ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v2.i2.a12

Al-Noor Journal for Information Technology and Cybersecurity

https://jncs.alnoor.edu.iq/

Generative AI for Relational Database Management: A Review of Natural Language Interface to convert text to SQL

[1]Yousif Baderaldeen Ahmed , [1]Rayan Yousif Yaqoub

[1]University of Mosul, College of Computer Science and Mathematics, Department of Computer Science, Iraq

Article information

Article history:
Received: October, 26,2025
Revised: November 16, 2025
Accepted: November 21, 2025

Keywords:
Generative AI
Relational Databases
Natural language Interface
SQL

Correspondence:

Yousif Baderaldeen Ahmed
yousif.24csp32@student.uomosul.edu.iq

Abstract

A big problem with Natural Language Interfaces to Databases (NLIDBs) is that they can't turn natural language queries into SQL instructions. The primary causes are linguistic ambiguity, schema complexity, and the challenges non-experts face in articulating relational data objectives. This paper examines the progression of Text-to-SQL methodologies from rule-based and statistical frameworks to deep neural architectures and ultimately to Large Language Models. It compares them to the Spider and WikiSQL benchmarks. Evidence indicates that LLMs, particularly GPT-4-class models, enhance execution accuracy via contextual reasoning and in-context learning; nonetheless, challenges persist in multilingual generalization, complex query management, and robustness across various database schemas. The study also shows how quick engineering, schema linking, and retrieval-augmented methods may fill these gaps and lower the cost of making queries. These improvements show that data interaction models are moving toward ones that are easier to use, more conversational, and allow for more than one way to participate. This makes databases easier for non-technical individuals to use and helps people in various fields make decisions based on data.

1_Introduction

Relational Database administration Systems (RDBMS) are essential for data administration, executing queries declaratively through SQL, A fundamental function of a DBMS is to convert SQL queries into physical execution plans, a process referred to as query optimization,This optimization is essential as the execution times for various physical designs, albeit producing the same outputs, can range significantly. The enhancement of modern database management systems' efficiency is a primary focus of database research, with query optimization serving as a crucial domain of progress, Efficient query optimization, frequently utilizing advanced information such as data dependencies, is essential for effective data processing [1].

Natural language inquiries are difficult to translate into SQL queries, especially for beginners, A model generates a SQL query from a plain language question, a database structure, and sometimes database information, When only the query is provided, models have a meagre 8.3% execution accuracy, but schema information and accurate SQL and foreign key data increase performance. Complex SQL features like consistent table aliasing conventions often differ from Codex output, making gold queries less accurate. Untrained users may struggle to discover and fix semantic errors like inappropriate groupings or unnecessary columns, which need a comprehensive understanding of the query's meaning and SQL standards [2].

Natural Language Interfaces for Databases (NLIDB) allow non-expert users to query relational databases using natural language rather than SQL. This strategy makes information more accessible by eliminating the need to understand database architectures or write sophisticated queries. NLIDBs transform natural language queries into formal database queries, making information retrieval more flexible and straightforward. The user-friendliness optimizes database utilization when form-based interfaces or complex query languages are unmanageable, especially on mobile devices[3].

Large Language Models (LLMs) have introduced a novel paradigm for Text-to-SQL operations, markedly enhancing the conversion of natural language inquiries into SQL queries, This breakthrough is ascribed to their capacity to learn from contextual examples, a process termed in-context learning, which allows them to produce accurate SQL queries without explicit task-specific instruction. Significantly, LLMs such as GPT-4 have attained superior execution accuracy on benchmarks like Spider, with innovations like DAIL-SQL revitalizing leaderboards, Prompt engineering, which includes question representation, example selection, and arrangement, is essential for optimizing the capabilities of LLMs in this field [4].

