Evaluating LLM Abilities to Understand Tabular Electronic Health Records: A Comprehensive Study of Patient Data Extraction and Retrieval

Jesus Lovon, Martin Mouysset, Jo Oleiwan, Jose G. Moreno, Christine Damase-Michel, Lynda Tamine

Evaluating LLM Abilities to Understand Tabular Electronic Health Records: A Comprehensive Study of Patient Data Extraction and Retrieval

Jesús Lovón-Melgarejo[1], Martin Mouysset[1], Jo Oleiwan[1], Jose G. Moreno[1], Christine Damase-Michel[2], and Lynda Tamine [1]

[1]Université Paul Sabatier, IRIT, Toulouse, France
[2]Centre Hospitalier Universitaire de Toulouse, CERPOP INSERM UMR 1295 - SPHERE team, Faculté de Médecine Université de Toulouse, Toulouse, France
{jesus.lovon, martin.mouysset, jo.oleiwan, jose.moreno,tamine}@irit.fr
{christine.damase-michel}@univ-tlse3.fr

Abstract. Electronic Health Record (EHR) tables pose unique challenges among which is the presence of hidden contextual dependencies between medical features with a high level of data dimensionality and sparsity. This study presents the first investigation into the abilities of LLMs to comprehend EHRs for patient data extraction and retrieval. We conduct extensive experiments using the MIMICSQL dataset to explore the impact of the prompt structure, instruction, context, and demonstration, of two backbone LLMs, Llama2 and Meditron, based on task performance. Through quantitative and qualitative analyses, our findings show that optimal feature selection and serialization methods can enhance task performance by up to 26.79% compared to naive approaches. Similarly, in-context learning setups with relevant example selection improve data extraction performance by 5.95%. Based on our study findings, we propose guidelines that we believe would help the design of LLM-based models to support health search.

Keywords: Large language models · Electronic Health Record (EHR)· tabular data· information retrieval · information extraction

1 Introduction

Large Language Models (LLMs)' applications on tabular data [24,5,32,8] (e.g., question-answering [7] and table search [26]) are beneficial in several domains such as finance [3] and health [23]. However, recent studies [5,24] show that the gap between tabular data and natural language significantly hinders LLM's performance on downstream tasks due to several challenges among which lack of standard adequate transformation from data to text, lack of grounding with prior knowledge, and lack of generalizability across data structures.
In particular, Electronic Health Record (EHR) tables include high-level of data heterogeneity, high dimensionality, and sparsity. Indeed, an EHR is a digital patient's medical history that contains vast amounts of heterogeneous tables (e.g.,

26 tables in MIMIC-III [10]) that comprise a temporal and longitudinal structure of patient visits. Each visit includes both administrative and clinical data related to a significant number of features with different natures (e.g., diagnosis, procedure and medication codes, dosages). While standard health-related tasks such as information extraction and retrieval require a synergic understanding of structure and content from one patient-entity view, previous work studied, agnostically to domain application, the capabilities of LLMs to understand separately table structure (e.g., table splitting and parsing [24]) and tasks (e.g., classification, question-answering [32,8]) using tables as a set of elements. Thus, it is still unclear whether previous findings on LLMs' understanding of tabular data are transferable to EHRs and patient-related tasks. Furthermore, there are so far no clear findings about the extent to which LLM's prompt elements such as *instruction*, *context*, and *demonstration* intertwine to jointly impact the performance of LLM's in health search tasks, namely extraction and retrieval. To sum up, there is a critical gap in the literature regarding standard best practices and guidelines for prompting LLMs on EHR-related tasks.

In this paper, we aim to fill this gap. To achieve this goal, we conduct extensive experiments using the publicly available MIMICSQL[1] [27] dataset, based on the MIMIC III benchmark dataset widely adopted in the literature [10]. By exploring the effect of *instruction*, *context*, and *in-context demonstration* on task performance, our study consists of the following highlights: 1) investigating the LLMs' understanding of the relationship between EHR structure and content by evaluating the joint impact of EHR serialization and medical feature selection; 2) analyzing the effect of providing guided vs. non-guided instructions regarding the task outcome; and 3) measuring the impact of in-context demonstration quality within an In Context Learning (ICL) setup on task performance. Our findings lead us to assess the following: 1) synergic comprehension of content and structure through EHR serialization and feature selection is significantly sensitive to prompt context with improvements up to 26.79%. In particular, LLM' self-generated EHR table descriptions are more impactful on task performance; 2) the use of guided instructions has minimal impact on task performance; 3) ICL positively impacts LLMs' performance, particularly for the extraction task, with the best results obtained by selecting examples that better match the query input rather than the patient; and 4) LLMs have more difficulty retrieving relevant tabular data than extracting relevant tabular data from EHRs. We summarize our contributions as follows:

- Through an extensive empirical study using standard medical datasets, we provide researchers and practitioners guidelines for suitably prompting LLMs on EHR-related tasks;
- We propose two new datasets MIMIC$_{ask}$ and MIMIC$_{search}$, based on the MIMICSQL dataset, to benchmark LLMs' on health data extraction and retrieval;

[1] https://github.com/wangpinggl/TREQS

– We provide for future work, code implementation for EHR data-to-text transformation techniques, instruction formatting, and patient example selection strategies[2], as well as corresponding baselines per task.

