Journal of KIIT. Vol. 24, No. 7, pp. 73-80, Jul. 31, 2026. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2026.24.7.73

도구 기반 LLM 에이전트의 자연어 질의 사전 검증 및 컨텍스트 토큰 감축 방안

변공규*, 최권택**, 유선진***

Pre-Validation of Natural Language Queries using Tool-Augmented LLM Agents with Context Token Reduction

Gongkyu Byeon*, Kwon-Taeg Choi**, and Sunjin Yu***

이 논문은 2025~2026년도 국립창원대학교 자율연구과제 연구비 지원으로 수행된 연구결과임
또한, 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No.RS-2024-00345763)

요 약

본 연구는 자연어 질의 검증을 SQL 생성 이전 단계의 독립적 문제로 정의하고, LLM과 외부 검증 도구를 결합한 에이전트 기반 사전 검증 방안을 제안한다. 제안 방법은 테이블 및 필드명을 스키마 비교로 판별하고, WHERE 조건절 값은 함수 기반 도구를 통해 선택적으로 검증함으로써 평균 컨텍스트 토큰을 기존 대비 약 92% 절감(11,382개 → 880개)하는 최적화를 달성하였다. 실험 결과, GPT-5-mini는 외부 도구 호출 수를 545회로 줄이고 효율성을 68.3%까지 향상시켜, GPT-4o-mini 대비 규칙 준수 및 운영 경제성 측면에서 우수한 성능을 나타냈다. 이러한 결과는 질의 사전 검증 단계가 Text-to-SQL 시스템의 신뢰성과 정확도를 향상하는 핵심 모듈이 될 수 있음을 시사한다.

Abstract

This study defines natural language query verification as an independent problem prior to SQL generation and proposes an agent-based pre-verification approach that combines Large Language Models (LLMs) with external verification tools. The proposed method identifies table and field names through schema comparison and selectively verifies WHERE clause values using function-based tools, achieving an optimization that reduces the average context token count by approximately 92%(from 11,382 to 880 tokens). Experimental results demonstrate that GPT-5-mini reduces the number of external tool calls to 545 and improves efficiency up to 68.3%, exhibiting superior performance over GPT-4o-mini in terms of rule compliance and operational cost-effectiveness. These findings suggest that the query pre-verification stage can serve as a core module for enhancing the reliability and accuracy of Text-to-SQL systems.

Keywords

natural language query validation, text-to-SQL, large language models, context optimization, artificial intelligence

* 국립창원대학교 문화융합기술협동과정 박사과정
- ORCID: https://orcid.org/0000-0001-8883-2925
** 강남대학교 소프트웨어응용학부 교수
- ORCID: http://orcid.org/0000-0001-5331-321X
*** 국립창원대학교 인공지능공학과 교수(교신저자)
- ORCID: https://orcid.org/0000-0001-9292-4099

· Received: Apr. 21, 2026, Revised: Jun. 22, 2026, Accepted: Jun. 25, 2026
· Corresponding Author: Sunjin Yu
Dept. of AI Engineering, 20 Changwondaehak-ro, Uichang-gu, Changwon-si, Gyeongsangnam-do, 51140, Korea
Tel.: +82-55-213-3098, Email: sjyu@changwon.ac.kr

Ⅰ. 서 론

자연어 기반 데이터 질의는 비전문가도 데이터베이스를 직접 활용할 수 있게 하는 핵심 기술이며, Text-to-SQL(Structured Query Language)은 이러한 요구를 실현하는 대표적 접근 방식이다. 최근 LLM(Large Language Model)의 발전으로 Text-to-SQL 시스템은 파인튜닝 없이도 높은 수준의 질의 해석 및 SQL 생성 성능을 보이고 있으며, 복잡한 데이터베이스 스키마와 다양한 사용자 표현을 처리할 수 있는 방향으로 빠르게 발전하고 있다[1]-[3].

그러나 실제 사용 환경에서 입력 질의는 항상 정제된 형태로 주어지지 않는다. 질의에는 오타, 존재하지 않는 테이블이나 필드의 참조, 실제 데이터베이스에 존재하지 않는 조건 값, 비현실적인 수치 조건, 혹은 시간 순서가 맞지 않는 제약이 포함될 수 있다. 이러한 오류는 표면적으로는 사소해 보일 수 있으나, SQL 생성 과정에서는 의미적으로 잘못된 질의를 정당화하는 환각의 직접적 원인이 된다.