Text-to-SQL conversion has made progress, especially with Generative AI and Large Language Models (LLMs), but some major issues remain, Linguistic ambiguity, contextual dependency, and SQL's complexity make natural language queries difficult to translate into SQL queries, especially for non-experts. Most cutting-edge solutions are for monolingual situations, making multilingual support difficult. Managing sophisticated and nested queries and schema complexity in databases with many connected tables and columns is difficult. To improve user engagement and support diverse query types, multimodal interfaces (text, audio, and visualization) will be used in future projects. Interactive conversational NLIDB systems that use LLMs' natural language understanding and generation will help non-SQL users access data more easily and efficiently.
The work empirically investigated numerous earlier research methodologies to create a thorough, systematic, and equitable LLM-based Text-to-SQL standard. Many Large Language Models (LLMs) assessed different question formats in zero-shot circumstances to find their merits and cons. In few-shot circumstances, selection and organization were examined. The study evaluated open-source LLMs for in-context learning and supervised fine-tuning using quick engineering methodologies. To find cost-effective, high-performance strategies with low token usage, many token efficiency-based methods were investigated.
Text-to-SQL algorithms were tested on Spider and Spider-Realistic datasets. Large, cross-domain Text-to-SQL dataset Spider has 8,659 training examples and 1,034 development samples from over 200 databases. Spider's training split occasionally provided Spider-dev and Spider-Realistic testing candidates. Published on November 20, 2023, the document indicates recent study and evaluation.
This study formalized question representation, in-context learning, and supervised fine-tuning and found serious faults. The report observed that most Text-to-SQL research developed and generalized question-to-SQL patterns utilizing encoder-decoder models. Despite gains, rapid engineering for LLM-based Text-to-SQL, including question representations, sample selection, and organization, required a full analysis. A Spider leaderboard record was set using DAIL-SQL, a novel Text-to-SQL prompt engineering technique Prompt engineering and LLM performance on Text-to-SQL operations were assessed using existing datasets for efficacy and efficiency. The research used previous findings to develop and validate a novel solution . [4]

2_ Methodology

This review followed a structured narrative methodology to synthesize recent advancements in NLIDB and Text-to-SQL systems. A systematic search was conducted across IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, and arXiv using keywords such as Text-to-SQL, NLIDB, Generative AI, and Large Language Models. The search covered studies published between 2020 and 2025. Inclusion criteria admitted papers that proposed or evaluated Text-to-SQL models, reported empirical results on benchmarks such as Spider, BIRD, or WikiSQL, or introduced new prompt-engineering, schema-linking, or retrieval-augmented techniques. Exclusion criteria removed theoretical-only papers, duplicates, studies lacking SQL generation, and works outside the 2020–2025 range. The initial search yielded approximately 140 papers; after duplicate removal and abstract screening, 78 remained. Following full-text evaluation, 33 studies met all criteria and were included in this review.

3-Background and Fundamentals

Relational Database Management Systems (RDBMS) are defined by a formally established relational data model, differentiating them from NoSQL databases due to their 'schema-on-write' characteristic, which necessitates the declaration of schemas prior to data storage, This approach is based on the mathematical concept of relations, usually depicted as tables, A relational schema consists of relation schemas, each detailing the connection name, attribute names, and their corresponding domains (types), Relationships among relations are formed implicitly by key propagating (one-to-one, one-to-many) or explicitly through distinct relation schemas for many-to-many relationships, Each relation possesses one or more properties that constitute a primary key, uniquely identifying tuples (rows), SQL as a widely utilized language, is frequently adapted to formulate languages of queries

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v2.i2.a12

for post-relational databases, encompassing systems that utilize NoSQL [5].

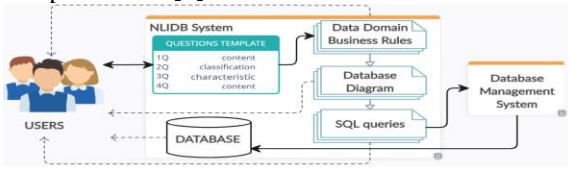
Natural Language Interfaces (NLIs) for database systems improve data accessibility. This method lets non-technical users interact with complex database systems using natural language instead than SQL, NLIs reduce data retrieval time by 37%[6], allowing business analysts to spend 42%[6] more time on analysis, 78%[6] of organizations believe improved data accessibility is crucial for data-driven decision-making, NLIs have also proved their value in healthcare and financial sectors, increasing non-technical self-service analytics usage by 63%[6], Large Language Models (LLMs) have strengthened the technological underpinning of modern Natural Language Interfaces (NLIs), increasing their functionality, These interfaces transform human-database interaction beyond access technology advances[6].

