Leveraging LLM-Generated Schema Descriptions for Unanswerable Question Detection in Clinical Data

Anonymous COLING 2025 submission

Abstract

Recent advancements in large language models (LLMs) have boosted research on generating SQL queries from domain-specific questions, particularly in the medical domain. A key challenge is detecting and filtering unanswerable questions. Existing methods often relying on model uncertainty, but these require extra resources and lack interpretability. We propose a lightweight model that predicts relevant database schemas to detect unanswerable questions, enhancing interpretability and addressing the data imbalance in binary classification tasks. Furthermore, we found that LLM-generated schema descriptions can significantly enhance the prediction accuracy. Our method provides a resource-efficient solution for unanswerable question detection in domain-specific question answering systems. The source code is available at https://anonymous.4open.science/r/sqd2024

1 Introduction

Developments in large-scale language models (LLMs) have enabled their application across diverse domains, achieving high performance in tasks like text summarization, question answering, and generating SQL queries for data extraction from relational databases(Li et al., 2023; Liu et al., 2023; Koreeda et al., 2023). LLMs excel across various tasks, often surpassing traditional methods, but the challenge of hallucination and inaccurate outputs persists, prompting ongoing research to enhance their reliability(Chen et al., 2023; Arabzadeh et al., 2022).

Recently, numerous studies have utilized electronic health record (EHR) data (Nayebi Kerdabadi et al., 2023; Shi et al., 2024) and text-to-SQL research has been studied for efficiently querying a patient's data stored in relational databases (Lee et al., 2022; Kim et al., 2024). In healthcare, identifying unanswerable questions is crucial to prevent serious consequences and ensure the accuracy and reliability of LLM-generated answers(Lee et al., 2024a). However, filtering unanswerable questions in healthcare domain is challenging due to data imbalance, which refers to an uneven distribution of observations between label classes. In the public healthcare dataset, the training data was imbalanced with less than 10% unanswerable questions, making unanswerable question detection challenging(Jo et al., 2024). To address this issue, prior methods have been developed to train text-to-SQL models and utilize uncertainty in the generation process (Lee et al., 2022; Wang et al., 2023). These approaches depend heavily on model performance and struggle to interpret why questions are unanswerable. Text-to-SQL models based on LLMs are also costly and have inconsistent filtering effectiveness. Thus, there's a need for a method that explicitly trained in a supervised manner and works independently of text-to-SQL models.

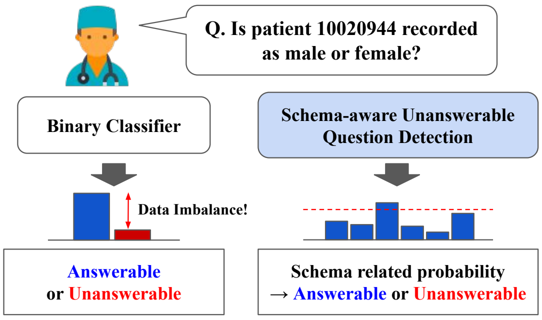
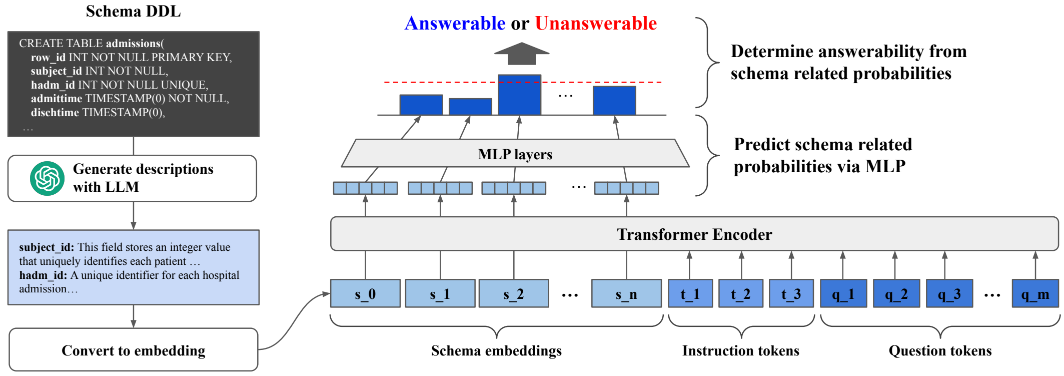