기존의 연구들은 주로 생성된 SQL의 문법적 오류를 수정하거나 실행 결과의 일관성을 검증하는 후처리 방식에 집중해 왔다[1]. 하지만 입력 단계의 오류가 걸러지지 않으면 의미적 오류를 완전히 배제하기 어렵다. 일부 사전 검증 시도는 데이터베이스의 모든 고윳값을 프롬프트에 포함하여 모델이 판단하게 했으나, 데이터 크기에 비례해 컨텍스트 토큰이 폭증하는 결과를 초래한다. 결론적으로, 비용 상승뿐만 아니라 모델의 컨텍스트 윈도우 한계를 초과하여 시스템 작동을 불가능하게 만든다.

따라서 본 연구는 자연어 질의 검증을 SQL 생성 이전 단계의 독립적 문제로 정의하고, 컨텍스트 최적화를 통해 시스템의 신뢰성과 운영 효율성을 동시에 확보하는 것을 목적으로 한다. 본 논문에서 제시하는 주요 연구 내용과 기여는 아래와 같다.

첫 번째로, 스키마 목록을 통한 직접 비교와 외부 도구를 활용한 선택적 검증을 명확히 분리한 프레임워크를 설계하였다. 이를 통해 질의 검증 단계가 단순한 전처리를 넘어, 생성되어서는 안 되는 오류 질의를 선제적으로 차단하는 독립적인 핵심 모듈로 기능하도록 하였다.

두 번째로, 데이터 주입 방식에 따른 토큰 소모량을 분석하여 본 프레임워크의 공헌도를 입증하였다. 모든 고윳값을 프롬프트에 직접 삽입하는 기존 방식은 평균 11,382개의 토큰을 소모하여 모델의 한계를 초과하는 문제가 발생했으나, 제안된 에이전트 구조는 필요한 시점에만 외부 검색 도구를 호출함으로써 평균 컨텍스트 토큰을 880개 수준으로 약 92% 이상 감축시켰다.

세 번째로, 기존 딕셔너리 기반 방식이 대용량 데이터에서 오타 검출 시 연산 병목을 유발하는 한계를 극복하기 위해 벡터 임베딩 기반 검색 도구를 도입하였다. 코사인 유사도 탐색을 통해 데이터베이스 내에 입력값이 존재하지 않더라도 의미적으로 유사한 수정 후보를 제안할 수 있는 실무적인 검증 환경을 구축하였다.

네 번째로, 단순 정확도 외에 도구 호출 효율성을 부가 지표로 설정하여 분석하였다. 실험 결과, GPT-4o-mini는 불필요한 도구 호출로 19.7%의 낮은 효율성을 보인 반면, GPT-5-mini는 호출 수를 545회로 줄이고 효율성을 68.3%까지 향상시켜 정책 준수 및 비용 절감 측면에서 성능을 증명하였다.

Ⅱ. 관련 연구

Text-to-SQL은 자연어 질의를 데이터베이스 질의 언어로 변환하는 의미론적 파싱의 한 분야로, 자연어처리와 데이터베이스 연구의 교차 영역에서 오랫동안 중요한 주제로 다뤄져 왔다. 특히, Spider 데이터셋의 등장은 복잡한 SQL 구조와 교차 도메인 일반화를 평가할 수 있는 표준 벤치마크를 제공함으로써 관련 연구를 가속하였다[3]. 최근에는 프롬프트 기반 추론, 인컨텍스트 러닝, 외부 스키마 정보 활용, LLM 기반 후처리 보정 등을 통해 파인튜닝 없이도 높은 성능을 달성하는 방향으로 연구가 전개되고 있다[4]-[6].

한편 기존 연구의 공통적 한계는 입력 자연어 질의가 이미 의미적으로 타당하고, 데이터베이스 스키마에 존재하는 엔티티만을 참조한다는 가정에 있다. Spider를 비롯한 주요 벤치마크 역시 정제된 질의-정답 쌍 중심으로 구성되어 있어, 실제 사용 환경에서 빈번히 발생하는 오타, 비존재 엔티티 참조, 잘못된 조건 값, 비현실적 제약 조건 등의 문제를 충

분히 반영하지 못한다[7].

