

# sLLM 기반의 경량화 Text-to-SQL 구현 방안

임재영, 윤수연\*  
국민대학교 소프트웨어융합대학원, \*국민대학교  
dlawowo01@kookimin.ac.kr, \*1004py@kookmin.ac.kr

## Approaches to Lightweight Text-to-SQL Implementation Based on sLLM

Jae Young Im, Soo-Yeon Yoon\*  
Kookmin Univ, \*Kookmin Univ.

### 요약

최근 대형 언어 모델(Large Language Model, LLM)을 활용한 Text-to-SQL 기술이 주목받고 있다. 그러나 LLM 기반의 Text-to-SQL 모델은 구축 및 유지보수에 많은 비용이 들며, 내부 스키마 정보를 공개해야 하는 보안 문제가 발생한다. 이러한 문제들을 해결하기 위해 sLLM(Smaller Large Language Model)을 활용하여 경량화된 Text-to-SQL 모델을 제안한다. 본 연구에서는 DAIL-SQL 과 Llama3-8B 모델을 사용하여 sLLM 기반의 경량화된 Text-to-SQL 모델을 구현하고, 이를 기존의 LLM 기반 모델과 비교하여 비용과 보안 측면에서의 우수성을 입증한다. 이러한 접근은 LLM의 기능을 유지하면서도 관련 비용을 절감하고 보안 문제를 개선할 수 있는 가능성을 보여준다.

### I. 서론

Text-to-SQL 기술은 자연어를 데이터베이스가 이해할 수 있는 SQL 쿼리로 변환하는 기술이다. 기존의 LLM은 학습된 데이터에서 답을 찾아 최신 정보를 반영하는 데 한계가 있었다. 이러한 한계를 극복하기 위해 Text-to-SQL 기술이 주목받고 있다. 특히 최근에는 LLM(Large Language Model)의 발전으로 Text-to-SQL 분야에서도 LLM을 기반으로 한 연구가 활발히 이루어지고 있다.

하지만 Large Language Model을 기반으로 한 Text-to-SQL 모델을 구축하는 것은 구축 및 유지보수 비용이 크며, 각 기업의 데이터베이스 내부 스키마 구조를 외부로 공개할 수 없다는 보안 문제로 인해 온라인 기반 Large Language Model을 사용하는 것은 제한적이다. 이에 본 논문에서는 sLLM 기반의 경량화 Text-to-SQL 모델을 제안한다.

### II. 본론

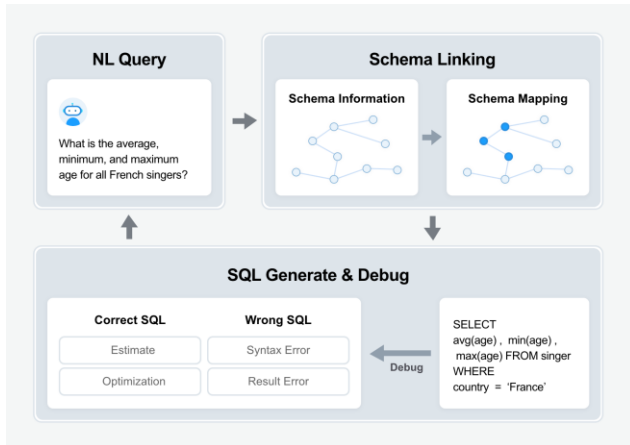
Text-to-SQL 기술은 초기에는 Seq2Seq 방식을 기반으로 질의어와 스키마 간 연관성을 찾아 Text-to-SQL을 구현하는 방식이었으나 최근에는 대규모 언어 모델(Large Language Model)을 기반으로 한 연구가 활발히 진행되고 있다. Spider 리더보드 기준으로 최상위 5개 중 4개(2024년 4월 12일 기준)는 LLM을 기반으로 한 Text-to-SQL 모델이다. 그러나 LLM 기반의 Text-to-SQL은 기업에서의 구현 및 유지보수에 많은 비용이 들며, 데이터베이스의 스키마 내부 구조를 개방형 LLM 모델에 학습시키는 것은 보안상 문제가 발생할 수 있다. 따라서 본 연구는 대용량 LLM 대신 Smaller Language Model(sLLM)을 활용한 경량화된 Text-to-SQL 모델을 제안한다.