Q. Is patient 10020944 recorded as male or female?
Binary Classifier
Schema-aware Unanswerable Question Detection
Data Imbalance!
Answerable or Unanswerable
Schema related probability → Answerable or Unanswerable

Figure 1: Comparison of binary classifiers and schema-aware unanswerable question detection

To overcome the aforementioned limitations, we propose **S**chema Aware Unanswerable **Q**uestion **D**etection (SQD) that leverages database schema to identify unanswerable questions, without relying on fine-tuned text-to-SQL models. Our approach mitigates the data imbalance problem by predicting the schema related to the question, rather than directly predicting the unanswerability of the



```
CREATE TABLE admissions(
    row_id INT NOT NULL PRIMARY KEY,
    subject_id INT NOT NULL,
    hadm_id INT NOT NULL UNIQUE,
    admittime TIMESTAMP(0) NOT NULL,
    dischtime TIMESTAMP(0),
 …
```

subject_id: This field stores an integer value that uniquely identifies each patient …
hadm_id: A unique identifier for each hospital admission…

Schema DDL
Generate descriptions with LLM
Convert to embedding
Answerable or Unanswerable
Determine answerability from schema related probabilities
MLP layers
Predict schema related probabilities via MLP
Transformer Encoder
s_0
s_1
s_2
s_n
t_1
t_2
t_3
q_1
q_2
q_3
q_m
Schema embeddings
Instruction tokens
Question tokens

Figure 2: The overall architecture of our proposed Schema Aware Unanswerable Question Detection

question. Figure 1 illustrates the distinction between our proposed method and a binary classification approach that directly predicts unanswerable questions. Binary classification methods are plagued by data imbalance, as the percentage of unanswerable data is very small. Conversely, our schema prediction-based method addresses this issue by making predictions across multiple schemas. Key features of our method include: (1) Predicting question-related schemas by fine-tuning pre-trained Transformer encoders. (2) Utilizing LLM-generated descriptions instead of schema Data Definition Language (DDL). Our proposed method outperforms existing approaches on clinical datasets and offers several advantages: (a) it does not require text-to-SQL training and (b) it provides explainability by demonstrating the relevance to the schema, which can guide users in real-world applications.

Our contributions in this work are as follows:

- We propose a Transformer encoder-based method that leverages question-schema relationships to effectively predict answerability in EHR relational databases.

- Our approach achieves higher performance by utilizing schema descriptions generated by the LLM, rather than directly using the DDL that defines the schema.

- Through multiple experiments, we demonstrate that our proposed method can effectively identify unanswerable questions and mitigate the data imbalance problem.

2 Method

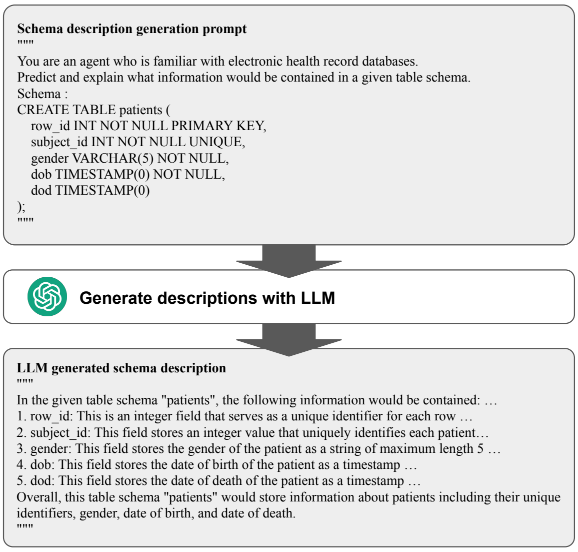
Figure 2 illustrates the overall framework of Schema Aware Unanswerable Question Detection (SQD). The proposed method consists of three main stages: (1) schema description generation and embedding extraction, (2) Transformer encoder-based question-schema related probability prediction, and (3) final answerability determination.