또한 일부 연구는 SQL 실행 과정에서 발생하는 구문 오류나 실행 오류를 수정하거나, 실행 결과를 바탕으로 SQL을 재작성하는 후처리 기반 접근을 제안하였다[8]. 이러한 방법은 실행 불가능한 SQL을 줄이고 정답률을 향상시키는 데 기여할 수 있으나, 입력 자연어 질의 자체가 잘못된 경우에는 한계를 갖는다. 즉, 생성된 SQL이 문법적으로 타당하고 실행 가능하더라도, 사용자의 의도와 일치하지 않는 의미적 오류가 남을 수 있다.

최근 LLM의 환각 문제를 완화하기 위해 RAG (Retrieval-Augmented Generation), 외부 도구 호출, 함수 기반 검증 등 다양한 방법이 제안되고 있으나, 이들 또한 대체로 모델의 출력 결과를 보완하는 데 초점을 둔다. 이에 비해 본 연구는 입력 오류가 환각적 SQL 생성의 근본 원인 가운데 하나라는 점에 주목하며, SQL 생성 이전 단계에서 자연어 질의의 유효성을 선별하는 문제에 접근한다.

III. 제안 시스템

3.1 문제 정의 및 전체 구조

본 연구에서 다루는 자연어 질의 검증 문제는 입력된 자연어 질의가 데이터베이스 스키마 및 의미 제약 조건과 정합적으로 대응 가능한지를 SQL 생성 이전 단계에서 판별하는 문제로 정의된다. 여기서 핵심은 검증의 목적이 질의를 곧바로 SQL로 변환하는 데 있지 않고, 해당 질의가 SQL로 변환 가능한 유효한 입력인지를 먼저 선별함으로써 모델의 환각 생성을 원천적으로 차단하는 데 있다. 본 연구가 제안하는 사전 검증 프레임워크의 핵심은 에이전트 기반의 '계층적 외부 도구 선택 호출' 구조이다. 본 프레임워크는 모든 데이터를 모델에 주입하는 대신, 입력된 질의에 대해 스키마 목록을 통한 직접 비교(테이블 및 필드명 검증)와 외부 함수 도구를 활용한 선택적 검증(WHERE 조건절 값 검증)을 분리하여 처리한다. 즉, 제안 시스템의 전체 파이프라인은 '자연어 질의 입력 → 에이전트 기반 계층적 오류 검증(스키마 대조 및 search_value 외부 도구 호출) → 정제된 컨텍스트 기반 의미 파싱(SQL 생성)'의 구조를 따른다. 이를 통해 질의 검증 단계를 독립적인 프레임워크로 확립하여 모델의 컨텍스트 한계 초과를 방지하고 의미적 환각을 선제적으로 차단한다.

3.2 오류 유형 체계

본 연구는 실제 환경에서 빈번히 발생할 수 있는 입력 오류를 네 가지 유형으로 아래 표 1과 같이 체계화하였다. 첫째, 오타 오류는 철자 수준의 실수로 테이블, 필드, 값이 잘못 표기된 경우이다. 둘째, 비존재 엔티티 오류는 스키마나 데이터에 존재하지 않는 요소를 참조하는 경우로, 테이블, 필드, WHERE 조건절 값 오류로 세분화한다. 셋째, 수치 제약 조건 위반 오류는 허용 범위를 벗어난 조건을 의미하며, 넷째, 시간 제약 조건 위반 오류는 시간 순서가 모순되거나 데이터 범위를 벗어나는 경우를 뜻한다. 이러한 구분은 오류 유형에 따라 필요한 검증 근거를 최적화하기 위함이다. 테이블과 필드명은 스키마 비교만으로 판별하여 불필요한 도구 호출을 방지하고, WHERE 조건절 값은 실제 데이터 존재 여부를 확인하기 위해 별도의 도구를 사용하는 등 검증 프로세스의 효율성을 높였다.

