Overview of the EHRSQL 2024 Shared Task on Reliable Text-to-SQL Modeling on Electronic Health Records

Gyubok Lee Sunjun Kweon Seongsu Bae Edward Choi
KAIST AI
{gyubok.lee, sean0042, seongsu, edwardchoi}@kaist.ac.kr

Abstract

Electronic Health Records (EHRs) are relational databases that store the entire medical histories of patients within hospitals. They record numerous aspects of patients' medical care, from hospital admission and diagnosis to treatment and discharge. While EHRs are vital sources of clinical data, exploring them beyond a predefined set of queries requires skills in query languages like SQL. To make information retrieval more accessible, one strategy is to build a question-answering system, possibly leveraging text-to-SQL models that can automatically translate natural language questions into corresponding SQL queries and use these queries to retrieve the answers. The EHRSQL 2024 shared task aims to advance and promote research in developing a question-answering system for EHRs using text-to-SQL modeling, capable of reliably providing requested answers to various healthcare professionals to improve their clinical work processes and satisfy their needs. Among more than 100 participants who applied to the shared task, eight teams completed the entire shared task processes and demonstrated a wide range of methods to effectively solve this task. In this paper, we describe the task of reliable text-to-SQL modeling, the dataset, and the methods and results of the participants. We hope this shared task will spur further research and insights into developing reliable question-answering systems for EHRs.

1 Introduction

Electronic Health Records (EHRs) store all types of medical events that occur in the hospital, including hospital admissions, diagnoses, procedures, prescriptions, and discharges. They replace traditional paper-based records and provide a centralized repository for patient data. Over the years, the widespread adoption of EHRs in hospitals has been shown to improve patient care, increase efficiency, and enhance coordination among healthcare professionals (Upadhyay and Hu, 2022; Mullins et al., 2020; Uslu et al., 2021). Although EHRs are a valuable source of patient data, the complexity of their data structures and the need for specialized skills, such as query languages like SQL, to extract and analyze the information, often hinder their effective utilization by healthcare professionals (Wang et al., 2020; Lee et al., 2022). These barriers lead to the underutilization of the full potential of EHRs in clinical practice and research.

An alternative way to utilize data stored in EHRs is to develop a question-answering (QA) system. QA systems provide a user-friendly interface that allows healthcare professionals to ask questions in natural language and receive relevant answers from the EHR data, without needing to know query languages or EHR database strctures. Specifically, text-to-SQL modeling is an effective approach for building QA systems for EHRs, which are typically relational databases. These models automatically convert natural language questions into their corresponding SQL queries, and then execute these queries on the database to obtain the final answer. With the impressive advances in large language models (LLMs), various high-performance text-to-SQL models have been introduced, which are accomplished through model fine-tuning (Scholak et al., 2021) or LLM prompting with demonstrations (Pourreza and Rafiei, 2024; Gao et al., 2023; Chang and Fosler-Lussier, 2023). If deployed with reliable performance, these models could significantly benefit healthcare professionals by allowing them to explore patient data more freely from the EHRs through natural language interactions.

Several datasets on question-answering for EHRs have been introduced, including MIMIC-SQL (Wang et al., 2020), emrKBQA (Raghavan et al., 2021), and EHRSQL (Lee et al., 2022). EHRSQL, in particular, poses unique challenges. It is the first dataset to compile a collection of questions that reflect the diverse needs of healthcare professionals, including physicians, nurses, and

Proceedings of the 6th Clinical Natural Language Processing Workshop, pages 644–654
June 21, 2024 ©2024 Association for Computational Linguistics