3.1_Evolution of Natural Language Interfaces for Databases: From Rules to Generative AI

From Rule-Based to Statistical Models: Rule-based systems, explicit protocols, lexicons, string matching, and regular expressions for data extraction were used to automate systematic reviews (SRs), Its static nature required manual update and lacked adaptive learning, This led to statistical models and machine learning (ML), which enabled adaptive learning and data pattern recognition, Machine learning models like SVM, LR, NB, K-Nearest Neighbors, Decision Trees, Random Forests, and LDA have become popular in screening and search tasks, improving efficiency by automating tedious processes,The models were trained using supervised, unsupervised, or semi-supervised (active learning) methods.

Deep Learning and the Rise of Generative AI: Deep learning (DL), a type of AI using multi-layered neural networks, improved the management of large datasets and complex problems, Using sophisticated pattern recognition algorithms, CNN and RNN, including LSTM and GRU, were used for data extraction, bias risk evaluation, and search,Transformer-based models like BERT and GPT, which use self-attention for complex text comprehension, have shown promise in text classification and data extraction, Large Language Models (LLMs) like ChatGPT in service request automation are still in their infancy, with next initiatives focusing on fine-tuning, query formulation, and structured domain knowledge[7].

Figure (1) shows the typical NLIDB system architecture. User inquiries in natural language are evaluated by the NLIDB system using question templates to determine purpose and relevant entities. These templates help with Data Domain Business Rules, Database Diagram understanding, and SQL query building. The Database Management System (DBMS) generates SQL commands to get accurate database results. This layered architecture shows the evolution of AI in database queries from rule-based expert systems based on carefully constructed linguistic and semantic rules to generative AI systems that determine contextual mappings from data. Syntactic parsing, semantic mapping, and rule-based template matching governed each element in early NLIDB systems. Since 2020, neural and generative methods like RAT-SQL, BRIDGE, and LLM-based Text-to-SQL have been used to translate natural language to SQL without manual rules, making it more adaptable to unfamiliar databases and linguistic diversity. This paradigm shift from symbolic to generative AI combines deterministic reasoning with data-driven contextual comprehension [8].



Figur1 : NLIDB system Architecture

3.2 Key challenges: linguistic ambiguity, context, multilingual support, schema complexity

A_Linguistic Ambiguity and: Linguistic ambiguity and contextual dependence make text-to-SQL semantic parsing difficult, Due to word and phrase contextual heterogeneity, translating natural language into database schema components is difficult, Understanding schema element placement in SQL clauses (SELECT, FROM, WHERE, etc.) is essential for accurate interpretation, Valid queries require accurate SQL Semantic Prediction (SSP) label prediction by the model, SSP mapping errors cause SQL generating issues,Complex and compositionally variable natural language queries require models to generalize across phrasings and query formats.

B_Multilingual Support: Text-to-SQL systems lack multilingual capability, a major concern. Many database schemata are in English, but customers often need to query them in other languages, Most modern Text-to-SQL solutions are monolingual, with natural language queries and database schemas in the same language, Moving from a monolingual system to a cross-lingual framework where natural language inquiry differs from database design is difficult, Traditional approaches sometimes use back-translations or machine translation, which are resource-intensive, inaccurate, and require high-quality translation systems ,Translations may differ from test instances, resulting in poor generality, FastRAT uses a cross-lingual multi-task pre-training architecture and distant supervision to improve semantic parsers for new languages without machine translation ,This strategy reduces encoder representation language distribution differences, improving cross-lingual competency.

C_Schema Complexity: Text-to-SQL methods struggle with schema complexity since modern databases have many interrelated tables and columns, The system must accurately recognize and correlate schema elements with plain language queries, which is harder in cross-database systems with unfamiliar schemas, Complex queries require efficient schema linking and correct JOIN relationships, FastRAT uses SQL Semantic Prediction (SSP) labels to contain schema element contexts, although constraints remain, Complex SQL queries with many clauses or nested sub-queries are difficult to reconstruct, highlighting the difficulty of managing SQL schema complexity[9].