표 1. 입력 오류 유형 체계 및 유형별 검증 근거
Table 1. Error typology and primary validation grounds

Error type	Sub-category	Primary validation grounds	Tool required
Typos	Misspelling of tables/fields/ values	Orthographic similarity & schema/value consistency	Optional
Non -existent entities	Table, field, WHERE value	Schema comparison or actual value existence	Only for WHERE values
Numerical constraint errors	Out of range, unrealistic conditions	Domain range & logical rules	Not required
Temporal constraint errors	Chronological reverse, out of range	Temporal order & timeline consistency	Not required

3.3 함수를 활용한 search_value 도구 설계

본 프레임워크의 핵심 도구는 WHERE 조건절에 사용된 값의 실재 여부를 확인하는 search_value 함수이다. 가능한 경우의 수가 매우 많은 조건절 값을 모두 모델 컨텍스트에 포함시키는 방식은 토큰 한계를 초과하여 불가능에 가깝기에, 이를 별도의 외부 도구로 분리하여 컨텍스트 최적화를 달성하였다. 본 연구에서는 세 가지 구현 방식을 비교 실험하였으며, 그 결과 완전 일치만 가능한 딕셔너리 기반 방식의 연산 병목과 오타 검출 한계를 확인하였다. 반면, 벡터 임베딩 기반 검색은 코사인 유사도를 통해 데이터베이스 내에 입력값이 없더라도 유사한 수정 후보를 제안할 수 있다. 구체적으로 search_value(table, column, keyword) 도구는 입력된 키워드와 해당 컬럼 내 값들의 임베딩 벡터 간 유사도를 계산한다. 에이전트는 사전 설정된 유사도 임계값을 초과하는 상위 K개(Top-K)의 후보 문자열 리스트를 반환받게 되며, 반환된 후보군이 없을 경우 완전한 비존재 오류로 판정하고, 후보군이 존재할 경우 질의의 문맥을 고려하여 가장 적절한 값으로 오타를 자동 수용하도록 논리적 제어 로직이 설계되었다

3.4 도구 호출 정책

ReAct 기반 에이전트의 자율성은 사실 검증 능력을 높이는 동시에 과도한 API 호출로 인한 오버헤드를 유발할 수 있다. 이를 제어하기 위해 본 연구는 도구 사용 범위를 명시적으로 제한하는 정책을 설계하였다. 핵심 원칙으로 테이블 및 필드 오류는 스키마 목록 비교로 즉시 차단하여 도구 호출을 금지하며, 오직 WHERE 조건절의 텍스트 값 검증에만 search_value를 활용하도록 제한한다. 수치 및 시간 제약 역시 논리 규칙과 메타데이터를 통해 판단하여 불필요한 호출 대상에서 제외한다. 이 정책은 에이전트가 ‘언제 도구를 사용해야 하는가’에 대한 규칙 준수 여부를 평가하는 기준이 되며, 실험을 통해 모델의 추론 성능에 따른 효율성 차이를 측정하는 핵심 지표로 활용된다.

IV. 실 험

4.1 실험 환경 및 오류 주입 데이터셋 구성

본 연구는 자연어 질의가 SQL 구조로 파싱되기 이전 단계에서 스키마 및 의미 제약 조건 위반 여부를 검증하는 기법의 실효성을 평가한다. 이를 위해 Spider 데이터셋을 기반으로 정상 질의와 동일한 비율의 오류 질의(오타, 비존재 엔티티, 수치 제약, 시간 제약)를 강제 주입하여 검증용 데이터셋을 구성하였다. 평가에 사용된 데이터셋은 Spider 데이터셋 전체가 아니며, 데이터베이스의 스키마 복잡도와 도메인 다양성을 균형 있게 반영할 수 있도록 7개의 대표적인 데이터베이스 환경을 무작위 추출하여 실험용 타겟 데이터셋으로 한정하였다. 평가 모델로는 상용적으로 사용하고 있는 GPT-4o-mini, GPT-5-mini 2종의 LLM을 선정하였으며, 평가 공정성을 위해 파인튜닝 없이 통일된 프롬프트와 검증 프레임워크 제어 환경을 적용하였다. 주요 평가 지표로는 단순 정확도보다 실제 환각 방지에 직결되는 오류 유형별 재현율을 핵심 지표로 설정하였으며, 에이전트 중심 구조의 실제 운영 가능성을 분석하기 위해 도구 호출 수, 효율성, 오버헤드 비율, 샘플별 효율성을 함께 측정하였다.

