SQL 합성을 SQL 검색으로 대체하는 Text-to-SQL 접근법

한을수 [O12] 이재길 [1]

[1] 한국과학기술원 소프트웨어 대학원 / [2] 삼성전자

sktan999@google.com, jaegil@kaist.ac.kr

An Approach to Replace SQL Synthesis with SQL Search for Text-to-SQL

Eulsu Han[O12] Jae-Gil Lee[1]

[1]School of Computing, KAIST / [2]Samsung Electronics

요 약

Text-to-SQL 은 데이터베이스 접근을 대중화하는 중요한 도구로 부상했지만, 현재 접근 방식은 일관성, 신뢰성, 성능 최적화, 활용성 측면에서 여러 과제에 직면해 있다. 본 논문에서는 Text-to-SQL 작업을 위한 새로운 SQL 검색 기반 방법을 소개한다. 제안 접근법은 데이터베이스의 최적화 방식 중 일부가 유사한 SQL 이 재사용 될 확률에 근거한다는 점에 기반하며, SQL 을 생성하지 않고 대규모 언어 모델을 활용하여 탐색한다. Spider 데이터셋과 Llama3-70B 모델로 92.9%의 일치 정확도를 달성하였고, 특히 SQL 쿼리 복잡도 대비 안정적인 성능을 보인다. 우리의 실험은 검색 방식의 활용 가능성을 보여준다. 이 SQL 검색 기반 방법은 실제 환경에서 실용적인 Text-to-SQL 적용 방향을 제시한다.

1. 서 론

Text-to-SQL 은 데이터베이스 접근을 대중화하는 중요한 도구로 부상했다. 이 기술은 사용자가 자연어로 SQL 쿼리를 작성할 수 있게 함으로써 데이터베이스 접근을 보다 직관적이고 사용자 친화적으로 만들며, 데이터 활용의 폭을 넓히는 데 크게 기여한다. 최근에는 대규모 언어 모델(LLM)을 활용한 In-context learning 방식이 주목받고 있으며, 이는 학습 없이도 우수한 성능을 발휘할 수 있음을 보여준다[1].

그러나 이러한 기술을 실제 환경에 적용하는 데는 여러 과제가 존재한다. 첫째, LLM 은 동일한 질의에도 결과를 예측하기 어렵고 SQL 인젝션[2] 등의 위험에 노출되어 있어 일관성과 신뢰성 문제를 가지고 있다. 둘째, 대규모 생산 시스템과 같이 운영 안정성이 중요한 경우에 필수적인 SQL 튜닝을 충분히 고려하지 않아 성능 최적화에 한계가 있다[3]. 셋째, 학습 기반 모델의 경우 데이터 의존성, 과적합 위험, 높은 학습 비용 및 해석 가능성 부족 등의 문제가 존재한다. 이러한 문제들은 Text-to-SQL 기술의 실제 환경 도입을 어렵게 만드는 주요 요인이 된다.

이러한 과제를 해결하기 위해 새로운 접근법을 제안한다. 제안 접근법은 조직 내에서 이미 검증된 SQL 쿼리가 존재한다는 가정을 한다. 이를 활용하여 새로운 SQL 쿼리를 생성하는 대신 가장 적합한 SQL 쿼리를 검색하는 방식이다. 이는 일관성과 성능을 보장하면서도, 자연어 처리 기술의 장점을 활용할 수 있는 방법이다.

본 연구의 주요 기여점은 다음과 같다. 첫째, SQL 검색 기반의 새로운 Text-to-SQL 방식을 제시한다. 둘째, 제안된 방법의 성능을 높이기 위한 실험과 결과를 제시한다. 셋째, 실제 환경에서의 Text-to-SQL 적용 가능성을 높이는 실용적인 접근법을 제공한다.

2. 관련 연구

Text-to-SQL 분야는 최근 몇 년 동안 자연어 처리 기술의 발전으로 인해 다양한 방식으로 연구되어 왔다.