Text-to-SQL은 <Figure 1>과 같은 절차에 따라 진행된다. 먼저 사용자가 질의한 자연어를 데이터베이스 스키마 구조와 매칭시킨다. 이를 통해 사용자가 질의한 스키마 구조를 분석하고 이를 바탕으로 SQL을 생성하며, SQL에 대한 평가를 진행한다. 평가를 진행할 때는 해당

\* First Author : Kookmin university, Graduate School of Software technology, dlawowo01@kookmin.ac.kr,

\* Corresponding Author : Kookmin university, Graduate School of Software technology, 1104py@kookmin.ac.kr, 정회원

SQL 이 오류인지 정상적으로 수행되는지 확인하고 이 중에서 올바른 SQL 에 대한 성능 평가를 수행한다.[1]



<Figure 1> [Text-to-SQL 수행 순서]

### III. 관련 기술

Text-to-SQL 변환 기술은 사용자의 자연어 질의를 SQL 쿼리로 전환하는데, 이를 위해 LLM(Large Language Models)을 활용하는 몇 가지 접근 방법이 있습니다.

첫 번째로는 Zero-shot learning 으로, 특별한 사전 훈련 없이도 태스크를 수행할 수 있게 설계되었다. GPT 와 같은 대규모 언어 모델을 사용하여 자연어 질의를 SQL 쿼리로 변환하지만, 이 방식은 데이터베이스의 스키마 정보를 반영하지 못하는 한계가 있다. 결과적으로 모델이 데이터베이스 스키마와의 관계를 명확히 인식하지 못하며, 이는 정확도 저하로 이어질 수 있다.

두 번째 방법은 Few-shot learning 으로, 이는 제한된 수의 샘플과 질문을 사용하여 모델을 학습시킨다. 이 방식은 zero-shot 에 비해 더 정밀한 데이터 학습을 가능하게 하며, 사용자 질의에 대한 이해도를 높이는 데 도움을 준다. 모델이 데이터베이스의 스키마 정보를 효과적으로 파악하고 SQL 쿼리 생성에 필요한 문맥적 뉘앙스를 포착할 수 있게 지원합니다.

마지막으로, Fine-tuning 방법이 있다. 이는 사전에 훈련된 LLM 에 특정 데이터베이스 스키마 구조를 추가로 학습시키는 과정을 포함한다. Fine-tuning 은 텍스트와 스키마 간의 관계를 더욱 정밀하게 매핑하며, 사용자의 질의에 대한 정확하고 효과적인 SQL 쿼리를 생성할 수 있도록 한다. 하지만, 이 방법은 상대적으로 높은 컴퓨팅 자원과 비용을 요구하는 단점이 있다.

### IV. 실험 방법 및 환경

Text-to-SQL 분야에서 주요한 벤치마크 데이터셋으로는 Spider 와 WikiSQL 이 대표적이다. WikiSQL 은 Wikipedia 의 24,241 개 테이블에 대한 80,654 개의 샘플을 포함한 데이터셋으로, 독립적인 단일 테이블에 대한 SQL 쿼리 쌍과 Select-Where 표현만을 다루는 구조를 갖추고 있다.

Spider[2]는 예일대 11 명의 대학생이 주석을 단 SQL 데이터 세트이며 10,181 개의 질문으로 이루어져 있는 복잡한 SQL 쿼리를 포함한 다중 테이블에 대한 SQL Query Pair 와 Having, Group by, limit, join 등 다양한 케이스가 존재한다. 난이도는 Easy, Medium, Hard, Extra Hard 로 나뉜다.

본 논문에서는 Spider 데이터셋을 활용하여 실험을 수행하고, 비교 모델로 DAIL-SQL 을 사용한다. DAIL-SQL 은 Dawei Gao 가 제안한 Text-to-SQL 변환 모델로써, Masked Question Similarity Selection (MQS)과 Query Similarity Selection (QRS) 두 가지를 결합하는 DAIL Selection 방법으로 구현된 Text-to-SQL 모델이다. MQS 는 도메인 간 정보의 부정적 영향을 최소화하기 위해 질문 내의 테이블 이름, 열 이름, 값 등을 마스크 토큰으로 대체하고, 이들의 임베딩 유사성을 k-Nearest Neighbor(kNN) 알고리즘을 통해 계산하는 방식이다. 이를 통해 다양한 도메인에서의 적용 가능성을 높이며, 적합한 SQL 를 효과적으로 선별할 수 있다. QRS 는 타겟 SQL 와 유사한 예제를 선택하는 과정에서,

사전 모델을 활용해 타겟 질문과 데이터베이스를 기반으로 초기 SQL을 생성하고, 이를 근사값으로 사용한다. 이후, SQL의 키워드를 기반으로 이진 이산 구문 벡터로 인코딩하고, 근사된 SQL와의 유사성 및 선택된 예제들 간의 다양성을 고려하여 최적의 예제를 선정한다.

sLLM은 2024년 4월에 발표한 Meta AI의 Llama3-8B를 사용한다. Llama3-8B는 기존 Llama2에서는 대용량 모델에만 적용되던 GQA(Grouped Query Attention)를 적용하고 있으며 Llama2 70B보다 성능이 뛰어나다. 본 논문의 실험에서는 DAIL-SQL + Llama3-8B를 사용하여 실험을 진행한다.

