Experience Retrieval-Augmentation with Electronic Health Records Enables Accurate Discharge QA

Justice Ou*
zo6@illinois.edu
University of Illinois Urbana-Champaign
Champaign, Illinois, USA

Tinglin Huang*
Yale University
New Haven, Connecticut, USA

Yilun Zhao
Yale University
New Haven, Connecticut, USA

Ziyang Yu
University of Waterloo
Waterloo, Ontario, Canada

Yuchen Kuang
Northeastern University
Boston, MA, USA

Yan Zeng
University of Pennsylvania
Philadelphia, Pennsylvania, USA

Peiqing Lu
Microsoft
Redmond, Washington, USA

Rex Ying
Yale University
New Haven, Connecticut, USA

Abstract

To improve the reliability of Large Language Models (LLMs) in clinical applications, retrieval-augmented generation (RAG) is extensively applied to provide factual medical knowledge. Beyond general medical knowledge, clinical case-based knowledge is also critical for effective medical reasoning, as it provides context grounded in real-world patient experiences. Motivated by this, we propose Experience Retrieval-Augmentation (ExpRAG) framework based on Electronic Health Record (EHR), explicitly leveraging **prompt optimization through retrieval methods** to dynamically construct informative prompts from similar patients' discharge reports. ExpRAG performs retrieval through a coarse-to-fine process, optimizing the prompt content by efficiently identifying similar patients, followed by extracting task-relevant clinical context. We conduct systematic ablation studies to evaluate different prompt-optimization strategies, including varying the number of retrieved patients, retrieval techniques, and the weighting of retrieved medical contexts. To evaluate RAG systems on EHR data including ExpRAG and medical agents, we introduce DischargeQA, a clinical QA dataset with 1,280 discharge-related questions across diagnosis, medication, and instruction tasks. Each problem is generated using historical EHR data to ensure realistic and challenging scenarios. Experimental results demonstrate that ExpRAG consistently outperforms traditional text-based rankers, achieving an average relative improvement of 5.2%, highlighting the importance of optimizing prompts with clinically relevant case-based knowledge for medical reasoning.

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '25, Toronto, ON, Canada
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3690624

CCS Concepts

• **Computing methodologies** → **Language resources**; **Natural language generation**; • **Applied computing** → **Health informatics**; • **Information systems** → **Question answering**; **Top-k retrieval in databases**.

Keywords

Electronic Health Record (EHR), Retrieval-Augmented Generation (RAG), LLM, Question answering

ACM Reference Format:
Justice Ou, Tinglin Huang, Yilun Zhao, Ziyang Yu, Yuchen Kuang, Yan Zeng, Peiqing Lu, and Rex Ying. 2025. Experience Retrieval-Augmentation with Electronic Health Records Enables Accurate Discharge QA. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '25).* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3690624

1 Introduction

Benefiting from pretraining on large-scale corpora, Large Language Models (LLMs) are capable of performing complex reasoning and have shown great promise in medical applications [21, 39]. One important application is inferring clinical conditions, including diagnosis and medication, which can be formulated as a question-answering (QA) task [2, 9, 31]. However, LLM agents often suffer from hallucinations and a lack of domain-specific knowledge, which limits their reliability in real-world medical applications.

To address this, prior studies have resorted to retrieving factual knowledge from open-ended databases to provide context, such as the description of drugs from Wikipedia [35, 37]. Such external knowledge enables LLMs to access general medical facts, thereby improving response accuracy. However, introducing such general facts cannot effectively help LLMs solve real clinical cases, which often involve coexisting clinical conditions. For example, as shown in Figure 1, adjusting a patient's warfarin dosage requires reasoning based on the specific clinical context, whereas conventional retrieval can only provide the standard dosage for warfarin, which is irrelevant in this case.

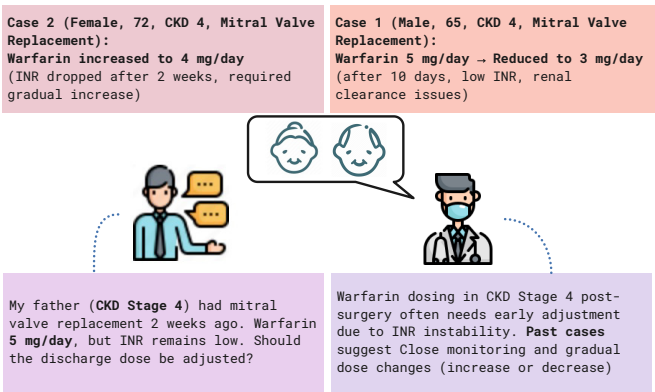

Figure 1: An illustrative example of utilizing experience from relevant clinical cases to support medical decision: adjusting a patient's warfarin dosage based on the specific clinical context rather than relying on a generic standard dose

In light of this, we argue that clinical case-based knowledge is crucial for effective medical reasoning. Experienced clinicians often rely on past cases with similar conditions to guide diagnosis, treatment decisions, and discharge planning. To systematically incorporate such case-based knowledge into LLM inputs, we frame retrieval-augmented generation (RAG) as a form of **prompt optimization**. Specifically, strategically selecting and embedding clinically relevant information retrieved from Electronic Health Records (EHR) optimizes prompts for improved reasoning performance. Motivated by this, we propose the **Experience Retrieval-Augmentation (ExpRAG)** framework, explicitly designed to optimize prompts by retrieving contextually rich clinical experiences from a large-scale EHR database, MIMIC-IV [16]. ExpRAG breaks down this retrieval-driven prompt optimization into two clear steps: (1) report ranking, which efficiently identifies patients with similar medical conditions by adjusting retrieval parameters such as similarity coefficients and the number of similar patients; and (2) experience retrieval, which optimizes the final prompt by extracting task-relevant context from these patients' discharge reports. This explicit prompt optimization ensures that the LLM's reasoning is consistently grounded in real-world clinical experiences.

To evaluate capability of ExpRAG and other RAG methods/agents in medical reasoning, we introduce DischargeQA, a clinical dataset including 1,280 QA pairs dedicated to discharge-related problems. The dataset primarily includes three types of problems: simulating the discharge process of final diagnosis, medication prescription and post-discharge instructions. For each problem, we follow the data structure of the discharge report and select the content preceding the question as the problem background to avoid label leakage. Additionally, we index the option candidates using EHR to generate contextually relevant options, ensuring a non-trivial and clinically meaningful challenge for the model.

We evaluate the performance of five different LLMs and compare ExpRAG with the text-based report ranker using DischargeQA. [1]

[1]Uploading to https://physionet.org/ , please check updates on Github: https://github.com/jou2024/EXPRAG

The results demonstrate the effectiveness of using EHR to retrieve relevant clinical experience, as it consistently improves the performance of LLM backbones and outperforms the text-based ranker with an average relative improvement of 5.2%. Our main contributions are summarized as follows:

- We propose ExpRAG, an EHR-based experience retrieval-augmentation framework, shedding light on the potential of leveraging past clinical cases to enhance LLM performance in medical reasoning tasks.
- We introduce DischargeQA, a medical QA dataset for discharge-related questions, designed to evaluate LLMs' ability to simulate the clinical decision-making process during patient discharge with a more challenging setup.
- Our results demonstrate the advantage of ExpRAG over the text-based ranker, highlighting the effectiveness of EHR in providing clinically meaningful context.

2 Related Work

Retrieval-Augmented Generation (RAG). RAG has become a key paradigm for overcoming the static knowledge limitations of LLMs by retrieving external information [6, 28]. Traditional RAG frameworks typically use dense retrieval methods to augment generative tasks [3, 36]. While effective in general QA, these methods often lack domain specificity, which is critical in healthcare [22]. Recent advancements like ClinicalRAG [22] and MIRAGE [35] address this by integrating structured EHR data and clinical notes for diagnosis and treatment planning. However, existing benchmarks primarily focus on isolated information retrieval [15, 18], overlooking the complexities of reasoning over patient histories and similar cases. To bridge this gap, our work extends RAG by combining structured EHR data with discharge summaries, enabling experience-driven reasoning for more realistic and reliable medical QA.

Medical QA Benchmark. EHRSQL [19] and DrugEHRQA [1] target structured data queries, with the former addressing SQL-based operations and the latter focusing on drug-related questions. EHRNoteQA [18] and RadQA [32] leverage clinician-verified QA pairs from discharge summaries and radiology reports, while MedQA [13], MedMCQA [26], and PubMedQA [14] evaluate LLMs with questions from medical exams or PubMed articles. Discharge-summary-focused datasets like emrQA [27] and CliniQG4QA [38] use discharge notes for QA tasks, and specialized datasets like RxWhyQA [5] and drug-reasoning QA [23] focus on specific question types like medication reasoning. Unlike these benchmarks, DischargeQA introduces an evaluation framework centered around the discharge process, simulating the clinical workflow from diagnosis inference to medication prescription and discharge instruction generation. Additionally, we leverage EHR to generate non-trivial, contextually relevant candidate options, providing a more challenging and realistic setup.