초기 Text-to-SQL 시스템은 주로 규칙 기반 방법과 의미 구문 분석 기술에 의존했다[4]. 이러한 접근 방식은 간단한 쿼리에는 효과적이었지만, 복잡한 다중 테이블 쿼리를 처리하는 데 어려움을 겪었다.

딥 러닝의 등장 이후 신경망 기반 접근법으로 Text-to-SQL 연구는 새로운 국면을 맞이하게 되었다. 예를 들어, SQLNet (Xu et al., 2017)은 강화 학습을 사용하지 않고도 구조화된 쿼리 생성이 가능함을 보여주었다[5]. SyntaxSQLNet(Yu et al., 2018)은 구문 트리 네트워크를 활용하여 복잡한 도메인 간의 쿼리를 처리할 수 있는 성능을 입증했다[6].

최근에는 미세 조정 모델과 LLM 을 활용한 연구가 주목받고 있다. PICARD(Scholak et al., 2021)는 미세 조정 모델을 사용하여 구문 제약을 적용한 자동 회귀 디코딩을 통해 Text-to-SQL 작업의 성능을 크게 향상시켰다[7]. PET-SQL(Li et al., 2024)은 Text-to-SQL 프레임워크에서 일관성과 최적화를

개선하기 위해 프롬프트 기반의 두 단계 접근법과 여러 LLM 을 사용한 Cross-consistency 로 최고 성능을 달성하였다[8].

한편, Tao et al. (2002)의 연구는 유사 SQL 하위 쿼리를 이용한 SQL Plan 최적화를 제안했다[9]. 이는 공용 데이터베이스에서 사용자가 원하는 SQL 이 이미 존재할 가능성이 높다는 점을 시사한다.

이러한 관련 연구들은 Text-to-SQL 분야의 발전 과정과 현재의 연구 동향을 보여준다. 본 연구에서 제안하는 접근법은 실제 환경에서의 적용 가능성을 높이는 새로운 방향을 제시한다.

3. 제안된 접근 방식

본 연구에서 제안하는 접근 방식은 일반적인 SQL 생성 방법에서 벗어나 RAG(Retrieval-Augmented Generation)와 유사한 검색 기반 방식을 활용한다[10]. 이 방법의 핵심 아이디어는 사전 검증된 SQL 과 이를 기반으로 만든 합성 질문, 그리고 벡터 스토어를 활용하여 사용자 질문을 가장 적절한 SQL 과 매칭하는 것이다. 제안된 방식은 준비 단계와 처리 단계의 두 주요 단계로 구성된다.

준비 단계에서는 기존 SQL 쿼리를 전처리하고 검색 가능한 벡터 스토어를 생성한다. 먼저 기존 SQL 쿼리를 변환하여 특정 값을 파라미터로 대체한다. 그 다음, 파라미터화된 각 SQL 쿼리에 대해 LLM 을 사용하여 k 개의 합성 질문을 생성한다. 마지막으로, 임베딩 모델을 사용하여 합성 질문을 벡터로 변환한 후, 이를 벡터 스토어에 저장하여 유사도 검색을 가능하게 한다.

처리 단계는 사용자가 질문을 제출할 때 시작된다. 먼저, 사용자의 질문을 준비 단계에서 사용된 것과 동일한 임베딩 모델을 사용하여 벡터 표현으로 변환한다. 그 다음, 벡터 스토어에서 가장 유사한 합성 질문을 식별하기 위해 최대 내적 연산 기반의 유사도 검색을 수행한다. 최종 단계에서는 유사도 검색의 상위 k 개 결과에서 가장 적절한 SQL 쿼리를 선택하기 위해 LLM 을 이용한 재순위화를 수행한다. 최종적으로 선택된 SQL 쿼리는 LLM 이 생성한 값으로 파라미터를 대체한다.

이 접근 방식의 장점은 검증된 SQL 을 활용함으로써 일관성과 성능을 보장하면서도, 자연어 기반의 유연성을 활용해서 검색할 수 있다는 점이다. 또한, 기존 SQL 쿼리를 재사용함으로써 새로운 SQL 을 생성하는 데 따르는 위험을 최소화하고, 데이터베이스 성능 최적화에 대한 고려사항을 자연스럽게 반영할 수 있다.