2.1 Schema description generation via LLM

The first step of our proposed method is utilizing LLMs to generate descriptions from Schema DDLs. Each Schema DDL provides data types and column names for defining database tables, but inferring specific column details remains difficult. Due to this problem, performance was not improved in previous studies using schema information in text-to-SQL(Lee et al., 2022). We attribute this primarily to the fact that pre-trained models are trained on plain text have a hard time inferring the contents of a table from DDL alone. For instance, the *subject_id* column in the *admissions* table could be interpreted in various ways without clear context. Therefore, we input each DDL into the LLM along with contextual prompts related to the EHR data to generate comprehensive descriptions. Figure 3 shows an example of schema description generation. LLMs can elucidate the meanings of abbreviations such as *dob* and *dod*, which are typically difficult to interpret. By providing contextual information, the LLM can generate more specific and useful descriptions for these columns. We use a pretrained encoder, such as T5, to convert the generated schema descriptions into embeddings, and then apply average pooling to obtain embeddings that encapsulate the semantic information for each schema.

2.2 Encoder-based question-schema related probability prediction

To determine the answerability, we input the schema embeddings generated in the previous step,



Schema description generation prompt

```
"""
You are an agent who is familiar with electronic health record databases.
Predict and explain what information would be contained in a given table schema.
Schema :
CREATE TABLE patients (
   row_id INT NOT NULL PRIMARY KEY,
   subject_id INT NOT NULL UNIQUE,
   gender VARCHAR(5) NOT NULL,
   dob TIMESTAMP(0) NOT NULL,
   dod TIMESTAMP(0)
);
"""
```

LLM generated schema description

```
"""
In the given table schema "patients", the following information would be contained: …
1. row_id: This is an integer field that serves as a unique identifier for each row …
2. subject_id: This field stores an integer value that uniquely identifies each patient…
3. gender: This field stores the gender of the patient as a string of maximum length 5 …
4. dob: This field stores the date of birth of the patient as a timestamp …
5. dod: This field stores the date of death of the patient as a timestamp …
Overall, this table schema "patients" would store information about patients including their unique
identifiers, gender, date of birth, and date of death.
"""
```

Figure 3: Schema description generation via LLM

along with the target question and instruction tokens, into another pre-trained Transformer encoder. As illustrated in Figure 2, each schema embedding, through attention mechanisms, learns its relationship with the question. The encoder outputs are used to predict the relevance of each schema. We use pretrained *t5-small* encoder in our experiments(Raffel et al., 2020).

Our method can be described as follows: let s_i be the embedding of the i-th schema, t_i be the i-th instruction token, and q_i be the i-th token of the target question. Input sequence is constructed as :

$$\text{Seq} = [s_1, s_2, \ldots, s_n,\ t_1, t_2, t_3,\ q_1, q_2, \ldots, q_m].$$

The Transformer encoder processes this sequence and outputs a embedding h_i for each schema s_i:

$$h_i = \text{Enc}(\text{Seq})_i,$$

where $\text{Enc}(\cdot)$ represents the Transformer encoder, and $(\cdot)_i$ indicates the output corresponding to the i-th output embedding. These embeddings h_i are then used to predict the relevance of each schema to the target question.