3 ExpRAG Framework

ExpRAG provides a comprehensive framework for retrieving relevant knowledge from the cohort, as shown in Figure 2. In this

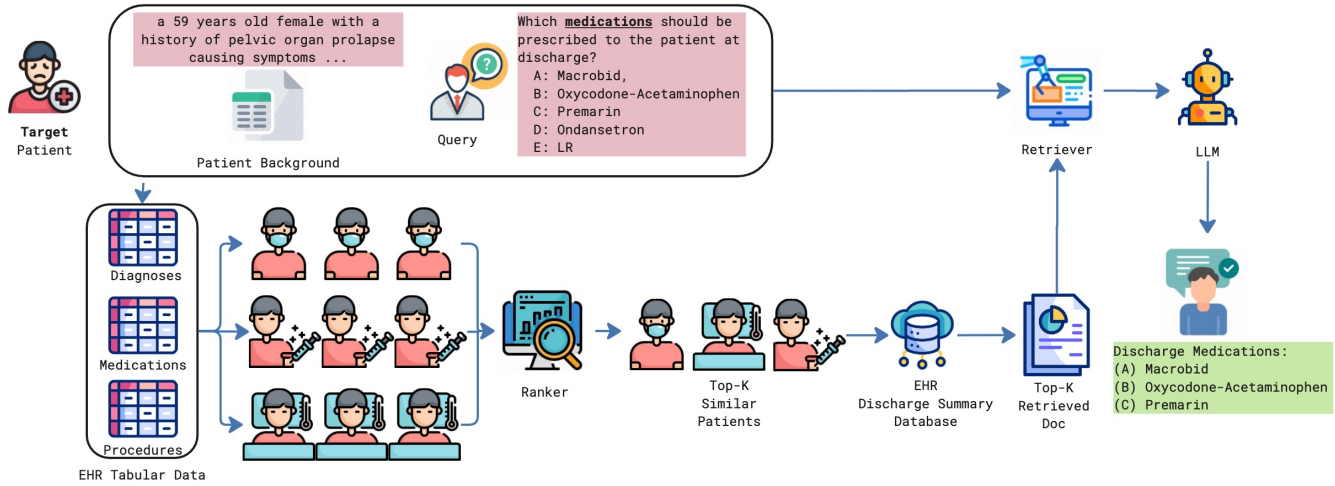

Figure 2: The overview of ExpRAG: Given a medical query and the patient's background, ExpRAG first indexes similar patients based on diagnosis, medication, and procedure similarity from the EHR. A text retriever is then applied to the discharge reports of the top-ranked similar patients to extract clinically relevant content, which is subsequently fed into the LLM to generate the answer.

section, we first formulate the problem that ExpRAG aims to tackle and then elaborate the two-step retrieval framework.

3.1 Task Formulation

A cohort contains a set of discharge report $\mathcal{D} = \{D_i\}_N$ where $D_i = \{d_j\}_M$ denotes the i-th report and d_j is j-th paragraph in D_i. Each report is a medical document that offers an overview of a patient's hospitalization. The goal of ExpRAG is to extract relevant content from $\mathcal{D}$ that helps LLM to effectively answer a given medical query q related to a specific patient p:

$$d_* = f_{\text{ExpRAG}}(p, q, \mathcal{D}) \tag{1}$$

The queries studied in this work focus on providing professional medical guidance for patients, including diagnosis, medication, and discharge instructions, thereby simulating realistic and practical clinical scenarios, as discussed in Section 4.

Different from the conventional RAG focusing on extracting factual concepts from open-ended databases, ExpRAG aims to utilize contextually-relevant clinical practice, inspired by how doctors collect and apply experience from past clinical cases. These two approaches rely on different reasoning procedures and knowledge sources, making them complementary to each other.

3.2 Coarse-to-Fine Retrieval Framework.

To retrieve information from the cohort, one naive solution is to concatenate all the reports into one document and apply a text retriever to extract relevant content, similar to the conventional RAG pipeline. However, a standard EHR cohort typically contains millions of hospital visits, making it impractical to exhaustively search over all reports.

To efficiently perform experience retrieval, ExpRAG applies a two-step framework which conduct the retrieval from a coarse to fine level:

Report Ranking. Before addressing a specific medical query, an intuitive assumption is that only patients with similar clinical histories, e.g., similar diseases or medications, can potentially provide meaningful guidance. In light of this, ExpRAG first employs a report ranker to efficiently discard unrelated cases and narrow down the candidate pool using the patient information:

$$\mathcal{D}' = f_{\text{Ranker}}(p, \mathcal{D}) \tag{2}$$

where $\mathcal{D}' = \{D_i\}_{N' \ll N}$ is a small subset of the selected discharge summaries. The ranker module will be scalable and enable effective utilization of patient context. Specifically, we introduce EHR as a knowledge base to facilitate the patient-level similarity measurement, as presented in Section 3.3.

Experience Retrieval. Based on the selected candidate pool, a sophisticated text retriever is capable of providing more accurate and dedicated clinical experience searching:

$$d_* = f_{\text{Retriever}}(q, \mathcal{D}') \tag{3}$$

Built on top of clinically relevant reports identified by the dedicated ranking approach, the retriever focuses on extracting content related to the medical query. We here apply existing text retrievers, such as auto-merging or BM25, during this phase.

3.3 EHR-Based Report Ranker f_{Ranker}

Electronic Health Record (EHR), as a structured data organization, typically consists of multiple tabular data, each recording specific medical information about patients. In this study, we focus on measuring the similarity between patients using the following three medical entities:

- Diagnosis: Identified disease assigned to a patient, represented by ICD-10 code.

- Medication: Prescribed drugs administered to a patient, recorded using NDC code.
- Procedure: Medical intervention, operation, or clinical process performed on a patient, represented by ICD-10 code.

Quantify the similarity between patients based on these three dimensions offers a comprehensive criterion for identifying clinically relevant reports.

For a patient p, these three medical entities are represented as sets E_p^{Diag}, E_p^{Med}, and E_p^{Proc}, respectively. Given two patients p and p', we first compute the set similarity between them using each medical information:

$$\tau_{\text{Diag}} = f_{\text{similarity}}(E_p^{\text{Diag}}, E_{p'}^{\text{Diag}}), \tag{4}$$

$$\tau_{\text{Med}} = f_{\text{similarity}}(E_p^{\text{Med}}, E_{p'}^{\text{Med}}), \tag{5}$$

$$\tau_{\text{Proc}} = f_{\text{similarity}}(E_p^{\text{Proc}}, E_{p'}^{\text{Proc}}) \tag{6}$$

where $f_{\text{similarity}}(\cdot, \cdot)$ is a set similarity metric, with the Jaccard Index applied in this study. Finally, these similarity metrics are aggregated using a weighted sum:

$$\tau = \lambda_1 \tau_{\text{Diag}} + \lambda_2 \tau_{\text{Med}} + \lambda_3 \tau_{\text{Proc}} \tag{7}$$

where $\lambda_{1/2/3}$ is the hyperparameter balancing the importance of each metric. We perform pairwise similarity comparisons between the query patient and other patients within the EHR, returning the discharge summaries of the top-k most similar patients as results.

Efficiency Analysis. The overall computation is practically efficient since the computation of Jaccard Index can be significantly accelerated with some libraries, such as Faiss [4] and NumPy [8]. Besides, indexing medical entities from tabular data enables fast lookups, further reducing computational overhead.

4 DischargeQA Dataset

To evaluate LLMs' capability in utilizing the retrieved experience, we construct a medical question-answering dataset specifically designed for discharge-related queries based on MIMIC-IV [16]. Each question in the dataset pertains to critical discharge information, including the patient's diagnosis, prescribed medications, and post-discharge instructions.

4.1 Dataset Introduction

Overview. As shown in Table 1, DischargeQA consists of a total of 1,280 QA pairs, each associated with a patient ID and corresponding clinical background. The questions in DischargeQA can be categorized into three main types:

- Diagnosis Inference: Questions related to identifying the patient's medical diagnosis.
- Medication Inference: Questions regarding the medications prescribed, including dosage, frequency, and purpose.
- Instruction Inference: Questions focused on discharge instructions, such as follow-up care, activity restrictions, and self-care guidelines.

These three categories collectively cover the key aspects of discharge-related patient care, requiring LLMs to perform non-trivial reasoning based on the given clinical background. Moreover, all these problems can potentially benefit from the retrieved experience, mimicking the way clinicians apply past clinical knowledge to make medical decisions during discharge.

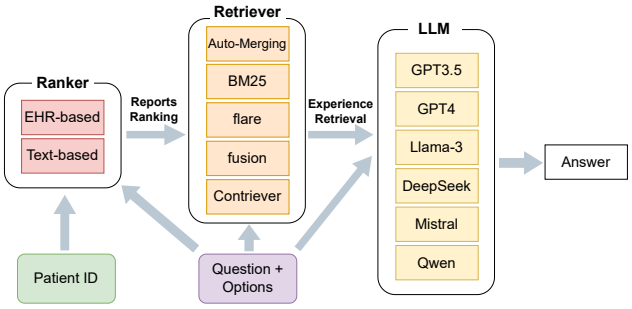

Figure 3: Inference pipeline of DischargeQA.

Evaluation Settings. Each problem in DischargeQA includes a problem description, the patient's clinical background, and multiple options for the LLM to choose from. While instruction inference uses a single-select setup, diagnosis and medication inference adopt a multi-select setup, requiring the model to select multiple options to answer the questions. Each option corresponds to a specific diagnosis or treatment. This multi-select format presents a more challenging and realistic setting for LLMs, as clinicians often need to identify and address multiple coexisting conditions.

The overall inference pipeline is presented in Figure 3, where we implement several components, including Ranker, Retrieval, and LLM agent, to support discharge-related QA using ExpRAG.

Comparison.

Compared with existing QA benchmarks that focus primarily on general clinical QA or information retrieval, DischargeQA centers on the discharge procedure, simulating a doctor's medical reasoning process: inferring the diagnosis from the clinical profile, prescribing appropriate treatments, and summarizing the condition—offering a more realistic scenario. Additionally, we utilize EHR to generate contextually relevant options, requiring LLMs to perform non-trivial reasoning to solve the tasks. More details are provided in Table 1.

4.2 Dataset Construction

Patients Filtering. We first filter out low-quality patient records in MIMIC-IV for various reasons. Starting with all 430,000 patients, we remove encounters without available discharge summaries, leaving 320,000 patients. Next, we filter out patients with fewer than 3 or more than 40 entries in any of the diagnosis, medication, or procedure records, resulting in a final dataset of 28,000 patients for generating QA pairs. For instruction inference, we further exclude patients with excessively short discharge summaries using GPT-4o, as explained in A.1.