그림 1 은 제안된 접근 방식의 전체적인 흐름을 보여준다. 준비 단계에서는 검증된 SQL 집합으로부터 LLM 을 통해 합성 질문을 생성하고, 이를 벡터 스토어에 저장한다. 처리 단계에서는 사용자 질문을 입력받아 벡터 스토어에서 유사한 질문을 검색하고, LLM 을 통해 최종 SQL 을 선택 및 파라미터를 생성한 값으로 대체한다. 이러한 과정을 통해 사용자의 자연어 질문을 적절한 SQL 쿼리로 변환할 수 있다.

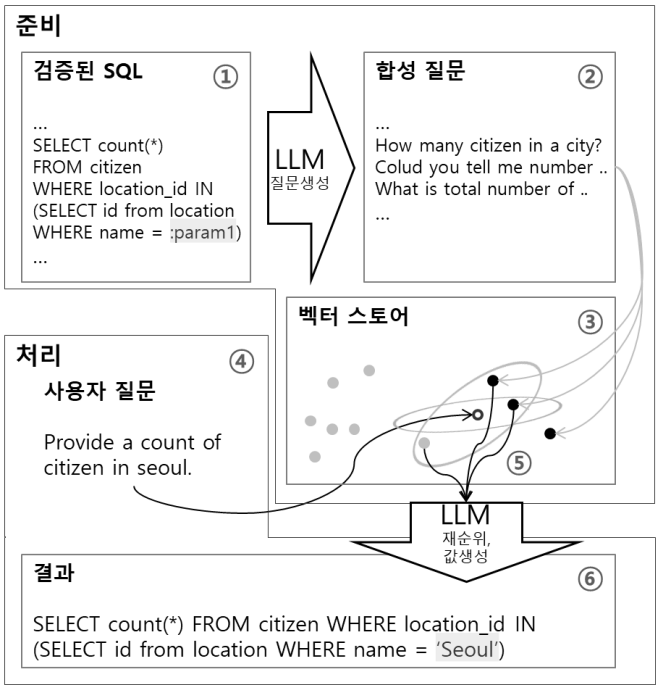

그림 1. 제안된 접근 방식의 흐름도

4. 실험 방법론

본 장에서는 제안된 SQL 검색 기반 Text-to-SQL 접근 방식의 효과성을 검증하기 위한 실험 설계에 대해 설명한다. 실험은 데이터셋 선택, 평가 지표 정의, 시스템 구성 세 가지 요소로 구성되었다.

4.1 데이터셋

본 실험에는 Spider 데이터셋을 사용하였다[11]. Spider 는 대규모 교차 도메인 Text-to-SQL 데이터셋으로, 총 10,181 개의 자연어 질문과 그에 맞는 답변 SQL 쿼리로 구성되어 있다. 이 중 평가를 위해 Spider 데이터셋의 테스트 세트를 사용했는데, 이는 1,034 개의 자연어 질문과 SQL 쿼리 쌍으로 이루어진다.

테스트 세트에 포함된 SQL 쿼리들을 검증된 SQL 집합으로 간주하여 시스템을 구성하였다. 이는 실제 환경에서 기존에 사용되던 검증된 SQL 쿼리들을 모사하기 위한 구성이다. 자연어 질문은 시스템의 성능을 평가하는 데 사용되었다.

4.2 평가 지표

시스템의 성능을 평가하기 위해 Spider 데이터셋에서 제시하는 두 가지 주요 평가 지표를 사용했다. 첫 번째는 일치 정확도로, 이는 시스템이 예측한 SQL 쿼리가 정답 SQL 쿼리와 일치하는 비율을 측정한다. 두 번째는 실행 정확도로, 시스템이 생성한 SQL 쿼리를 실제로 실행했을 때 정답 쿼리와 정확히 일치하는 결과를 산출하는 비율을 측정한다.

