LTRC-IIIT at MEDIQA-SYNUR 2026: Benchmarking a Fully Local, Training-Free RAG Pipeline

Aashwin Vaish, Dipti Misra Sharma
Language Technologies Research Center, KCIS
IIIT Hyderabad, India
aashwin.vaish@research.iiit.ac.in, dipti@iiit.ac.in

Abstract

In this paper we present our solution to MEDIQA-SYNUR 2026 shared task organized at LREC-ClinicalNLP workshop. The goal of the task is to populate Electronic Health Record (EHR) flowsheets using the transcriptions of nurse dictations, to alleviate the extensive manual labor associated with sifting through large flowsheets of clinical concepts. We propose a modular architecture combining heuristic-driven Retrieval-Augmented Generation (RAG) with grammar-constrained decoding on an open-weight, quantized, 8B-parameter model (Llama 3.1 Instruct). Our system achieves an F1 score of 0.57, significantly trailing the initial zero-shot experiments with GPT-4o (F1 $\approx$ 0.88) and placing it towards the lower end of the current leaderboard. We conduct a failure analysis of this approach while establishing a baseline for privacy-preserving, zero-shot documentation assistants.

Keywords: Clinical Documentation, Local LLMs, Zero-Shot Extraction, Data Privacy, Negative Results

1. Introduction

While the digitization of healthcare has expedited several aspects of clinical care, it has also placed additional burdens of documentation on the clinicians and the nurses (Budd, 2023). One such documentation task involves filling Electronic Health Records (EHR) flowsheets that record aspects of patient care like vitals, doctor's assessments, intake/output levels etc. These flowsheets often span multiple tabs with hundreds of ontological concepts that need to be sifted through to find the relevant fields. Naturally, automation of this task by populating the EHR flowsheet using nurse dictations will improve the time that nurses can devote to patient care. To encourage research on this front, the MEDIQA-SYNUR shared task was proposed (George Michalopoulos, 2026).

While the recent advances of commercially deployed LLMs including the likes of OpenAI's GPT and Google's Gemini have shown impressive results on information extraction in healthcare (Umeton et al., 2023), such solutions have increased concerns of data privacy and sovereignty over local open-weight solutions.

Furthermore, due to limited availability of public patient data for training on this task, any comprehensive LLM training task incurs a significant cost. So the question we ask is -

How effectively can we automate flowsheet extraction using only local, general-purpose resources without task-specific training?

We present a training free pipeline designed to run on consumer hardware. Our approach utilizes a standard retrieval architecture (Lewis et al., 2020) enhanced by heuristics, decomposing enumerable fields into individual vectors for retrieval and using deterministic rules to link dependent variables (e.g., units of measurement). For generation, we employ Llama 3.1 (8B Instruct)(Dubey et al., 2024) with grammar-constrained decoding to enforce schema compliance without fine-tuning. Our results yield an F1 score of 0.57, placing our method at the lower end of the current leaderboard. However, this lower performance comes with distinct operational advantages:

1. **Privacy:** Data never leaves the local machine.
2. **Adaptability:** The system is purely zero-shot, handling a new flowsheet requires only updating the JSON schema, not retraining.
3. **Accessibility:** The pipeline runs on standard hardware, lowering the barrier to entry for resource-constrained institutions.

In this work, we detail our heuristic retrieval strategies and provide a transparent error analysis, identifying the specific reasoning gaps in smaller scale LLMs that currently prevent them from matching supervised performance in the medical domain.

2. Methodology

Our pipeline consists of two distinct stages: a retrieval module designed to filter relevant flowsheet rows from a potential set of hundreds, and a generation module that extracts values using a local instruction LLM. The architecture prioritizes recall in the retrieval stage and precision in the generation stage, operating entirely on consumer-grade hardware (i.e., a single NVIDIA GPU).

Proceedings of the 8th Workshop on Clinical Natural Language Processing (Clinical NLP) @ LREC 2026, pages 136–140
16 May 2026.