4.2 실험 1: 컨텍스트 최적화 및 search_value 도구 최적화

전체 데이터베이스의 고윳값을 LLM의 텍스트 컨텍스트에 직접 삽입하는 방식은 토큰 한계 초과와 성능 저하를 초래한다. 아래 표 2와 같이 본 연구는 이를 해결하기 위해 에이전트가 필요할 시기에 외부 도구를 호출하는 구조를 설계하여 평균 컨텍스트 토큰을 기존 11,382개에서 880개 수준으로 약 92% 감소시키는 성과를 거두었다. 특히 비교군으로 설정된 '10-Sample Selection'은 각 데이터베이스 컬럼별로 존재하는 전체 고윳값 중에서 대표성을 띠는 상위 빈도값이 아닌, 데이터의 다양성을 반영하기 위해 무작위로 10개씩을 추출하여 프롬프트에 고정 삽입하는 방식을 의미한다. 도구의 최적 구

현 방식을 결정하기 위해 7개의 DB 환경에서 세 가지 방식을 비교한 결과, 딕셔너리 기반 검색은 완전 일치 판별에는 유리하나 대용량 데이터에서 오타 검출 연산 병목이 발생하였다. 단순 문자열 필터는 속도 면에서 장점이 있으나 동의어나 철자 오타를 안정적으로 검출하지 못했다. 반면 벡터 임베딩 검색은 코사인 유사도 기반 탐색을 통해 데이터베이스에 없는 값에 대해서도 의미적 교정 제안을 제공할 수 있어 정확도와 확장성 측면에서 가장 실용적인 구현 방식으로 확인되었다.

표 2. 데이터 주입 방식에 따른 평균 컨텍스트 토큰 소모량 비교
Table 2. Comparison of average context token consumption by data injection method

Data injection method	Avg. token count	Min tokens	Max tokens	Accuracy & result analysis
Full sampling (Baseline)	11,382	789	99,148	Exceeds model context limits & causes bottlenecks
10 -sample selection	1,376	771	2,556	Incomplete validation due to data constraints
Statistical info only	1,055	756	1,703	Incapable of identifying text errors (e.g., typos)
Proposed agent framework	880	647	1,359	Maintains high accuracy via embedding search

4.3 실험 2: 전체 검출 성능 및 오류 및 위치 정확도 분석

자연어 질의에 내포된 4가지 주요 오류 유형(오타, 비존재 엔티티, 수치 제약, 시간 제약)에 대한 LLM의 검출 능력을 심층적으로 평가하기 위해, 모델별로 오류 유형 및 오류 위치에 따른 세부 정확도를 표 3과 같이 분석하였다.

첫 번째, 전체 검출 성능 비교에서 종합적인 검출 성능을 분석한 결과, GPT-5-mini는 GPT-4o-mini 대비 높은 성능 향상을 보였다. GPT-4o-mini는 주입된 오류에 대해 오류 유형 정확도 83.4%, 오류 위치 정확도 59.4%를 기록하는 데 그쳤다. 반면, GPT-5-mini는 오류 유형 정확도를 95.3%까지 끌어올렸으며, 특히 오류 위치 정확도에서 기존 대비 약 33.3% 향상된 92.7%를 달성하여 오류가 발생한 지점을 식별해 내는 신뢰성을 입증하였다.

두 번째, 오류 유형별 검출 한계선 분석에서 오타 항목은 기존 분석에서 오타 오류는 전반적으로 높은 검출 성능을 보인다고 확인된 바와 같이, GPT-4o-mini(93.1%)와 GPT-5-mini(95.3%) 모두 90% 이상의 우수한 정확도를 유지하였다. 비존재 엔티티 항목은 LLM이 사전 학습된 일반 지식과 주어진 DB의 외재적 스키마 지식을 혼동하여 가장 취약한 오류 유형으로 관찰되었던 영역이다. 실제로 통계 지표상 GPT-4o-mini는 비존재 엔티티 검출 정확도가 30.8%에 불과하여 환각에 매우 취약한 한계를 노출했다. 그러나 GPT-5-mini는 96.1%로 대폭 개선하며, 비존재 엔티티로 인한 환각 문제를 원천적으로 차단하는 능력이 향상되었음을 보여주었다. 수치 및 시간 제약은 명시적 제약 조건 간의 논리적 불일치를 판단해야 하며, GPT-4o-mini는 수치 제약 28.9%, 시간 제약 66.7%의 성능을 기록했다. 반면 GPT-5-mini는 수치 제약 86.4%, 시간 제약 100.0% 검출률을 달성하며, 언어적 자연스러움을 넘어선 규칙 기반의 논리 추론 능력이 안정적으로 작동함을 입증했다.