Each schema embedding is processed by multi-layer perceptron (MLP) layers, and the model is trained using mean squared error (MSE) loss. A schema is labeled as 0 if it is not utilized by the SQL query to answer the given question and as 1 if it is utilized. Consequently, unanswerable questions are labeled as 0 for all schemas. Mathematically, our method can be described as follows:
For a given question, the schema is labeled as 0 if it is not related with the question and as 1 if it is related. For unanswerable questions, all schemas are labeled as 0:

$$y_i = \begin{cases} 1 & \text{if schema } i \text{ is related to the question,} \\ 0 & \text{if schema } i \text{ is not related to the question.} \end{cases}$$

For unanswerable questions, all schema labels are 0:

$$y_i = 0 \quad \forall i$$

Let $MLP_i(\cdot)$ be the multi-layer perceptron layer for each i-th schema. The prediction for each schema $\hat{y}_i$ is given by:

$$\hat{y}_i = MLP_i(h_i)$$

The model is trained using mean squared error (MSE) loss. Given the true label y_i for each schema, the MSE loss $\mathcal{L}$ is calculated as:

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

2.3 Answerability determination

Finally, the answerability of a question is determined through schema relevance probabilities. A question is considered answerable if the highest probability among all schema relevance probabilities exceeds a predefined threshold. The threshold is chosen to maximize the F1 score on the validation set. If the maximum relevance probability exceeds the threshold, the question is deemed answerable. Otherwise, the question is considered unanswerable:

$$\text{Answerable} = \begin{cases} \text{Yes} & \text{if } \max_i(\hat{y}_i) > \text{threshold} \\ \text{No} & \text{if } \max_i(\hat{y}_i) \leq \text{threshold} \end{cases}$$

3 Experiment

In this section, we describe the experimental environment and results. The proposed method is compared with other models to answer the following research questions.

RQ1 Can the proposed method improve unanswerable question detection performance over existing baselines?

RQ2 How do the combinations of special tokens affect performance?

RQ3 How does the LLM generated schema descriptions impact the performance of the proposed method?

3.1 Experimental Settings

This section describes the major details of the experimental setup. However, we have not included all the details due to lack of space, and the detailed experimental setup and data are publicly available in our online repository [1].

Dataset We conducted an experiment using the EHRSQL-2024 dataset[2] (Lee et al., 2022, 2024b,a), a representative medical text-to-SQL dataset that includes unanswerable questions often overlooked in existing datasets. To the best of our knowledge, this is the only publicly available text-to-SQL dataset in the healthcare domain that includes unanswered questions.

Evaluation Metrics We adopted binary classification evaluation metrics, specifically F1 score and AUC, to address label imbalance effectively. Additionally, to provide a comprehensive comparison of model characteristics, we included Accuracy, Recall, and Precision metrics. In our evaluation, we treated the *unanswerable* case as *True* and the *answerable* case as *False*, allowing us to calculate these metrics accurately.

Baselines We compared our method to (1)latest LLMs using a zero-shot approach with DDL and target questions and (2) fine-tuned T5-based binary classifiers, as well as to methods proposed in previous works(Lee et al., 2022; Kim et al., 2024). To address data imbalance, we included a version of the T5 binary classifier trained with evenly sampled data. All binary classifiers utilized the same *t5-small* backbone as our model.

3.2 Experiment Results

Table 1 presents the results of our experiment addressing **RQ1**. Our model surpasses all of the baselines in F1, AUC, and ACC, demonstrating its robustness to data imbalance. While some baselines exhibited higher Recall and Precision, their predictions were biased, highlighting an advantage of our proposed method, which facilitates more balanced inference. We also demonstrate explainability of our mothod in Appendix B.

We conducted an ablation study to address **RQ2**. The results, presented in Table 2, indicate that each of our proposed factors contributes to performance improvement. Notably, the absence of schema tokens results in a significant performance reduction, underscoring the effectiveness of our schema-aware approach.

