Journal of KIIT. Vol. 23, No. 7, pp. 123-132, Jul. 31, 2025. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2025.23.7.123

NL2SQL 정확도 향상을 위한 GPT-4o 오류 패턴 기반 프롬프트 설계

변공규*, 최권택**, 유선진***

Prompt Engineering for Improving NL2SQL Accuracy based on GPT-4o Error Patterns

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

이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임
(RS-2024-00345763)

요 약

본 연구는 GPT-4o를 활용하여 자연어 질의를 SQL로 변환하는 시스템을 제안한다. 기존 데이터 분석 방식은 SQL을 활용한 데이터 전문가의 수작업이 필수적이었으므로 비전문가가 사용하기 어렵다. 본 시스템은 오류 패턴을 기반으로 수정된 프롬프트를 통해 SQL 쿼리 및 분석 코드를 자동 생성하여 실행할 수 있도록 설계한다. 실험 결과, 제안 시스템은 기존 방식에 비해 평균 22.9%가 높은 변환 정확도를 기록하였으며, LIKE 연산의 경우 54.33%에서 88.87%로 크게 향상되었다. 또한, 자연어 기반 질의 변환의 성능을 평가하기 위해 난이도별 SQL 변환 실험을 수행하였고, 기존 방식 대비 유의미한 정확도 향상을 기록하였다. 따라서, 본 연구는 비전문가도 데이터 분석을 쉽게 수행할 수 있는 기틀을 마련하고 데이터 활용의 효율성을 높일 것으로 기대된다.

Abstract

This study proposes a system that utilizes GPT-4o to convert natural language queries into SQL. Traditional data analysis methods require manual work using SQL or Python, making them difficult for non-experts to use. The proposed system automatically generates and executes SQL queries and analysis code through prompt optimization based on error patterns. Experimental results indicate that the proposed system achieved an average of 22.9% improvement in conversion accuracy over existing methods. Notably, the accuracy of the LIKE operation increased substantially, rising from 54.33% to 88.87%. Therefore, this study lays the groundwork for enabling non-experts to perform data analysis more easily and is expected to enhance the overall efficiency of data utilization.

Keywords

natural language to SQL, large language models, prompt engineering, gpt-4o, error pattern analysis

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

· Received: Mar. 28, 2025, Revised: Jun. 11, 2025, Accepted: Jun. 14, 2025
· Corresponding Author: Sunjin Yu
Dept. of Culture Technology, Changwon National University, 20 Changwondaehak-ro, Uichang-gu, Changwon-si, Gyeongsangnam-do, 51140, Korea
Tel.: +82-55-213-3098, Email: sjyu@changwon.ac.kr

Ⅰ. 서 론

빅데이터 시대의 도래와 함께 대규모 데이터셋이 지속적으로 생성되고 있으며, 이에 따라 데이터 기반 의사결정의 중요성이 점차 부각되고 있다[1][2]. 그러나 방대한 데이터에서 의미 있는 정보를 빠르고 정확하게 추출하기 위해서 SQL(Structured Query Language), 컴퓨팅 언어, 통계학 등의 기술이 필요하다. 이에 따라 데이터 분석 경험이 부족한 사용자들에게 높은 진입 장벽을 경험하게 한다[3][4].

최근에는 이러한 문제를 완화하는 방식으로 데이터베이스에 접근할 수 있는 NL2SQL(Natural Language to SQL) 기술이 주목받고 있다[5]. NL2SQL은 사용자가 자연어로 입력한 질의를 SQL 쿼리로 자동 변환하여 관계형 데이터베이스에서 원하는 정보를 추출할 수 있도록 하는 기술이다[6]. 이로써 비전문가도 복잡한 문법에 대한 지식 없이 데이터를 탐색하고 분석할 수 있으며, 기업 내 다양한 데이터 활용의 폭을 넓힐 수 있게 된다[7]. NL2SQL 시스템은 기존의 단어 기반 검색보다 높은 수준의 문맥 이해와 의미 해석 능력이 요구되며, 최근 고성능 LLM(Large Language Model)의 등장으로 그 성능이 크게 향상되었다[8]. 특히, 구조화된 데이터 스키마와 질의 의도를 이해하고, 문법적으로 알맞은 SQL 쿼리를 생성할 수 있어 실무 환경에서 활용 가능성이 높다.

이에 본 연구에서는 대표적 LLM 중 하나인 GPT-4o를 활용하여 자연어 질의를 SQL로 변환하는 NL2SQL 시스템을 구현하고, 해당 시스템의 정확성과 실용성을 데이터셋과 시나리오를 통해 정량적으로 평가한다. 이를 통해 자연어 인터페이스 기반 데이터 분석 환경의 가능성을 모색하고, 비전문가의 데이터 접근성과 분석 역량 향상에 기여하고자 한다.