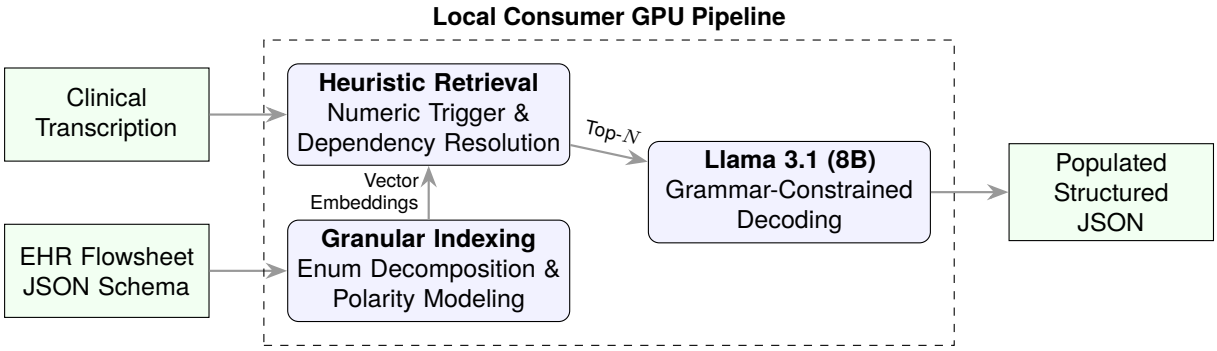

Figure 1: Architecture of the zero-shot, training-free RAG pipeline. The system operates entirely locally, utilizing heuristic-augmented retrieval to filter the JSON schema before passing candidates to the quantized 8B-parameter LLM.

2.1. Granular State Indexing

The easiest approach to make an RAG system would be to embed entire database rows as single vectors. In the context of flowsheet schemas, this obscures a critical semantic distinctions. For example, a row defining "Weightbearing Status" might contain the Enum options *["Full", "Partial", "Non-Weightbearing"]*. Embedding the row metadata alone fails to capture the semantic distance between these mutually exclusive states. To address this, we implement a state indexing strategy. Instead of a single vector per row, we generate multiple vectors based on the field type:

- **Categorical Fields:** We decompose Enum options into individual "child" vectors (e.g., *"Weightbearing: Non-Weightbearing"*). This allows the retriever to match specific patient states rather than generic field names.
- **Boolean Polarity:** For binary fields (Yes/No), we explicitly model polarity. Rather than embedding generic "Yes" and "No" tokens, which introduce noise, we generate positive assertions (e.g., *"Edema present"*) and negative assertions (e.g., *"No Edema"*).

All child vectors map back to their original parent Row ID, allowing us to retrieve the full schema definition if any specific state is detected.

2.2. Heuristic Enhanced Retrieval

Clinical transcriptions often have high information density and a disjointed syntax, along with disfluencies. After splitting on common disfluencies (e.g., "um", "uh"), we apply a sliding window chunking strategy, splitting text on commas rather than sentences. This reduces vector dilution, ensuring that short, distinct observations (e.g., *"lungs clear"*, *"heart regular"*) are represented by sharp, high-magnitude vectors.

Using the *sentence-transformers/all-MiniLM-L6-v2* embedding model, we augment the standard cosine similarity search (retrieving the top $N = 15$ candidates) with two deterministic heuristics to address common vector space failures:

2.2.1. Type Aware Retrieval

Vector models frequently fail to associate raw numbers (e.g., *"88"*, *"38.5"*) with their corresponding clinical concepts (e.g., *Heart Rate*, *Temperature*). We implement a **Numeric Trigger**: if a text chunk contains a digit, the system executes a parallel search restricted solely to fields defined as `NUMERIC` in the schema. The top-scoring numeric field is forced into the candidate list, prioritizing type compatibility over semantic similarity.

2.2.2. Deterministic Dependency Resolution

Certain fields, such as units of measurement (e.g., *Temperature Unit*), are rarely explicitly vocalized by clinicians but are structurally required by the EHR. Pure vector search often fails to retrieve these dependent variables. We address this via **Dependency Resolution**: if a primary numeric field is retrieved (e.g., *Temperature*), its associated metadata fields defined in the schema are automatically injected into the context window, bypassing the retrieval process entirely.