Background Generation. To enable LLMs to make realistic medical decision, it is necessary to offer a clinical background of the patient along with the question. To **avoid label leakage** during

	Task	Response Type	#Query	Example Question	Practical Significance	Background Source	Option Source
DischargeQA	Diagnosis Inference	Multi-select	436	"Which diagnoses should be documented in the patient's discharge summary?"	Reflects a doctor's process of identifying all relevant diagnoses based on clinical profile.	Clinical profile	Discharge Report & EHR
	Medications Inference	Multi-select	444	"Which medications should be prescribed to the patient at discharge?"	Simulates the doctor's task of ensuring correct medications are prescribed based on hospital treatment.	Clinical profile & In-hospital progress	Discharge Report & EHR
	Instructions Inference	Single-select	400	"What is the best instruction for this patient?"	Mimics the final step of doctors' advising patients with appropriate post-discharge care instructions.	Clinical profile & In-hospital progress	Discharge Report & AI
EHRNoteQA	Clinical Inference	Single-select & Open-Ended	962	"What was the treatment provided for the patient's left breast cellulitis?"	Extract and answer based on content from full discharge notes	Full clinical notes	Discharge Report & AI
CliniQG4QA	Retrieval	Text Span	1,287	"Why has the patient been prescribed hctz?"	Retrieve the related content as answer from report	Full clinical notes	/

Table 1: Comparison of DischargeQA and the previous EHR-related QA benchmarks.

Model	Context	Instruction	Diagnosis		Medication	
		Acc(%)	Acc(%)	F1	Acc(%)	F1
Mistral-7b	Direct-Ask	67.0	16.03	0.488	1.13	0.362
	bge-small-en	**70.0**	14.67	0.490	0.90	0.356
	ExpRAG $_{EHR}$	69.0	13.79	**0.505**	1.13	**0.371**
Deepseek-R1-8B	Direct-Ask	70.0	10.78	0.363	0.69	0.217
	bge-small-en	72.0	12.84	**0.381**	1.58	0.230
	ExpRAG $_{EHR}$	**75.3**	11.01	0.379	0.9	**0.241**
Qwen3-30B-A3B	Direct-Ask	90.8	15.60	0.415	1.13	0.280
	bge-small-en	93.8	18.12	0.502	2.25	**0.366**
	ExpRAG $_{EHR}$	**95.3**	17.43	**0.528**	1.35	0.355
GPT-3.5	Direct-Ask	73.0	18.50	0.498	1.15	0.234
	bge-small-en	78.8	15.60	0.405	0.68	0.317
	ExpRAG $_{EHR}$	**79.5**	18.81	**0.504**	1.80	**0.371**
GPT-4o	Direct-Ask	90.0	9.86	0.510	3.65	0.486
	bge-small-en	90.3	8.26	0.493	4.95	0.601
	ExpRAG $_{EHR}$	**91.3**	**21.33**	**0.530**	**9.68**	**0.638**

Table 2: Performance Comparison Across Multiple LLMs using DischargeQA

the context generation, we propose leveraging the structured format of the discharge summary, which consists of the following components:

- Clinical profile: Essential patient demography, the presenting condition, and initial clinical assessments.
- In-hospital progress: The interventions, therapies, and the patient's clinical progress during hospitalization.
- Discharge plan summary: The details of discharge diagnosis and medication, and instructions for post-hospital care.

An illustrative example can be found in Appendix A.3. These three components represent the core of discharge decision-making procedure, aligning with the three main problem types in DischargeQA. Accordingly, we present the summarized sections preceding the questions as context. For example, the clinical profile serves as the background for diagnosis-related questions, while the in-hospital progress is additionally included for medication-related questions. Notably, basic patient demographic information is always included as part of the contextual background.

Option Generation. For diagnosis and medication inference, we utilize EHR to generate non-trivial candidate options by extracting all associated diagnoses and medications of a patient and feeding them into GPT-4o to select contextually relevant candidates, ensuring a challenging selection process. For instruction inference, GPT-4o first summarizes the key points of the ground-truth answers

and then applies permutations to generate plausible yet incorrect candidate options. Notably, EHR tabular data usually has **very limited overlap** with our task for discharge, such as in-hospital medications, due to professionalism and use cases, have much limitation than discharge prescriptions. For all tasks, the correct options are directly extracted from the discharge reports.

5 Experiments

In this section, we evaluate ExpRAG on DischargeQA with five LLMs, comparing it with the text-based report rankers. We conduct experiments to systematically evaluate various prompt optimization strategies within ExpRAG. Specifically, we explore how altering prompt content through different retrieval parameters—such as the number of retrieved patients (§ 5.3), different retrieval methods (§ 5.3), and weighting coefficients of clinical entity similarity (§ 5.3)—affect the downstream clinical QA performance of LLMs.

5.1 Comparison of LLMs

We evaluated the performance of 4 state-of-the-art LLMs of varying scales, ranging from close-source to 8 billion parameters open-source model, on three clinical tasks: discharge instructions, diagnosis, and medications. These models included GPT-3.5 [24], GPT-4o [25]), Mistral-7B [11], and two thinking models—Deepseek-R1-8B [7] and Qwen3-30B-A3B [33].

Hyperparameters. We also present the results of LLMs with ExpRAG. The default balancing coefficients $\lambda_{1/2/3}$ are set to 1/3 each and auto-merging [20] is used as the retriever. The number of similar patients is set as 15.

Metrics. We report accuracy across all the tasks, calculated as the percentage of correctly answered questions. Note that for multi-select problems, a question is considered answered correctly only when all correct options are selected. We additionally report F1 scores for the multi-select problems to provide a more comprehensive analysis of LLM performance on these challenging tasks.

Results. Table 2 summarizes the performance of all LLMs with ExpRAG on the three clinical tasks. GPT-4o consistently achieves the best performance across tasks, with an absolute improvement of 13% over GPT-3.5 on instruction inference problems. Additionally, the multi-select tasks (diagnosis and medication) prove significantly more challenging, as most LLMs achieve accuracy below 20%, indicating the limited medical reasoning capabilities of current large language models, the value of challenge of DischargeQA.

5.2 Comparison of Report Ranker

To verify the effectiveness of ExpRAG and the utilization of EHR, we implement baselines that perform report ranking solely based on text, i.e., a text-based ranker, as a key part in other traditional RAG methods. Using embedding model `bge-small-en-v1.5` [34], a popular model due to its light weight and high correlation scores in Table 4, we embed queries (question, options, background) to be computed similarity with each discharge report embedding. The top-k similar reports are then retrieved, followed by a text retriever to extract the relevant information.

Rankers	Instruction	Diagnosis		Medication	
	Acc	Acc	F1	Acc	F1
bge-small-en	78.8	15.60	0.405	0.68	0.317
all-MiniLM-L6	79.3	17.43	0.476	1.13	0.316
paraphrase-L3	77.5	18.81	0.492	1.58	0.330
ExpRAG $_{EHR}$	**79.5**	**18.81**	**0.504**	**1.80**	**0.371**

Table 3: Performance Comparison with Embedding Models as Reranker vs EHR-based Reranker.

Rankers	Pearson	Spearman
bge-small-en-v1.5	0.639	0.623
all-MiniLM-L6	0.640	0.618
paraphrase-MiniLM-L3	0.478	0.481
ExpRAG $_{EHR}$	**0.669**	**0.648**

Table 4: Retrieval Performance Comparison of Rerankers using Pearson and Spearman Correlation.

The results are presented in Table 2. We observe that ExpRAG outperforms the text-based ranker in most cases, achieving an average relative improvement of 5.2%. Notably, the EHR-based ranker leverages structured EHR data for ranking, eliminating the need for an embedding process and thereby enabling a more efficient pipeline.

As shown in Table 3, we compare our EHR-based ranker with two additional sentence-embedding models: `all-MiniLM-L6-v2` and `paraphrase-MiniLM-L3-v2` [29], using GPT3.5 as backbone LLM. As a result, ExpRAG $_{EHR}$ outperforms all text-based variants in terms of all the metrics.

Retrieval Correlation Comparison. We conducted an experiment comparing patients similarity scores generated by the EHR-based method with those reliably annotated by LLMs. A higher correlation between the two indicates stronger retrieval performance by the EHR-based approach. Specifically, we randomly sample 100 target patients from DischargeQA, as detailed in Appendix A.8. Each target–candidate pair is scored by GPT4o-mini on the three modalities (diagnoses, procedures, prescriptions) with a single "overall" similarity. For each ranker (i.e., three text-embedding models and our EHR-based ExpRAG), we compute Pearson and Spearman correlation between its scores and annotations, and average over all 100 targets (Table 4). Over the strongest text-embedding baselines,the results by ExpRAG confirm that explicit EHR similarity provides more faithful retrieval signals than generic sentence embeddings.

Model	Instruction	Diagnosis		Medication	
	Acc	Acc	F1	Acc	F1
Uniform	79.5	18.81	0.504	1.8	0.371
Task-focused	77.5	10.91	0.377	0.91	0.322
Complementary	76.8	18.18	0.446	2.73	0.305

Table 5: Performance Comparison for Coefficients Distributions DischargeQA

# Similar Patients	Instruction	Diagnosis		Medication	
	Acc	Acc	F1	Acc	F1
$k = 5$	80.00	19.04	0.511	1.80	0.366
$k = 10$	78.75	18.58	0.511	1.35	0.371
$k = 15$	79.50	18.81	0.504	1.80	0.371
$k = 20$	80.75	19.50	0.524	1.80	0.377
$k = 25$	82.25	19.27	0.515	1.35	0.352

Table 6: GPT-3.5 performance with different k.

5.3 Systematic Study of Prompt Optimization Methods

In this section, we systematically study various prompting methods and optimization strategies employed within the ExpRAG framework, without introducing additional experiments beyond those reported. Our systematic study covers the following distinct prompting approaches:

(1) **QA-Generation Prompts:** Prompts for generating diagnosis and instruction QA-pairs, evaluated by examining the generated question-answer sets' quality and clinical validity (see Appendix A.6 for exact prompts).
(2) **Instruction Filtering Prompt:** An evaluation prompt for assessing instruction completeness and clinical safety by ensuring instructions clearly state required actions and warning signs (see Appendix A.7, Figure 10).

Additionally, to evaluate prompt optimization effectiveness, we conduct experiments by adjusting the following retrieval-based parameters, directly impacting the prompts' informational content:

- **Retrieval Methods:** BM25, Contriever, Auto-merging, and others, comparing different strategies for selecting relevant context to form optimized prompts.
- **Number of Retrieved Patients (k):** Evaluating how changing the quantity of retrieved contextual information optimizes prompt effectiveness.
- **Balancing Coefficients ($\lambda_1, \lambda_2, \lambda_3$):** Investigating how different clinical entity similarities (diagnosis, medications, procedures) impact the prompts' clinical relevance and reasoning performance.