Model	F1	AUC	ACC	Recall	Precision
LLaMA3.1 (8B)	0.5487	0.8077	0.8252	0.5322	0.5662
Gemma2 (9B)	0.5425	0.8396	0.7095	0.8627	0.3957
gpt-4o-mini	0.5614	0.8099	0.7335	0.8541	0.4181
gpt-4o	0.7778	0.8855	0.9177	0.7210	0.8442
Binary-classifier (T5)	0.6340	0.7339	0.8908	0.4721	**0.9649**
Binary-classifier (T5, balance)	0.6759	0.7586	0.8993	0.5236	0.9531
Entropy-based filtering (T5)	0.6148	0.7770	0.8260	0.6953	0.5510
ProbGate (gpt-3.5-turbo)	0.6917	0.8854	0.8243	**0.9871**	0.5324
SQD (Ours)	**0.8547**	**0.9062**	**0.9426**	0.8640	0.8455

Table 1: Performance of SQD compared to baselines

Model	F1	AUC	ACC	Recall	Precision
SQD (Ours)	**0.8547**	**0.9062**	**0.9426**	**0.8640**	**0.8455**
w/o Prompt Tokens	0.7919	0.8573	0.9212	0.7511	0.8373
w/o Schema Tokens	0.5739	0.7846	0.7506	0.8412	0.4356
w/o LLM Generated Discriptions	0.7826	0.8493	0.9186	0.7339	0.8382

Table 2: Ablation study

Discription	F1	AUC	ACC	Recall	Precision
gpt-3.5-turbo	**0.8547**	0.9062	**0.9426**	0.8640	**0.8455**
LLaMA 3	0.8462	**0.9207**	0.9349	**0.8970**	0.8008
DDL	0.8184	0.8815	0.9289	0.8026	0.8348
w/o Discription	0.7826	0.8493	0.9186	0.7339	0.8382

Table 3: Impact of LLM generated schema descriptions

To address **RQ3**, we experimented with different descriptions to generate schema embeddings. We discovered that LLMs, such as *gpt-3.5-turbo* and *LLaMA 3* [3], outperformed the DDL-based methods in comparison. These results demonstrate that the capabilities of LLMs can be harnessed for various tasks without the need for resource-intensive processes like manual fine-tuning. Notably, as illustrated in Figure 3, the LLMs have the potential to be utilized in a variety of domains, as even relatively small models can provide explanations for complex technical terms that are otherwise difficult to comprehend.

4 Conclusion

We have proposed a lightweight schema-aware unanswered question detection method that leverages the capabilities of LLMs. Through experiments on healthcare domain data, we demonstrate that our method is robust to data imbalance and achieves more balanced performance compared to the existing methods. We have also analyzed the types of questions that are difficult to detect, providing insights and directions for future research. Our work illustrates how LLMs can be more efficiently utilized across diverse domains.

[1] https://anonymous.4open.science/r/sqd2024

[2] https://github.com/glee4810/ehrsql-2024

[3] https://llama.meta.com/llama3/

5 Limitations

Our proposed method can detect unanswered questions more effectively than existing methods. However, our method performs poorly on some samples. Figure 4 presents a sample of these undetected unanswerable questions. These examples typically include words that are directly related to the schema, such as “patient,” and the questions often have a very specific purpose, such as asking for a number. In such cases, even human judgment struggles to determine whether the question is answerable, as the answerability can only be confirmed by executing the SQL query to check if the relevant data exists. Thus, it is essential to collaborate with an agent capable of executing SQL in an environment similar to the actual database to classify the answerability of these challenging cases accurately.

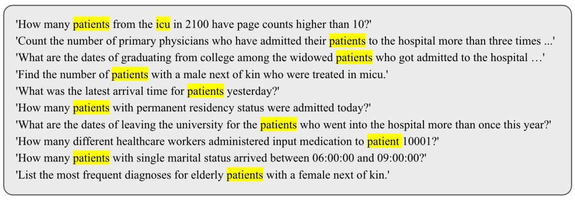
Another limitation of our study is the use of a single dataset. Currently, text-to-SQL datasets containing unanswered questions are very rare, and EHRSQL-2024 is the only dataset in the medical domain. This lack of data can be addressed in the future as more applications are adopted in industry.

Furthermore, our study does not address whether our method retains its generalised reasoning ability when new schemas are added. However, we believe that the method proposed in this study can be effectively applied to the domain of general unanswered question detection. It is possible to improve the generalisation performance by augmenting the data using LLM or introducing a pre-learning model, and to evaluate schemas and question types not seen in the training phase. We intend to overcome these limitations in future research.