본 논문은 이후 다음과 같은 구조로 구성된다. Ⅱ장에서는 자연어 질의에서 SQL로의 변환 기술의 연구 동향과 대형 언어 모델을 활용한 데이터 분석 자동화 관련 선행 연구를 정리하여 본 연구의 이론적 기반을 제시한다. Ⅲ장에서는 GPT-4o 기반 NL2SQL 시스템의 구조를 설명하고, 제안 방식인 오류 패턴 기반 프롬프트 설계 방식 및 각 구성 모듈의 동작 원리와 구현 방법에 대해 기술한다. Ⅳ장에서는 제안된 시스템의 정확도 향상을 검증하기 위해 다양한 실험을 수행하고, 도메인별, 질의 난이도별, SQL 연산 유형별 성능 평가 결과를 통계적으로 분석한다. Ⅴ장에서는 연구 결과를 종합하여 제안 시스템의 효과성과 한계를 정리하고, 향후 연구 방향과 개선 방안을 제안한다.

Ⅱ. 배경 이론

2.1 자연어 질의에서의 SQL 변환 연구

자연어를 SQL로 변환하는 연구는 NL2SQL로 불리며, 다양한 기계 학습과 심층 신경망 모델이 적용되었다. 기존 연구에서는 규칙 기반 접근 방식, 템플릿 매칭, 신경망 번역 모델 등을 활용하여 SQL 변환 성능을 개선하는 방법이 제안되었다[9].

R. Sun et al.[10]는 자연어 처리 모델 PaLM을 퓨샷 프롬프트(Few-shot prompt) 기법을 활용하여 실행 기반 오류 필터링을 수행하고, 일관성 디코딩(Consistency decoding)을 적용하여 변환 성능을 향상시키는 방법을 제안한다. 특히, 관련 데이터베이스 요소를 정확하게 선택하는 기술을 통해 SQL 생성의 정확도를 높이는 데 초점을 맞추었다.

L. Shi et al.[11]는 GPT-3, GPT-4, 기타 LLM을 활용한 Text-to-SQL 성능 향상 기법을 분석한다. 연구에서는 프롬프트 엔지니어링(Prompt engineering), 미세 조정(Fine-tuning), 평가 기법 등을 비교하며 LLM이 SQL 변환에서 높은 성능을 보이도록 조정하는 다양한 방법을 논의하였다.

H. Yoon et al.[12]는 사전 학습된 언어 모델을 기반으로 한국어 자연어 질의를 SQL로 변환하는 모델을 개발한다. 특히, 트랜스포머(Transformer) 기반 아키텍처를 적용하여 기존 규칙 기반 접근법보다 높은 변환 정확도를 기록했으며, 다국어 환경에서도 적용 가능한 NL2SQL 모델 개발 가능성을 제시한다.

2.2 LLM 기반 데이터 분석 자동화 연구

LLM의 발전과 함께 데이터 분석 자동화 분야에서도 GPT 계열 모델을 활용한 연구가 진행되고 있

다. LLM은 자연어 처리 능력을 기반으로 SQL 생성뿐만 아니라 데이터 분석 및 해석 과정에서도 중요한 역할을 한다.

L. Wang et al.[13]는 LLM을 활용한 ADA (Automated Data Analysis) 분야의 주요 연구 동향을 다룬다. 특히, AutoML 시스템과 LLM을 결합한 데이터 분석 자동화 및 멀티모달(Multi-modal) 데이터를 처리하는 AI 에이전트 개발을 주요 사례로 제시하며, LLM 기반 분석 자동화의 가능성과 한계를 탐색한다.

Z. Tan et al.[14]는 LLM이 데이터 주석 과정에서 비용 절감 및 효율성 향상을 기대할 수 있으며, LLM을 활용한 고품질 데이터 라벨링 자동화와 평가 기준 마련의 필요성을 언급한다. 본 연구는 LLM이 기존 데이터 라벨링 방식 대비 얼마나 정확한 주석을 제공할 수 있는지 비교하고, Human-in-the-loop 방식과의 조합을 통해 데이터 주석 품질을 높이는 방법을 탐구한다.

이처럼 기존 연구들은 NL2SQL, LLM 기반 데이터 분석 자동화 등의 분야에서 다양한 접근 방식을 제안하고 있다[15]. 특히, 최근 연구들은 GPT 계열 모델을 활용하여 데이터 분석의 접근성을 높이고, 자동화된 데이터 처리 및 SQL 변환 성능을 개선하는 방향으로 발전하고 있다. 본 연구는 이러한 선행 연구를 기반으로 GPT-4o를 활용하여 NL2SQL 변환 성능을 향상시키는 시스템을 개발하는 것을 목표로 한다.