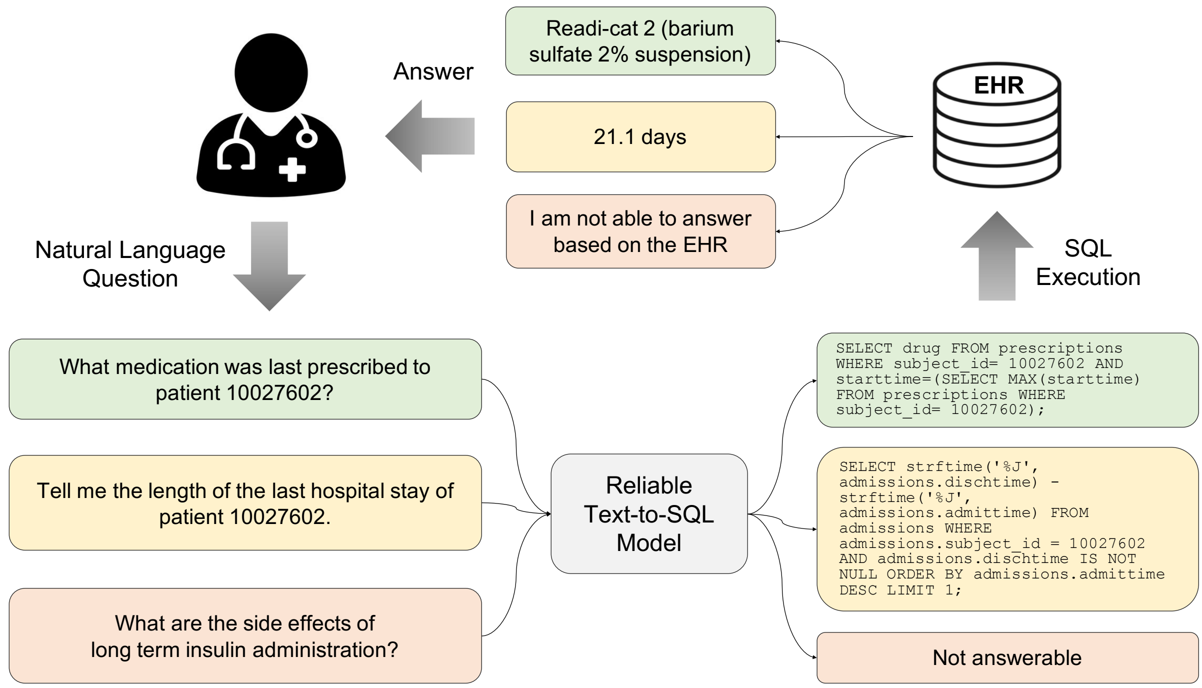

Figure 1: Overview of reliable text-to-SQL modeling on EHRs. For any input questions, a reliable text-to-SQL model should accurately predict SQL queries for what it can and abstain from what it cannot, such as for intrinsically unanswerable questions or ones that are likely incorrect by the model. Successfully developing such a model can serve as a valuable tool for healthcare professionals in hospitals for better accessibility of patient data and assistance of clinical decision-making.

hospital administrative staff. It contains extensive use of time expressions and includes SQL queries of increased complexity, which better reflect the real needs of a hospital setting. The SQL queries are linked to two open-source EHR databases[1], MIMIC-III (Johnson et al., 2016) and eICU (Pollard et al., 2018), retaining incompatible ones as unanswerable questions in the dataset (used to test a model's ability to abstain). Starting from their collected real-world questions, this shared task presents more up-to-date changes to the text-to-SQL modeling (use of MIMIC-IV and new paraphrases for questions) and more challenging problem settings (new data splitting and additional unanswerable questions). The dataset for this shared task is publicly available at `https://github.com/glee4810/ehrsql-2024`. The shared task platform is hosted on Codabench at `https://www.codabench.org/competitions/1889/`.

In this paper, we present the EHRSQL 2024 shared task and its dataset in Sections 2 and 3, respectively. Section 4 introduces the evaluation metric and baseline model for the task. Section 5 describes methods proposed by the participating teams and discusses interesting findings from the official results.

[1]SQL queries are database-dependent, meaning that even though a question attempts to retrieve the same information, the location of that information can vary across databases. For example, to list all drugs in MIMIC-III, you would use `SELECT drug FROM prescriptions`, whereas in eICU, it would be `SELECT drugname FROM medication`.

