The 6th Clinical Natural Language Processing Workshop at NAACL 2024

ProbGate at EHRSQL 2024: Enhancing SQL Query Generation Accuracy through Probabilistic Threshold Filtering and Error Handling

Sangryul Kim[1], Donghee Han[2], Sehyun Kim[3]

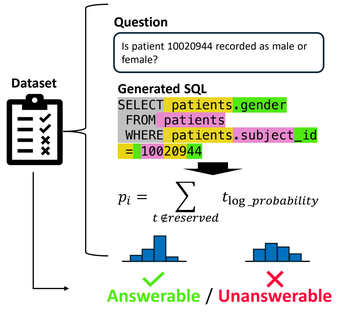
[1]KAIST AI, [2]KAIST Graduate School of Data Science, [3]KAIST Bio and Brain Engineering

Code & Paper

ABSTRACT

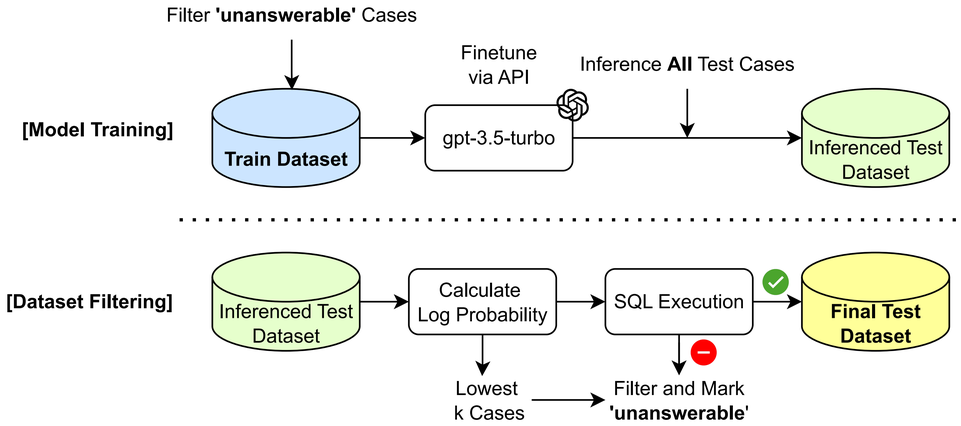
Recently, deep learning-based language models have significantly enhanced text-to-SQL tasks, with promising applications in retrieving patient records within the medical domain. One notable challenge in such applications is discerning unanswerable queries. Through fine-tuning model, we demonstrate the feasibility of converting medical record inquiries into SQL queries. Additionally, we introduce an entropy-based method to identify andfilter out unanswerable results. We further enhance result quality by filtering low-confidence SQL through log probability-based distribution, while grammatical and schema errors are mitigated by executing queries on the actual database. We experimentally verified that our method can filter unanswerable questions, which can be widely utilized even when the parameters of the model are not accessible, and that it can be effectively utilized in practice.

ProbGate



This averaged value is considered representative of the log probability for one test dataset. Consider the lowest 'k' datasets as unanswerable.

Main Method



✓ We propose a two-step filtering method here to distinguish unanswerable SQL queries that were incorrectly generated due to hallucination.

✓ The first is a filtering method called "**ProbGate**" which uses log probability, and the second involves filtering through SQL Execution to verify any syntax errors in the generated SQL.

Results on shared task test set

Model	Rs(0)	RS(5)	RS(10)	RS(N)
gpt-3.5-turbo FT	73.52	-58.87	-191.25	-30826.47
gpt-3.5-turbo FT + ProbGate(t=450)	79.43	73.01	66.58	-1420.57
gpt-3.5-turbo FT + ProbGate(t=450) + GEF	79.78	75.92	72.06	-820.22
gpt-3.5-turbo FT + ProbGate(t=425) + GEF	**81.92**	**78.06**	74.21	**-818.08**

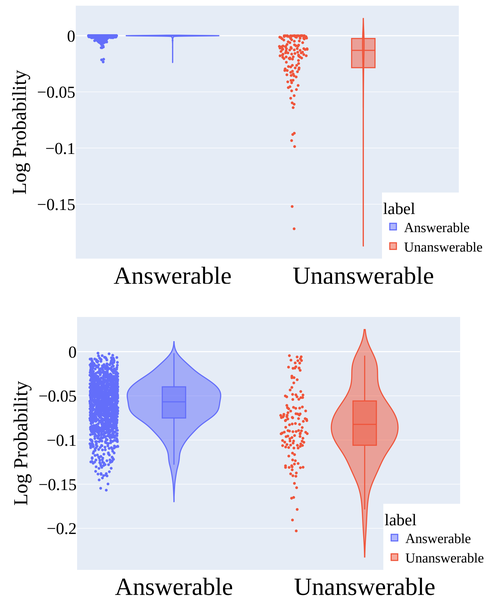
Why ProbGate ?

✓ The trained model can effectively filter answerable and unanswerable data across different data distributions stably.

Why GEF(Grammatical Errors Filtering) ?

✓ Grammatical errors can ultimately only be detected by actual execution; SQLs that pass this stage are syntactically correct, executable, and likely to achieve high accuracy.

Analysis



Up - Log Probability Distribution of the Fine-Tuned Model

Down - Log Probability Distribution of the Unfine-Tuned Model

- We compare the log probability distribution between 30% mixed answerable and unanswerable data using a model trained on about 70% answerable data from the train dataset.
- A clear log probability difference between answerable and unanswerable questions is evident, and this distinction is more pronounced in the fine-tuned model.