2.3. Constrained Generative Extraction

The final extraction is performed by **Llama-3.1-8B-Instruct**, quantized to 8-bit precision. While 8B-parameter models are generally capable of zero-shot extraction, they are prone to outputting invalid JSON and inventing values for unmentioned fields.

We mitigate these issues through **Grammar-Constrained Decoding**. We utilize the Grammar-Based Network Format (GBNF) engine within the `llama.cpp` serving infrastructure. Rather than relying on the LLM to learn JSON syntax through

prompt examples, the GBNF engine dynamically masks the model's output logits during inference. It assigns a zero probability to any token that violates the provided JSON schema. This strictly limits generation to valid brackets, schema-defined keys (e.g., `id` and `value`), and appropriate string enclosures, ensuring 100% syntactical validity without fine-tuning.

To control hallucination, we employ a strict "Silence Constraint" in the system prompt, instructing the model to output a null value if a field is not explicitly supported by the text. This shifts the burden of decision-making from implicit inference to explicit extraction.

2.3.1. Generation Prompt

```
You are a strict data extraction
engine.
RULES:
1. Output specific JSON only.
2. SILENCE CONSTRAINT: If a field
from the schema is NOT mentioned,
do NOT include it.
3. Select value(s) ONLY if they
appear in text or are HEAVILY
implied by BASIC LOGIC.
4. FORMAT: List of objects with
"id" and "value". All values must
be strings.

CONTEXT:
{transcription}

TARGET SCHEMA:
{candidates}
```

2.4. Deterministic Post-Processing

Despite strict JSON structural constraints, the generative model occasionally exhibits semantic slippage by paraphrasing rigid `SINGLE_SELECT` and `MULTI_SELECT` Enum values (e.g., generating *"weightbearing"* instead of the required *"weightbearing as tolerated"*). Because downstream EHR databases require exact string matches, we implement a two-step deterministic post-processing layer to sanitize the LLM outputs.

The system uses a localized vector-similarity check. We embed both the LLM's predicted string and all valid Enum options using the same *all-MiniLM-L6-v2* model utilized in the retrieval stage. The predicted value is then deterministically mapped to the valid option exhibiting the highest cosine similarity.

To prevent aggressive misclassification of genuine hallucinations, we enforce a strict minimum similarity threshold. If the highest-scoring option falls below this threshold (<0.6), the extracted value is discarded. This operation runs in $O(1)$ time relative to the database size, as it only embeds the isolated schema options rather than querying the global vector store, thereby correcting near-miss extractions without adding significant computation overhead.

3. Experiments and Results

3.1. Evaluation Setup

We evaluated our local-first extraction pipeline against the MEDIQA-SYNUR dataset (Corbeil et al., 2025). The dataset contains a concept schema in a JSON format that specifies a mixture of `NUMERIC`, `STRING`, `SINGLE_SELECT`, and `MULTI_SELECT` variables. We measured the system performance using the shared task evaluation script. Additionally, we measured performance of the retrieval step independently from the generation step to better analyze the impact of either step.

3.2. Retrieval-Stage Performance

To isolate the effectiveness of our heuristic-augmented vector search, we evaluated the Retrieval-Augmented Generation (RAG) filter independently of the downstream LLM. The heuristic based retrieval module achieved a Precision of 0.35, a Recall of 0.87, and an F1 score of 0.50. This high-recall, low-precision profile is an intentional architectural trade-off. In the context of EHR flowsheet extraction, missing a relevant clinical observation (a false negative) at the retrieval stage is a catastrophic failure, as the downstream LLM should not extract data it is never shown. But on the other hand, an irrelevant schema row (a false positive) merely consumes additional context window tokens.

By setting generous retrieval limits and applying our "Numeric Boost" heuristic, the retrieval system successfully surfaces 87% of the necessary schema context required to process the transcription. The burden of precision is then intentionally offloaded to the generative stage. The LLM acts as a secondary precision filter, using its semantic understanding and strict "Silence Constraints" to ignore the irrelevant candidate rows (the 65% noise generated by the 0.35 precision rate), ultimately raising the end-to-end system precision to 0.64.