4_literature review

Gao et al. (2024) benchmarked LLM-based Text-to-SQL systems to address the lack of a comprehensive framework for efficient and cost-effective solutions, Rapid engineering methods including question formats, example selection, and organizational approaches were compared, as well as open-source LLMs with supervised fine-tuning, The Spider and Spider-Realistic datasets were used to introduce DAIL-SQL, which achieved 86.6% execution accuracy on the Spider leaderboard with improved token efficiency, The novel DAIL-SQL technology and extensive benchmarking are strengths, In-context learning is limited by open-source LLMs' tendency to overfit after fine-tuning[4].

Shi et al. (2025) conducted a comprehensive text-to-sql LLM survey, The paper details fast engineering and fine-tuning techniques, phases, and applications, It ranks benchmarks by LLM progress and highlights breakthroughs such LLM-based approaches' improved performance and generalization, Precision LLM-based approaches boost SOTA performance, as evidenced by SPIDER execution accuracy, The most advanced Text-to-SQL models (e.g., GPT-4) achieve 54.89% execution accuracy on the BIRD dataset, far behind human performance (92.96%), demonstrating ongoing challenges and a large gap between academic progress and practical application, The unique instruction-following paradigm and consistent architecture are benefits, but data needs, closed-source model privacy issues, and real-world schema complexity are weaknesses, Despite progress, LLM accuracy on real-world datasets like BIRD and Spider 2.0 remains low, demonstrating a substantial gap between LLM performance and human talent[10] .

Chen, Zhang, and Deng (2022) explored text-to-SQL advances that encode, parse, and decode natural language into SQL queries, The procedure evaluated dataset, encoding, decoding, learning, and evaluation, Used: Academic, IMDB, WikiSQL, Spider, Text-to-SQL research expansion, data augmentation, and graph-based methods are important, The standard warns that naive execution accuracy may misidentify semantically diverse SQL queries. Precision String match requirements are overly strict since strings can mean the same, Spider metrics like Exact Set Match (ESM) and databases that separate predict SQL from gold queries simplify problems. Simpler SQL queries help models perform better, according to the essay, A thorough procedural review is good, Cross-domain generalization requires durable, user-focused, multilingual solutions. Traditional metrics prefer accurate set matching for complex queries due to accuracy difficulties[11].

Tai et al. (2023) studied how Chain-of-Thought (CoT) prompting affects LLM reasoning in text-to-SQL processing, They used Chain-of-Thought (CoT) and Least-to-Most prompting approaches and introduced Question Decomposition Prompting (QDecomp) and QDecomp+InterCOL, Codex on Spider, Spider Realistic, and other datasets showed that repetitive prompting is often unnecessary and that complex reasoning techniques may propagate errors, QDecomp+InterCOL outperformed alternatives with test-suite accuracies of 68.4% on Spider and 56.5% on Spider Realistic, demonstrating resilience and reducing error propagation, Methodical investigation and improved accuracy are strengths, although Codex reliance and limited robustness testing across database situations are weaknesses[12].

ASTRES, developed by Shen et al. (2024) improves Text-to-SQL semantic parsing through retrieval-augmented generation, They approximate SQL queries using normalized Abstract Syntax Trees for re-ranking and an efficient, schema-parallelizable approximator, A hybrid schema selection method dynamically removes schema items and selects values, reducing LLM workload, ASTRES performs well on SPIDER and CSPIDER, improving execution and exact match accuracy (86.6% EX, 77.3% EM on the SPIDER development set with GPT-4), Erroneous approximators and schema selection recollection issues are limitations[13].

Zhang et al. (2024) thoroughly studied NLI tabular data access and visualization, Research studied approaches from rule-based systems to neural networks to Large Language Models, Performance was assessed using WikiSQL, Spider, and nvBench. Key findings include semantic parsing's role in formalizing natural language and querying and visualization's interaction, Accuracy varies substantially by approach and dataset, Tables exhibit execution accuracy (EX%), exact match (EM%), and overall accuracy (Acc.%) for WikiSQL, Spider, and nvBench text-to-sql and text-to-vis datasets, Spider

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v2.i2.a12

EM% is 24.8% (SyntaxSQLNet) to 85.96% (UnifiedSKG), while WikiSQL EX% is 40.7% (GNN) to 92.3% (X-SQL), While neural networks' data dependency and LLM interpretability are shortcomings, LLMs' robust generalization and rule-based systems' interpretability are strengths [14].