2 Prompting LLMs on tabular data

Early work dealing with data-to-text generation focused on the design of suited structure-aware encoder-decoder architectures [4,17,15,18] and specific pretraining strategies (e.g., TaPas [4], TUTA [29], and UTP [1]). Recent efforts leverage the strengths of decoder-only architectures through LLMs [5]. The core underlying issue is the design of appropriately structured prompts composed of linear texts describing the input data and in-context demonstrations for helping LLMs understand the outputs [30].
To set up effective techniques that convert tabular data into linear texts fed to LLMs, previous work has relied on 1) hand-crafted templates using JSON, HTML, XML, and X-separate formats [21,24,16]; 2) embedding-based serialization techniques that rely on table encoders (e.g., UniTabPT [20] and TableGPT [6]); 3) graph-based serialization techniques that convert a table into a tree represented as a tuple fed to the LLMs [35]; and 4) LLM self-generated table description [24]. Overall, research findings show that LLMs' performance is heavily sensitive to prompt formats [24,21,8], and that most LLMs struggle to handle high-dimensional tables due to the long context they induce. This leads to a significant challenge in balancing between effectiveness and efficiency in terms of memory and computational cost [14,25]. ICL [30] has also been shown to be impactful on the performance of tasks involving tabular data regardless of downstream tasks [24,2,8]. It has been shown that performance is optimized with a limited number of examples [2]. Zhao et al. [35] and Ye al. [32] also demonstrated the benefits of applying chain of thought reasoning (COT) to enhance search performance on tables.

3 Study Design

3.1 Tasks

We focus on tasks leveraging a repository R that contains raw tabular data related to n EHRs represented using a reference set of demographic and clinical features $F = \{f_1, \dots f_k\}$. The EHR of patient p_i (with $1 \leq i \leq n$) can be formalized as a reference table T_i structured using a subset of features $F^{p_i} \subseteq F$ where $F^{p_i} = \{f_1^{p_i}, \dots f_{k_i}^{p_i}\}$, with k_i is the number of EHR features in T_i. We assume that feature names are natural language strings (e.g., "age", "blood pressure"). In practice, each table T_i is built upon a subset of tables in R. We formally define two pilot tasks that we address in our work.

[2] `https://github.com/jeslev/llm-patient-ehr`

- *Extraction*: Given the EHR of patient $\mathbf{p}_i$ represented with table T_i, the goal is to generate an answer $\mathbf{a}$ to a natural language query $\mathbf{q}$ based on information in T_i, e.g., **q:** *"specify the primary disease and icd9 code of patient id 1875"*;

- *Retrieval*: The goal is to provide a list of EHR tables $T_1, \ldots, T_m$ from the repository R, that are relevant to a given natural language query $\mathbf{q}$, e.g., **q:** *"Which male patients had done the lab test renal epithelial cells?"*.

3.2 Prompt design

Prompt format. We consider hard prompts as triplets following the generic format composed of the concatenation of elements in the form <`Instruction [Demonstration] Context>`, where the symbol `[...]` indicates that the occurrence of the element is optional. The element order of the format follows recommendations from previous findings [5,25], where:

- `Instruction` refers to a short textual description of the task I_e or I_r for respectively the *extraction* and *retrieval* task.
- `Context` includes two elements: 1) the natural language description C_i such as $C_i = \phi(T_i)$ where T_i is the EHR tabular form of the input patient $\mathbf{p}_i$, ϕ is a table serialization function; and 2) query $\mathbf{q}$ involved in the task (§3.1).
- `Demonstration` comprises examples appended to the prompt within the ICL setting. Formally, we build a database E of labeled examples (q, p, a) and (q, a) for respectively the *extraction* and *retrieval* task, where a is the gold answer for query q, and p is the target patient for the *extraction* task. The core component of the demonstration selection strategy relies on a retrieval function σ which provides high-quality examples from database E to be fed as demonstrations to the LLM.

Prompting strategies. Our strategies are given by the multiple configurations of the prompt format defined as follows.

Instruction. We follow recommendations from previous work that emphasize the positive impact of guided instructions [22]. Specifically, we explore using *Non-Guided* instructions vs. *Guided* instructions through a step-by-step description of how the model should analyze the input to provide the expected output.

Context. To investigate to which extent LLMs can understand EHR structure we explore a set of SOTA table serialization functions ϕ on EHR table structure.

- *Templates (txt)* [8,16]: rows are converted into sentences using a simple text template for each column;
- *X-separate (xsep)* [25,21]: rows are line-separated, columns separated with a special character. Particularly, we use HTML tags;
- *Self-generate (sgen)* [28,24]: following previous work, we prompt a LLM to self-generate from an input EHR table a textual description with relevant features in function of a given question.

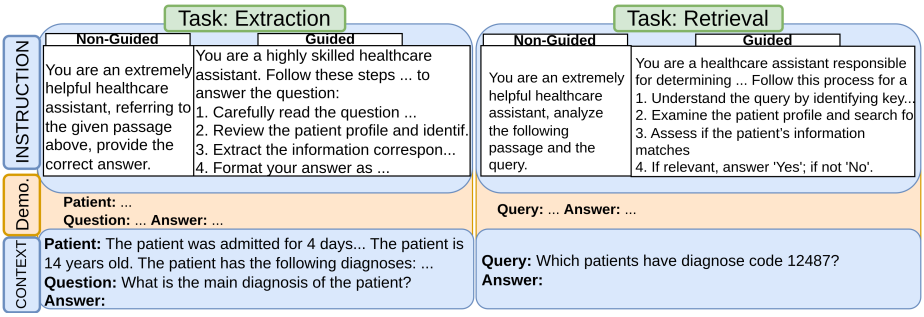

Fig. 1: Illustration of the prompts used for the *extraction* and *retrieval* tasks, including *Guided* vs. *Non-Guided* instructions, and patient with *txt* (left) serializations.

To synergically consider EHR table structure and content, we jointly explore a table structure serialization strategy ϕ with table feature selection of a subset of medical features $F^{p_i} \subseteq F$ as follows:

- *All (all):* all features and associated longitudinal values;
- *All aggregated (all_{avg})*: all features and associated aggregated (averaged) longitudinal values;
- *Random (rnd)*: random features and associated longitudinal values;
- *Random aggregated (rnd_{avg})*[3]: random features and associated aggregated (averaged) longitudinal values.