2 Task - Reliable Text-to-SQL Modeling

The goal of the EHRSQL 2024 shared task is to develop a reliable QA system for EHRs, specifically through text-to-SQL modeling. Reliability is crucial for the deployment of AI systems, especially in critical domains like hospitals, where incorrect predictions can have severe consequences. The term reliability in question answering refers to the system's preference for abstention over providing an incorrect answer (Whitehead et al., 2022; Chen et al., 2023; Lee et al., 2024). In this shared task, we adopt the definition of *reliable text-to-SQL* from TrustSQL (Lee et al., 2024), which first expands the scope of reliability to include unanswerable questions. A reliable text-to-SQL model should not only correctly generate SQL queries, providing utility, but also abstain from answering questions that are likely to be incorrect or are unanswerable, thereby

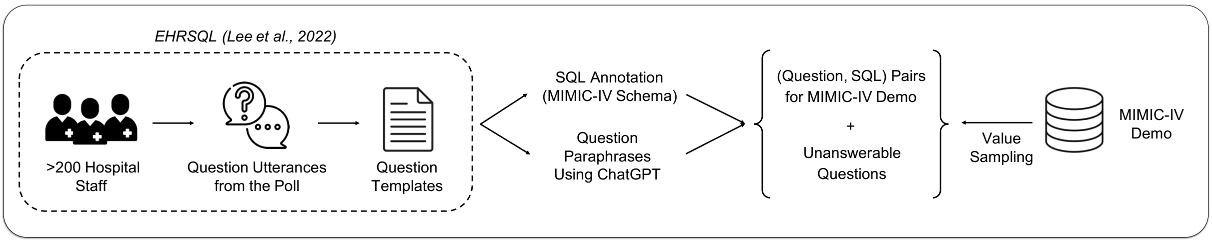

Figure 2: Data construction process of the EHRSQL shared task.

minimizing harm. This objective contrasts with most other text-to-SQL tasks, where the primary focus is to maximize SQL generation performance for answerable questions only. Further discussion on specific scenarios of measuring reliability for text-to-SQL is explained in Section 4.1.

3 Dataset Construction

In this section, we outline the key steps to generate data for the shared task. The overall data construction pipeline is illustrated in Figure 2. Each subsection provides a detailed explanation of each step.

3.1 Question Templates from EHRSQL

To construct the shared task data, we started from the pool of questions that reflect the real needs of diverse healthcare professionals in EHRSQL (Lee et al., 2022). This dataset is derived from the results of a poll participated in by more than 200 professionals at a university hospital in South Korea. The collected questions are those that the professionals would ask an AI speaker if it could access and synthesize structured information stored in EHRs (i.e., records in tabular form). The authors then translated the raw question utterances and removed duplicate ones to distill them into question templates. This shared task leverages the question templates collected in EHRSQL to generate diverse and realistic question-SQL pairs.

3.2 SQL Queries linked to MIMIC-IV Demo

Unlike the original EHRSQL dataset whose SQL queries are based on value-shuffled MIMIC-III and eICU[2], this shared task uses the demo version of MIMIC-IV[3] (Johnson et al., 2020) to construct question-SQL pairs. The demo version, containing records of 100 patients from the full MIMIC-IV database, has the same database schema as the full MIMIC-IV and is openly-available for anyone who is interested in using the dataset without special training[4]. Since the demo database schema is identical to the full database, the same query can be used to retrieve information from both the full and demo versions.

3.3 New Question Paraphrases

We found that the style and naturalness of paraphrases generated by current LLMs, like ChatGPT, surpass the paraphrases in EHRSQL, which are produced through both human and machine efforts. To improve the quality of the paraphrases for each question template, we employed ChatGPT to generate new paraphrases that are more natural and conversational. We then manually reviewed all new paraphrases to ensure they maintain the intended meaning of the original question templates.

3.4 Challenging Unanswerable Questions

A recent study revealed that unanswerable questions in the EHRSQL dataset can mostly be filtered out using a combination of N-gram and beam search score filtering (Yang et al., 2024). This is primarily because the unanswerable questions in EHRSQL were collected erroneously due to human errors during the polling process[5], resulting in limited diversity. To increase the difficulty of the task, we combined the original unanswerable questions with those from the EHRSQL portion of TrustSQL (Lee et al., 2024), which contains adversarially created unanswerable questions, such as those referring to non-existing columns and requests that exceed SQL functionalities.

[2]This process was done to further de-identify the question-SQL pairs for public release. Please refer to more detailed reasons in the original paper.

[3]https://physionet.org/content/mimic-iv-demo/2.2/

[4]The full MIMIC-IV dataset requires researchers to complete the Collaborative Institutional Training Initiative (CITI) training before accessing the data.