3.3. Baseline Performance

Our zero-shot, locally hosted pipeline achieved a Precision of 0.6, a Recall of 0.55, and an F1 score of 0.57. For context, the current best score on the shared task leaderboard (F1 = 0.84) was around the same as the prelimnary experiments conducted by the organizers using GPT-4o, which achieved

an F1 score of approximately 0.88 in zero shot setting (Corbeil et al., 2025). While our system trails the organizers baseline by a significant margin (Table 1), it establishes a functional baseline for a fully zero-training, privacy-preserving architecture. More importantly, analyzing the delta between our 0.57 F1 and the 0.84 baseline reveals specific limitations of using compact LLMs for deterministic extraction.

3.4. Failure Analysis

To understand the performance gap, we conducted a manual qualitative analysis of the 204 missed occurrences in our validation set (the test set references had not been released at the time of writing). The errors largely fell into three distinct categories inherent to compact LLMs:

1. The semantic gap in abstract strings: Of the missed variables, a significant portion were generic `STRING` fields, specifically abstract categories like *"Secondary Diagnosis"*.

While the RAG filter easily retrieved concrete fields (e.g., *"Heart Rate"*), it struggled to map specific diseases mentioned in the text (e.g., *"Sepsis"*) to the abstract schema field without explicit anchor examples in the vector space.

2. Format slippage and enum paraphrasing: Despite grammar-constrained decoding ensuring valid JSON, the 8B-parameter model frequently paraphrased strict Enum values.

For example, generating *"weightbearing"* instead of the required *"weightbearing as tolerated"*.

While our deterministic post-processing layer (Section 2.4) successfully mitigated the majority of these paraphrasing errors by mapping them back to valid Enums via cosine similarity, extreme deviations (e.g. an entire sentence being picked from the dictation) or highly truncated generations occasionally fell below our safety threshold and remained penalized as false negatives.

3. The "Silence Constraint" Failure: LLMs exhibit a strong completion bias (Kumar et al., 2025). When presented with a flowsheet containing binary fields (e.g., *"Violence Checklist: [Yes, No]"*), the model feels compelled to select an option even if the clinical transcript is completely silent on the matter. Despite explicit prompt instructions to omit unmentioned fields, the model occasionally hallucinated *"No"* or *"None"*. This contributed heavily to precision degradation.

3.5. Throughput and Hardware Efficiency

While trailing in accuracy, our system excelled in operational efficiency. By implementing continuous batching with a shared context window across multiple execution slots, the entire pipeline (embedding generation, heuristic vector search, and 8-bit quantized LLM generation) ran satisfactorily on consumer-grade hardware. (NVIDIA 2080 Ti GPUs).

4. Conclusion and Future Work

In this work, we explored the feasibility of a fully local, zero-shot extraction pipeline for automating EHR flowsheet documentation. By combining granular state indexing and heuristic-augmented retrieval with grammar-constrained decoding on an 8B-parameter LLM, we established a privacy-preserving baseline that requires no task-specific fine-tuning or cloud API reliance. While our F1 score of 0.57 trails state-of-the-art closed-weight commercial AI systems, our failure analysis highlights the exact boundaries of compact LLMs in medical contexts: they excel at strict, explicit extraction but falter at implicit clinical reasoning and semantic gap bridging. However, the system's ability to run locally on consumer-grade hardware with high throughput (via continuous batching) makes it an operationally viable foundation for privacy-restricted environments. Future work will focus on narrowing the reasoning gap without sacrificing the zero-shot adaptability of the pipeline. We plan to explore multi-agent reflection loops, where a secondary local model verifies the logical consistency of the extracted flowsheet and the integration of lightweight, domain-specific knowledge graphs into the retrieval step to better anchor abstract clinical concepts (e.g., mapping specific diseases to "Secondary Diagnosis" fields).