III. 시스템 개발

3.1 전체 시스템 구성

표 1은 자연어 질의를 SQL로 변환하는 본 연구의 자동화 시스템과 그 세 가지 주요 모듈 구성을 보여준다. 각 모듈은 특정한 기능을 수행하여 SQL 변환의 정확성을 높이고, 실행 결과를 검증하는 역할을 한다. 자연어-SQL 변환 모듈은 사용자가 입력한 자연어 질의를 GPT-4o를 이용해 SQL 쿼리로 변환하는 핵심 역할을 수행하며, SQL 검증 모듈은 생성된 SQL의 정확성을 평가하고 WHERE 절과 JOIN 조건을 분석하여 오류를 수정한다. 실행 결과 비교 모듈은 변환된 SQL을 실제 데이터베이스에서 실행한 후 정답 SQL과 비교하여 변환 정확도를 검증하는 기능을 담당한다.

자연어 질의가 입력되면 테이블 정보와 함께 GPT-4o에 전달되며, GPT-4o는 입력된 정보를 기반으로 SQL을 생성한다. 이후 SQL 검증 모듈이 변환된 SQL을 검토하고 오류를 수정하며, 마지막으로 SQL을 실행하여 결과를 확인하고 정답 SQL과 비교하여 일관성을 평가한다.

3.2 오류 패턴 기반 프롬프트 제안

자연어 질의를 SQL로 변환하는 본 연구의 접근 방식은 프롬프트 구성, GPT-4o 기반 추론, SQL 결과의 검증 및 보정의 세 단계로 구성된다. 기존 연구들이 파인튜닝이나 외부 지식 기반 RAG (Retrieval-Augmented Generation)를 통해 성능을 개선하는 반면, 본 연구는 추가 학습 없이 프롬프트 설계만으로 NL2SQL의 정확도 성능을 향상시키는 전략에 초점을 맞추었다는 점에서 차별성을 가진다.

우선, 학습에 사용하는 데이터 중에서 오류가 발생한 SQL 쿼리들을 수집하고, 각 쿼리를 자연어 질의, 정답 SQL, 생성된 SQL의 세 가지 구성요소로 페어링하여 분석하였다. 이때 단순한 정답 불일치가 아닌 실행 결과 기준으로 정답과 다른 결과를 출력하는 경우를 오류로 판단하고, 해당 오류의 원인을 간단한 설명과 함께 기록하였다.

표 1. 전체 시스템 구성
Table 1. System configuration

Module name	Role
Natural language to SQL conversion module	Converts natural language queries into SQL using GPT-4o.
SQL validation module	Reviews WHERE clauses and JOIN conditions for errors and corrections.
Execution result comparison module	Compares the generated SQL with the correct SQL and evaluates performance.

모든 오류를 프롬프트에 일괄 반영할 경우 토큰 수 제한 및 API 비용 측면에서 비효율이 발생하기 때문에, 수집된 오류들 중 반복적으로 발생하는 오류 패턴을 중심으로 그룹화하였다. 그 후 빈도수가 높은 상위 n개의 오류 그룹(본 실험에서는 n=20)을 선정하고, 각 그룹에 대해 GPT 모델을 활용하여 공통적인 오류 원인과 이를 수정할 수 있는 보편적인 규칙을 자동 추출하였다. 예를 들어, 날짜 관련 검색 시 날짜 연산자나 범위 표현 오류, 문자열 비교 시 불필요한 조사(예: “을”, “를”)가 포함되는 문제, 숫자형과 문자열형 데이터 구분 오류 등이 대표적인 패턴으로 확인되었다.

이와 같이 정제된 보편적 오류 규칙은 최종 프롬프트에 직접 반영되었다. 프롬프트는 테이블 구조, 주요 컬럼 정보, 변환 규칙, 예시 데이터뿐 아니라, 자주 발생하는 오류 유형을 사전에 방지할 수 있도록 설계된 연산 방식, 조건 설정, 주의 사항 등을 포함한다. 예를 들어 “지난 1개월 동안 서울에서 가장 많이 판매된 상품은?”과 같은 질의가 주어질 경우, 날짜 비교를 위한 범위 설정 방식, 정렬 기준 설정, 문자열 필터링 방식 등을 명확히 제시함으로써 모델이 더 정확한 SQL을 생성할 수 있도록 유도한다.

SQL 쿼리는 단순히 문법적으로 맞는지 여부로만 정확성을 판단할 수 없기 때문에, 생성된 쿼리를 실제 실행하고 그 결과가 정답 SQL의 결과와 동일한지 여부를 기준으로 성능을 평가하였다. 이를 통해 프롬프트 적용 전후의 성능 차이를 정량적으로 비교하였고, 프롬프트 설계만으로도 모델의 SQL 생성 정확도가 유의미하게 향상됨을 확인하였다.

Ⅳ. 실험 및 성능 평가

본 장에서는 제안된 자연어 변환의 성능을 검증하기 위해 정량적인 정확도 평가를 중점적으로 수행한다. 실험은 크게 (1) 도메인별 NL2SQL 시스템의 성능 평가, (2) 질의 변환 정확도 비교, (3) 난이도별 정확률 분석을 측정하는 세 가지 실험으로 구성되었다. 이를 통해 기존 방식과 비교하여 제안 시스템이 얼마나 효과적으로 데이터를 분석할 수 있는지 정량적으로 검토한다.