[5]The poll participants were initially provided with examples of inappropriate questions for the system, including those requiring external knowledge, ambiguous or qualitative statements, and questions about the reasons behind certain clinical decisions.

	Dev Phase		Test Phase
	Train	Valid	Test
Answerable question template	100 (100 seen)	134 (100 seen + 34 unseen)	134 (100 seen + 34 unseen)
Answerable samples	4674	931	934
Unanswerable samples	450	232	233
Total samples	5124	1163	1167

Table 1: Data statistics for the shared task. All text-to-SQL data used in the shared task is based on MIMIC-IV.

3.5 New Data Split

In real-world scenarios, text-to-SQL models can encounter questions that are answerable based on the EHR schema but have not been seen in the training set (unseen SQL with respect to the training set). This situation can lead to increased confusion for the model in distinguishing between answerable and unanswerable questions. Unlike the original EHRSQL, where answerable questions were split in an identically distributed (IID) manner, we split the shared task data to include both seen and unseen question templates (or SQL structures) in the validation and test sets. For unanswerable questions, the original unanswerable questions from EHRSQL were distributed across all splits (training, validation, and test), while new unanswerable questions were added exclusively to the validation and test sets to increase the task's difficulty. Each of these splits has a 20% proportion of unanswerable questions. Table 1 shows the number of question templates and the size of each data split[6]. The training and validation sets were made available during the development phase (Jan 29, 2024 - Mar 26, 2024), and the test set was made available for the three-day test phase (Mar 26, 2024 - Mar 28, 2024).

[6]Even if the MIMIC-IV demo includes only 100 patients, a wide variety of question templates can exist. Consider patient ID 100 and two question templates: 'What is patient 100's gender?' and 'What is patient 100's last blood pressure?' The data splitting in text-to-SQL for EHRs does not have to be done by patient, such as 'What is patient 100's gender?' in the training set and 'Tell me patient 200's sex?' in the validation set, because the task could become relatively easy. Instead, it might include 'What is patient 100's gender?' in the training set and 'What is patient 100's last blood pressure?' in the validation set. A more challenging and realistic goal of text-to-SQL is to assess how well the model can generate SQL queries for both question templates (or SQL structures) that it has seen and those it has not seen. In this example, we show four question samples with two question templates.

4 Evaluation

4.1 Evaluation Metric

We chose the evaluation metric that best aligns with the purpose of our shared task: to build reliable text-to-SQL models aimed at accurately predicting correct SQL queries and identifying unanswerable questions, while minimizing incorrect SQL predictions and the wrongly classifying unanswerable questions as answerable. More concretely, we adopt the Reliability Score (RS) for reliable text-to-SQL (Lee et al., 2024), formally written as follows:

$$RS(c)(x) = \begin{cases} 1 & \text{if } x \in \mathcal{Q}_{ans}; g(x) = 1; Acc(x) = 1, \\ 0 & \text{if } x \in \mathcal{Q}_{ans}; g(x) = 0, \\ -c & \text{if } x \in \mathcal{Q}_{ans}; g(x) = 1; Acc(x) = 0, \\ -c & \text{if } x \in \mathcal{Q}_{una}; g(x) = 1, \\ 1 & \text{if } x \in \mathcal{Q}_{una}; g(x) = 0, \end{cases} \tag{1}$$

where $\mathcal{Q}_{ans}$ and $\mathcal{Q}_{una}$ denote answerable and unanswerable questions, respectively. g(x) = 1 implies that the model selects its SQL generation as the final answer, whereas g(x) = 0 implies that the model abstains. Acc(x) indicates the accuracy of the generated SQL, based on execution accuracy, which is determined by whether the answers returned by the ground-truth and predicted SQL queries match.

The RS has five different cases for assigning the score:

- A score of 1 is assigned if SQL is correctly generated by the model for answerable questions.
- A score of 0 is assigned if the model abstains from generating SQL for answerable questions.
- A score of $-c$ is assigned if the model predicts incorrect SQL for answerable questions.
- A score of $-c$ is assigned if the model attempts to predict SQL for unanswerable questions.
- A score of 1 is assigned if the model accurately detects unanswerable questions by abstaining.