"Semantic Synthesis," developed by Jha and Anand (2025) converts natural English into precise SQL queries, The approach uses Generative AI, LLMs, transformers, and an easy-to-use interface, Key findings show that the approach democratizes database engagement, provides detailed algorithmic explanations, and is adaptable across fields, Increased accessibility, transparency, and user-centered design are strengths, Constrained areas include informal language, generative model interpretability, error management, security, and user education, The study shows precise query development, improving database interface for non-SQL users[15].

Kim et al. (2020) critically examined Natural Language to SQL (NL2SQL) technologies' current state and restrictions, Over more than 10 benchmarks, eleven modern methodologies were tested, including a unique semantic equivalence assessment tool, The study used WikiSQL, ATIS, and Spider databases to show that accuracy metrics are often misleading, Strong points include a consistent evaluation framework and improved validation, Existing benchmarks are limited and rule-based and deep learning systems struggle with unseen data or complex queries, The validation tool had 99.61% accuracy, while the NL2SQL technique was inconsistent and often failed in complex real-world circumstances[16].

Shen and Kejriwal (2024) studied SelECT-SQL, a novel Text-to-SQL in-context learning method, The methodology included chain-of-thought (CoT) prompting, self-correction, and ensemble techniques with GPT-3.5-Turbo as the core large language model, Spider development set was used for testing, SelECT-SQL outperformed GPT-3.5-Turbo and GPT-4 with 84.2% execution accuracy, Innovative performance, economic efficiency, and modular architecture are strengths, GPT's tendency to make incorrect nested queries, string matching issues, and performance decrease on complex schemas are limitations, The research shows that the suggested strategy improves Text-to-SQL precision and efficienc[17].

Chafik et al. (2025) studied text-to-SQL security in light of huge language model vulnerabilities, SQLSHIELD, a novel dataset of malicious and benign NLQ-SQL pairs, and two transformer-based models, SQLPROMPTSHIELD and SQLQUERYSHIELD, were fine-tuned to detect harmful prompts and SQL queries. The models were in an agent-based text-to-SQL system. SQLPROMPTSHIELD has 0.997 accuracy and SQLQUERYSHIELD 0.998 accuracy, indicating a 70% security improvement and good detection model precision, Strong security with low overhead and a large dataset are strengths, Limitations include poor attack methodology coverage, source dataset bias, and the need for continual upgrades to address evolving threats[18].

A systematic review of Text-to-SQL by KANBUROĞLU and TEK (2024) used the PRISMA approach to analyze challenges and models, Current literature, WikiSQL, Spider, and assessment metrics were examined, Key findings highlighted ordering and schema representation issues and showed that execution-guided decoding improves model correctness, ChatGPT had 73.83% Spider execution accuracy in the LLM research,The strengths are LLM concentration and detailed review, LLMs are promising for cross-linguistic tasks, but non-English datasets are few, Finally, the study discusses future research in this dynamic topic[19].

GSN Murthy et al. (2025) studied an AI-driven translator that transforms natural language to database queries, The approach used advanced NLP and LLMs to analyze user queries and generate appropriate MySQL and MongoDB SQL or NoSQL instructions, System modules include user authentication, database selection, query processing, execution, and result presentation, Significant findings show that natural language inputs can be converted into executable queries, enhancing data analytics, customer service, and business intelligence productivity for non-technical users, Strengths include increased accessibility, precision, efficiency, and security, It ensures precise query construction and provides explicit error messages for reliability[20].

GSN Murthy et al. (2025) studied an AI-based translator that transforms natural language to database queries, User queries were evaluated and MySQL and MongoDB SQL or NoSQL commands generated using advanced NLP and LLMs, The system modules handle user authentication, database selection, query processing, execution, and result presentation, Several studies show that natural language inputs can be converted into executable queries, increasing data analytics, customer service, and business intelligence for non-technical users, Improved accessibility, accuracy, efficiency, and security, Dependability is ensured via precise query formulation and unambiguous error alerts[21].