## V. 실험 결과

실험 결과는 EM(exact match accuracy)와 EX(Execution accuracy) 두 가지로 구분된다. EM은 예측 SQL과 정답 SQL이 얼마나 SQL 키워드가 같은지 측정하는 것이고 EX는 실제 실행한 결과가 동일한지 평가한다.

Model	EM	EX
DAIL-SQL + Llama3-8B	46.8	<b>69.2</b>
DAIL-SQL + GPT-4(1 shot)[4]	62.1	80.2
DIN-SQL + GPT-4[3]	60	85.3
RESDSL-3B + NatSQL[5]	72.0	79.9

<Table 1> [DAIL-SQL + Llama3-8B 실험 결과와 기존 Text-to-SQL 모델의 정확도]

<Table 1>에서 DAIL-SQL + Llama3-8B는 EX 69.2으로 비교 모델에 비해 준수한 성능을 보여줬다. 이 결과는 sLLM 기반으로 한 경량화 Text-to-SQL 접근법 또한 효과적임을 입증하였다.

	EASY	MEDIUM	HARD	EXTRA	ALL
<b>EX</b>	85.1	74.4	61.5	39.8	68.2
<b>EM</b>	85.1	46	33.3	14.5	46.8

<Table 2> [DAIL-SQL + Llama3-8B의 SQL 난이도 별 결과]

	Prediction	Gold
<b>EASY</b>	SELECT AVG(LifeExpectancy) FROM country WHERE Continent = 'Africa' AND Region = 'Central'	SELECT avg(LifeExpectancy) FROM country WHERE Region = "Central Africa"
<b>MEDIUM</b>	SELECT COUNT(DISTINCT continent) FROM countrylanguage WHERE language = 'Chinese'	SELECT COUNT(DISTINCT Continent) FROM country AS T1 JOIN countrylanguage AS T2 ON T1.Code = T2.CountryCode WHERE T2.Language = "Chinese"
<b>HARD</b>	SELECT p.birth_date FROM poker_player p ORDER BY p.earnings ASC LIMIT 1	SELECT T1.Birth_Date FROM people AS T1 JOIN poker_player AS T2 ON T1.People_ID = T2.People_ID ORDER BY T2.Earnings ASC LIMIT 1
<b>EXTRA</b>	SELECT state FROM VOTES GROUP BY state ORDER BY COUNT(*) DESC LIMIT 1	SELECT T1.area_code FROM area_code_state AS T1 JOIN votes AS T2 ON T1.state = T2.state GROUP BY T1.area_code ORDER BY count(*) DESC LIMIT 1

<Table 3> [DAIL-SQL + Llama3-8B의 실패 사례]

실패 사례 분석 결과, 모델은 테이블 간의 정확한 관계를 파악하는 데 어려움을 겪었으며, 이는 다중 테이블을 포함하는 JOIN 연산을 정확하게 표현하지 못하는 주요 원인으로 확인되었다. <Table 2>에서 보듯이, 복잡성이 증가함에 따라 모델의 성능 저하가 두드러지게 나타났다.

이러한 문제를 해결하기 위해 모델에 대한 Fine-tuning이 필요하다. Fine-tuning 과정을 통해 Llama3-8B 모델은 데이터베이스 스키마 간의 관계를 더 명확하게 학습할 수 있으며, 이는 JOIN과 같은 복잡한 SQL 연산을 정확하게 처리하는 데 도움을 줄 것이다.

현재 연구에서는 Fine-tuning 없이 sLLM 모델을 Few-shot learning 으로 구현하여 Text-to-SQL 변환을 시도했다. 이 접근 방식은 초기 학습에는 유리하지만, 특히 복잡한 질의 처리에 있어서는 한계를 보이는 것으로 나타났다. 따라서 추후 연구에서는 Fine-Tuning 을 통한 성능 개선이 필요하며, 이는 Text-to-SQL 변환의 정확도를 높이는 데 중요한 역할을 할 것이다.