세 번째, 오류 위치별 검증 정밀도 분석에서 본 프레임워크는 테이블과 필드명 오류는 스키마 목록 비교로 판별하고, WHERE 조건절 값 오류는 별도의 search_value 도구를 선별적으로 사용하는 체계로 설계되었다. 위치 파악 정확도에서 GPT-4o-mini는 테이블 위치 인지율이 19.5%에 불과하고 WHERE 조건절 판별 역시 54.8%로 저조하여, 검증 초점이 필드(80.4%)에만 과도하게 편중되는 양상을 보였다. 반면 GPT-5-mini는 테이블(92.5%), 필드(92.5%), WHERE 조건절(93.3%) 모든 영역에서 높은 정확도를 기록하였다. GPT-5-mini가 프롬프트 상의 제어 정책을 일관되게 준수하여 도구 사용 시점과 스키마 확인 시점을 명확히 분리함으로써, 각 위치별 검증 근거를 매핑하고 시스템을 고도화했음을 뒷받침한다.

표 3. GPT-4o-mini 및 GPT-5-mini 오류 탐지 성능 분석
Table 3. Error Detection Performance Analysis of GPT-4o-mini and GPT-5-mini

Category	Subcategory	GPT-4o-mini	GPT-5-mini
Overall	Injected errors	1384	1495
	Detected errors	1400	1856
	Type accuracy	83.4% (1154/1384)	95.3% (1425/1495)
	Location accuracy	59.4% (822/1384)	92.7% (1386/1495)
Error type	entity_typo	93.1% (1067/1146)	95.3% (1037/1088)
	entity_nonexist	30.8% (20/65)	96.1% (323/336)
	numeric_range	28.9% (37/128)	86.4% (38/44)
	temporal_range	66.7% (30/45)	100.0% (27/27)
Location	table	19.5% (58/297)	92.5% (540/584)
	field	80.4% (529/658)	92.5% (456/493)
	where	54.8% (235/429)	93.3% (390/418)

4.4 에이전트의 도구 호출 효율성 분석

본 연구에서는 제안된 프레임워크 내에서 에이전트가 도구 사용 정책을 얼마나 엄격히 준수하는지 평가하기 위해, 도구 호출 효율성 분석을 수행하였다. 이를 위해 먼저 전체 실험 데이터셋 내에서 이론적으로 도구 호출이 반드시 필요한 시점과 불필요한 시점을 구분하여 분석의 기준이 되는 기준을 아래 표 4와 같이 구성하였다.

표 4. 도구 호출 효율성 분석을 위한 기준 구성
Table 4. Ground truth configuration for tool-calling efficiency analysis

Category	Total entity errors	WHERE errors (Tool required)	Table errors (Tool not required)	Field errors (Tool not required)
Count	1,573	372	659	542

위 구성에 따르면, 전체 1,573건의 엔티티 오류 중 where 조건절 값 오류에 해당하는 372건만이 search_value 도구 호출이 필요한 대상이다. 반면, 테이블(659건) 및 필드(542건) 관련 오류는 에이전트가 보유한 스키마 정보만으로도 판별하므로, 도구를 호출하지 않는 것이 이상적이다.

따라서 본 실험에서는 '이상적 도구 호출 수'인 372회를 기준으로, 실제 각 모델이 수행한 호출 횟수와의 차이를 측정함으로써 효율성과 오버헤드 비율을 산출하였다. 이러한 기준 데이터 구성은 단순히 오류를 찾아내는 정확도를 넘어, 에이전트가 자원을 얼마나 최적화하여 사용하는지를 정량적으로 평가하는 핵심 지표가 된다.

4.4.1 모델별 도구 호출 행태 분석

표 5와 같이 GPT-4o-mini는 총 엔티티 오류 1,573건(WHERE 372건, 테이블 659건, 필드 542건)에 대해 총 1,887회의 도구를 호출하여 19.7%의 낮은 효율성을 보였다. 원인으로는 스키마만으로 판단 가능한 필드/테이블 오류까지 도구를 활용하여 검증하려는 과다 호출 패턴에 기인하며, 결과적으로 80.3%의 높은 오버헤드 비율을 기록하였다. 반면 GPT-5-mini는 실제 도구 호출 수를 545회로 억제하며 68.3%의 향상된 효율성을 달성하였다. 오버헤드 호출수는 173회로 급감하였으며, 완전 효율성 샘플 수는 241개로 집계되어 프롬프트 상의 제어 정책을 GPT-4o-mini보다 훨씬 일관되게 준수함을 증명하였다.