TABL1 Summery of Comparison literature review

Authors & Year	Title/Focus	Methodology	Key Datasets	Accuracy/Performance	Strengths	Limitations	Key Contributions
Gao et al. (2024)	Systematic benchmark evaluation of LLM-based Text-to-SQL solutions	Extensive comparisons of prompt engineering methods, supervised fine-tuning of open-source LLMs	Spider, Spider-Realistic	86.6% execution accuracy on Spider leaderboard	Comprehensive benchmarking, novel DAIL-SQL solution	Potential overfitting of open-source LLMs, reduced in-context learning capabilities	DAIL-SQL framework with improved token efficiency
Shi et al. (2025)	Comprehensive survey of LLMs for Text-to-SQL	Review of prompt engineering and fine-tuning approaches	SPIDER, BIRD	GPT-4: 54.89% on BIRD (vs 92.96% human performance)	Novel instruction-following paradigm, uniform architecture	Data requirements, privacy concerns, complexity of real-world schemas	Comprehensive LLM survey and performance analysis
Deng, Chen, and Zhang (2022)	Comprehensive survey on Text-to-SQL advances	Review of encoding, translating, and decoding methods	Academic, IMDB, WikiSQL, Spider	Better performance on shorter SQL queries	Systematic review, detailed coverage of techniques	Poor cross-domain generalization, need for robust multilingual systems	Evaluation metric improvements (ESM over naive execution accuracy)
Tai et al. (2023)	Chain-of-Thought prompting for Text-to-SQL	Adapted CoT and Least-to-Most prompting, proposed QDecomp+InterCOL	Spider, Spider Realistic	QDecomp+InterCOL: 68.4% (Spider), 56.5% (Spider Realistic)	Systematic exploration, improved accuracy, reduced error propagation	Reliance on Codex, limited robustness testing	Question Decomposition Prompting (QDecomp) method
Shen et al. (2024)	ASTRES framework for retrieval-augmented generation	Normalized Abstract Syntax Trees, schema-parallelizable approximator	SPIDER, CSPIDER	86.6% EX, 77.3% EM on SPIDER dev set (GPT-4)	State-of-the-art performance, hybrid schema selection	Potential bias from inaccurate approximators, schema selection recall challenges	ASTRES framework with retrieval-augmented generation
Zhang et al. (2024)	Survey on Natural Language Interfaces for tabular data	Analysis of evolution from rule-based to neural to LLM-based approaches	WikiSQL, Spider, nvBench	WikiSQL: 40.7%-92.3% EX%, Spider: 24.8%-85.96% EM%	Strong LLM generalization, interpretable rule-based systems	Neural network data dependency, LLM interpretation challenges	Comprehensive NLI evolution analysis
Jha & Anand (2025)	Semantic Synthesis approach	Integration of Generative AI, LLMs, transformers with user-friendly UI	Not specified	High accuracy in query generation	Enhanced accessibility, transparency, user-centric design	Colloquial language challenges, security concerns, need for user education	Democratized database interaction system
Kim et al. (2020)	Comprehensive NL2SQL technology assessment	Experimentation with 11 techniques across 10+ benchmarks	WikiSQL, ATIS, Spider	Validation tool: 99.61% accuracy, NL2SQL methods varied significantly	Unified evaluation framework, improved validation methodology	Narrow benchmark scope, challenges with unseen data and complex queries	Novel validation tool for semantic equivalence
Shen and Kejriwal (2024)	SelECT-SQL in-context learning solution	Chain-of-thought prompting, self-correction, ensemble methods	Spider development set	84.2% execution accuracy	State-of-the-art performance, cost-efficiency, modular design	Performance drops on complex schemas, string matching issues	SelECT-SQL framework outperforming GPT-3.5 and GPT-4
Chafik et al. (2025)	Security enhancement in text-to-SQL systems	SQLSHIELD dataset creation, transformer-based detection models	SQLSHIELD (custom dataset)	SQLPROMPTSHIELD: 0.997 accuracy, SQLQUERYSHIELD: 0.998 accuracy	70% security enhancement, robust security with low overhead	Incomplete attack coverage, bias propagation, need for continuous updates	SQLSHIELD security framework
Kanburoğlu and Tek (2024)	Methodical Text-to-SQL review using PRISMA	Systematic literature review, LLM performance evaluation	WikiSQL, Spider	ChatGPT: 73.83% execution accuracy on Spider	Systematic review methodology, LLM focus	Scarcity of non-English datasets	PRISMA-based systematic review framework
GSN Murthy et al. (2025)	AI-powered natural language to database query translator	Advanced NLP and LLMs for MySQL and MongoDB query conversion	Not specified	Successful conversion with high accuracy	Improved accessibility, accuracy, efficiency, robust security	Not explicitly mentioned	Multi-database (SQL/NoSQL) natural language interface
Kombade et al. (2020)	Natural language and speech to SQL conversion	Multinomial Logistic Regression for query type prediction	Generated training data	98.65% query type prediction accuracy	User-friendly interface, speech input support, broad applicability	Not explicitly detailed	Speech-enabled natural language to SQL system