Demonstration selection. Previous work showed the impact of in-context example quality on downstream task performance [19,34]. We explore two main strategies of demonstration selection by varying the core retrieval function σ:

- *Patient-based retrieval function* σ_p which retrieves high-quality examples from E based on patient similarity. This strategy is applicable for the *extraction* task which involves a patient input description $\mathbf{p}$ (§3.1).
- *Query-based retrieval function* σ_q which retrieves high-quality examples from E based on query similarity. This strategy is applicable for the *extraction* and *retrieval* tasks which both involve an input query $\mathbf{q}$ (§3.1).

Figure 1 summarizes the prompt formats used w.r.t each studied task (§3.1).

3.3 Datasets

Our study uses the MIMICSQL dataset [27], based on the MIMIC III dataset, which contains de-identified EHRs from 48,520 ICU patients over a decade (2001-2012), structured into 26 tables. Each EHR record includes demographics and medical features (age, laboratory measurements, diagnoses, etc.). MIMIC III provides detailed time-series clinical features (e.g., blood pressure, heart rate) with variable time stamps (second, minutes), and formats, leading to a high level

[3] For both random approaches, we kept 60% of the total features

	# patients(n)	# features(k)	# k/n	# train	# dev	# test
MIMIC$_{ask}$	100	5414	34	861	96	372
MIMIC$_{search}$	4000	19970	557	2204	368	$1101^{full} \mid 250^{small}$
MIMICSQL	46520	32340	3912	8000	1000	1000

Table 1: Statistics of MIMIC$_{ask}$, MIMIC$_{search}$ and MIMICSQL datasets.

of heterogeneity and data sparsity. The MIMICSQL dataset [27] is a question-SQL pair dataset based on MIMIC III, to perform the Question-to-SQL generation task in the healthcare domain. It comprises 10,000 questions, expressed in natural language and SQL queries using 5 tables from the original database (Demographics, Diagnosis, Procedure, Prescriptions, and Laboratory tests).

We used the MIMICSQL dataset to perform the *extraction* and *retrieval* tasks and created the new MIMIC$_{ask}$ and MIMIC$_{search}$ datasets. We focused on the questions related to **single** and **multiple** patients to explore LLM abilities to comprehend EHRs, and omitted general questions that target database-level facts. To build the ground truth, we evaluated the golden SQL queries provided in the original MIMICSQL dataset and generated corresponding question-query pairs by converting the SQL queries into their natural language form. We created upon the MIMICSQL dataset the MIMIC$_{ask}$ using **single** patient questions and MIMIC$_{search}$ datasets using **multiple** patient questions for the *extraction* and *retrieval* tasks respectively. For the MIMIC$_{ask}$ dataset, we cleaned the gold SQL answer by removing duplicates, serializing, and concatenating the features according to the format *"column name: value"*. For the *retrieval* task, we adapt the original question into queries by using rule-based query reformulations to target patients as outputs (e.g., transforming *"Count the male patients that had done the lab test..."* into *"Which male patients had done the lab test..."*).

Finally, we created training, validation, and test datasets for each task, ensuring, among other factors, that there was no overlap of patients between the training and test sets. For the *extraction* task, we used the originally sampled question answer about 100 random patients from MIMICSQL. For the *retrieval* task, we increased the corpus to 4000 random patients and created two versions, *small* and *full*, based on the number of test queries (1101 and 250 respectively). We used the MIMIC$_{search}$ *small* dataset for our study and shared both versions with the community for future work. Table 1 presents the statistics of the MIMIC$_{ask}$ and MIMIC$_{search}$ datasets.

3.4 Experimental Setup

LLM Setups. For our evaluation we selected two main LLMs previously applied in patient-related tasks [16]: Llama2-7B[4] (Llama) and Meditron-7B (Meditron)[5]. The latter leverages further pre-training on medical PubMed scientific corpus. We particularly used a 4-bit quantization configuration and a maximum context

[4] `https://huggingface.co/meta-llama/Llama-2-7b-chat-hf`
[5] `https://huggingface.co/malhajar/meditron-7b-chat`

length of 4096 tokens. Additionally, we explored the performance of the LLMs under a fine-tuning approach using the LoRa optimization strategy.

To perform the *extraction* task, we follow previous work [33] and provide the question (query) $\mathbf{q}$ and the serialized text-based description $\phi(T_i)$ of EHR of patient $\mathbf{p}_i$ to the LLM to generate the answer $\mathbf{a}$. Regarding the *retrieval* task (§3.1), we follow previous work [36] and employ a zero-shot LLM-based pointwise re-ranker. A list of k-top relevant patients to the input query q is retrieved in the first stage with an unsupervised retriever and then a LLM is prompted to generate whether the input candidate patient is relevant to the query. The re-ranking stage is repeated for each candidate relevant EHR $\phi(T_x)$, for $x = 1 \dots k$ rounds. At each round, we provided the LLM with query q, concatenated with the text-based description $\phi(T_x)$ (§ Figure 1).

Dense retrievers. In our work, the retrieval function σ (§3.2) relies on an off-the-shelf dense unsupervised retriever, agnostic to all the pilot tasks. Since we focus, on the joint impact of EHR structure and content on task performance, the retriever selects the top-K examples from E based on several EHR representations similar to those used for serialization based on function ϕ (§3.2). Following recent work [13], we encoded queries and text-based EHR using the Dragon+ encoder [12] in a pre-processing step with the training set of the corresponding task. Then, we used FAISS, implemented through the Pyserini framework [11], to index and retrieve the most relevant examples based on similarity. In addition, we design the random selection *Random* as a lower-bound baseline to evaluate the effectiveness of the feature selection strategies.