6 Ethics Statement

Potential Risks Our study was conducted using a limited dataset and does not guarantee the integrity of the proposed method in real-world applications. When applying our proposed method to databases containing patient-specific records, consideration should be given to the sensitive information that may be contained in the data, and our findings do not guarantee the completeness of the inferred results.

Use of Scientific Artifacts Our research leveraged open-source tools including PyTorch (Paszke et al., 2019) and scikit-learn (Pedregosa et al., 2011), alongside pre-trained language models such as T5, LLaMA3.1, LLaMA3, and Gemma2 obtained via the Huggingface (Wolf et al., 2019) library. For experiments involving LLMs, we utilized OpenAI’s API under their sharing and publication policy (OpenAI, 2022).



'How many patients from the icu in 2100 have page counts higher than 10?'
'Count the number of primary physicians who have admitted their patients to the hospital more than three times ...'
'What are the dates of graduating from college among the widowed patients who got admitted to the hospital …'
'Find the number of patients with a male next of kin who were treated in micu.'
'What was the latest arrival time for patients yesterday?'
'How many patients with permanent residency status were admitted today?'
'What are the dates of leaving the university for the patients who went into the hospital more than once this year?'
'How many different healthcare workers administered input medication to patient 10001?'
'How many patients with single marital status arrived between 06:00:00 and 09:00:00?'
'List the most frequent diagnoses for elderly patients with a female next of kin.'

Figure 4: Samples incorrectly classified as answerable.

Use of AI Assistants We only used ChatGPT to provide a better expression and to refine the wording. Some of the code used in the experiment was written with the assistance of Copilot.

References

Negar Arabzadeh, Mahsa Seifikar, and Charles L.A. Clarke. 2022. Unsupervised question clarity prediction through retrieved item coherency. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 3811–3816, New York, NY, USA. Association for Computing Machinery.

Yuyan Chen, Qiang Fu, Yichen Yuan, Zhihao Wen, Ge Fan, Dayiheng Liu, Dongmei Zhang, Zhixu Li, and Yanghua Xiao. 2023. Hallucination detection: Robustly discerning reliable answers in large language models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 245–255, New York, NY, USA. Association for Computing Machinery.

Yongrae Jo, Seongyun Lee, Minju Seo, Sung Ju Hwang, and Moontae Lee. 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. *arXiv preprint arXiv:2405.11162*.

Sangryul Kim, Donghee Han, and Sehyun Kim. 2024. Probgate at ehrsql 2024: Enhancing sql query generation accuracy through probabilistic threshold filtering and error handling. *arXiv preprint arXiv:2404.16659*.

Yuta Koreeda, Terufumi Morishita, Osamu Imaichi, and Yasuhiro Sogawa. 2023. Larch: Large language model-based automatic readme creation with heuristics. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 5066–5070, New York, NY, USA. Association for Computing Machinery.

Gyubok Lee, Woosog Chay, Seonhee Cho, and Edward Choi. 2024a. 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.

Gyubok Lee, Sunjun Kweon, Seongsu Bae, and Edward Choi. 2024b. Overview of the ehrsql 2024 shared task on reliable text-to-sql modeling on electronic health records. *Preprint*, arXiv:2405.06673.

Lei Li, Yongfeng Zhang, and Li Chen. 2023. Prompt distillation for efficient llm-based recommendation. In *CIKM*.

Bulou Liu, Yiran Hu, Qingyao Ai, Yiqun Liu, Yueyue Wu, Chenliang Li, and Weixing Shen. 2023. Leveraging event schema to ask clarifying questions for conversational legal case retrieval. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 1513–1522, New York, NY, USA. Association for Computing Machinery.

Mohsen Nayebi Kerdabadi, Arya Hadizadeh Moghaddam, Bin Liu, Mei Liu, and Zijun Yao. 2023. Contrastive learning of temporal distinctiveness for survival analysis in electronic health records. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 1897–1906, New York, NY, USA. Association for Computing Machinery.