This systematic analysis explicitly frames the EXPRAG retrieval parameters and prompt structures as forms of prompt optimization, allowing us to understand which factors most significantly impact clinical QA performance.

GPT-3.5 is applied as the backbone by default.

Balancing Coefficients. As shown in Equ. 4, we introduce three coefficients to balance the similarity computed based on diagnosis, medication, and procedures. We apply an equal distribution by default and explore the effect of using different weighting strategies here:

- Task-focused weighting: Assign a weight of 1 to the task-relevant similarity measure and 0 to the others. For example, $\lambda_1 = 1, \lambda_2 = 0, \lambda_3 = 0$ for diagnosis inference.
- Complementary weighting: Assign a weight of 1 to the two less relevant similarity measures while setting the task-relevant measure to 0. For example, $\lambda_1 = 0, \lambda_2 = 1, \lambda_3 = 1$ for diagnosis inference.

The results are shown in Table 5. We can find that complementary weighting can achieve the best performance in most cases, demonstrating that information from multiple clinical dimensions can provide a more comprehensive context.

Top-k Patients. We vary the number of retrieved similar patients k on QA performance. As shown in Table 6, different tasks have varies trends. For example, on Instruction task, accuracy starts to increases with larger number. While for Diagnosis and Medication tasks, beyond $k = 20$, performance fluctuates, suggesting that while more retrieved candidates provide useful context, excessive retrieval may introduce irrelevant or conflicting information, leading to slight declines in accuracy.

5.4 Case Study

We perform a focused case study on a Discharge Diagnosis task from DischargeQA. Specifically, we analyze a patient (ID: 20453584) presenting with bilateral ulnar paresthesias and neck pain. Similar patients are identified by matching ICD/NDC codes from structured EHR data (Figure 4). Reviewing discharge summaries of these similar patients revealed shared key diagnostic features—including cervical disc herniation, spinal stenosis, and upper extremity neurological symptoms—which substantially clarified the target patient's clinical picture. For instance, similar patients exhibiting C6-C7 disc herniations and spinal stenosis provided critical evidence, improving the interpretation of the target patient's symptoms and supporting a more accurate final diagnosis. We elaborate the explanation in Appendix A.9.

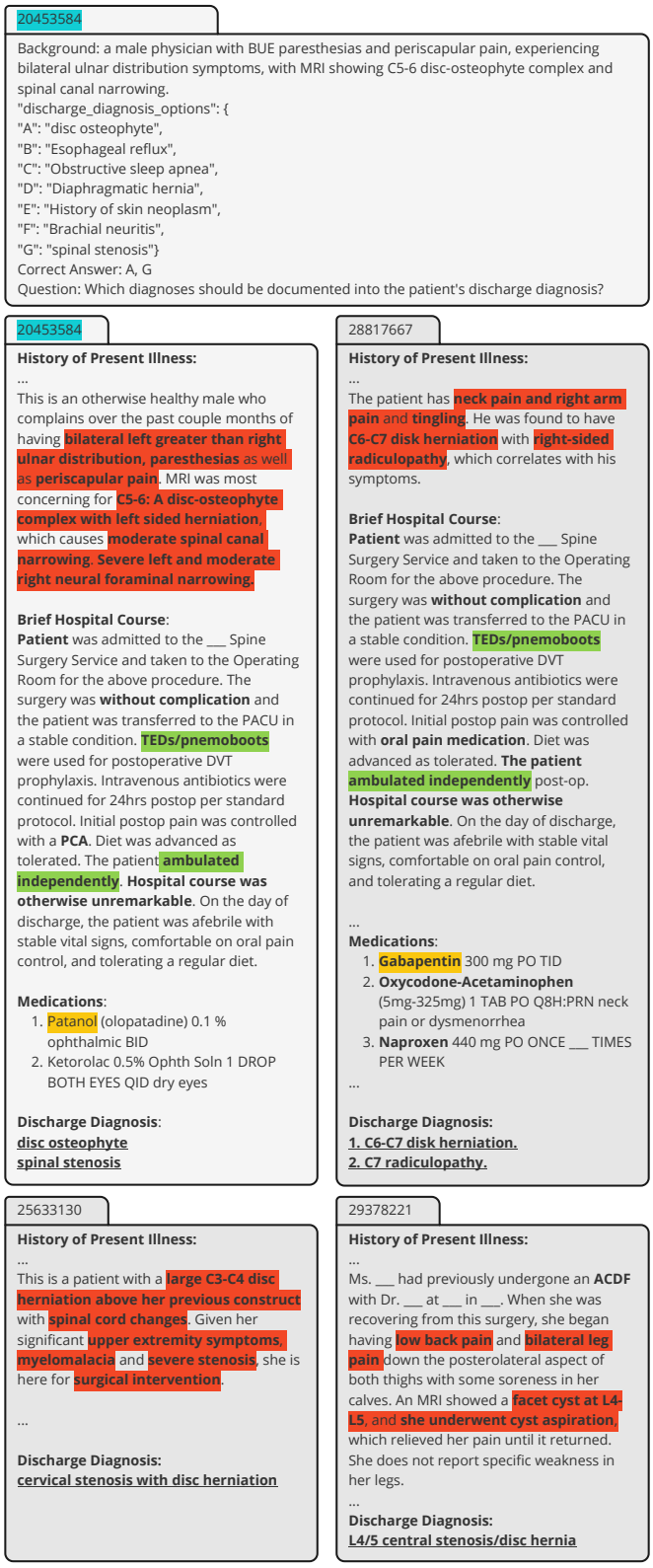

Figure 4: Comparison of Similar Patients.

6 Conclusion

Inspired by the importance of experience in clinical decision-making, we propose a novel coarse-to-fine retrieval framework, ExpRAG, to utilize knowledge from similar patient records. Specifically, we introduce EHR as a knowledge basis and employ a reliable similarity measurement algorithm to narrow down the candidate pool to have relevant and useful content. Evaluated on our curated DischargeQA, ExpRAG consistently improves the performance of various LLMs, highlighting the potential of leveraging past experience to enhance model performance on medical QA.

Limitations While EHR provides abundant medical information, such as lab test results, our proposed ExpRAG currently utilizes only diagnosis, medication, and procedures as an initial exploration, which provides valuable insights and promising directions for future research. Additionally, DischargeQA currently consists solely of multi-option questions, which can be enhanced to be open-ended to comprehensively evaluate the generative capabilities of LLMs, When more economical and accurate LLMs are developed.

References

[1] Jayetri Bardhan, Anthony Colas, Kirk Roberts, and Daisy Zhe Wang. 2022. DrugEHRQA: A Question Answering Dataset on Structured and Unstructured Electronic Health Records For Medicine Related Queries. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 1083–1097. https://aclanthology.org/2022.lrec-1.117

[2] Zeming Chen, Alejandro Hernández-Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. MEDITRON-70B: Scaling Medical Pretraining for Large Language Models. arXiv:2311.16079 [cs.CL]

[3] Jacob Devlin, Ming-Wei Chang, et al. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[4] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. *arXiv:2401.08281* (2024). arXiv:2401.08281 [cs.LG]

[5] Jungwei Fan. 2019. Annotating and characterizing clinical sentences with explicit why-QA cues. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. 101–106.

[6] Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *AAAI*, Vol. 32.

[7] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

[8] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825 (Sept. 2020), 357–362. doi:10.1038/s41586-020-2649-2

[9] Tinglin Huang, Syed Asad Rizvi, Rohan Krishna Thakur, Vimig Socrates, Meili Gupta, David van Dijk, R Andrew Taylor, and Rex Ying. 2024. HEART: Learning better representation of EHR data with a heterogeneous relation-aware transformer. *Journal of Biomedical Informatics* 159 (2024), 104741.

[10] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118* (2021).

[11] Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *ArXiv* abs/2310.06825 (2023). https://api.semanticscholar.org/CorpusID:263830494

[12] Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983* (2023).

[13] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences* 11, 14 (2021), 6421.

[14] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2567–2577.

[15] Alistair Johnson, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. 2023. MIMIC-IV-Note: Deidentified free-text clinical notes (version 2.2). https://doi.org/10.13026/1n74-ne17

[16] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. 2023. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific data* 10, 1 (2023), 1.

[17] Alistair E. W. Johnson et al. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data* 3 (2016), 160035. https://doi.org/10.1038/sdata.2016.35

[18] Sunjun Kweon, Jiyoun Kim, Heeyoung Kwak, Dongchul Cha, Hangyul Yoon, Kwanghyun Kim, Jeewon Yang, Seunghyun Won, and Edward Choi. 2024. EHRNoteQA: An LLM Benchmark for Real-World Clinical Practice Using Discharge Summaries. In *Proceedings of NeurIPS 2024 (Datasets and Benchmarks)*. https://doi.org/10.48550/arXiv.2402.16040

[19] Gyubok Lee, Hyeonji Hwang, Seongsu Bae, Yeonsu Kwon, Woncheol Shin, Seongjun Yang, Minjoon Seo, Jong-Yeup Kim, and Edward Choi. 2022. EHRSQL: A Practical Text-to-SQL Benchmark for Electronic Health Records. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 15589–15601. https://proceedings.neurips.cc/paper_files/paper/2022/file/643e347250cf9289e5a2a6c1ed5ee42e-Paper-Datasets_and_Benchmarks.pdf

[20] Jerry Liu. 2022. LlamaIndex. https://github.com/jerryjliu/llama_index

[21] Lei Liu, Xiaoyan Yang, Junchi Lei, Xiaoyang Liu, Yue Shen, Zhiqiang Zhang, Peng Wei, Jinjie Gu, Zhixuan Chu, Zhan Qin, et al. 2024. A Survey on Medical Large Language Models: Technology, Application, Trustworthiness, and Future Directions. *arXiv preprint arXiv:2406.03712* (2024).