Baselines. We established various baselines for each of the tasks studied in our work: 1) for the *extraction* task, we evaluated different generative models, including T5[6] and BART[7] in a zero-shot setup (T5_0 and BART_0), as well as fine-tuned versions in the target task (T5_{ft} and BART_{ft}); 2) for the *retrieval* task, we considered both sparse rankers such as BM25, and dense rankers: MonoBERT and MonoT5, which were finetuned for the target task. Finally, we reproduced the TREQS model [27], executed the generated SQL queries based on our test sets, and post-processed them following the original work [27]. In case of execution errors due to syntactically incorrect generated queries, we considered them as empty outputs. We then adapted the final SQL answer to match the expected output for our tasks (§3.1).

Metrics. We used standard evaluation metrics appropriate for each task: 1) for the *extraction* task, we used Rouge-1 (R-1) and BERT score (B_{score} referring to the F-1 score); 2) for the *retrieval* task, we used MAP and Recall@100 (R).

		Llama					Meditron				
F^p	ϕ	Extraction		Retrieval		Δ%	Extraction		Retrieval		Δ%
		B_{score}	R-1	MAP	R		B_{score}	R-1	MAP	R	
all	txt	56.18	22.84	9.30	32.19	+26.79	56.21	23.26	8.31	29.01	+21.53
	xsep	57.10	20.97	<u>9.80</u>	<u>33.39</u>	+27.64	52.10	14.94	7.65	27.44	+11.12
	sgen	<u>57.80</u>	23.25	**9.84**	**33.62**	+11.11	52.47	17.51	7.63	24.67	+4.59
all_{avg}	txt	56.86	<u>23.28</u>	8.25	27.70	+22.44	<u>57.26</u>	**25.89**	**10.34**	**32.09**	+21.88
	xsep	57.30	21.84	7.98	28.35	+19.06	54.88	19.85	8.14	29.03	+23.40
	sgen	**58.46**	**24.36**	8.52	32.00	+7.01	**57.79**	<u>23.95</u>	8.05	29.10	+6.62
rnd	txt	51.60	12.69	8.72	28.83	-	53.61	14.09	8.27	25.07	-
	xsep	52.14	12.94	8.98	25.71	-	50.53	11.60	7.32	25.39	-
	sgen	55.89	18.16	9.21	31.67	-	51.57	15.08	7.19	26.14	-
rnd_{avg}	txt	51.76	12.91	8.34	27.73	-	53.66	14.66	<u>10.30</u>	<u>30.91</u>	-
	xsep	52.54	12.75	8.17	28.87	-	50.83	12.69	7.69	23.53	-
	sgen	56.58	20.63	7.90	32.39	-	56.72	20.40	7.53	29.02	-

Table 2: Evaluation on the joint impact of EHR feature selection F^p, and EHR structure serialization ϕ. Δ shows global improvement across tasks and metrics per setting (row) w.r.t their color corresponding baseline (all vs rnd , all_{avg} vs rnd_{avg}) . We report **best** and <u>second best</u> values per metric (column).

4 Results

4.1 Leveraging tabular EHR structure and content

The results of our exploration of the joint impact of EHR feature selection (F^p) and structure serialization (ϕ) on *extraction* and *retrieval* performances using the Llama and Meditron models are shown in Table 2. At first glance, we can see that the optimal setting for Llama is using all features with *sgen* serialization (F^p =all, ϕ =sgen), achieving 3 out of 4 top scores, and Meditron using all aggregated features and *txt* serialization (F^p =all_{avg}, ϕ =txt).

Furthermore, we can notice that both models are sensitive to structure with a wide improvement variation range across feature selection strategies, with Δ from 7.01 to 27.64, and from 4.59 to 23.40 for Llama and Meditron, respectively. Also, the highest improvements are observed for Llama when all values are used (F^p =all and F^p =all_{avg}), while for Meditron only when these values are averaged (F^p =all_{avg}). Focusing on the variability of this impact on performance across tasks, we can surprisingly see that the *retrieval* task seems more difficult for both models. For instance Meditron reaches a minimal improvement of 0.39% (10.34 vs 10.30) for *retrieval* vs. 1.75% (52.47 vs 51.57) for *extraction* w.r.t their baselines. Similarly, Llama drops performance by -2.33% (7.98 vs 8.17) for *retrieval* while improving at least 3.32% (58.46 vs 56.58) for *extraction*. This could be explained by the fact that *retrieval* intrinsically requires more abilities to comprehend, at a coarse-grained level, the EHR structure and content to answer patient-profile-oriented queries, while *extraction* questions explicitly focus

[6] `https://huggingface.co/google/flan-t5-large`
[7] `https://huggingface.co/facebook/bart-base`

(F^p,ϕ)	Llama								Meditron							
	(all, sgen)				(all_{avg},sgen)				(all_{avg},txt)				(all_{avg},sgen)			
	Extraction		**Retrieval**		**Extraction**		**Retrieval**		**Extraction**		**Retrieval**		**Extraction**		**Retrieval**	
I_e/I_r	B_{score}	R-1	MAP	R	B_{score}	R-1	MAP	R	B_{score}	R-1	MAP	R	B_{score}	R-1	MAP	R
Guided	**57.92**	**23.44**	9.56	33.42	**58.93**	**24.98**	**9.22**	31.32	57.26	25.71	**12.07**	**32.54**	**57.82**	23.77	7.98	28.95
Non-Guided	57.80	23.25	**9.84**	**33.62**	58.46	24.36	8.52	**32.00**	57.26	**25.89**	10.34	32.09	57.79	**23.95**	**8.05**	**29.10**

Table 3: Evaluation on the impact of guideline instructions on extraction and retrieval performance. **Bold** reports best score per metric (column).

on fine-grained specific patient features. Thus, LLMs struggle to match the query with relevant passages corresponding to 'cells' in the EHR table. However, both models achieve top performances on both tasks when using all features under different metrics. Specifically, by cross-linking EHR structure, feature selection and task performance, we highlight from Table 2 that the Llama model with F^p ={all,all_{avg}} and *sgen* method consistently outperforms other settings, with one exception in the *retrieval* task where *xsep* also achieves high scores. In contrast, Meditron shows a clear positive trend for F^p =all_{avg} with *txt* and *sgen* methods. The preference of Meditron for *txt* over *sgen* suggests that textual medical knowledge captured by Meditron from the literature corpus endows it with better abilities to leverage EHR-related tasks with this same text format *txt*. Interestingly, by comparing F^p =all and F^p =rnd_{avg}, we can see that Meditron exhibits a close gap in performance between these two settings, with better performance with F^p =rnd_{avg} in the *retrieval* task. This suggests that averaging longitudinal values positively impacts Meditron's performance, even using *random* features.