4.3 시스템 구성

제안된 접근 방식의 성능을 최적화하기 위해 다양한 시스템 구성을 실험했다. 주요 구성 요소와 실험 변수는 다음과 같다.

- 언어 모델: 합성 질문 생성을 위해 Llama3 8B 와 Llama3 70B 두 가지 크기의 모델을 비교 실험
- 임베딩 모델: 벡터화를 위해 all-MiniLM-L6-v2(23M)와 GIST-Embedding-v0(109M)를 사용
- 합성 질문 수: 각 SQL 쿼리에 대해 1 개에서 20 개까지의 합성 질문을 생성하며 최적의 수를 탐색
- 벡터 스토어: MIPS 기반의 유사도 검색을 지원하는 Meta 의 FAISS 를 사용
- 프롬프트 유형: 기본, 다양성, 스키마, 난이도의 네 가지 프롬프트 유형을 제안하고 실험
- SQL 전처리: 원본, 마스킹, 일반화, 마스크 채우기 등 다양한 전처리 방식을 적용하여 효과 분석

이러한 다양한 구성 요소와 변수들을 조합하여 실험을 진행함으로써, 제안된 접근 방식의 최적 성능을 도출하고자 하였다.

5. 실험 결과 및 분석

본 장에서는 제안된 접근 방식의 최적 구성을 위한 실험 결과와 성능 평가 결과를 제시한다.

5.1 최적 모델 구성

실험	최적화 방식	실험 결과
프롬프트 유형	다양성(10-shot)	75.6 → 82.0
LLM	Llama3 70B	67.7 → 72.7
임베딩 모델	GIST-Embedding-v0	74.4 → 80.9
질문 전처리	전처리 없음	-
합성 질의 수	11 개	74.6 → 83.1

표 1. 최적화를 위한 실험 결과

제안 방식을 최적화하기 위해 다양한 실험을 진행했으며, 그 결과는 표 1 에 요약되어 있다. 프롬프트 유형 실험에서는 SQL 쿼리의 파라미터 예시와 다양성을 고려한 질문을 few-shot 형태로 추가한 방식이 가장 효과적이었으며, 기존 대비 6.4%p 의 성능 향상을 보였다. 언어 모델의 경우, Llama3 70B 가 8B 모델에 비해 약 5%p 높은 성능을 나타냈다. 임베딩 모델 실험에서는 GIST-Embedding-v0 가 all-MiniLM-L6-v2 보다 약 6.5%p 우수한 결과를 보여주었다.

프롬프트 유형에서는 다양성을 높이고 합성 질문을 일반화하는 방식이 높은 성능을 보였다. 반면, 사용자 질문 전처리에서는 전처리를 하지 않는 방식이 가장 높은 정확도를 달성했다. SQL 특성 상 질문으로 변환하는 과정에서는 일반화 대상이 명확했다. 그러나 사용자 질문을 일반화하는 과정에서는 일반화 대상이 불명확했다. 예를 들어, 사용자 질문 '서울의 인구는 얼마인가요?'에서 일반화 대상이 '서울'인지 '인구'인지 모호했다. 이로 인해 일반화를 통한 사용자 질문 전처리 시 정확도가 10.5%p 감소했다. 합성 질의 개수에 관해서는, 3 개까지는 정확도가 선형적으로 상승했으나 그 이후로는 정체되며 11 개 이후 감소하는 양상을 보였다. 이러한 실험을 통해 각 요소별 최적의 구성을 도출할 수 있었으며, 이를 바탕으로 전체 시스템의 성능을 최적화할 수 있었다.