4.1 실험 데이터

본 연구에서는 AI-Hub에서 제공하는 자연어 기반 질의 검색 생성 데이터를 실험 데이터로 사용한다. 본 연구에 사용된 데이터셋은 자연어 질의와 그에 상응하는 SQL 질의로 구성되어 있으며, 공공기관 데이터베이스를 활용해 NL2SQL 모델 개발을 위해 구축되었다.

본 데이터는 표 2와 같이 SQLite, SQL, JSON의 세 가지 형식으로 제공된다. SQLite 파일은 실제 데이터베이스 테이블을 포함하여 모델 학습 및 평가 환경을 제공하며, SQL 파일은 자연어 질의에 대응하는 SQL 질의를 포함하고 있다. JSON 파일은 데이터의 라벨링 정보와 메타데이터를 포함하여 질의 유형과 난이도 등의 정보를 제공한다.

해당 데이터는 서울시 열린데이터광장과 공공데이터포털에서 제공하는 행정 분야의 정형 텍스트 및 테이블 데이터를 기반으로 수집되었으며, 산업과 경제(22.18%), 보건(21.32%), 환경(19.29%), 문화와 관광(13.21%), 일반행정(9.98%), 교육(5.87%), 교통(4.66%), 복지(3.49%)의 여덟 개 도메인으로 구성된다.

표 2. 데이터셋의 메타데이터 구조표
Table 2. Metadata structure table for datasets

Data category	Korean	Data type	Text
Data format	txt (SQLite, SQL, JSON)	Data source	Structured text data and table data from Open Data Plaza and Public Data Portal
Labeling type	Question-Answering (Natural language)	Labeling format	JSON
Data utilization service	Semantic parsing	Data construction year / Volume	2022 / Labeled Data: 111,152 cases

총 6,401개의 데이터베이스와 111,152건의 라벨링 데이터로 이루어진 데이터셋은 학습(80%), 검증(9.9%), 테스트(10.1%) 데이터로 분할되었다. 본 연구에서는 테스트 데이터셋을 활용하여 모델 성능을 평가한다. 메타데이터는 질의 난이도(하, 중, 상, 최상)와 질의 유형을 포함한다. 하 난이도의 질의는 단순 SELECT 문(23.09%), 중 난이도는 WHERE 및 GROUP BY 포함(45.37%), 상 난이도는 서브쿼리 및 다중 테이블 조인(25.37%), 최상 난이도는 고난도 SQL 질의(6.17%)로 구성된다. 질의 유형은 수량(21.49%), 사물과 속성(20.15%), 인물과 조직(19.73%), 장소(19.44%), 일시(19.19%)의 비율을 차지한다.

4.2 도메인별 NL2SQL 시스템 성능 평가

본 실험은 프롬프트를 활용한 방식과 기존 일반 방식과 비교 실험을 수행하였다. 평가 항목으로는 정확률(%)을 기준으로 기존 방법과 비교 분석하였으며, 8개의 도메인에서 분석 성능을 측정하였다.

실험 결과, 표 3과 같이 제안 시스템은 기존 방식 대비 평균 22.9% 향상된 88.32%의 정확도를 기록하였다. 특히, 문화(31.82%) 및 산업(30.88%) 도메인에서 가장 큰 성능 향상을 보였다. 기존 방식의 평균 정확도는 65.42%였으며, 모든 도메인에서 p-value < 0.001을 기록하여 통계적으로 유의미한 차이가 있음을 확인하였다. 오차 감소율 분석 결과, 산업(47.40%), 문화(45.59%), 교육(36.88%) 도메인에서 가장 큰 폭의 오차 감소율을 보였으며, 상대적 향상 배율 분석에서는 문화(1.54배), 산업(1.50배), 교육(1.37배) 도메인이 가장 높은 향상 배율을 기록하였다.

Cohen의 d는 두 집단 간 평균 차이를 표준편차로 나누어 계산하는 효과 크기(Effect size) 지표로, 통계적으로 유의미한 차이가 실질적으로도 의미 있는지를 판단하는 데 활용된다. 본 연구에서는 제안한 시스템과 기존 방식 간 성능 차이의 크기를 정량적으로 평가하기 위해 대응표본 t-검정과 함께 Cohen의 d 값을 산출하였다. 일반적으로 Cohen의 d 값이 0.2 이상이면 작은 효과, 0.5 이상이면 중간 효과, 0.8 이상이면 큰 효과로 해석된다. Cohen's d 효과 크기는 문화(1.75), 산업(1.68), 교육(1.45) 도메인에서 가장 높은 효과 크기를 보이며, 기존 방식 대비 제안 시스템이 큰 성능 차이를 나타내는 것으로 분석되었다. 모든 도메인에서 Cohen's d 값이 1.0 이상을 기록하여 실질적으로 유의미한 차이를 확인하였다. 이를 종합하면, 제안된 프롬프트 방식은 기존 일반 방식 대비 신뢰도 높은 데이터 분석을 수행할 수 있으며, 특히 복잡한 데이터 구조를 가지는 문화 및 산업 도메인에서 효과적인 성능 향상을 보였다. 또한, 오차 감소율, 상대적 향상 배율, 효과 크기 등 다양한 지표에서 기존 방식 대비 우수한 분석 성능을 입증하였다.