Overall this first exploration confirms findings from previous works about the sensitivity of LLM prompts on the performance of tabular tasks [24,21,8]. It also reveals the following insights: 1) patient data *retrieval* is a more difficult task than patient data *extraction* for both LLMs; 2) Llama, a general domain LLM, lean to require all patient (*all*) salient features (*sgen*) while Meditron comprehends simple concatenation (*txt*) of averaged patient feature values (all_{avg}) to perform both *extraction* and *retrieval* tasks.
In the following sections, we chose the best settings from Table 2: (F^p =all, ϕ =sgen) and (F^p =all_{avg}, ϕ =sgen) for Llama; and(F^p =all_{avg}, ϕ =txt) and (F^p =all_{avg}, ϕ =sgen) for Meditron.

4.2 Guiding task completion

Next, we evaluate the impact of the *instruction* component, I_e and I_r, by comparing two types: *Guided* vs. *Non-Guided*, based on the optimal settings identified in Section 4.1. Table 3 shows that the choice of instruction type has minimal impact on task performance, with no consistent improvement observed across all metrics. Llama leverages *Guided* instructions, achieving higher performance in 5 out of 8 metrics, compared to Meditron, which only shows improvements in 3 out of 8 metrics. This suggests that Llama benefits slightly more from explicit guidance, whereas Meditron's performance may be hindered when adding detailed

		ICL w/ Llama (Lma_{icl})								ICL w/ Meditron (Med_{icl})							
	(F^p,ϕ) I_e/I_r	(all, sgen) Non-Guided				(all_{avg}, sgen) Guided				(all_{avg}, txt) Guided				(all_{avg}, sgen) Non-Guided			
σ	#ex	**Extraction**		**Retrieval**		**Extraction**		**Retrieval**		**Extraction**		**Retrieval**		**Extraction**		**Retrieval**	
		B_{score}	R-1	MAP	R	B_{score}	R-1	MAP	R	B_{score}	R-1	MAP	R	B_{score}	R-1	MAP	R
-	0	57.80	23.25	**9.84**	**33.62**	58.93	24.98	9.22	31.32	57.26	25.71	**12.07**	32.54	**57.79**	**23.95**	**8.05**	**29.10**
Patient σ_p	1	59.47	26.88	N/A	N/A	60.75	30.61	N/A	N/A	57.08	24.28	N/A	N/A	56.61	21.84	N/A	N/A
	2	59.97	27.16	N/A	N/A	60.51	29.68	N/A	N/A	54.60	21.36	N/A	N/A	51.10	15.40	N/A	N/A
	3	60.75	28.75	N/A	N/A	60.75	29.53	N/A	N/A	54.92	22.61	N/A	N/A	50.72	14.51	N/A	N/A
Query σ_q	1	60.82	28.90	8.06	29.50	60.36	30.61	7.84	29.82	**60.69**	**33.18**	10.15	34.64	52.58	18.49	7.08	24.33
	2	61.42	30.04	8.07	29.43	61.90	31.87	8.81	30.80	59.16	28.09	7.57	20.97	51.28	15.81	6.88	23.48
	3	**61.83**	**31.48**	8.28	31.95	**62.44**	**32.39**	8.85	34.17	54.77	22.88	7.29	23.34	51.05	15.95	7.08	26.68
Random	1	58.66	25.33	7.87	29.63	59.89	28.55	7.98	29.60	57.80	26.75	10.92	**35.64**	53.69	19.26	7.22	24.85
	2	59.90	26.60	7.87	30.20	59.80	27.76	8.67	31.15	56.86	24.74	7.88	21.33	51.91	15.96	6.80	23.49
	3	59.75	26.69	8.48	31.15	60.91	28.90	**9.42**	**35.54**	52.40	17.47	7.77	24.92	50.86	15.06	7.13	27.54

Table 4: Evaluation on demonstration selection strategies and number of examples within an ICL setting. We report **best** and second best values per metric (column).

Guided instructions. When comparing across tasks, we observe that *Guided* instructions are particularly beneficial for the *extraction* task (I_e), with 5 out of 8 metrics showing improvement, compared to 3 out of 8 for *retrieval* (I_r). Notably, *extraction* shows an average B_{score} improvement of 0.27%, a trend inconsistent with other metrics. In conclusion, the choice of instruction type, I_r and I_e, is heavily dependent on the model and setting with however limited impact on performance, unlike what has been found in previous work [22].
For the next sections, we keep the following best-performing settings: (F^p =all, ϕ =sgen, *Non-Guided*) and (F^p =all_{avg}, ϕ =sgen, *Guided*) for Llama; and (F^p =all_{avg}, ϕ =txt, *Guided*) and (F^p =all_{avg}, ϕ =sgen, *Non-Guided*) for Meditron.

4.3 Selecting demonstrations

Now, we evaluate the impact of demonstration quality in an ICL setup on task performance. To this end, we varied the demonstration retrieval function σ (§3.2) and the number of demonstrations fed to the LLM, $k = \{1, 2, 3\}$. To fit the LLM's context length constraints, we truncate patient serializations as necessary. We compare the ICL performance w.r.t scores obtained in Section 4.2 with no demonstrations ($k = 0$). Our results are reported in Table 4.