5-Datasets and Benchmarks

WikiSQL (2017) [22]: The 2017 introduction of this extensive, cross-domain text-to-SQL dataset has transformed natural language processing research , It comprises tables from the English Wikipedia, along with natural language inquiries pertaining to them and corresponding SQL queries, The dataset is intended for single-turn text-to-SQL tasks, producing a SQL query from a natural language question without conversational context. WikiSQL has 80,654 natural language queries and 24,241 experimental and evaluative tables. The dataset consists of 56,355 training records, 8,421 development records, and 15,878 evaluation records.

Spider (2018) [19]: The Spider (2018) dataset is essential for Text-to-SQL jobs that need complex and cross-domain semantic processing, There were 10,181 natural language searches and 5,693 complex SQL queries from 200 databases in 2018,

Complex SQL query components like JOIN, NESTED searches, GROUP BY, and ORDER BY clauses are rare in single-domain datasets, Spider's intricacy requires Text-to-SQL models to generalize across film databases, location, and sports, Its cross-domain property makes it a tough model robustness and adaptability benchmark.

The BIRD dataset (2023) [23]: The Brid is a comprehensive dataset designed for large-scale database text-to-SQL tasks, effectively connecting academic research with industrial applications,It comprises 12,751 text-to-SQL pairs across 95 databases, totaling 33.4 GB and covering 37 unique professional domains.

Other Text-to-SQL Benchmarks and Query Accuracy Evaluation [24]: WikiSQL and Spider are not the sole text-to-SQL benchmarks, other others possess distinct characteristics and evaluation criteria, Domain-specific datasets such as ATIS, GeoQuery, Restaurants, Academic, IMDb, Yelp, Scholar, Advising, MIMICSQL, FIBEN, and SEDE pertain to a singular database,These datasets exhibit commendable performance within their respective domains; nonetheless, they are smaller and less generalizable than WikiSQL or Spider.

Model constraints, domain maturity, and execution precision Due to real-world data's complexity, execution accuracy varies widely, BIRD, a benchmark of 12,751 text-to-SQL pairs spanning 95 databases and 37 domains, addresses noisy or missing data, natural language alignment with external information, and efficient SQL generation for large datasets, Complex, noisy metadata and 18-21 SQL token queries increase variation.

Existing LLM models, including complex versions like GPT-4, have performance gaps, such as 54.89% execution accuracy compared to 92.96% for humans.

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v2.i2.a12

Inconsistent information makes database architecture and relationships harder for models to grasp. These models struggle with complex queries with many joins, nested subqueries, and conditional statements.

The BIRD benchmark shows the field's sophistication as standards rise. This benchmark emphasizes real-world situations to link academic research to practice.

Model validation is complex when confidence intervals and Valid Efficiency Score (VES) are used to quantify correctness and computing efficiency [23].

TABL2 Summery of Comparison of SQL Benchmark Datasets

Dataset	Year	Question	Tables /Databases	Key Focus	Complexity Level	Domain Scope	Examples	Main Advantage	Main disadvantage
WikiSQL	2017	80,654	24,241 tables	Single-table, simple queries	Low	General (Wikipedia-based)	Various Wikipedia topics	Simple and easy for basic text-to-SQL tasks.	Lacks multi-table and conversational complexity.
Spider	2018	10,181	200 databases (5,693 unique queries)	Cross-domain, complex SQL	High	Cross-domain	Movies, geography, sports, etc.	Covers complex, multi-domain SQL queries for strong generalization.	Hard to achieve high accuracy due to query and schema complexity.
BIRD	2023	12,751	95 databases (33.4 GB data)	Real-world, large-scale	Very High	Professional domains	37 distinct professional fields	Focuses on real-world, noisy, and complex databases for practical text-to-SQL use.	Low model accuracy compared to humans, showing task difficulty.
Others	Various	<1,000 typically	Domain-specific	Specialized domains	Variable	Single/specific domains	ATIS (flights), GeoQuery (geography), Restaurants, Academic	Includes complex SQL with numerical computations and variable declarations, reflecting real-world use.	Lacks standard train/test splits, making fair model comparison difficult.