4.3 난이도별 정확률 분석

본 실험은 질의 변환 정확도를 난이도별로 구분하여 일반 방식과 제안 방식의 비교 실험을 수행하였다. 실험 데이터는 표 4와 같이 난이도를 Easy, Medium, Hard, Extra Hard 네 가지로 나누어 분석하였으며, 기존 방식과 제안 시스템 간의 정확도 차이를 비교하였다.

표 3. 도메인별 NL2SQL 시스템의 성능 개선 비교 결과
Table 3. Domain-wise comparison of NL2SQL system performance improvements

Domain	Base accuracy (%)	Proposed accuracy (%)	Improvement rate (%)	Error reduction rate (%)	Relative improvement factor	Number of SQL statements	Cohen's d effect size	p-value (t-test)
Health	63.69	82.31	18.62	29.37	1.29	1,937	1.15	< 0.001
Administration	66.66	89.5	22.84	34.29	1.34	300	1.3	< 0.001
Culture	59.4	91.22	31.82	45.59	1.54	1,100	1.75	< 0.001
Industry	62.33	93.21	30.88	47.4	1.5	693	1.68	< 0.001
Welfare	74.49	89.26	14.77	22.73	1.2	149	1.1	< 0.001
Environment	68.72	90.3	21.58	32.42	1.31	598	1.35	< 0.001
Transportation	66.66	79.91	13.25	19.82	1.2	249	1	< 0.001
Education	65.38	89.86	24.48	36.88	1.37	286	1.45	< 0.001

표 4. 난이도별 정확률 분석 결과
Table 4. Accuracy analysis results by difficulty level

Difficulty level	Base accuracy (%)	Proposed accuracy (%)	Improvement rate (%)	Improvement rate (%)	Relative improvement factor	Cohen's d effect size	p-value (t-test)
Easy	72.66	89.44	16.78	X	X	0.9	< 0.001
Medium	61.02	88.58	27.56	35.54	1.45	1.25	< 0.001
Hard	50.23	86.63	36.41	48.44	1.73	1.75	< 0.001
Extra hard	69.57	86.96	17.39	24.99	1.25	1.05	< 0.001

난이도는 증가할수록 데이터 처리 및 분석 과정이 복잡해지며, 시스템이 정확한 결과를 도출하는 데 더욱 높은 성능이 요구된다.

기존 방식은 Easy 난이도에서는 비교적 높은 정확도를 유지했으나, 난이도가 증가함에 따라 성능이 급격히 저하되었다. 반면, 제안 시스템은 난이도가 증가하더라도 안정적인 성능을 유지하는 것으로 나타났다. 특히 Hard 및 Extra Hard 난이도에서 기존 방식과 큰 성능 차이를 기록하며, 복잡한 분석에서도 강한 성능을 유지할 수 있음을 입증하였다. 실험 결과, Hard 난이도의 경우 기존 방식(50.23%)과 제안 시스템(86.63%) 간의 차이가 36.41%로 가장 컸다. Extra Hard 난이도의 경우에 기존 방식(69.57%)과 비교하여 제안 시스템(86.96%)이 17.39%의 향상을 기록하며, 높은 난이도에서 높은 정확도를 유지하였다.

오차 감소율 분석 결과, Medium 난이도에서 35.54%, Hard 난이도에서 48.44%, Extra Hard 난이도에서 24.99%의 오차 감소율을 기록하였다. 상대적 향상 배율 분석에서도, Medium 난이도에서 1.45배, Hard 난이도에서 1.73배, Extra Hard 난이도에서 1.25배 높은 성능을 보였으며, 난이도가 증가할수록 성능 향상이 더욱 두드러졌다.

성능 차이가 통계적으로 유의미한지를 확인하기 위해 대응 표본 t-검정을 수행한 결과, 모든 난이도에서 $p<0.001$을 기록하여 통계적으로 유의미한 차이가 있음을 확인하였다. Hard 난이도의 경우 Cohen's d 값이 1.75로 가장 높은 효과 크기를 보였으며, Extra Hard 난이도에서도 1.05를 기록하여 실질적인 성능 향상이 있음을 나타냈다.

특히 Hard 및 Extra Hard 난이도에서도 높은 정확도를 유지하며, 오차 감소율과 상대적 향상 배율에서도 우수한 성능을 보였다.