5.1 성능 결과

		개발(%)		실험(%)	
		일치	실행	일치	실행
미세조정	T5-SR[12]	77.2	-	**72.4**	-
	RESDSQL-3B+NatSQL[13]	**80.5**	**84.1**	72.0	79.9
LLM 기반	PET-SQL	-	-	66.6	**87.6**
	DAIL-SQL+GPT-4+Self-Consistency	-	83.6	-	86.6
	제안 접근법	-	-	92.9	89.5

표 2. 기존 모델과 제안 접근법 정확도 비교

표 2 는 제안된 모델과 기존 최신 모델들의 성능을 비교한 결과를 보여준다. 본 연구의 모델은 92.9%의 일치 정확도와 89.5%의 실행 정확도를 달성했다. 이는 학습을 통한 미세조정 모델(84.1%) 수치나 GPT-4 를 사용한 LLM 기반 모델(87.6%) 같은 최신 모델들보다 높은 성능이다. 특히 주목할 점은, 본 연구에서는 사전 학습된 Llama3 70B 모델과 109M 파라미터의 GIST-Embedding-v0 임베딩 모델만을 사용했다는 점이다. 이는 GPT-4 나 학습을 필요로 하는 다른 모델들에 비해 상대적으로 적은 자원으로 우수한 성능을 달성했음을 의미한다. 다만, 실험 데이터의 SQL 쿼리를 사용하여 실험 환경이 다르기 때문에 이 결과는 실험이 제시한 특정 상황에서만 의미를 가진다.

추가적으로, 생성 모델은 동일 결과를 산출하는 다양한 SQL 쿼리가 존재하여 실행 정확도가 일치 정확도를 상회한다. 반면, 제안된 접근법은 검증된 SQL 쿼리의 표현만을 사용하므로 일치 정확도가 실행 정확도를 초과한다. 이 차이는 LLM 이 파라미터 값 생성 시 대소문자 구분이나 유사 표현(예: 'Male', 'male', 'M') 간 혼동에서 기인한다. 테이블 컬럼 내 데이터를 프롬프트에 추가시키는 등의 기법을 통해 실행 정확도를 일치 정확도 수준으로 향상시킬 수 있다.

5.2 SQL 쿼리 난이도에 따른 성능

난이도	PET-SQL	제안 접근법	Δ
쉬움	**93.2**	92.7	-0.5
중간	90.7	**94.4**	+3.7
어려움	84.9	**93.1**	+8.2
매우 어려움	75.9	**89.2**	+13.3

표 3. SQL 난이도에 따른 정확도 비교

SQL 쿼리 난이도별 성능 분석 결과는 표 3 에 제시되어 있다. Spider 데이터셋의 난이도 분류에 따른 비교 결과, 제안 방식은 난이도 증가에 따라 최신 모델들과의 성능 격차가 확대되었다. 쉬움 난이도에서는 SoTA 모델 대비 0.5%p 낮은 성능을 보였으나, 매우 어려움 난이도에서는 13.3%p 높은 성능을 달성했다.

SoTA 모델의 쉬움과 매우 어려움 난이도 간 성능 차이가 17.3%p 인 반면, 제안 방식은 3.5%p 에 그쳤다. 이는 제안된 검색 기반 방식이 생성 기반 방식보다 난이도에 따른 성능 저하가 현저히 적음을 시사한다.

이러한 결과는 제안된 SQL 검색 기반 접근법이 Text-to-SQL 작업에서 높은 정확도와 안정성을 제공하며, 특히 복잡한 쿼리에 대한 우수한 성능이 실제 응용 환경에서의 유효성을 입증한다.

6. 논의 및 결론

본 연구에서 제안한 SQL 검색 기반 Text-to-SQL 접근 방식은 기존의 SQL 생성 방식과 차별화된 관점에서 문제를 해결하고자 하였다. 실험 결과를 통해 이 접근 방식의 효과성과 잠재력이 확인되었으며, Text-to-SQL 분야에 새로운 시사점을 제공한다.

주요 연구 결과는 다음과 같다. Spider 데이터셋에서 92.9%의 정확한 일치 정확도와 89.5%의 실행 정확도를 달성했다는 점과 복잡한 쿼리에서 더 높은 성능 향상을 보였다. 특히 매우 어려움 난이도에서 기존 SoTA 모델 대비 13.3%p 높은 성능을 보여주었다.

이러한 결과는 실제 업무 환경에서의 적용 가능성을 시사한다. 사전 검증된 SQL 쿼리를 활용함으로써 생성된 SQL 쿼리의 정확성과 안정성을 보장할 수 있으며, 기존 모델을 도입하기 위한 질문과 SQL 데이터셋을 제작하는 데 드는 자원을 절감할 수 있다.