표 5. GPT-4o-mini와 GPT-5-mini의 도구 호출 효율성 성능 비교
Table 5. Comparison of tool-calling efficiency performance between GPT-4o-mini and GPT-5-mini

Performance metric	GPT-4o-mini	GPT-5-mini	Improvement
Actual calls	1,887	545	71% reduction
Efficiency (%)	19.7%	68.3%	3.5x improvement
Overhead calls	1,515	173	88.6% reduction
Perfect efficiency	150/979	241/380	4x improvement

4.4.2 종합 비교 및 시사점

모델 간 비교 분석 결과, GPT-5-mini는 GPT-4o-mini 대비 도구 호출 수를 71% 감소시키고 효율성을 3.5배 향상했으며, 오버헤드는 88.6% 줄였

다. 완전 효율성 또한 약 4배 개선되어 도구 사용 규칙 준수 측면에서 안정된 행동을 보였다. 두 언어 모델 간의 검출 성능 및 도구 호출 효율성 비교는 특정 모델의 단편적 우수성을 강조하기 위함이 아니다. 본 프레임워크가 제안하는 '계층적 에이전트 제어 전략'이 실효성을 거두기 위해서는 LLM이 주어진 도구 호출 정책을 엄격히 준수하는 능력이 필수적이며, 이를 통계적으로 입증하기 위한 보조 지표로서 모델 비교를 수행한 것이다. 결론적으로 단순 스키마 대조는 경량 모델이 전처리하고, 복잡한 제약 위반이나 모호한 사례는 규칙 준수율이 높은 고성능 모델에 위임하는 하이브리드 전략이 프레임워크의 운영 경제성을 극대화할 수 있다. 결론적으로 상용 검증 파이프라인 설계 시, 모든 질의를 고성능 모델에 위임하기보다 간단한 스키마 오류는 경량 모델과 하드 필터로 우선 처리하고, 복잡한 값 검증이나 모호한 사례만 고성능 모델로 위임하는 계층적 하이브리드 제어 전략이 운영 효율성 측면에서 실용적일 것으로 판단된다.

V. 결 론

본 논문은 자연어 질의 검증을 SQL 생성 이전 단계의 독립적인 문제로 규정하고, 대규모 언어 모델과 외부 검증 도구를 결합한 에이전트 기반 사전 검증 프레임워크를 제안하였다. 기존 Text-to-SQL 연구가 주로 SQL 생성 이후의 후처리 보정에 집중해 온 것과 달리, 본 연구는 입력 질의의 유효성을 먼저 판별하는 단계가 시스템 신뢰성 확보와 환각 방지에 필수적임을 강조하였다. 실험 결과, 제안한 프레임워크는 오타, 비존재 엔티티, 수치 및 시간 제약 위반 등 다양한 오류 유형에 대해 우수한 사전 검출 성능을 보였다.

본 프레임워크는 질의 처리 시간의 단축보다는, 대규모 데이터 주입 시 발생하는 컨텍스트 윈도우 초과 현상을 방지하고 의미적 오류를 선제적으로 차단하는 데 목적이 있다. 외부 도구 호출에 따른 일부 지연 시간이 존재하지만, 평균 컨텍스트 토큰을 약 92%(11,382개 → 880개) 절감함으로써 토큰 폭증으로 인한 운영 비용을 억제하고 시스템의 안정성과 경제성을 극대화하였다는 점에서 실용적 의의가 크다.

비록 질의 검증과 SQL 생성 단계 간의 종단간 평가는 향후 과제로 남겨두었으나, 사전 검증 단계의 고도화가 최종 성능에 미치는 긍정적 파급 효과는 명확하다. GPT-5-mini가 달성한 96.1%의 비존재 엔티티 검출률은 모델이 존재하지 않는 스키마를 참조하여 발생하는 의미적 환각을 원천 차단함으로써, 최종 생성되는 SQL의 실행 정확도를 향상시키는 직접적인 기반이 된다.

다만, 본 연구는 통제된 실험 환경을 위해 4가지 오류 유형을 단일하게 구성하여 성능을 분석하였다는 한계가 있다. 실제 사용자 환경에서는 여러 오류가 얽힌 복합 오류 양상이 나타날 수 있으며, ReAct 기반 에이전트가 이를 순차적으로 처리하는 과정에서 잦은 도구 호출로 인한 토큰 오버헤드가 발생할 수 있다. 따라서 이를 효율적으로 해결하기 위한 다중 도구 병렬 호출 정책 설계가 향후 연구로 요구된다.