4.4 SQL 연산별 질의 변환 정확도 평가

본 실험에서는 다양한 SQL 연산이 포함된 질의를 제안 시스템이 얼마나 정확하게 변환할 수 있는지를 평가하였다. SQL 변환 정확도는 연산 유형에 따라 차이가 발생하며, 본 실험에서는 SQL WHERE, SQL WHERE1, SQL WHERE2, SQL WHERE3, SQL LIKE, SQL GROUP BY 등 6가지 주요 연산 유형을 선정하여 분석을 수행하였다.

기존 방식과 제안 시스템 간의 질의 변환 성공률을 비교하기 위해 동일한 데이터셋을 사용하였으며, 변환된 SQL이 실행 가능한지를 기준으로 정확도를 측정하였다.

실험 결과, 제안 시스템은 모든 SQL 연산에서 기존 방식 대비 높은 변환 정확도를 기록하였다.

표 5와 같이 WHERE 조건문 변환 성능 분석 결과, SQL WHERE(76.92%), SQL WHERE1(90.80%), SQL WHERE2(79.54%), SQL WHERE3(80.00%)에서 기존 방식 대비 15~20%의 정확도 향상을 보였으며, 특히 SQL WHERE2(비교 연산자 포함 WHERE 조건)의 경우 20.45%의 향상률을 기록하며 가장 높은 성능 개선을 나타냈다.

LIKE 연산 변환 성능 분석 결과, 기존 방식이 54.33%의 정확도를 기록한 반면, 제안 시스템은 88.87%의 정확도를 기록하며 34.54%의 향상률을 보였다. 이는 제안 시스템이 자연어 질의를 보다 정확한 SQL 패턴으로 변환할 수 있음을 의미하며, 특히 문자열 패턴 검색에서 기존 방식 대비 높은 변환 신뢰도를 제공함을 시사한다.

표 5. SQL 연산별 질의 변환 정확도 결과
Table 5. SQL operation-specific query conversion accuracy results

SQL operation type	SQL operation type	Proposed accuracy (%)	Improvement rate (%)	Error reduction rate (%)	Relative improvement factor	Cohen's d effect size	p-value (t-test)
SQL WHERE	61.54	76.92	15.38	24.99	1.25	1.15	< 0.001
SQL WHERE1	72.41	90.8	18.39	33.28	1.25	1.3	< 0.001
SQL WHERE2	59.09	79.54	20.45	38.63	1.35	1.5	< 0.001
SQL WHERE3	60	80	20	33.33	1.33	1.45	< 0.001
SQL LIKE	54.33	88.87	34.54	54.07	1.63	1.85	< 0.001
SQL GROUP BY	73.91	84.78	10.87	16.66	1.15	0.95	< 0.001

오차 감소율 분석 결과, SQL LIKE(54.07%), SQL WHERE2(38.63%), SQL WHERE3(33.33%)에서 가장 큰 폭의 오차 감소율을 기록하였다. 상대적 향상 배율 분석 결과, SQL LIKE(1.63배), SQL WHERE2(1.35배), SQL WHERE3(1.33배)에서 가장 높은 배율을 기록하여, 해당 연산 유형에서 제안 시스템이 특히 강점을 가진다는 점을 확인하였다.

통계적 유의성을 검증하기 위해 Cohen's d 효과 크기를 분석한 결과, SQL LIKE(1.85), SQL WHERE2(1.50), SQL WHERE3(1.45)에서 가장 높은 효과 크기를 기록하였다. 또한, 기존 방식과 제안 시스템 간의 변환 정확도 차이가 통계적으로 유의미한지를 검증하기 위해 대응 표본 t-검정을 수행한 결과, p-value < 0.001을 기록함을 확인하였다.

이와 같은 결과를 종합적으로 고려할 때, 제안된 SQL 질의 변환 시스템은 WHERE 조건문 및 LIKE 연산과 같이 변환 난이도가 높은 SQL 연산에서 기존 방식 대비 높은 정확도를 기록하였으며, 오차 감소율과 상대적 향상 배율 분석에서도 기존 방식 대비 우수한 성능을 입증하였다.

4.5 실험 결과 요약

본 연구에서는 GPT-4o에 프롬프트를 활용한 NL2SQL 시스템을 개발하고, 기존의 일반 GTP-4o 데이터 분석 방식과 비교하여 성능을 검증하였다. 실험 결과, 아래 그림 1과 같이 제안된 시스템은 기존 방식 대비 전반적으로 높은 정확도를 기록하였으며, 특히 복잡한 데이터 환경에서도 높은 성능을 유지할 수 있음을 확인하였다.