그러나 본 접근 방식에는 명확한 한계점이 존재한다. 기존 쿼리에 대한 의존성으로 인해 새로운 형태의 쿼리가 필요한 경우 해답을 제공할 수 없다. 또한, 사용자의 질문이 모호하거나 특수한 도메인 지식이 필요한 경우 정확한 매칭에 어려움이 있을 수 있다.

이러한 한계점을 고려하여 다음과 같은 향후 연구 방향을 제시한다. 첫째, 검색 기반 방식과 생성 기반 방식을 결합한 하이브리드 모델을 개발하여 기존 쿼리로 해결 불가능한 새로운 유형의 질문에도 대응할 수 있다. 둘째, 사용자 상호작용을 통한 점진적이고 지속적인 시스템 성능 개선 방안을 모색할 수 있다. 셋째, 모호하거나 특수 도메인 지식에 대응하기 위해 사용자 질문 전처리 방식이나, 합성 질문 생성 방법을 연구할 수 있다. 이러한 접근은 시스템의 적용 범위를 확장하고 정확도를 향상시키는 데 기여할 것으로 기대된다.

결론적으로, 본 연구에서 제안한 SQL 검색 기반 Text-to-SQL 접근 방식은 높은 정확도와 효율성을 보여주며, 특히 복잡한 쿼리에 대해 강점을 가진다. 이는 실제 업무 환경에서 Text-to-SQL 시스템의 적용 가능성을 크게 높여준다. 향후 연구에서는 이 접근 방식의 한계를 극복하고 더욱 발전된 형태의 시스템을 개발하는 데 초점을 맞출 필요가 있다. 이를 통해 자연어로 데이터베이스와 상호작용하는 방식이 더욱 보편화되고, 궁극적으로는 데이터 활용의 장벽을 없애는 것에 기여할 수 있을 것이다.

참고문헌

[1] Gao, D. et al., "Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation", arXiv preprint arXiv:2308.15363, 2023.

[2] Pedro, R. et al., "From prompt injections to sql injection attacks: How protected is your llm-integrated web application?", arXiv preprint arXiv:2308.01990, 2023.

[3] Khushairi, N.M. et al., "Database performance tuning methods for manufacturing execution system", World Appl. Sci. J, 30(30), 91-99, 2014.

[4] Li, F. and Jagadish, H.V., "Constructing an interactive natural language interface for relational databases", Proc. VLDB Endowment, 8(1), 73-84, 2014.

[5] Xu, X. et al., "Sqlnet: Generating structured queries from natural language without reinforcement learning", arXiv preprint arXiv:1711.04436, 2017.

[6] Yu, T. et al., "Syntaxsqlnet: Syntax tree networks for complex and cross-domaintext-to-sql task", arXiv preprint arXiv:1810.05237, 2018.

[7] Scholak, T. et al., "PICARD: Parsing incrementally for constrained auto-regressive decoding from language models", arXiv preprint arXiv:2109.05093, 2021.

[8] Li, Z. et al., "PET-SQL: A Prompt-enhanced Two-stage Text-to-SQL Framework with Cross-consistency", arXiv preprint arXiv:2403.09732, 2024.

[9] Tao, Y. et al., "Exploiting common subqueries for complex query optimization", Proc. 2002 Conf. Centre for Advanced Studies on Collaborative research, 12, 2002.

[10] Lewis, P. et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks", Adv. Neural Inf. Process. Syst., 33, 9459-9474, 2020.

[11] Yu, T. et al., "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task", arXiv preprint arXiv:1809.08887, 2018.

[12] Li, Y. et al., "T5-SR: A Unified Seq-to-Seq Decoding Strategy for Semantic Parsing", ICASSP 2023, 1-5, 2023.

[13] Li, H. et al., "Resdsql: Decoupling schema linking and skeleton parsing for text-to-sql", Proc. AAAI Conf. Artif. Intell., 37(11), 13067-13075, 2023.