## VI. 결론

이 연구에서는 LLM 기반의 Text-to-SQL 기술의 한계를 극복하기 위해, sLLM 을 활용한 경량화된 Text-to-SQL 모델을 제안하였다. 연구 결과에 따르면, sLLM 기반의 Text-to-SQL 모델은 기존 LLM 기반 모델에 비해 준수한 성능을 유지하면서도 구축 및 유지보수 비용을 크게 절감하고, 보안 문제를 개선할 수 있는 가능성을 입증하였다.

특히, Meta AI 의 최신 모델인 Llama3-8B 를 사용한 실험은 비용 효율성과 보안성을 유지하면서 기존의 LLM 모델들과 경쟁할 수 있는 성능을 보여주었다. 이는 Text-to-SQL 기술의 상용화 및 다양한 데이터베이스 환경에서의 활용 가능성을 시사한다.

후속 연구에서는 sLLM 의 향상된 성능을 목표로 하며, Fine-tuning 을 통해 데이터베이스 테이블 간의 정확한 관계를 학습함으로써, 성능이 더욱 향상된 경량화 Text-to-SQL 모델을 개발할 예정이다. 이 과정은 모델의 성능을 높이는 동시에, 다양한 데이터베이스 환경에서의 범용성을 강화할 것으로 기대된다.

## 참 고 문 헌

[1] Bin Zhang, Yuxiao Ye, Guoqing Du, Xiaoru Hu, Zhishuai Li, Sun Yang, Chi Harold Liu, Rui Zhao, Ziyue Li and Hangyu Mao, "Benchmarking the Text-to-SQL Capability of Large Language Models: A Comprehensive Evaluation", arXiv, 2403.02951, 2024.

[2] Tao Yu, Rui Zhang and Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang and Dragomir Radev, "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task", arXiv, 1809.08887, 2019.

[3] Mohammadreza Pourreza and Davood Rafiei, "DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction", arXiv, 2304.11015, 2023.

[4] Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian and Bolin Ding and Jingren Zhou, "Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation", arXiv, 2308.15363, 2023.

[5] H. Li, J. Zhang, C. Li, and H. Chen, "RESDSL: Decoupling Schema Linking and Skeleton Parsing for Text-to-SQL", AAAI, vol. 37, no. 11, pp. 13067-13075, Jun. 2023.

[6] 윤훈상, 허재혁, 김정섭, and 강필성, "다국어 BERT 를 활용한 한국어 자연어 질의의 SQL 변환," 대한산업공학회지, vol. 48, no. 1, pp. 91-104, 2022

[7] Trummer and Immanuel, "Demonstrating GPT-DB: Generating Query-Specific and Customizable Code for SQL Processing with GPT-4", Proc. VLDB Endow., 16, 12, pp.4098- 4101, 2023

[8] Vanessa Câmara, Rayol Mendonca-Neto, André Silva, and Luiz Cordovil-Jr, "DBVinci - towards the usage of GPT engine for processing SQL Queries", Proceedings of the 29th Brazilian Symposium on Multimedia and the Web, pp.91- 95, 2023

[9] Yangjun Wu and Han Wang, "Schema-Aware Multi-Task Learning for Complex Text-to-SQL", arXiv, 2403.09706, 2024

[10] Immanuel Trummer, "CodexDB: Generating Code for Processing SQL Queries using GPT-3 Codex", arXiv, 2204.08941, 2022

[11] Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov and Matthew Richardson, "RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers", arXiv, 1911.04942, 2021

[12] Haoyang Li, Jing Zhang, Hanbing Liu, Ju Fan, Xiaokang Zhang, Jun Zhu, Renjie Wei, Hongyan Pan, Cuiping Li and Hong Chen, "CodeS: Towards Building Open-source Language Models for Text-to-SQL", arXiv, 2402.16347, 2024

[13] P. Wang, T. Shi, and C. K. Reddy, "Text-to-SQL Generation for Question Answering on Electronic Medical Records," in Proceedings of The Web Conference 2020 (WWW '20), Taipei, Taiwan, pp. 350-361, 2020

[14] X. Deng, A. H. Awadallah, C. Meek, O. Polozov, H. Sun, and M. Richardson, "Structure-Grounded Pretraining for Text-to-SQL," in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), doi: 10.18653/v1/2021.naacl-main.105, 2021

[15] X. Dong, C. Zhang, Y. Ge, Y. Mao, Y. Gao, L. Chen, J. Lin, and D. Lou, "C3: Zero-shot Text-to-SQL with ChatGPT," arXiv, 2307.07306, 2023