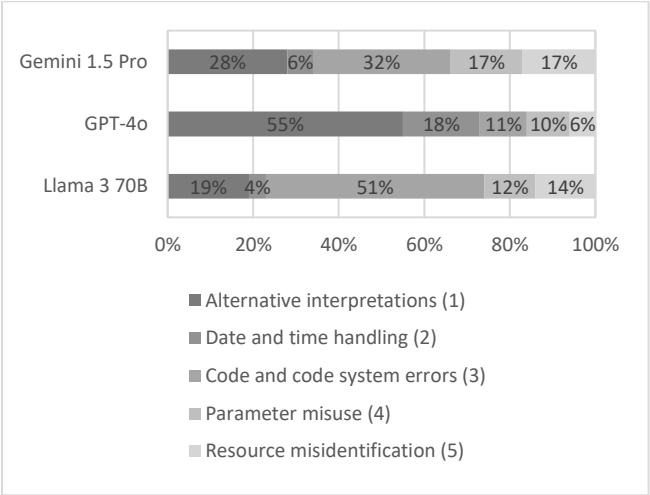

Fig. 2: Relative distribution of semantic error categories

V. Discussions

The conducted experiments show that LLMs can reliably generate syntactically valid FHIR REST queries from natural language, with accuracy further boosted by structured prompts, few-shot examples, and simple feedback-loops. Semantic correctness remained the harder problem: even top models (GPT-4o, Gemini 1.5 Pro, Llama 3 70B) made clinically meaningful mistakes – especially around medical code systems, date-handling, and implementation-specific assumptions. This pattern aligns with broader observations in clinical NLP that surface-form correctness (well-formed output) is easier to achieve than task-grounded validity (correct retrieval conditioned on standards/terminologies and site-specific conventions) [13].

Since the design of this study with focus on prompt- and feedback-based experiments, new technologies have emerged: a) RAG pipelines that inject structured context at inference time [6] and b) agentic workflows that iteratively plan, call tools, and verify intermediate results [16]. Medical agent benchmarks emphasize multi-step, EHR-based tasks and show that agentic systems can outperform static prompting on complex sequences, including FHIR-backed interactions. This work, therefore, serves as a prompt-centric baseline that is complementary to, and can be extended by, these pipelines.

The largest observed share of semantic errors are misused SNOMED/LOINC codes, ambiguous resource parameters, and relative time misinterpretations. These are precisely the

kinds of mistakes that retrieval-grounding can possibly mitigate by injecting the FHIR specification documentation, site-specific implementation notes, and the correct coding system at generation time. Systematically integrating RAG (with caching, schema-aware chunking, and strict citation of the retrieved snippet) is an immediate path to reducing these error classes. Recent healthcare-focused RAG reviews argue the same: hallucination reduction and semantic robustness hinge on good retrieval design and implementation [6].

When queries depend on medical ontologies or multi-hop relations (e.g., mapping colloquial problem names to SNOMED/LOINC, traversing from encounter to observation to code system), graph-structured RAG can prove beneficial. Agentic pipelines like AMG-RAG can build or maintain medical knowledge graphs and use graph-conditioned retrieval and transparent reasoning to improve factual accuracy and interpretability [15].

The feedback-loop experiments based on HTTP error codes and messages already simulate a narrow slice of agentic behavior. Full agent systems can expand this into plan-act-observe-reflect patterns. They could propose a query, execute it, read the server response, check coverage and repair if needed. Benchmarks like MedAgentBench evaluate these approaches in real and simulated FHIR environments. In line with our research, they show that agentic patterns, including tool-use and multi-step approaches, can outperform purely single-shot prompt-based methods [16].

To integrate tool-use for production systems, protocols like MCP are emerging to reliably expose tools (FHIR servers, terminology services) to LLMs with consistent schemas and security postures. These are crucial when moving from a sandbox CDR to hospital infrastructure. Early landscape and safety analysis highlight early adaptation and the need for defense-in-depth when giving models tool access [21].