[22] Yuxing Lu, Xukai Zhao, and Jinzhuo Wang. 2024. ClinicalRAG: Enhancing Clinical Decision Support through Heterogeneous Knowledge Retrieval. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, Sha Li, Manling Li, Michael JQ Zhang, Eunsol Choi, Mor Geva, Peter Hase, and Heng Ji (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 64–68. doi:10.18653/v1/2024.knowllm-1.6

[23] Sungrim Moon, Huan He, Heling Jia, Hongfang Liu, Jungwei Wilfred Fan, et al. 2023. Extractive Clinical Question-Answering With Multianswer and Multifocus Questions: Data Set Development and Evaluation Study. *JMIR AI* 2, 1 (2023), e41818.

[24] OpenAI. 2022. CHATGPT: Optimizing language models for dialogue. https://openai.com/blog/chatgpt/

[25] OpenAI. 2024. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/

[26] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning*. PMLR, 248–260.

[27] Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrqa: A large corpus for question answering on electronic medical records. *arXiv preprint arXiv:1809.00732* (2018).

[28] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language Models as Knowledge Bases? arXiv:1909.01066 [cs.CL]

[29] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084

[30] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[31] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine* (2025), 1–8.

[32] Sarvesh Soni, Meghana Gudala, Atieh Pajouhi, and Kirk Roberts. 2022. RadQA: A Question Answering Dataset to Improve Comprehension of Radiology Reports. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 6250–6259. https://aclanthology.org/2022.lrec-1.672

[33] Qwen Team. 2025. Qwen3. https://qwenlm.github.io/blog/qwen3/

[34] Shitao Xiao, Zheng Liu, Peitian Zhang, et al. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. *arxiv:2309.07597* (2023).

[35] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking Retrieval-Augmented Generation for Medicine. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 6233–6251. doi:10.18653/v1/2024.findings-acl.372

[36] Lee Xiong, Chenyan Xiong, Ye Li, et al. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *ICLR*.

[37] Rui Yang, Yilin Ning, Emilia Keppo, Mingxuan Liu, Chuan Hong, Danielle S Bitterman, Jasmine Chiat Ling Ong, Daniel Shu Wei Ting, and Nan Liu. 2025. Retrieval-augmented generation for generative artificial intelligence in health care. *npj Health Systems* 2, 1 (2025), 2.

[38] Xiang Yue, Xinliang Frederick Zhang, Ziyu Yao, Simon Lin, and Huan Sun. 2021. Cliniqg4qa: Generating diverse questions for domain adaptation of clinical question answering. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 580–587.

[39] Yanxin Zheng, Wensheng Gan, Zefeng Chen, Zhenlian Qi, Qian Liang, and Philip S Yu. 2024. Large language models for medicine: a survey. *International Journal of Machine Learning and Cybernetics* (2024), 1–26.

A Appendix

A.1 DischargeQA

To evaluate LLMs in real-world clinical scenarios, we constructed a dataset combining structured tables from MIMIC-IV and unstructured discharge notes from MIMIC-IV-note, totaling over 140 000 patient records. The whole process is shown in 5

Patient selection. As in 4.2, a valid pool of candidates for target or similar patients is the key point for retrieving information from similar patients.

Starting from the 430 k patients in MIMIC-IV, we retain (i) encounters with a discharge summary, (ii) 3–40 rows in each of `diagnoses`, `prescriptions`, and `procedures`, and (iii) discharge notes that contain ≥ 4 instruction bullet points identified by GPT-4o. This yields 28 000 admissions, each can give enough structured items and narrative content to form challenging QA pairs, so that there are enough information to compare to other patients, and to have candidate options as wrong answer if there is no overlap in discharge reports.

Note segmentation and exposure. Each discharge summary is heuristically split into seven sections *(Patient Demography, Presenting Condition, Clinical Assessment, Treatment Plan, In-Hospital Progress, Discharge Summary, Post-Discharge Instructions)* and mapped to three temporal phases: **pre-diagnosis**, **in-hospital**, and **post-discharge**, matching the 3 sections **Clinical profile**, **In-hospital**, and **Discharge Plan** as Figure 7. For every task we reveal only the phases that would have been available to the clinician at decision time, preventing label leakage.

Problem Design with EHR.

- **Golden answers** are clinician-authored discharge–note items.
- **Distractors** are drawn from the same patient's structured tables (drugs for medication, ICD-coded diagnoses for diagnosis).
- Overlaps between structured candidates and golden answers are merged via GPT-4o.
- Unlike single-choice formats, both diagnosis and medication tasks use *multi-select*, requiring selection of *all* correct items, thus raising task difficulty.

Construction Prompt Design. Prompts to generate options are carefully crafted to combine EHR tabular data and golden answer from discharge note. As an example, the prompt as Figure 8 defines the role of the model as a clinician and provides three key components: the list of discharge diagnoses, a database of historical diagnoses, and summarized background information extracted from the discharge summary. The task requires the model to identify which diagnoses should be included in the discharge summary by reasoning through the given data. Correct options are derived from the discharge diagnosis list, while incorrect but plausible options are generated from the diagnoses database. To ensure realism, GPT-4o is used to handle overlap, summarize long diagnoses, and align outputs with clinical expectations, providing a rigorous framework for evaluation.

A.2 Compare Two Rankers using similar patients

Shown as Figure 6.

A.2.1 Conventional RAG. A traditional RAG pipeline for clinical question answering consists of:

(1) Encoding a query (e.g., a discharge planning question) into an embedding vector.
(2) Chunking a large corpus into smaller text passages and embedding each chunk.
(3) Finding top-k relevant text chunks by comparing similarity scores between query embeddings and chunk embeddings.
(4) Retrieving the most relevant text passages.
(5) Feeding the retrieved text to the LLM for answer generation.

However, applying this approach to EHR discharge summaries presents challenges: **Inefficiency**: The MIMIC-IV discharge summary corpus is over 4GB in size, making fine-grained chunk retrieval computationally expensive. **Loss of Context**: Traditional chunking disrupts the continuity of patient history, making it difficult for LLMs to infer longitudinal medical decisions. **Scattered Information:** Important information may be spread across multiple notes, making individual paragraph retrieval suboptimal.

A.2.2 Text-Based Ranker. Instead of retrieving individual paragraphs or sentences, we treat each patient's discharge summary as a retrievable document and adapt the RAG pipeline accordingly:

Query Encoding Convert the question, options and target patient's background into an embedding vector.

Patient-Level Indexing Store each patient's discharge summary as a separate retrievable document and compute its embedding.

Find Similar Patients Using Embeddings Rank top-k similar patients from all patients based on similarity between the query embedding and patient embeddings.

Retrieve Top-K Patient Summaries Select the top-k most similar patient discharge summaries as a document source, like "experience".

Run RAG on Retrieved Summaries Use the retrieved summaries as references for LLM-based answer generation. Traditional RAG is running by embedding the query and the document source, retrieving chunks as reference, and answering query to generate the correct answer.

By preserving full discharge summaries, Text-based retrieval maintains patient-level coherence while improving retrieval efficiency.

A.2.3 EHR-Based Ranker. To further improve retrieval relevance, we introduce an EHR-based Ranker that utilizes structured patient data for retrieving better discharge summaries as the "experience" document source:

Identify the Target Patient's Condition Extract structured EHR tabular data (ICD codes for diagnoses, prescriptions, and procedures) from the target patient.

Find Similar Patients Using Structured EHR Data Compute similarity between the target patient's structured data and all other

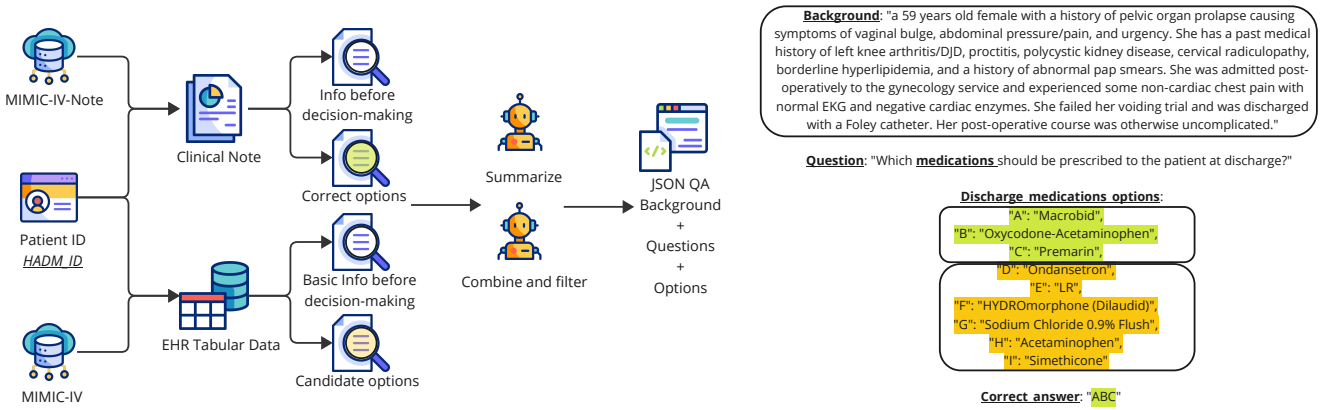

Figure 5: DischargeQA generation workflow.

patients based on shared ICD codes in MIMIC-IV, after scanning all recorded patient data. Jaccard index is used to calculate the similarity.

Retrieve Top-K Similar Patients Select the top-k most similar patients based on their structured medical records.

Extract Their Discharge Summaries Retrieve the corresponding discharge summaries for the top-k similar patients.

Run RAG on Retrieved Summaries Feed the retrieved summaries into an LLM for answer generation, same as the last step in Text-Based Ranker.

By incorporating structured EHR data, this method ensures that retrieval is clinically relevant, going beyond semantic text similarity. This process resembles an experience-based search: identifying similar past experiences and adapting them to solve the problem at hand.

A.3 Dataset Details

MIMIC-IV[2] This database [15–17] contains information on over 40,000 patients admitted to the critical care units at Beth Israel Deaconess Medical Center from 2001 to 2012 and has been widely used in prior research. The database is publicly available for research purposes, with strict de-identification protocols to protect patient privacy, making it a valuable resource for developing and evaluating machine learning models in healthcare. The data is hierarchically organized, with each patient record comprising multiple encounters, each containing various entities such as demographics, medications, diagnoses, procedures, and lab results. Additionally, the database includes unstructured data, such as discharge reports and X-ray images, with each admission marked by a date and timestamp.