The first surprising observation from these results is that the Meditron model fails to leverage the ICL setup across most settings. The best scores are obtained in 5 out of 8 metrics with $k = 0$ demonstrations, leading to dropping performance up to 39.60% in MAP and 12.24% in B_{score} with ICL. This behavior suggests that demonstrations do not bring to Meditron additional relevant external knowledge beyond the domain-specific internal knowledge acquired during fine-tuning. Analyzing ICL's impact at the task level, we observe that the *extraction* task benefits more from the ICL setup. On the contrary, the *retrieval* task consistently underperforms compared to zero-shot, with a performance drop of

Extraction									
Models	$T5_0$	$BART_0$	$T5_{ft}$	$BART_{fr}$	TREQS	Lma^*	Med^*	Lma^*_{ft}	Med^*_{ft}
$\mathbf{B}_{score}$	46.07	48.50	53.41	83.94	23.68	62.44	60.69	**84.79**	56.04
R-1	4.92	2.19	28.07	67.18	13.21	32.39	33.18	**74.47**	11.90

Retrieval								
Models	BM25	MonoB	MonoT5	TREQS	Lma^*	Med^*	Lma^*_{ft}	Med^*_{ft}
MAP	35.26	10.19	38.49	43.99	9.84	12.07	11.33	**44.34**
R	49.37	35.43	53.01	52.16	33.62	32.54	47.95	**53.38**

Table 5: Results for patient-related tasks using different SOTA models and best settings found for LLMs. All baselines were evaluated using (F^p =all_{avg}, ϕ =txt, Non-Guided). We report **best** scores per metric (row).

-9.5% between the best ICL setting and $k = 0$, except for one Llama setting (F^p =all_{avg}, ϕ =sgen, Guided), which shows an improvement of 2.17% MAP. These results suggest that LLMs struggle to reason over patient EHRs in *retrieval* tasks, even with demonstrations, whereas their performance on *extraction* tasks consistently improves, with a $+5.95\%$ B_{score} increase using query-based demonstrations. This aligns with our previous findings (§4.1) about LLMs' challenges in comprehending EHR structure and content for *retrieval* tasks.

Regarding the impact of the demonstration, selection functions σ, we can observe that query-based demonstrations (σ_q) obtain the highest scores in 9 out of 16 metrics, while patient-based demonstrations (σ_p) obtain lower performance than random demonstrations. We further analyze the optimal number of demonstrations, focusing only on query-based demonstrations (σ_q). We can observe that Llama benefits mainly from $k = 3$ demonstrations, while Meditron with $k = 1$. In general, we observe improvements up to 10.81% in B_{score} when finding the optimal number of demonstrations for the *extraction* task.

To better showcase this non-intuitive finding about the variability in the impact of ICL across tasks, we analyze selected examples of queries for each of the *extraction* and *retrieval* tasks based on the trends observed in Table 4. Figure 2 shows a pair of such examples. We can observe that for the *extraction* task, *query-based* demonstrations share relevant features (highlighted) between the question and the patient's EHR profile (*"admission time", "was admitted"*), guiding the LLM to find relevant information, particularly challenging for numerical information. In contrast, *random* demonstrations lean to focus on irrelevant features, leading to nonfactual information generation. For the *retrieval* task, we can observe that adding more demonstrations introduces more heterogeneous information into prompts (*"diagnosis heart valve", "diagnosed with anemia"*), and conditioning the binary answer to the example output (*"Output: No relevant"*), which appears to hinder the LLM's performance, whereas single clear EHR profiles obtain better relevance scores. Overall our results assess that ICL significantly improves performance on *extraction* tasks using query-based demonstration selection, but *retrieval* benefits more from zero-shot setups.

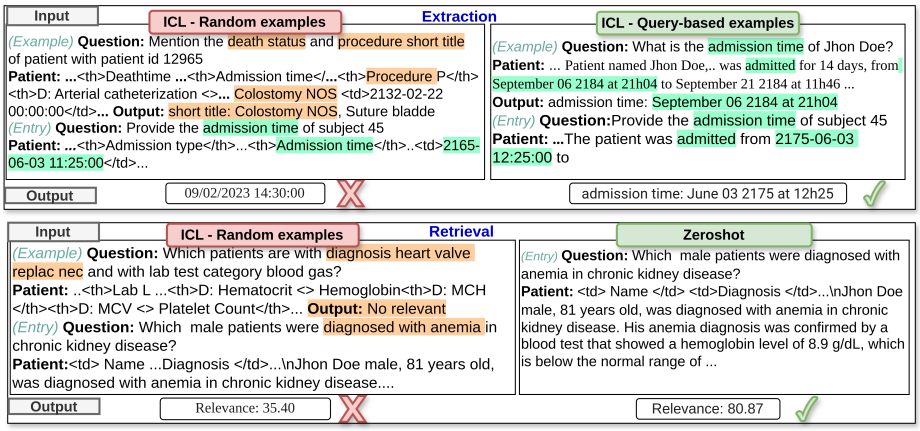

Fig. 2: (top) Example of random and query-based demonstrations in an ICL setup for *extraction*. (bottom) Example of ICL and zeroshot setup for *retrieval*. Highlighted the features (and values) referenced in demonstration and input.

5 Comparative evaluation and guidelines

To better support our final guidelines, we compare in Table 5, the optimal settings previously obtained, denoted Lma* and Med* for Llama and Meditron, with: 1) different state-of-the-art models per task and 2) their fine-tuned counterparts, Lma^*_{ft} and Med^*_{ft}, using the new MIMIC_{ask} and MIMIC_{search} datasets.