The overall RS is calculated by taking the average of sample-level scores, represented in percentages. The penalty of c is chosen depending on the reliability requirements of the model. A penalty of 0 (RS_0) means no punishment for incorrect

	Team	Affiliation	Paper	Code
1	LG AI Research & KAIST	LG AI Research & KAIST, South Korea	Jo et al. (2024)	[1]
2	PromptMind	-	Gundabathula and Kolar (2024)	[2]
3	ProbGate	KAIST, South Korea	Kim et al. (2024b)	[3]
4	KU-DMIS	Korea university, South Korea	Kim et al. (2024a)	[4]
5	AIRI NLP	AIRI, Russia	Somov et al. (2024)	[5]
6	LTRC-IIITH	IIIT Hyderabad, India	Thomas et al. (2024)	[6]
7	Saama Technologies	Saama Technologies, USA	Jabir et al. (2024)	[7]
8	TEAM_optimist	SUST, Bangladesh	Joy et al. (2024)	[8]

[1] https://github.com/sylee0520/ehrsql-2024 (private)
[2] https://github.com/satyakesav/ehrsql-clinicalnlp-2024 (private)
[3] https://github.com/venzino-han/probgate_ehrsql
[4] https://github.com/Chanwhistle/EHRSQL_NACCL
[5] https://github.com/runnerup96/EHRSQL-text2sql-solution
[6] https://github.com/jr-john/ehrsql_2024 (private)
[7] https://github.com/upjabir/ehrsql_2024
[7] https://github.com/joy-2019331037/nlpConference

Table 2: Participating teams, affiliation, paper, and code.

predictions, a penalty of 10 (RS_10) represents a moderately rigorous scenario, and a penalty of N (RS_N), where N refers to the evaluation data size, is the most rigorous scenario in which even a single mistake outweighs all correct predictions and abstentions. The maximum possible RS is 100%, and the minimum possible scores vary depending on the penalties: 0 for $c = 0$; -1000% for $c = 10$; $-100N\%$ for $c = N$. The main evaluation metric for the shared task is RS(10), where every ten accurate predictions weigh the same as one incorrect prediction.

4.2 Code Verification and Fact Sheet

The participants shared their code and the fact sheet following the instructions reported in Appendix A. The purpose of the fact sheet was to collect a brief summary of participants' methods, including any use of pre-trained models or external data. For code verification, participants had the option to submit their code either via email or through GitHub repositories. These repositories could be public or private, as long as access was granted to the task organizers. Upon receiving the submissions, we conducted a careful review to ensure that the provided code and the methods described in the fact sheets are consistent.

4.3 Baseline Model

For the baseline, we employ the simplest method, denoted as ABSTAIN-ALL, which abstains from answering all questions. Evaluating in the RS, abstaining from all questions results in an overall score of 20%. This score is earned by correctly abstaining from answering unanswerable questions. This is not a trivial score, particularly as the penalty for incorrect predictions increases, which can severely harm the overall score.

5 Official Results

5.1 Participating Teams

The EHRSQL shared task attracted over 100 participants from both academia and industry. Of these, 8 teams submitted their code and fact sheet. Table 2 lists the participating teams, their affiliations, the code submission status (not all of which is publicly available), and their working papers.

5.2 Methods and Results

Table 3 presents the official results for each team, along with short descriptions of their methods. The proposed methods can be categorized into two types: unified and pipeline-based approaches ('Modeling Type' in Table 3). The unified approach leverages LLMs to perform both SQL generation and abstention, while the pipeline-based approach

	Team	RS_0	RS_10	RS_N	Modeling Type	Ensemble	Fine-tuned	Model Used
1	LG AI Research & KAIST	88.17	81.32	-711.83	Unified	Yes	No	ChatGPT
2	PromptMind	82.6	74.89	-817.4	Unified	Yes	Yes	GPT-4, ChatGPT, Claude Opus
3	ProbGate	81.92	74.21	-818.08	Unified	No	Yes	ChatGPT
4	KU-DMIS	72.07	59.21	-1427.93	Unified	No	Yes	ChatGPT
5	AIRI NLP	68.89	44.04	-2831.11	Pipeline	No	Yes	T5-3B, Logistic Regression
6	LTRC-IIITH	66.84	43.7	-2633.16	Pipeline	No	Yes	SQLCoder-7b-2
7	Saama Technologies	53.21	36.08	-1946.79	Pipeline	Yes	Yes	Decision Trees, CodeLlama-7b, ChatGPT
8	TEAM_optimist	14.14	-713.37	-84.9K	Unified	No	No	SQLCoder-7b-2
-	ABSTAIN-ALL	20.0	20.0	20.0	No	No	No	-