OpenAI. 2022. Introducing chatgpt. `https://openai.com/blog/chatgpt`. Last accessed on 2024-01-15.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

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.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce Ho, Carl Yang, and May D Wang. 2024. Ehragent: Code empowers large language models for complex tabular reasoning on electronic health records. *arXiv preprint arXiv:2401.07128*.

Bing Wang, Yan Gao, Zhoujun Li, and Jian-Guang Lou. 2023. Know what I don't know: Handling ambiguous and unknown questions for text-to-SQL. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5701–5714, Toronto, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

A Dataset

We share more detailed information about the dataset here. Table 4 provides a descriptive summary of the dataset we used. Notably, the valid and test datasets exhibit a higher rate of unanswerable questions compared to the training dataset.

Dataset	Train	Valid	Test
Number of questions	5124	1163	1167
Unanswerable question ratio	0.0878	0.1995	0.1997
Total number of schema	17	17	17
Average number of schema per question	2.441	2.193	2.163

Table 4: Statistics of datasets.

EHRSQL2024 is the only publicly available sql-to-text data we have, including unanswerable samples. Experimenting with a wider variety of data would help validate our methodology, but we are limited by the current publicly available data. We hope that this study will encourage academics to contribute to public datasets with more diverse unanswerable questions in the future.

Figure 5 presents the proportion of questions associated with each schema in the training set used in our experiments. In a binary classification approach, the percentage of unanswerable questions is less than 9%. However, by predicting schema-level relevance, this imbalance is mitigated. When schema-level relevance is converted to binary data, the proportions of relevant and irrelevant questions are 14.4% and 85.6%, respectively. This distribution is less imbalanced and yields more robust results by enabling multifaceted assessments across various schemas. We propose a way to leverage this question-schema relationship to effectively predict answerability.

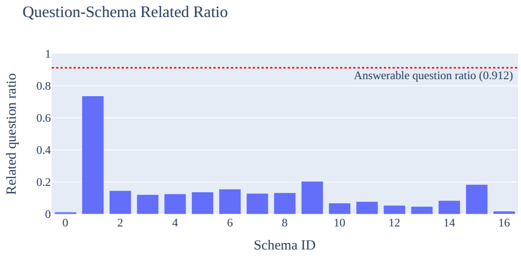
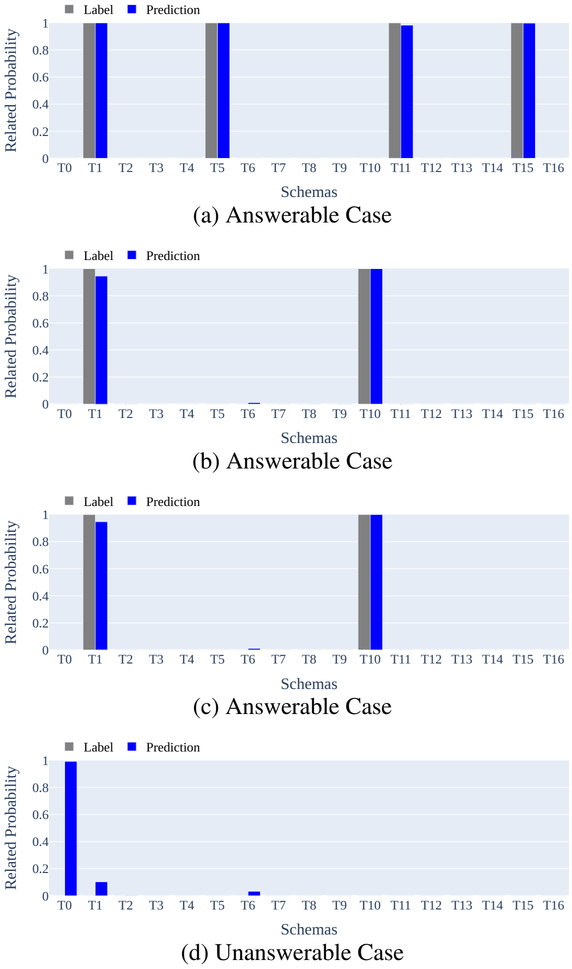


Figure 5: Question-schema related ratio.