The qualitative breakdown highlights three key areas where targeted engineering can improve performance. First, for medical codes and code systems, code selection should be routed through a terminology service with value-set constraints rather than relying on the model to guess. This process should be grounded in retrieval, prefer explicit "system code" pairs, and fail when ambiguity remains, in line with best practices in clinical AI system design [13]. Second, for date and time logic, relative time expressions should be converted into absolute ranges using deterministic helper tools, with unit tests in place to handle edge cases such as time zones and inclusive or exclusive bounds. Third, for implementation variance across FHIR installations, site-specific quirks, such as custom name components or non-standard search parameters, should be surfaced through retrieval and validated before query execution.

Larger models dominated in absolute performance, but small and medium models benefited most from structure and feedback. This is in line with recent reports that tool-use and retrieval disproportionately help smaller models close the gap. In regulated settings, arguments towards smaller models include data residency, privacy, and vendor lock-in. Our results suggest that with good retrieval and agentic repair workflows, smaller open-source models can be viable for query-generation tasks at lower cost.

Two risks even persist when the syntax is perfect: a) Wrong but plausible queries returning incomplete data and b) Drifts in specification of different FHIR versions or implementation. Current reviews stress that clinical LLM systems must adopt grounded evaluation, error taxonomies, and human-in-the-loop guardrails before deployment. Benchmarks that simulate realistic EHR workflows (ordering, chart review, cohorting) are a step forward but should be coupled with site-level validations and prospective usability testing with clinicians and domain experts [6][16].

The evaluation is prompt-centric and offline. Models have no access to external tools, terminology services, or documentation beyond prompt examples. Syntactic and semantic scoring was binary, which is simple and transparent but can obscure partial correctness or clinically acceptable alternatives. Future work could adopt graded or multi-class scoring to capture nuances. Finally, results target FHIR R4 and a specific server. Generalization to other versions should be tested.

A practical path forward is to wrap this baseline with several approaches. Use RAG over FHIR specifications, local implementation guides and value sets. Apply more sophisticated agent-based repair loops that verify and iteratively refine queries using server responses. Develop MCP-style tool adapters to standardize secure access to FHIR, terminology, logging, and policy engines. A more fine-grained evaluation that distinguishes minor and major semantic errors. Public resources like MedAgentBench [16] can anchor comparisons, while graph-aware RAG [15] can target ontology-heavy requests.

VI. Conclusion

The results show that large language models can achieve high syntactic accuracy in generating FHIR REST queries from natural language without access to external tools, retrieval mechanisms, or integration protocols. This demonstrates that with well-structured prompts, few-shot examples, and simple feedback-loops, technically valid queries are possible out-of-the box from current LLMs. The study provides a transparent, reproducible, and cost-effective baseline for further evaluation. At the same time, the error analysis and feedback-loop experiments, particularly those leveraging HTTP error codes and messages, empirically highlight where tool use, RAG, and agentic workflows can bring immediate benefits. By exposing these limits of static prompting and showing how iterative repair reduces errors, the study motivates the integration of terminology services, site-specific retrieval, and protocol-based tool adapters such as the MCP. As such, this work not only benchmarks current capabilities for further evaluation but also charts a clear path for enhancing semantic robustness and clinical reliability in next-generation FHIR query systems.

References

[1] Gematik, *Digital Health and Interoperability in Germany: Digital Health und Interoperabilität in Deutschland.* [Online]. Available: https://www.ina.gematik.de/themenbereiche/digital-health-und-interoperabilitaet-in-deutschland

[2] HL7 International, *FHIR Summary.* [Online]. Available: https://hl7.org/fhir/summary.html

[3] M. Kim, T. Stennett, D. Shah, S. Sinha, and A. Orso, "Leveraging Large Language Models to Improve REST API Testing," in *Proceedings of the 2024 ACM/IEEE 44th International Conference on Software Engineering: New Ideas and Emerging Results*, Lisbon Portugal, 2024, pp. 37–41.

[4] Y. Song *et al.,* "RestGPT: Connecting Large Language Models with Real-World RESTful APIs," doi: 10.48550/ARXIV.2306.06624.