References

[1] C. Zhu, Y. Lin, Y. Cai, and Y. Li, "SCoT2S: Self-Correcting Text-to-SQL Parsing by Leveraging Large Language Models", Computer Speech & Language, Vol. 95, Art. no. 101865, Jan. 2026. https://doi.org/10.1016/j.csl.2025.101865.

[2] B. Wang, X. Yu, and X. Zheng, "SQL Statement Generation Enhanced Through the Fusion of Large Language Models and Knowledge Graphs", Electronics, Vol. 15, No. 2, Art. no. 278, Jan. 2026. https://doi.org/10.3390/electronics15020278.

[3] T. Yu, et al., "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task", Proc. EMNLP, Brussels, Belgium, pp. 3911-3921, Oct.-Nov. 2018. https://doi.org/10.18653/v1/D18-1425.

[4] H. M. Pandey, et al., "GEMMA-SQL: A Novel Text-to-SQL Model Based on Large Language Models", Applied Artificial Intelligence, Vol. 39, No. 1, Art. no. 2587371, Nov. 2025. https://doi.org/10.1080/08839514.2025.2587371.

[5] C. Li, Y. Shao, Y. Li, and Z. Liu, "SEA-SQL: Semantic-Enhanced Text-to-SQL with Adaptive Refinement", Frontiers of Computer Science, Vol. 20, No. 3, Art. no. 2003602, Oct. 2026. https://doi.org/10.1007/s11704-025-41136-3.

[6] P. Ghosh, et al., "SQLGenie: A Practical LLM-based System for Reliable and Accurate Natural Language to SQL Conversion", ACL Industry Track, Vienna, Austria, pp. 1004-1012, Jul. 2025. https://doi.org/10.18653/v1/2025.acl-industry.71.

[7] N. Wretblad, et al., "Understanding the Effects of Noise in Text-to-SQL: An Examination of the BIRD-Bench Benchmark", Proc. ACL 2024, Bangkok, Thailand, pp. 356-369, Aug. 2024. https://doi.org/10.18653/v1/2024.acl-short.34.

[8] Y. Yang, et al., "Automated Validating and Fixing of Text-to-SQL Translation with Execution Consistency", Proc. of the ACM on Management of Data, New York, United States, Vol. 3, No. 3, pp. 1-28, Jun. 2025. https://doi.org/10.1145/3725271.

저자소개

변 공 규 (Gongkyu Byeon)

2016년 2월 : 경상국립대학교 미술교육학과(학사)
2022년 2월 : 국립창원대학교 문화융합기술협동과정(공학석사)
2022년 3월 ~ 현재 : 국립창원대학교 문화융합기술협동과정 박사과정
관심분야 : 대형언어모델, 증강/가상현실, 실감콘텐츠

최 권 택 (Kwon-Taeg Choi)

2006년 2월 : 연세대학교 컴퓨터공학과(공학석사)
2011년 2월 : 연세대학교 컴퓨터공학과(공학박사)
2016년 3월 ~ 현재 : 강남대학교 소프트웨어응용학부 교수
관심분야 : 가상현실, 증강현실, 모바일컴퓨팅, 기계학습, HCI

유 선 진 (Sunjin Yu)

2005년 3월 : 고려대학교 전자정보공학(공학사)
2006년 2월 : 연세대학교 생체인식공학(공학석사)
2011년 2월 : 연세대학교 전기전자공학(공학박사)
2011년 1월 ~ 2012년 5월 : LG전자기술원 미래IT융합연구소 선임연구원
2012년 5월 ~ 2013년 2월 : 연세대학교 전기전자공학과 연구교수
2013년 3월 ~ 2016년 8월 : 제주한라대학교 방송영상학과 조교수
2016년 9월 ~ 2019년 8월 : 동명대학교 디지털미디어공학부 부교수
2019년 9월 ~ 2024년 9월 : 국립창원대학교 문화테크노학과 부교수
2024년 10월 ~ 2025년 2월 : 국립창원대학교 문화테크노학과 정교수
2025년 3월 ~ 2026년 2월 : 국립창원대학교 메타융합콘텐츠학부/인공지능공학과 정교수
2026년 3월 ~ 현재 : 국립창원대학교 인공지능공학과 정교수
관심분야 : 컴퓨터비전, 증강/가상현실, HCI