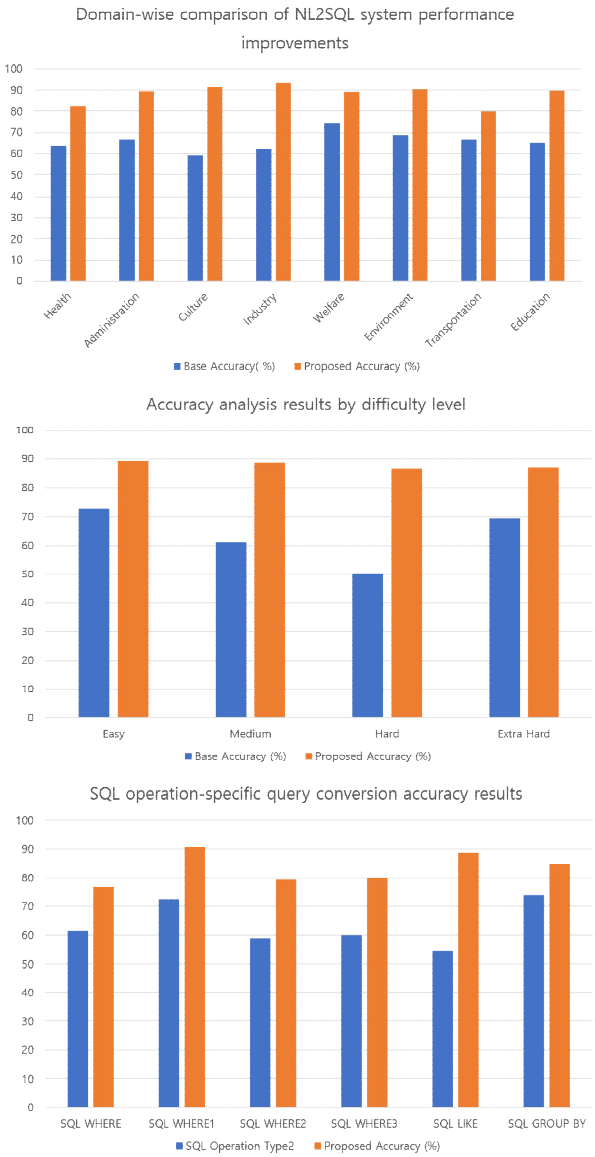

그림 1. 도메인·난이도·SQL 연산 유형에 따른 NL2SQL 변환 정확도 비교
Fig. 1. Comparing NL2SQL transformation accuracy by domain, aifficulty, SQL operation typetems

실험 데이터 구성 분석(4.1)은 AI Hub에서 제공하는 NL2SQL 데이터를 활용하여 실험을 수행하였으며, 6,401개의 데이터베이스와 111,152건의 라벨링 데이터를 포함하는 데이터셋을 기반으로 평가를 진행하였다. SQL 변환 난이도는 하(23.09%), 중(45.37%), 상(25.37%), 최상(6.17%)으로 분류되며, 본 연구에서는 이 난이도를 반영하여 다양한 질의 변환 성능을 평가하였다.

탐색적 데이터 분석 성능 평가(4.2)는 기존 방식 대비 평균 22.9%의 정확도 향상을 기록하였으며, 문화(31.82%) 및 산업(30.88%) 도메인에서 가장 큰 성능 향상을 보였다.

난이도별 질의 변환 정확도 평가(4.3)는 기존 방식은 난이도가 증가할수록 정확도가 급격히 저하되었으나, 제안 시스템은 Hard(36.41%) 및 Extra Hard(17.39%) 난이도에서도 높은 정확도를 유지하며 신뢰도 높은 분석을 수행하였다.

SQL 연산별 질의 변환 정확도 평가(4.4)는 WHERE, LIKE, GROUP BY 등의 SQL 연산에서 최대 34.54%의 정확도 향상을 기록하였으며, 특히 패턴 매칭을 수행하는 LIKE 연산에서 가장 큰 개선이 이루어졌다.

V. 결 론

본 연구에서는 GPT-4o 기술을 활용하여 자연어의 SQL 변환 성능을 향상시키는 방법을 제안하였다. 실험 결과, 제안된 시스템은 기존 방식 대비 평균 22.9%의 정확도 향상을 기록하였으며, 특히 복잡한 SQL 연산이 포함된 질의 변환 성능에서도 큰 폭의 개선이 이루어졌음을 확인하였다.

SQL 연산별 변환 성능 분석에서는 WHERE, LIKE, GROUP BY 등의 연산에서 성능 향상이 이루어졌으며, 특히 패턴 매칭을 수행하는 LIKE 연산에서 가장 큰 개선이 나타났다.

그러나 본 연구는 일부 복잡한 질의에서 GPT-4o가 부정확한 SQL을 생성하는 문제가 발생하는 한계를 가진다. 특히 다중 조건이 포함된 질의의 경우 변환 오류율이 상대적으로 높게 나타났다. 이는 GPT-4o가 복잡한 논리 연산을 처리하는 데 한계를 가질 수 있음을 시사하며, 향후 프롬프트 최적화 및 보완 모델 결합을 통해 개선이 필요하다.