[5] W.-S. Jian *et al.,* "Voice-based control system for smart hospital wards: a pilot study of patient acceptance," *BMC Health Serv Res*, vol. 22, no. 1, 2022, doi: 10.1186/s12913-022-07668-1.

[6] L. M. Amugongo, P. Mascheroni, S. Brooks, S. Doering, and J. Seidel, "Retrieval augmented generation for large language models in healthcare: A systematic review," *PLOS Digital Health*, vol. 4, no. 6, e0000877, 2025, doi: 10.1371/journal.pdig.0000877.

[7] F. Gaber *et al.,* "Evaluating large language model workflows in clinical decision support for triage and referral and diagnosis," *npj Digit. Med.*, vol. 8, no. 1, p. 263, 2025, doi: 10.1038/s41746-025-01684-1.

[8] X. Hou, Y. Zhao, S. Wang, and H. Wang, "Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions," Mar. 2025. [Online]. Available: http://arxiv.org/pdf/2503.23278

[9] Model Context Protocol, *Model Context Protocol - Model Context Protocol.* [Online]. Available: https://modelcontextprotocol.io/

[10] A. Torab-Miandoab, T. Samad-Soltani, A. Jodati, and P. Rezaei-Hachesu, "Interoperability of heterogeneous health information systems: a systematic literature review," *BMC Med Inform Decis Mak*, vol. 23, no. 1, 2023, doi: 10.1186/s12911-023-02115-5.

[11] M. Ayaz, M. F. Pasha, M. Y. Alzahrani, R. Budiarto, and D. Stiawan, "The Fast Health Interoperability Resources (FHIR) Standard: Systematic Literature Review of Implementations, Applications, Challenges and Opportunities," *JMIR Med Inform*, vol. 9, no. 7, e21929, 2021, doi: 10.2196/21929.

[12] S. Bedi *et al., A Systematic Review of Testing and Evaluation of Healthcare Applications of Large Language Models (LLMs)*, 2024.

[13] Y. Li, H. Wang, H. Z. Yerebakan, Y. Shinagawa, and Y. Luo, "FHIR-GPT Enhances Health Interoperability with Large Language Models," *NEJM AI*, vol. 1, no. 8, 2024, doi: 10.1056/aics2300301.

[14] HealthSage AI, *HealthSage AI LLM - from clinical note to FHIR.* [Online]. Available: https://github.com/HealthSage-AI/healthsage-ai-llm

[15] M. R. Rezaei, R. S. Fard, J. L. Parker, R. G. Krishnan, and M. Lankarany, "Agentic Medical Knowledge Graphs Enhance Medical Question Answering: Bridging the Gap Between LLMs and Evolving Medical Knowledge," Feb. 2025. [Online]. Available: https://arxiv.org/pdf/2502.13010

[16] Y. Jiang *et al.,* "MedAgentBench: A Realistic Virtual EHR Environment to Benchmark Medical LLM Agents," Jan. 2025. [Online]. Available: http://arxiv.org/pdf/2501.14654

[17] Sithursan Sivasubramaniam, Cedric Osei-Akoto, Yi Zhang, Kurt Stockinger, and Jonathan Fuerst, "SM3-Text-to-Query: Synthetic Multi-Model Medical Text-to-Query Benchmark," *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. [Online]. Available: https://openreview.net/forum?id=Pm0UzCehgB#discussion

[18] J. Delaunay, D. Girbes, and J. Cusido, "Evaluating the Effectiveness of Large Language Models in Converting Clinical Data to FHIR Format," *Applied Sciences*, vol. 15, no. 6, p. 3379, 2025, doi: 10.3390/app15063379.

[19] T. Brown *et al.,* "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020, doi: 10.5555/3495724.3495883.

[20] Ollama, *Ollama.* [Online]. Available: https://ollama.com/

[21] B. Radosevich and J. Halloran, "MCP Safety Audit: LLMs with the Model Context Protocol Allow Major Security Exploits," Apr. 2025. [Online]. Available: http://arxiv.org/pdf/2504.03767