Limitations

While this work establishes a baseline for local flowsheet extraction, several limitations must be acknowledged:

- **Performance Gap:** The 0.57 F1 score indicates that the system is not yet suitable for fully autonomous deployment. It is strictly an assistive technology requiring human verification.
- **Reasoning Constraints:** Our pipeline relies heavily on explicit mentions in the text. It struggles to infer values that a human clinician would consider "obvious" context (e.g., inferring that a patient walking unassisted implies a negative finding for fall risk).
- **Hardware Requirements:** Although technically "consumer-grade", our continuous batching setup utilizes two NVIDIA 2080 Ti GPUs (approx. 22GB VRAM). Leaving the possibility

System	Precision	Recall	F_1 score
Our Local Pipeline (Using Llama 3.1 8B)	0.60	0.55	0.57
GPT-4o Zero-Shot (Corbeil et al., 2025)	–	–	**0.88**

Table 1: Performance metrics of the retrieval stage, the full end-to-end local pipeline, and the commercial model GPT-4o zero-shot baseline.

for more aggressive quantization and serialization (as opposed to parallization) under further constraints. While vastly more accessible than the enterprise clusters required for 70B+ parameter models, this hardware footprint still exceeds the capacity of, say, individual practices, necessitating a dedicated local server architecture.

- **Language and Scope:** The pipeline was designed and evaluated exclusively on English-language clinical transcriptions and standard US-based EHR flowsheet schemas. Its robustness against heavily accented speech transcriptions or non-standard abbreviations remains untested due to the partially synthetic nature of the dataset.

Ethics Statement

The automation of medical documentation introduces significant ethical considerations, primarily concerning patient safety and data privacy.

Data Privacy: The primary motivation for this architecture is the protection of Protected Health Information (PHI). By utilizing a 100% local execution pipeline, patient data never traverses external networks or third-party APIs (e.g., OpenAI, Anthropic), ensuring strict compliance with data sovereignty regulations such as HIPAA and GDPR.

Patient Safety and Automation Bias: Errors in EHR flowsheets can lead to incorrect clinical decisions and patient harm. Because our system exhibits a known performance gap (F1 = 0.57) and compact LLMs are prone to occasional "hallucinations" (inventing positive or negative findings when none are mentioned), this system must strictly be deployed under a *Human-in-the-Loop* paradigm. The system is designed to draft flowsheet entries, not finalize them. Clinicians must review, edit, and authorize all generated structured data before it is committed to the patient's permanent medical record. Mitigating automation bias, where clinicians might blindly trust the AI's output to save time, is critical and requires careful UI/UX design in the final clinical application.

5. Bibliographical References

Jeffrey Budd. 2023. Burnout related to electronic health record use in primary care. *Journal of primary care & community health*, 14:21501319231166921.

Jean-Philippe Corbeil, Asma Ben Abacha, George Michalopoulos, Phillip Swazinna, Miguel Del-Agua, Jerome Tremblay, Akila Daniel, Cari Bader, Kevin Cho, Pooja Krishnan, Nathan Bodenstab, Thomas Lin, Wenxuan Teng, Francois Beaulieu, and Paul Vozila. 2025. Empowering healthcare practitioners with language models: Structuring speech transcripts in two real-world clinical applications.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Meta AI*.

Cari Bader Nate Bodenstab Asma Ben Abacha George Michalopoulos, Jean-Philippe Corbeil. 2026. Overview of the mediqa-synur 2026 shared task on observation extraction from nurse dictations. In *Proceedings of the 8th Clinical Natural Language Processing Workshop, ClinicalNLP@LREC 2026, Palma, Mallorca, Spain, May 16, 2026*. Association for Computational Linguistics.

Charaka Vinayak Kumar, Ashok Urlana, Gopichand Kanumolu, Bala Mallikarjunarao Garlapati, and Pruthwik Mishra. 2025. No llm is free from bias: A comprehensive study of bias evaluation in large language models.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Renato Umeton, Anne Kwok, Rahul Maurya, Domenic Leco, Naomi Lenane, Jennifer Willcox, Dana-Farber Committee, and Jason Johnson. 2023. Gpt-4 in a cancer center: Institute-wide deployment challenges and lessons learned.