향후 연구에서는 프롬프트 엔지니어링 최적화 및 실시간 데이터 처리 성능 개선을 통해 실무 적용성을 더욱 강화할 필요가 있다. 이를 통해 데이터 분석의 효율성을 높이고, 데이터 기반 의사결정을 지원하는 데 기여할 것을 기대한다.

References

[1] K. Kim and S. Kim, "Strategies for the Use of Artificial Intelligence in the Public Sector: Focused on a Large language model based Generative artificial intelligence", korean policy sciences review, Vol. 28, No. 2, pp. 25-45, Jun. 2024. https://doi.org/10.31553/kpsr.2024.6.28.2.25.

[2] S. Bong, J. Lee, H. Park, S. Chae, and S. Lee, "Implementation of a Schedule Management Planner using ChatGPT", Journal of the IIBC, Vol. 24, No. 6, pp. 223229, Jan, 2024. https://doi.org/10.7236/JIIBC.2024.24.6.223.

[3] M. C. Data, M. Komorowski, D. C. Marshall, J. D. Salciccioli, and Y. Crutain, "Exploratory data analysis", Secondary analysis of electronic health records, pp. 185-203, Sep. 2016. https://doi.org/10.1007/978-3-319-43742-2_15

[4] I. Okpala, A. Golgoon, and A. R. Kannan, "Agentic AI Systems Applied to tasks in Financial Services: Modeling and model risk management crews", arXiv preprint arXiv:2502.05439, Feb. 2025. https://doi.org/10.48550/arXiv.2502.05439.

[5] L. Shi, Z. Tang, N. Zhang, X., Zhang, and Z. Yang, "A survey on employing large language models for text-to-sql tasks", arXiv preprint arXiv:2407.15186, Nov. 2024. https://doi.org/10.48550/arXiv.2407.15186.

[6] X. Liu, et al., "A Survey of NL2SQL with Large Language Models: Where are we, and where are we going?", arXiv preprint arXiv:2408.05109, Mar. 2024. https://doi.org/10.48550/arXiv.2408.05109.

[7] T. Ren, et al., "Purple: Making a large language model a better sql writer", IEEE 40th International Conference on Data Engineering (ICDE), Utrecht, Netherlands, pp. 15-28, May 2024. https://doi.org/10.1109/ICDE60146.2024.00009.

[8] C. Lee and J. Kim, "An Exploratory Case Study on Design of Carrer Simulation Game using GPT", Vol. 25, No. 2, pp. 221-230, Apr. 2025. Journal of the IIBC, https://doi.org/10.7236/JIIBC.2025.25.2.221.

[9] S. Nielsen, "Management accounting and the concepts of exploratory data analysis and unsupervised machine learning: a literature study and future directions", Journal of Accounting & Organizational Change, Vol. 18, No. 5, pp. 811-853, Oct. 2022. https://doi.org/10.1108/JAOC-08-2020-0107

[10] R. Sun, et al., "Sql-palm: Improved large language model adaptation for text-to-sql (extended)", arXiv preprint arXiv:2306.00739, Mar. 2024. https://doi.org/10.48550/arXiv.2306.00739.

[11] L. Shi, Z. Tang, N. Zhang, X. Zhang, and Z. Yang, "A survey on employing large language models for text-to-sql tasks", arXiv preprint arXiv:2407.15186, Nov. 2024. https://doi.org/10.48550/arXiv.2407.15186.

[12] H. Yoon, J. Heo, S. Kim, and P. Kang, "Text-to-SQL for Korean Language based on Multilingual BERT", Journal of the Korean Institute of Industrial Engineers, Vol. 48, No. 1, pp. 91-104, Feb. 2022. https://doi.org/10.7232/JKIIE.2022.48.1.091.

[13] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, and J. Wen, "A survey on large language model based autonomous agents", Frontiers of Computer Science, Vol. 18, No. 6, pp. 186345, Mar. 2024. https://doi.org/10.1007/s11704-024-40231-1.

[14] Z. Tan, A. Beigi, S. Wang, R. Guo, A. Bhattacharjee, B. Jiang, and H. Liu, "Large language models for data annotation: A survey", arXiv prints, arXiv-2402, Jun. 2024. https://doi.org/10.48550/arXiv.2402.13446.

[15] P. Ma, R. Ding, S. Wang, S. Han, and D. Zhang, "Demonstration of InsightPilot: An LLM-empowered automated data exploration system", arXiv preprint arXiv:2304.00477, Nov. 2023. https://doi.org/10.48550/arXiv.2304.00477.

저자소개

변 공 규 (Gongkyu Byeon)

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

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)

2003년 8월 : 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 ~ 현재 : 국립창원대학교 문화테크노학과
교수
관심분야 : 컴퓨터비전, 증강/가상현실, HCI