Table 3: Official results. ABSTAIN-ALL is the baseline for the shared task, explained in Section 4.3. ‘Ensemble’ denotes the use of any ensemble methods. ‘Fine-tuned’ indicates whether any pre-trained models were further trained for SQL generation or abstention purposes. ‘Pipeline-based’ means the use of multiple methods in a sequence, such as a pipeline that consists of an answerability detector, an SQL generator, and subsequently an SQL error detector.

involves building a series of specialized, smaller models, such as SQLCoder or T5-3B, to ensure reliability as one system. The overall observation is that 1) methods that fall under the unified approach tend to outperform those in the pipeline-based approaches; 2) most teams chose to fine-tune LLMs on the training data, either general-purpose (e.g., ChatGPT) or code-specialized models (e.g., CodeLLama), highlighting the importance of domain-specific fine-tuning for adapting LLMs to this task; 3) teams with smaller discrepancies between the RS with different penalties (e.g., the gap between RS(0) and RS(10)) tend to rank higher, indicating that minimizing incorrect SQL predictions through effective abstention mechanisms is crucial for this task. Detailed discussions of each method by category are provided in the following paragraphs.

Unified approach. Five teams utilized methods under the unified approach. The LG AI Research & KAIST team achieved the best results, scoring 81.32 in RS(10) by using self-training LLMs (Amini et al., 2022; Yuan et al., 2024) with pseudo-labeling for unanswerable questions. The PromptMind team implemented an ensemble of LLMs, including fine-tuned ChatGPT, GPT-4, and Claude Opus. They selected SQL generation as the final prediction only if all three models unanimously agreed; otherwise, they would abstain. For SQL generation, they employed two retrievers (one for the general domain and another for the medical domain) to retrieve similar question-SQL pairs from the training set. The ProbGate team employed fine-tuned ChatGPT with log-probability thresholding and error handling for abstention, where the threshold was set heuristically based on the ratio of unanswerable questions in the validation set. The KU-DMIS team took a two-stage method. First, they generated question-SQL pairs to align the test set distribution with the training set using question templates from the original EHRSQL. Then, they fine-tuned ChatGPT on this newly generated dataset. Abstention was achieved by sampling multiple SQL predictions for each input question and abstaining if the outputs were not consistent. Lastly, the TEAM_optimist team used SQLCoder-7b-2 for direct generation of SQL and abstention labels (null) through in-context learning.

Pipeline-based approach. Alternatively, three teams adopted the pipeline-based approach. The AIRI NLP team used a two-stage method: initially using logistic regression to detect unanswerable questions, then generating SQL with a fine-tuned T5-3B (Raffel et al., 2020), and finally checking the executability of the generated SQL for final abstention. The LTRC-IIITH team used two different SQLCoder-7b-2 models, one for detecting unanswerable questions and the other for SQL generation. For final abstention, they utilized the log-probabilities from the SQL generator to detect potential errors in SQL generation, followed by an executability check of the SQL. The Saama Technologies team began with an ensemble of unanswerable question detectors, including multinomial naive bayes, SGD classifier, CatBoost (Prokhorenkova et al., 2018), and CodeLlama-7b (Roziere et al., 2023). They then generated SQL using CodeLlama-7b, and finally used a ChatGPT-based answer selector for final abstention.

6 Conclusion

With the increasing volume of data stored in EHRs and the impressive advances in LLMs, the EHRSQL 2024 shared task offered an opportunity to develop and test participants' creative methods to building reliable QA systems on EHRs using text-to-SQL modeling. The dataset for this shared task presents unique challenges, including questions that extensively use time expressions and the increased complexity of SQL queries, which more accurately reflect the real needs of a hospital setting. It also includes challenging unanswerable questions that should be avoided. This distinguishes the task from most other text-to-SQL challenges, as reliable text-to-SQL models must not only generate correct SQL queries, providing utility, but also abstain from answering questions that are likely incorrect or unanswerable, thereby minimizing harm.