We observe that for both tasks, fine-tuned LLMs achieve the highest scores, demonstrating their adaptability to leverage EHR tabular data when using optimal feature selection and serialization techniques. Specifically, for the *extraction* task, we observe that explored LLMs perform better than all baselines, with the best score for Lma* with $\text{B}_{score} = 62.44$, except BART_{ft} with $\text{B}_{score} = 83.94$. Interestingly, Med^*_{ft} shows a performance decrease compared to the zero-shot counterpart Med*, highlighting the challenge of fine-tuning LLMs for complex tasks [31]. Additionally, we can see that the TREQS model achieves low $\text{B}_{score}(23.68)$, mainly due to frequent errors in retrieving specific columns from the EHRs. For the *retrieval* task, we note that studied LLMs only outperform MonoBERT, though BM25 and MonoT5 surpass them, emphasizing the difficulty of *retrieval* tasks for LLMs as shown in the literature [9]. Notably, only Med^*_{ft} exhibits a better task comprehension, outperforming other models. Interestingly, the TREQS performance trend is reversed compared to the *extraction* task, by achieving the second-best performance. This suggests model limitations to handle the sparsity of features to retrieve fine-grained clinical features in complex tabular data while being optimal to retrieve general coarse-grained conditions. In summary, our exploration provides the following guidelines about the use of LLMs for tabular EHR-related *extraction* and *retrieval* tasks:

1. Context is improved when using all available EHR features, leading to better task performance. If longitudinal values are present in the EHRs, the best performance is reached by feature value aggregation;
2. The best EHR serialization method is based on the LLM self-generated EHR tabular descriptions, particularly for zero-shot LLMs. Medical pre-trained LLMs can handle naive template-based serialization;
3. In an ICL setting, demonstration selection based on queries is more effective for extraction as the number of examples increases. Unlikely, the retrieval task better leverages zero-shot setups;
4. Finetuned LLMs with basic data-to-text EHR serialization methods achieve the best performance across tasks against fine-tuned pre-trained models.

6 Conclusion

In this paper, we explored prompt techniques of LLMs on tabular EHR extraction and retrieval. Our results showed a trend toward retrieval being more challenging than extraction. Our study has also shown that LLMs' performance on both tasks is particularly impacted by feature selection, serialization methods, and the quality of in-context demonstrations with significant levels of variations across tasks and backbone LLMs. Overall, our findings provide actionable insights for optimizing LLM performance on tasks involving tabular EHR data.
Our work has some limitations. We only consider two LLMs based on the open Llama model. Thus, the generalization of these guidelines to other model architectures with different sizes and training data remains unexplored. Moreover, for computational limitations, we restricted the exploration to the optimal settings identified, omitting possible causality relationships that other settings would have revealed. In future work, we will extrapolate our study to predictive EHR-related tasks and investigate to which extent extraction, retrieval, and prediction tasks can be used as control tasks for assessing data privacy protection in LLMs.

Acknowledgments. This work has been supported by the In-Utero project funded by HDH (France) and FRQS (Canada). This work was also granted access to the HPC resources of IDRIS under the allocation 2024-AD011015371 made by GENCI.

References

1. Chen, N., Shou, L., Gong, M., Pei, J., You, C., Chang, J., Jiang, D., Li, J.: Bridge the gap between language models and tabular understanding. arXiv preprint arXiv:2302.09302 (2023)
2. Chen, W.: Large language models are few(1)-shot table reasoners. In: Findings of EACL (2023)
3. Deng, X., Bashlovkina, V., Han, F., Baumgartner, S., Bendersky, M.: What do llms know about financial markets? a case study on reddit market sentiment analysis. In: Companion Proceedings of the ACM Web Conference 2023. p. 107–110. WWW '23 Companion, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3543873.3587324
4. Deng, X., Sun, H., Lees, A., Wu, Y., Yu, C.: Turl: Table understanding through representation learning. ACM SIGMOD Record **51**(1), 33–40 (2022)
5. Fang, X., Xu, W., Tan, F.A., Zhang, J., Hu, Z., Qi, Y., Nickleach, S., Socolinsky, D., Sengamedu, S., Faloutsos, C.: Large language models on tabular data–a survey. arXiv preprint arXiv:2402.17944 (2024)
6. Gong, H., Sun, Y., Feng, X., Qin, B., Bi, W., Liu, X., Liu, T.: TableGPT: Few-shot table-to-text generation with table structure reconstruction and content matching. In: Scott, D., Bel, N., Zong, C. (eds.) Proceedings of the 28th International Conference on Computational Linguistics. pp. 1978–1988. International Committee on Computational Linguistics, Barcelona, Spain (Online) (Dec 2020). https://doi.org/10.18653/v1/2020.coling-main.179
7. Haug, T., Ganea, O.E., Grnarova, P.: Neural multi-step reasoning for question answering on semi-structured tables. In: Pasi, G., Piwowarski, B., Azzopardi, L., Hanbury, A. (eds.) Advances in Information Retrieval. pp. 611–617. Springer International Publishing, Cham (2018)
8. Hegselmann, S., Buendia, A., Lang, H., Agrawal, M., Jiang, X., Sontag, D.A.: Tabllm: Few-shot classification of tabular data with large language models. In: AISTATG. vol. abs/2210.10723 (2022)
9. Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., Zhao, W.X.: Large language models are zero-shot rankers for recommender systems. In: European Conference on Information Retrieval. pp. 364–381 (2024)
10. Johnson, A., T., Pollard, T., Shen, L., al.: Mimic-iii, a freely accessible critical care database. Sci Data **3** (2016)
11. Lin, J., Ma, X., Lin, S.C., Yang, J.H., Pradeep, R., Nogueira, R.: Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In: Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021). pp. 2356–2362 (2021)
12. Lin, S.C., Asai, A., Li, M., Oguz, B., Lin, J., Mehdad, Y., Yih, W.t., Chen, X.: How to train your dragon: Diverse augmentation towards generalizable dense retrieval. In: Findings of the Association for Computational Linguistics: EMNLP 2023. pp. 6385–6400 (2023)
13. Lin, X.V., Chen, X., Chen, M., Shi, W., Lomeli, M., James, R., Rodriguez, P., Kahn, J., Szilvasy, G., Lewis, M., et al.: Ra-dit: Retrieval-augmented dual instruction tuning. In: The Twelfth International Conference on Learning Representations (2023)
14. Liu, S.C., Wang, S., Chang, T., Lin, W., Hsiung, C.W., Hsieh, Y.C., Cheng, Y.P., Luo, S.H., Zhang, J.: JarviX: A LLM no code platform for tabular data analysis and

optimization. In: Wang, M., Zitouni, I. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track. pp. 622–630. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.emnlp-industry.59