Structure of Discharge Report As shown in Figure 7, a discharge summary follows a structured format that provides a comprehensive overview of a patient's hospitalization and care journey. It is typically divided into three main parts: (1) The Clinical Profile, which includes essential patient information, presenting conditions, and the initial clinical assessment upon admission; (2) The In-Hospital Progress, which documents the treatment plan, administered therapies, and the patient's progress throughout the hospital stay; and (3) The Discharge Plan Summary, which summarizes the patient's discharge status, prescribed medications, and detailed post-discharge instructions for ongoing care.

[2]https://mimic.mit.edu/docs/iv/

Retriever	Instruction Acc
Auto Merging	79.5
Sentence Window	74.5
BM25	68.0
BM25+	69.0
Flare	74.5
Contriever	69.0

Table 7: Performance Comparison of Retrievers.

A.4 Retriever

A.4.1 Retriever Experiments. Our experiments include: Auto-merging [20], sentence-window [20], BM25 [30], BM25+ (the combination of BM25 and word embeddings), Contriever [10], and flare [12]. More details regarding the methods can be found in Appendix A.4. As shown in Table 7, auto-merging, sentence-window, and flare achieve the best performance, highlighting the effectiveness of context-aware retrieval. However, Contriever, which utilizes unsupervised dense representations, also underperforms in our cases, suggesting the need for medical domain-specific fine-tuning.

Auto-merging Auto-merging retrieval in RAG by LlamaIndex [20] hierarchically structures documents into parent and child nodes, allowing for the retrieval of larger, more coherent context by merging child nodes into parent nodes when multiple related chunks are relevant to a query

Sentence-window Sentence-window retrieval by LlamaIndex [20] parses documents into individual sentences with surrounding

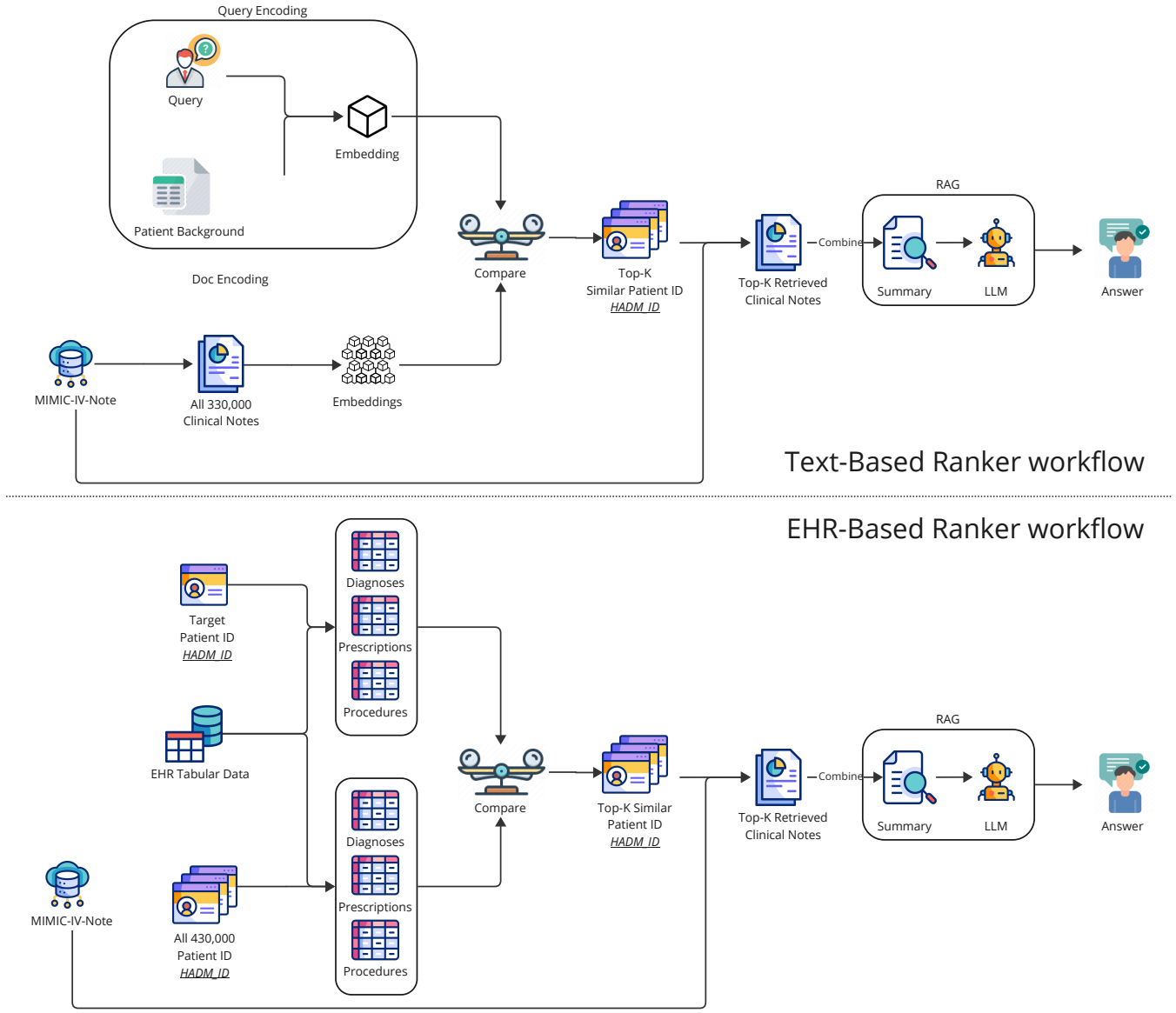

Figure 6: Text-Based Ranker vs EHR-Based Ranker workflow.

context, enabling fine-grained retrieval while maintaining local coherence by including a window of adjacent sentences

BM25 & BM25+ BM25 (Best Match 25) is a ranking function used in information retrieval that scores documents based on the frequency of query terms within them, taking into account document length and term frequency saturation. BM25+ by LlamaIndex [20] combines retrieval methods BM25 and vector-based retrieval, to leverage the strengths of both approaches. This hybrid technique allows for capturing both keyword relevance and semantic similarity, often using algorithms like Relative Score Fusion to re-rank and merge results from different retrievers, resulting in more accurate and comprehensive search outcomes.

Flare FLARE (Forward-Looking Active REtrieval) enhances RAG by enabling the language model to anticipate future content needs, iteratively predicting upcoming sentences and retrieving relevant information when encountering low-confidence tokens, thus improving response accuracy and contextual relevance

----------Section 1 : Clinical profile----------
[----------Patient Demography----------]
Name: ___ Admission Date: ___ Discharge Date: ___
Date of Birth: ___ Sex: F
Service: Surgery
Allergies: Penicillins, ACE Inhibitors
Attending: ___

[----------Presenting Condition----------]
Chief Complaint: Patient admitted with nausea and abdominal pain.
History of Present Illness: Patient presented with a history of nausea, vomiting, and abdominal pain for several days. Laboratory evaluation was unremarkable. CT scan showed a small bowel obstruction secondary to a Richter's hernia.
Past Medical History: HTN, HLD, Asthma, Tenosynovitis, Obesity.
Social History: ___
Family History: Non-contributory.

[----------Clinical Assessment----------]
Physical Exam:
- Vital signs: Temp 97.8°F, BP 130/65, HR 72, RR 16.
- Abdomen: Distended, possible tender mass below the umbilicus without erythema or guarding.

Pertinent Results:
- Labs: WBC 8.8, Hgb 13.1, Hct 37.7.
- CT Abdomen: Richter's hernia causing small-bowel obstruction, no perforation.

----------Section 2: In-Hospital Progress----------

[----------Treatment Plan----------]
Major Surgical or Invasive Procedure: Exploratory laparotomy, lysis of adhesions, hernia repair with mesh.

[----------In-Hospital Progress----------]
Brief Hospital Course:
Patient underwent surgery for small bowel obstruction secondary to Richter's hernia. Postoperative recovery was uneventful. Diet was gradually advanced, and pain was well-controlled.
Medications on Admission: Fluticasone, HCTZ, Naproxen, Prochlorperazine, Valsartan.

----------Section 3: Discharge Plan----------
[----------Discharge Summary----------]
Discharge Medications:
1. Oxycodone-Acetaminophen for pain, as needed.
2. Docusate Sodium, twice daily.
3. Ciprofloxacin, twice daily for 2 weeks.

Discharge Disposition: Home with services.
Discharge Diagnosis: Small bowel obstruction due to Richter's hernia.
Discharge Condition: Stable.

[----------Post-Discharge Instructions----------]
Discharge Instructions:
- Avoid driving or operating machinery while taking pain medications.
- Resume regular home medications.

Call your doctor or visit the ER if:
- You have chest pain, persistent vomiting, dehydration, or a fever above 101.5°F.
- There is redness, swelling, or drainage from the incision.

Activity:
- No heavy lifting for 6 weeks. Moderate exercise allowed, no abdominal exercises.

Wound Care:
- Shower allowed, no baths or swimming. Steri-strips will fall off on their own.

Follow-Up Instructions: ___

Figure 7: An example of discharge report, which can be split into 3 sections and 7 subsections: clinical profile, in-hospital progress, and discharge plan summary. Note: Some pertinent results from exams are before diagnosis, while some are after diagnosis or after procedures.

Contriever Contriever is a single-tower dense retrieval model that employs self-supervised contrastive learning to enhance document embeddings for retrieval tasks. It encodes both queries and documents using the same encoder, producing dense vector representations. The model utilizes a self-supervised contrastive learning approach with a loss function that optimizes embeddings by comparing relevant passages to negative (irrelevant) ones

A.5 LLM Backbone

A.5.1 Proprietary Models. We selected GPT-3.5 and GPT-4o as representative models due to their extensive real-world adoption and practical relevance demonstrate the applicability of our framework in everyday clinical settings.