The shared task attracted over 100 participants from academia and industry, with 8 teams ultimately submitting their code and fact sheets. As a novel task at the intersection of the NLP and clinical domains, it inspired a variety of proposed methods. These included self-training LLMs through pseudo-labeling, ensembling of different LLMs, generating synthetic question-SQL pairs to handle distribution shifts from training to test sets, leveraging log-probabilities for abstention, and pipeline-based approaches with specialized models for correct SQL generation and abstention. We hope that this shared task, emphasizing reliability, will encourage further research into building QA systems for EHRs that can truly serve as valuable tools for healthcare professionals in hospitals, improving clinical decision-making, facilitating research, and enhancing patient care quality. Future research directions include expanding reliable question answering for EHRs to multimodal settings by incorporating clinical notes, X-ray images, and ECG signals.

Limitations

This shared task does not represent all types of answerable and unanswerable questions encountered in hospital settings. Additionally, this shared task employs MIMIC-IV as the EHR database, which is not a universally accepted EHR schema, and the databases are preprocessed for the QA task by eliminating duplicate values across different tables to reduce ambiguity. Lastly, further experiments are necessary for newly proposed LLMs, since most methods, including text-to-SQL generation and abstention, depend heavily on the underlying LLMs.

Acknowledgments

We would like to thank Asma Ben Abacha from Microsoft Health AI and the ClinicalNLP organizers for their feedback and support for the EHRSQL 2024 shared task. We also thank all the participating teams who contributed to the success of this shared task by proposing their interesting methods and showing strong engagement. This work was supported by the Institute for Information communications Technology Promotion(IITP) grant (No.2019-0-00075) and the National Research Foundation of Korea (NRF) grant (NRF-2020H1D3A2A03100945) funded by the Korea government (MSIT).

References

Massih-Reza Amini, Vasilii Feofanov, Loic Pauletto, Emilie Devijver, and Yury Maximov. 2022. Self-training: A survey. *arXiv preprint arXiv:2202.12040*.

Shuaichen Chang and Eric Fosler-Lussier. 2023. How to prompt llms for text-to-sql: A study in zero-shot, single-domain, and cross-domain settings. In *NeurIPS 2023 Second Table Representation Learning Workshop*.

Jiefeng Chen, Jinsung Yoon, Sayna Ebrahimi, Sercan Arik, Tomas Pfister, and Somesh Jha. 2023. Adaptation with self-evaluation to improve selective prediction in llms. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5190–5213.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. 2023. Text-to-sql empowered by large language models: A benchmark evaluation. *arXiv preprint arXiv:2308.15363*.

Satya Gundabathula and Sriram Kolar. 2024. Promptmind team at ehrsql-2024: Improving reliability of sql generation using ensemble llms. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, Mexico City, Mexico. Association for Computational Linguistics.

Mohammed Jabir, Kamal Kanakarajan, and Malaikannan Sankarasubbu. 2024. Saama technologies at ehrsql 2024: Sql generation through classification answer selector by llm. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, Mexico City, Mexico. Association for Computational Linguistics.

Yongrae Jo, Seongyun Lee, and Minju Seo. 2024. Lg ai research kaist at ehrsql 2024: Self-training large language models with pseudo-labeled unanswerable questions for a reliable text-to-sql system on ehrs. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, Mexico City, Mexico. Association for Computational Linguistics.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. 2020. Mimic-iv. *PhysioNet. Available online at: https://physionet. org/content/mimiciv/1.0/(accessed August 23, 2021)*, pages 49–55.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Sourav Bhowmik Joy, Rohan Redwan, Argha Pratim Saha, Minhaj Ahmed, Utsho Das, and Partha Sarothi Bhowmik. 2024. Team optimist at ehrsql 2024: Text-to-sql generation using large language model for ehr analysis. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, Mexico City, Mexico. Association for Computational Linguistics.

Chanhwi Kim, Hajung Kim, Hoonick Lee, Jiwoo Lee, Kyochul Jang, Kyungjae Lee, Gangwoo Kim, and Jaewoo Kang. 2024a. Ku-dmis: Generating sql query via question templatization in ehr. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, Mexico City, Mexico. Association for Computational Linguistics.