15. Liu, T., Wang, K., Sha, L., Chang, B., Sui, Z.: Table-to-text generation by structure-aware seq2seq learning. In: AAAI Conference on Artificial Intelligence (2017), https://api.semanticscholar.org/CorpusID:7672408
16. Lovon-Melgarejo, J., Ben-Haddi, T., Di Scala, J., Moreno, J.G., Tamine, L.: Revisiting the MIMIC-IV benchmark: Experiments using language models for electronic health records. In: Demner-Fushman, D., Ananiadou, S., Thompson, P., Ondov, B. (eds.) Proceedings of the First Workshop on Patient-Oriented Language Processing (CL4Health) @ LREC-COLING 2024. pp. 189–196. ELRA and ICCL, Torino, Italia (May 2024), https://aclanthology.org/2024.cl4health-1.23
17. Puduppully, R., Dong, L., Lapata, M.: Data-to-text generation with entity modeling. In: Korhonen, A., Traum, D., Màrquez, L. (eds.) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2023–2035. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1195
18. Rebuffel, C., Soulier, L., Scoutheeten, G., Gallinari, P.: A hierarchical model for data-to-text generation. In: Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42. pp. 65–80. Springer (2020)
19. Rubin, O., Herzig, J., Berant, J.: Learning to retrieve prompts for in-context learning. In: Carpuat, M., de Marneffe, M.C., Meza Ruiz, I.V. (eds.) Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 2655–2671. Association for Computational Linguistics, Seattle, United States (Jul 2022). https://doi.org/10.18653/v1/2022.naacl-main.191
20. Sarkar, S., Lausen, L.: Testing the limits of unified sequence to sequence llm pretraining on diverse table data tasks. In: NeurIPS 2023 Second Table Representation Learning Workshop (2023)
21. Singha, A., Cambronero, J., Gulwani, S., Le, V., Parnin, C.: Tabular representation, noisy operators, and impacts on table structure understanding tasks in llms. In: Table Representation Learning Workshop at NeurIPS 2023 (December 2023)
22. Slack, D., Singh, S.: Tablet: Learning from instructions for tabular data. arXiv preprint arXiv:2304.13188 (2023)
23. Steinberg, E., Jung, K., Fries, J.A., Corbin, C.K., Pfohl, S.R., Shah, N.H.: Language models are an effective representation learning technique for electronic health record data. Journal of Biomedical Informatics **113**, 103637 (2021). https://doi.org/https://doi.org/10.1016/j.jbi.2020.103637
24. Sui, Y., Zhou, M., Zhou, M., Han, S., Zhang, D.: Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining. p. 645–654. WSDM '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3616855.3635752
25. Sui, Y., Zou, J., Zhou, M., He, X., Du, L., Han, S., Zhang, D.: Tap4llm: Table provider on sampling, augmenting, and packing semi-structured data for large language model reasoning. In: Findings of the Association for Computational Linguistics: EMNLP 2024 (2024)

26. Trabelsi, M., Chen, Z., Zhang, S., Davison, B.D., Heflin, J.: Strubert: Structure-aware bert for table search and matching. In: Proceedings of the ACM Web Conference 2022. p. 442–451. WWW '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3485447.3511972
27. Wang, P., Shi, T., Reddy, C.K.: Text-to-sql generation for question answering on electronic medical records. In: Proceedings of The Web Conference 2020. p. 350–361. WWW '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3366423.3380120
28. Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N.A., Khashabi, D., Hajishirzi, H.: Self-instruct: Aligning language models with self-generated instructions. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 13484–13508. Association for Computational Linguistics, Toronto, Canada (Jul 2023). https://doi.org/10.18653/v1/2023.acl-long.754
29. Wang, Z., Dong, H., Jia, R., Li, J., Fu, Z., Han, S., Zhang, D.: Tuta: Tree-based transformers for generally structured table pre-training. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. p. 1780–1790. KDD '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3447548.3467434
30. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., hsin Chi, E.H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., Fedus, W.: Emergent abilities of large language models. ArXiv **abs/2206.07682** (2022)
31. Yang, H., Zhang, Y., Xu, J., Lu, H., Heng, P.A., Lam, W.: Unveiling the generalization power of fine-tuned large language models. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). pp. 884–899 (2024)
32. Ye, Y., Hui, B., Yang, M., Li, B., Huang, F., Li, Y.: Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 174–184. SIGIR '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3539618.3591708
33. Yu, W., Iter, D., Wang, S., Xu, Y., Ju, M., Sanyal, S., Zhu, C., Zeng, M., Jiang, M.: Generate rather than retrieve: Large language models are strong context generators. In: The Eleventh International Conference on Learning Representations (2023)
34. Zhang, Y., Feng, S., Tan, C.: Active example selection for in-context learning. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 9134–9148. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022). https://doi.org/10.18653/v1/2022.emnlp-main.622
35. Zhao, B., Ji, C., Zhang, Y., He, W., Wang, Y., Wang, Q., Feng, R., Zhang, X.: Large language models are complex table parsers. In: Conference on Empirical Methods in Natural Language Processing (2023)
36. Zhuang, S., Zhuang, H., Koopman, B., Zuccon, G.: A setwise approach for effective and highly efficient zero-shot ranking with large language models. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 38–47 (2024)