GPT3.5 & GPT-4o GPT-3.5, developed by OpenAI and released in November 2022, was followed by GPT-4o, also created by OpenAI

and launched on May 13, 2024, marking a significant advancement with its ability to process and generate outputs across text, audio, and image modalities in real time.

A.5.2 Open-source Models. Qwen3, Deepseek-R1 and Mistral were included as leading open-source models to benchmark the generalizability and robustness of our approach. Specifically, Qwen3 and Deepseek-R1 are thinking models, which is an essential feature to tackle challenges in our tasks.

Deepseek-R1-8B DeepSeek-R1 is a powerful 671 billion parameter language model developed by DeepSeek AI, from which we use DeepSeek-R1-Distill-Llama-8B was derived as a more efficient 8 billion parameter version, distilled from the original model's knowledge to offer improved performance on reasoning tasks while maintaining computational efficiency

Mistral Mistral-7B-Instruct-v0.3 is an advanced instruction-tuned language model featuring an extended vocabulary of 32,768 tokens, support for the v3 Tokenizer, and function calling capabilities, enabling more versatile and complex interactions compared to its predecessors

Qwen3-30B-A3B Qwen3-30B-A3B is developed by Alibaba Cloud, released at April 2025. It is a dense-and-MoE model that has a "thinking mode" for deep reasoning, math, and coding. It outperforms earlier Qwen generations on logical reasoning, code generation, tool-using agent tasks, and human-preference benchmarks, while supporting 100 + languages for instruction following and translation. In our work we leverage its thinking mode to maximize accuracy on complex reasoning tasks.

A.5.3 Medical Models. **Baichuan-M1** Trained from scratch on 20 T tokens that mix high-quality clinical and general texts, Baichuan-14B-M1 is the first open-source 14 B-parameter LLM purpose-built for medicine. It incorporates specialised heads, enabling fine-grained reasoning that matches general-domain peers on standard benchmarks yet surpasses models 5× larger on medical tasks. An updated architecture with longer-context handling further improves comprehension of lengthy clinical narratives and complex patient histories.

We have experiments shown in A.5.3. Baichuan-14B-M1 exhibits the same limitations we observed in other compact models. When additional context from EHR- or text-based retrieval is supplied, the model occasionally confuses the target patient with the retrieved similar cases. Under RAG settings, this confusion produces > 5 % invalid responses—outputs that either fail our answer-parsing patterns or omit a decisive answer—leading to a noticeable overall drop in accuracy. Despite these invalid cases, Baichuan-M1 using EHR-Based ExpRAG still surpasses other approaches on the Medication task, underscoring its strength in pharmacological reasoning even when other aspects falter.

UltraMedical Llama-3-8B-UltraMedical is derived from Meta's Llama-3-8B and further tuned on the 410 K-example UltraMedical dataset (synthetic + curated), an 8 B-parameter biomedical specialist. It tops MedQA, MedMCQA, PubMedQA and MMLU-Medical, handily beating larger general models such as GPT-3.5 and Meditron-70B, and rivals domain-tuned Flan-PaLM and OpenBioLM-8B. The model targets exam-style question answering, literature comprehension and clinical-knowledge retrieval, making advanced medical NLP accessible at modest compute cost.

Model	Context	Instruction	Diagnosis	Medication
		Acc(%)	F1	F1
Baichuan	Direct-Ask	89.8	0.381	0.256
	Text-based	87.8	0.377	0.277
	ExpRAG $_{EHR}$	86.3	0.373	**0.285**

Table 8: Performance from Biomedical Model, with 5% invalid answer on Text-based and EHR-based

UltraMedical's 70 B variant has a 8 k-token context window. When we integrated the model into our retrieval-augmented generation pipeline, more than 50 % of the assembled prompts—target patient record + similar-patient excerpts + task instruction—were longer than 8 k tokens. These over-length inputs had to be truncated or rejected, which systematically stripped clinical details from half of our queries and degraded answer quality. Because this hard context limit blocked reliable end-to-end evaluation, we discontinued UltraMedical-70B for our study despite its otherwise strong domain results.

A.6 Prompts

We provide detailed textual forms and analyses of the prompting methods systematically studied in this work.

Diagnosis and Medication QA Generation Prompt. Figure 8 presents the prompt used for systematically generating question-answer pairs for the Diagnosis task. The same structured prompt design is also utilized to generate options for the Medication task. This prompt explicitly instructs the LLM to:

(1) Clearly differentiate between correct diagnoses or medications and clinically relevant distractors derived from the patient's EHR data.
(2) Summarize and simplify overly long distractor options to maintain consistent length and clarity.
(3) Ensure generated distractors remain realistic and contextually relevant, thereby posing a meaningful evaluation challenge to clinical reasoning capabilities of LLMs.

This structured generation process systematically ensures high clinical validity and challenging multi-choice QA items, as discussed in Sections **??** and **??**.

Instruction QA Generation Prompt. Figure 9 shows the prompt designed to generate valuable multiple-choice questions and answers for evaluating discharge and follow-up instruction generation. This prompt methodically instructs the LLM to:

(1) Formulate clinically relevant and actionable questions directly answerable from the discharge context.
(2) Ensure questions are framed to explicitly guide clinical decision-making evaluation (e.g., identifying best instructions or follow-up care recommendations).

(3) Generate multiple-choice options (ABCDE) with a carefully crafted correct option and four distractors. The distractors systematically introduce clinically plausible variations by altering or replacing specific points of the correct option, thus testing LLM's detailed clinical reasoning skills.

Instruction-Filtering Prompt (Quality Evaluation). Figure 10 illustrates the explicit evaluation prompt designed for assessing the clinical utility and safety of generated discharge and follow-up instructions. This prompt implements a systematic evaluation method by mandating explicit criteria:

(1) Each instruction must clearly specify actionable steps ("what to do").
(2) Instructions must explicitly mention clinically significant "warning signs" requiring immediate professional medical contact.
(3) Instructions that merely describe clinical events without actionable or safety-related guidance are explicitly rejected.
(4) Complex instructions must be split into simpler, clear, keyword-oriented points to facilitate downstream usage and retrieval.

To further highlight the clinical and operational benefits of this instruction-filtering prompt, we provide a detailed analysis below (Appendix A.7).

A.7 Detailed Analysis of Instruction-Filtering Prompt

The instruction-filtering prompt (Figure 10) systematically enforces clinically rigorous evaluation criteria on generated discharge instructions. Here we explicitly analyze the effectiveness of its design criteria.

Explicit Clinical Criteria. This prompt explicitly mandates the presence of actionable steps ("what to do") and critical warning signs ("warning signs") in every discharge instruction. These criteria directly address common real-world clinical communication failures, significantly reducing risks associated with ambiguous or unsafe patient instructions.

Simplification of Complex Instructions. Instructions identified as overly complex are explicitly required to be split into distinct, simpler actionable points. For instance, a complex instruction stating "Your skin or the whites of your eyes become yellow" is systematically split into clearly actionable, distinct points such as "Skin turns yellow" and "Eyes turn yellow," each assigned concise and clinically relevant keywords (e.g., "jaundice," "yellow skin," "yellow eyes"). This design greatly enhances interpretability, retrieval accuracy, and downstream automation potential.

Automated and Scalable Quality Control. The structured JSON evaluation format provided by this prompt significantly simplifies the automation of post-generation validation processes. By explicitly delineating clear pass/fail/select outcomes and providing detailed structured outputs, it greatly improves the reliability, consistency, and scalability of automated instruction-generation pipelines for clinical decision support systems.

Overall, the explicit, structured, and clinically-driven criteria embedded in the instruction-filtering prompt represent a robust and systematic approach toward improving the clinical quality, clarity, safety, and operational effectiveness of generated discharge instructions. This analysis reinforces the value and necessity of rigorous prompt optimization and structured evaluation methods within clinical prompt-generation frameworks.

A.8 Detailed Setup for Ranker Evaluation

Patient and Candidate Sampling. We first randomly select 100 target patients from the MIMIC-IV cohort. For each target, we construct 100 candidate patients by:

- Sampling 20 uniformly at random from the filtered step described in 4.2 from full dataset (to include unrelated cases).
- Sampling 80 from a restricted pool formed by taking the patients (per target) with non-zero EHR similarity in either of three modalities (diagnoses, procedures, prescriptions).

Annotation with GPT4o-mini. Each target–candidate pair is judged by GPT4o-mini along the three modalities. We average these three scores to obtain a single "ground truth" similarity for correlation.

Ranking Methods.

(1) **Text embeddings:** Cosine similarity over discharge summaries using three pretrained models (`bge-small-en-v1.5`, `all-MiniLM-L6-v2`, `paraphrase-MiniLM-L3-v2`).
(2) **EHR-based EXPRAG:** Jaccard similarity on code sets (`ICD/NDC`) for each modality, aggregated with equal weights.

Correlation Metrics. For each ranker and target patient, we produce a ranked list of 100 candidates. We then compute Pearson's ρ and Spearman's τ between the ranker's scores and the GPT4o-mini annotations, and average each metric across all 100 targets.

Results. Table 4 (main text) reports the final average correlations, demonstrating that EHR-based EXPRAG best aligns with human-like judgments.

A.9 Details in Case Studies

We conduct a case study focusing on Discharge Diagnosis in DischargeQA. We examine the discharge diagnosis of a target patient (ID: 20453584) who presented with bilateral ulnar paresthesias and neck pain. We compare this patient with similar patients, who are selected by comparing ICD/NDC codes from EHR tabular data. Fig4 shows one example question in DischargeQA and the similarities in the discharge reports between the target patient and the similar patients with IDs 25633130, 29378221 and 28817667. Upon reviewing the discharge summaries of the similar patients, it became clear that several key diagnostic features were shared with patient 20453584:

- **Disc Herniation**: Both the target patient and similar patients had disc herniations, with the target patient experiencing a C5-6 disc-osteophyte complex and the similar patients exhibiting C3-C4 and C6-C7 herniations.
- **Spinal Stenosis**: Many of the similar patients displayed **spinal stenosis**, which was consistent with the target patient's symptoms of narrowing of the spinal canal and foraminal narrowing.
- **Upper Extremity Symptoms**: The target patient reported **bilateral ulnar paresthesias**, which mirrored the bilateral symptoms observed in several similar patients, such as neck pain radiating to the arms and tingling in the extremities.