B Explainability

Our proposed method estimates the relevance of each schema to a question, which can increase its explainability. Figure 6 illustrates the real schema relevance compared to the results predicted by our model. (a), (b), and (c) are answerable cases, and we observe that our model predicts the same results as the actual label values in all cases. (d) is unanswerable but incorrect as our model predicts that the question is relevant for the “T0” table. “T0” is the "patients" table, and the question was ”Show the average age of patients with a female first child”. For this question, even if we knew the existence of the “patients” table, it would be difficult for a human to determine, and would only be known by performing actual data exploration. These results suggest that there is a need for research into agents that can interact directly with databases. Our method provides human-understandable inference result and a variety of possibilities to complement the methodology.



(a) Answerable Case

(b) Answerable Case

(c) Answerable Case

(d) Unanswerable Case

Figure 6: Caption for Figure 3

C LLM Generated Schema Descriptions

We expose some of the schema descriptions generated from the DDLs via *gpt-3.5-turbo*. The full data is available in our online repository.

C.1 "patients" Table Schema

> In the given table schema "patients", the following information would be contained:
> 1. row_id: This is an integer field that serves as a unique identifier for each row in the table. It is marked as NOT NULL, meaning it must have a value for each record, and is set as the PRIMARY KEY, ensuring its uniqueness.
> 2. subject_id: This field stores an integer value that uniquely identifies each patient. It is marked as NOT NULL and UNIQUE, indicating that each patient must have a subject_id, and no two patients can have the same subject_id.
> 3. gender: This field stores the gender of the patient as a string of maximum length 5 characters (e.g., 'Male' or 'Female'). It is marked as NOT NULL, meaning it must have a value for each record.
> 4. dob: This field stores the date of birth of the patient as a timestamp with precision up to seconds. It is marked as NOT NULL, indicating that the date of birth must be recorded for each patient.
> 5. dod: This field stores the date of death of the patient as a timestamp with precision up to seconds. It is not marked as NOT NULL, allowing for cases where the patient's date of death may not be known or recorded.
> Overall, this table schema "patients" would store information about patients including their unique identifiers, gender, date of birth, and date of death (if available).

C.2 "admissions" Table Schema

> In the given table schema "admissions", the following information would be contained:
> 1. row_id: This is a unique identifier for each row in the table.
> 2. subject_id: The identifier for the patient associated with the admission.
> 3. hadm_id: A unique identifier for each hospital admission.
> 4. admittime: The timestamp when the patient was admitted to the hospital.
> 5. dischtime: The timestamp when the patient was discharged from the hospital.
> 6. admission_type: The type of admission (e.g., emergency, elective).
> 7. admission_location: The location where the patient was admitted (e.g., emergency room, transfer from another hospital).
> 8. discharge_location: The location where the patient was discharged to (e.g., home, skilled nursing facility).
> 9. insurance: The type of insurance coverage the patient has.
> 10. language: The language spoken by the patient.
> 11. marital_status: The marital status of the patient.
> 12. age: The age of the patient at the time of admission.
> Additionally, there is a foreign key constraint on the subject_id column, referencing the subject_id column in the "patients" table. This ensures referential integrity between the two tables, linking patient information across the database.

D Hyperparameters

The following describes the hyperparameter values used in the experiments. The learning rate was set to three different values: 5×10^{-4}, 1×10^{-4}, and 1×10^{-3}. We trained the model in 100 epochs. The batch size was fixed at 32. The threshold was experimented with five values, 0.1, 0.2, 0.3, 0.4 and 0.5. Finally, the number of prompt tokens was set to 1, 2, and 3 for the experiments. We report the highest performance among several parameter combinations.