Sangryul Kim, Donghee Han, and Sehyun Kim. 2024b. Probgate at ehrsql 2024: Enhancing sql query generation accuracy through probabilistic threshold filtering and error handling. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, Mexico City, Mexico. Association for Computational Linguistics.

Gyubok Lee, Woosog Chay, Seonhee Cho, and Edward Choi. 2024. Trustsql: A reliability benchmark for text-to-sql models with diverse unanswerable questions. *arXiv preprint arXiv:2403.15879*.

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. *Advances in Neural Information Processing Systems*, 35:15589–15601.

Alexandra Mullins, Renee O'Donnell, Mariam Mousa, David Rankin, Michael Ben-Meir, Christopher Boyd-Skinner, and Helen Skouteris. 2020. Health outcomes and healthcare efficiencies associated with the use of electronic health records in hospital emergency departments: a systematic review. *Journal of Medical Systems*, 44(12):200.

Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. 2018. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):1–13.

Mohammadreza Pourreza and Davood Rafiei. 2024. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. *Advances in Neural Information Processing Systems*, 36.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Preethi Raghavan, Jennifer J Liang, Diwakar Mahajan, Rachita Chandra, and Peter Szolovits. 2021. emrKBQA: A clinical knowledge-base question answering dataset. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 64–73, Online. Association for Computational Linguistics.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. Picard: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901.

Oleg Somov, Elena Tutubalina, and Alexei Dontsov. 2024. Airi nlp team at ehrsql 2024: T5 and logistic regression to the rescue. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, Mexico City, Mexico. Association for Computational Linguistics.

Jerrin John Thomas, Pruthwik Mishra, Dipti Sharma, and Parameswari Krishnamurthy. 2024. Ltrc-iiith at ehrsql 2024: Enhancing reliability of text-to-sql systems through abstention and confidence thresholding.

In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, Mexico City, Mexico. Association for Computational Linguistics.

Soumya Upadhyay and Han-fen Hu. 2022. A qualitative analysis of the impact of electronic health records (ehr) on healthcare quality and safety: Clinicians' lived experiences. *Health Services Insights*, 15:11786329211070722.

Aykut Uslu, Jürgen Stausberg, et al. 2021. Value of the electronic medical record for hospital care: update from the literature. *Journal of medical Internet research*, 23(12):e26323.

Ping Wang, Tian Shi, and Chandan K Reddy. 2020. Text-to-sql generation for question answering on electronic medical records. In *Proceedings of The Web Conference 2020*, pages 350–361.

Spencer Whitehead, Suzanne Petryk, Vedaad Shakib, Joseph Gonzalez, Trevor Darrell, Anna Rohrbach, and Marcus Rohrbach. 2022. Reliable visual question answering: Abstain rather than answer incorrectly. In *European Conference on Computer Vision*, pages 148–166. Springer.

Yongjin Yang, Sihyeon Kim, SangMook Kim, Gyubok Lee, Se-Young Yun, and Edward Choi. 2024. Towards unbiased evaluation of detecting unanswerable questions in EHRSQL. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

A Code Submission and Fact Sheet Template

Fact Sheet for EHRSQL-2024 Shared Task:

I. Team name

- Username on Codalab:
- Team leader affiliation:
- Team leader email:
- Name of other team members (and affiliation):
- Team website URL (if any):

II. Contribution

- **Title of the contribution**
  - Provide a brief summary of the method and contributions.
- **Representative image / workflow diagram of the method**
  - An image (or several images) to support method description to better understand the approach and model pipeline. You can refer to these images in the method description part.
- **Detailed method description**
  - Provide a technical and detailed description of the method and contributions. The explanations must be self-contained and one must be able to reproduce the approach by reading this section.
- **Shared task results**
  - RS_0:
  - RS_5:
  - RS_{10}:
  - RS_N:
- **Final Remarks**
  - Please identify the pros and cons (if any) of the proposed approach.

III. Additional method details

- Did you use any pre-trained model?
- Did you use external data?
- Did you perform any data augmentation?
- At the test phase, did you use the provided validation set as part of your training set?
- Did you use any regularization strategies/terms?

- Did you use handcrafted features?
- Did you use any domain adaptation strategy?

IV. Code Repository

- Link to a code repository with complete and detailed instructions so that the results obtained on Codabench can be reproduced.
- If private repo, share the repo with `glee4810`