Results: By comparing the discharge summaries, key features from similar patients that influenced the diagnosis of the target patient:

- Similar patients with **C6-C7 disc herniation** and **radiculopathy** helped to refine the target patient's diagnosis, suggesting that similar nerve root involvement could explain the upper extremity symptoms.
- The presence of **spinal stenosis** and **neural foraminal narrowing** in several patients guided the understanding of the target patient's potential nerve compression, which contributed to the diagnosis of **spinal stenosis**.

The comparison to similar patients led to a more precise discharge diagnosis for the target patient, which included a C5-6 disc-osteophyte complex with associated **spinal canal narrowing** and **neural foraminal narrowing**. These insights allowed the LLMs to confirm the target patient's diagnosis, which aligned with options **A** and **G** — **disc osteophyte** and **spinal stenosis**.

Prompt: generate options for Diagnosis task

Role: You are a doctor evaluating the Discharge Diagnosis of a patient.

Task: Your objective is to review the discharge diagnosis provided in the discharge summary and determine whether these diagnosis are suitable for the patient's treatment plan. The correct options are based on the diagnosis listed in the discharge summary, while incorrect options are derived from diagnoses table that are not part of the discharge diagnosis but may have been found during the hospital stay.

Discharge Diagnosis:
----info starts----
{discharge_diagnosis}
----info ends----

Diagnoses Database Info:
----info starts----
{diagnoses}
----info ends----

Also, please review the provided background info from other part of the Discharge Summary, which can be summarized (keep important info) to be background info, but do not put any diagnosis decision into it.
----background info starts----
{discharge_summary}
----background info ends----

Please provide a multi-answer true/false response for the following question:

Question:
Which discharge diagnosis were made for the patient at discharge?

Answer Options:
Provide a list of diagnosis in JSON format.
Each diagnosis should be marked as "True" if it was in the discharge diagnosis and "False" if it was only found in the diagnoses history but not listed as a discharge diagnosis.

Instruction:
1. List all items in the diagnosis and assign one option letter (from A to Z then a to z) to each non-repeated one
2. Review all items provided in the diagnoses database one by one. If the item is also listed by discharge diagnosis, or equivalent or very close meaning, then the "correct_answer" should have the letter of this item, and this item should be the same way as described in "Discharge Diagnosis"
3. If the item is from "Diagnoses Database Info" only but not in "Discharge Diagnosis", and the name is too long, please summarize it to be less than 10 words

Output Format:
Provide your responses in JSON format as follows:
{
"Reason": "<Explain how you combine equivalent diagnosis from both info sides to which options, and which options are from which info>"
"background": "{background} + <Your summary from other parts of the Discharge Summary, do not put diagnosis info into it, try to include as much important info as possible>",
"discharge_diagnosis_options": {
"A": "<diagnosis name>",
"B": "<diagnosis name>",
"C": "<diagnosis name>",
...
},
"correct_answer": "<String of options representing correct diagnosis, e.g., 'ACD'>"
}

Figure 8: Prompt design example for Diagnosis tasks

Prompt: generate options for Instruction task

Role and Task

Role: You are a doctor and a professional discharge summary writer in a hospital.

Task: Your objective is to use existing discharge summaries to generate valuable questions and corresponding answers (refined from the context of the discharge summary) that will evaluate the ability of large language models (LLMs) to generate discharge/follow-up instructions.
The LLM to be evaluated will be provided with a discharge summary where certain parts in "Discharge Instructions" need to be filled in. The LLM should select the correct option as all or part of the instructions based on the information provided and the experience inferred from similar patients.

Info retrieved

Info for background from Discharge Summary:
----info starts----
{discharge_summary}
----info ends----

DO

"Good Points" for correct instructions:
----points starts----
{points_of_instructions}
----points ends----

DON'T DO

"Other Points to avoid" including in options:
----points to avoid starts----
{points_to_avoid}
----points to avoid ends----

Please review the provided background info, which can be summarized to the background, and key points for instructions from the target patient

Detailed Instructions

Ensure:
1. Question
 - Can be answered directly from context
 - Consider the category of each point to organize questions. Recommend but not limit to topics: Follow-Up Care Recommend[illegible]plications Post-Discharge; Predicting Readmission Risk, etc.
   - Formulate it For example, instead of asking, "What are the specific...", you should ask, "What are the best..."
   - Value: Your questions should guide the reader of clinic notes to have meaningful info, and be valuable for evaluating the LLM’s reasoning and generation capabilities.
2. Points:
   - "warning_signs" are points once which is found, doctor or other people in hospital should be contacted
   - Filter out bad points. For example, "Contact Dr__" is not helpful to evaluate LLM because we have no info from "__"; if there is any "_" part as info erased, do not use this part as point to distinguish options, such as "_ lbs" or "_ days", no need to alter it to "6 lbs" or "3 weeks" for incorrect options.
3. Generate Answer Options (ABCDE): For each question, generate five answer options:
   - A correct answer [A]: Create the correct answer by combining all the good points provided
   - Four incorrect answers [BCDE]: Generate four additional options by altering or replacing 1 or 2 "good points" while keeping the other points the same as [A].
     - "altering" means changing the important number or keyword of that point
     - "replacing" means: Make sure options [CE] to have replaced points which are completely generated by you and:
       (1) not in the context of the background of discharge summary
       (2) follow the correct clinic knowledge and health knowledge but are not applicable to the patients mentioned in the background.
       (3) have the same topic of the original point of [A] but different details. For example, if [A] refers medicine, then [CE] should also refer different medicine.
       (4) have the ability to confuse LLM to make the correct choice without patient's background.
       (5) are more specific, comprehensive and offer more guidelines than option [A].
       (6) make sure are not in "Other Points to avoid" provided above
     - Not the same point is chosen in [D] and [E] to be altered or replaced
     - Not the same 2 points are chosen in [B] and [C] to be altered or replaced
     -
   - Make sure options can not contain any patients' specific information of the patient, such as disease, surgery, drugs, past history, such as gout, fever, incision.
   - Make sure options are general enough and do not involve proper nouns.
4. Error Reason: For each incorrect answer, include an explanation of which point is altered/replaced and why.

Define Output Format

Output Format:
Provide your responses in JSON format, as shown below:
```
{
  "hadm_id": {hadm_id},
  "background": "{background} <Your summary from other parts of the Discharge Summary>",
  "qa_pair": {
     "reason_of_value": "<Why this question is valuable to LLM evaluation>",
     "selected_points": {
       "point1": {
       "index": <The index from provided info>
       "summary": <The summary of the first key point of the context"
       }
       "...": "<Additional key points>"
     },
     "question": "<The reasoning-based question you are posing>",
     "error_reasons": {
       "B": "<Explanation of which 2 points altered in option B and why>",
       "C": "<Explanation of which 2 points replaced in option C and why>",
       "D": "<Explanation of which 1 point altered in option D and why>",
       "E": "<Explanation of which 1 point replaced in option E and why>"
     },
     "answer_options": {
       "A": "<Correct answer with no more than 5 points>",
       "B": "<Incorrect option, with 2 points altered, same number of points as A>",
       "C": "<Incorrect option, with 2 points replaced>",
       "D": "<Incorrect option, with 1 point altered, same number of points as A>",
       "E": "<Incorrect option, with 1 point replaced>"
     }
  }
}
```

Figure 9: Prompt design example for Instruction tasks

Prompt: Instruction Filtering

```
Task:
You are evaluating a discharge or follow-up instruction.
Your task is to assess whether the instruction is sufficient or inadequate based on the following criteria and return
your evaluation in JSON format.

Evaluation Criteria:
1. If the instruction content is empty (e.g., "Follow-up instruction: ___"), mark it as "Fail."
2. A good instruction must include at least one of both:
   - What the patient should do ("what to do").
   - Warning signs that require contacting a doctor ("warning signs").
3. A common poor instruction only describes "what happened" without providing actionable steps ("what to do") or
warnings ("warning signs").
4. If the instruction includes both "what happened" and "what to do"/"warning signs":
   - Identify and extract the points related to "what to do" and/or "warning signs."
   - List these points in the "selected_content" field in your JSON output.
5. For each point in "what to do" and "warning signs":
   - If there is only 1 point, or it is too general without useful information, mark it as "Fail."
   - For each point, provide:
      (1) A concise summary (few words).
      (2) A list of equivalent or closely related keywords.
      (3) The original sentence (if clear), or a rewritten instruction.

Ensure:
Split complex points into simple, concise points so that each can be represented by selecting 1 or at most 2
keywords from a list of equivalent terms(list of keywords).
Example 1 "sentence": "Your skin, or the whites of your eyes become yellow.", I cannot use "skin" or "eyes" as key
word to represent the point. So this point should be split to 2 points like below:
   {
      "summary": "Skin turns yellow",
      "keywords": ["yellow skin", "jaundice"],
      "sentence": "Your skin becomes yellow."
   },
   {
      "summary": "Eyes turn yellow",
      "keywords": ["yellow eyes", "whites of eyes yellow", "jaundice"],
      "sentence": "The whites of your eyes become yellow."
   }
Example 2 "sentence": "You must continue to wear the clamshell brace when out of bed at all times.". It should be
split to 2 points "wear the clamshell brace" and "wear the clamshell brace when out of bed at all times"

Here is the context:
---context starts---
{context}
---context ends---

Provide your evaluation in JSON format, as shown below:
{
  "reason": <Explanation for your decision>,
  "evaluation": <Pass, Fail, or Select>,
  "selected_content": <If "Select", include relevant points; otherwise, leave empty>,
  "points": {
    "what_to_do": [
      {
        "summary": <A few words summarizing the action>,
        "keywords": [<List of equivalent keywords>],
        "sentence": <The instruction sentence>
      }
      ...
    ],
    "warning_signs": [
      {
        "summary": <A few words summarizing the warning sign>,
        "keywords": [<List of equivalent keywords>],
        "sentence": <The warning sentence>
      }
      ...
    ]
  }
}
```

Figure 10: Prompt design to filter discharge instructions