6_Generative AI Models for NLIDB

6.1_early Seq2Seq models: : Typically constructed using recurrent neural networks (RNNs) such as Long Short-Term Memory (LSTM) networks or Gated Recurrent Units (GRUs), the architecture comprises an encoder that transforms the input sequence into a fixed-length context vector and a decoder that produces the output sequence from this context vector, This design was essential for jobs like machine translation, where an input sentence in one language is encoded and subsequently decoded into another language ,This fundamental concept involves an encoder processing the input sequence into a fixed-length context vector, and a decoder then generating an output sequence from that vector.

6.2_AI model: This method generates SQL queries from natural language instructions, particularly voice input, but does not mention or explain fine-tuned Large Language Models (LLMs) like ChatGPT, Codex, or LLaMA for text-to-SQL synthesis, NLP and Neural Networks are used to create a Long Short-Term Memory (LSTM) model to predict SQL queries from natural language input, An LSTM model trained on natural language input generates a query skeleton, Translations show the model's accuracy at 93.47%. Designed to help users. Voice commands can do SQL queries for non-sql users.

6.3_As for hybrid models: This method describes a system that tokenizes, stop word eliminates, and lemmatizes natural language input before feeding it into the model,The system relies on an LSTM model to determine user intent and deliver a 'tag' for the desired function or query, A dataset's skeleton structured query is processed with database metadata to create an executable SQL query by this tag, This method uses a generative LSTM model for intent recognition and skeletal query retrieval and structured components like predefined skeletal queries and database metadata for rule-based or grammar-guided generation, but implicitly through the dataset architecture, The system's dataset contains 'patterns' (possible user words) and'responses' (basic SQL queries), showing a template-driven or rule-based connection[25].

6.4_Transformer-based models: T5, BERT-to-SQL, and GPT-3/4 represent significant NLP advances and have potential in many disciplines, including database interactions, These designs, from T5 and BERT-to-SQL for natural language-to-SQL translation to GPT-3/4 for more adaptable models, can alter user engagement with databases, They can improve natural language searches and enable complex query building[26].

7-Applications

7.1_Business Intelligence (BI) systems: Intelligent ChatBots, Robotics, and Business Intelligence require NLIDB systems,These technologies allow non-technical people to easily access and extract data from databases, making them ideal for departments that need data but lack SQL skills, The basic idea is to convert natural language queries into a format a Database Management System (DBMS), usually SQL, can understand, execute, and return the answers to the user. This approach makes data access easy for non-SQL users[27].

7.2_Advancements in Healthcare Data Analysis: Artificial intelligence and machine learning are profoundly revolutionizing healthcare through the facilitation of sophisticated data searching and analysis. These technologies are essential for predictive capabilities in early illness detection, improving cancer treatment, and transforming patient diagnoses through thorough evaluations of AI's influence on medical diagnosis. AI-driven innovations facilitate vaccine creation and enhance brain health analysis breakthroughs. Additionally, AI and NLP applications are being investigated in COVID-19 rehabilitation, while AI-driven automation is transforming industrial processes, perhaps yielding indirect advantages for healthcare data management.

7.3_Impact on Education and Beyond: The main focus is aerodynamic efficiency, but AI and machine learning affect other fields, including education, AI apps are helping youngsters with learning disabilities, showing the technology's adaptability to data-driven problems, Aerodynamic form optimization shows AI's ability to quickly absorb and understand complex data patterns, which could improve educational data querying, personalize learning, and boost administrative efficiency, The theme of using AI to improve performance and efficiency applies to healthcare and education.[28].

7.4_Support for non-technical users in organizations: AI-driven query optimization vastly increases dashboard responsiveness and data engagement for non-technical business users, NLP is vital for turning user-intended actions into effective SQL queries, especially in self-service Business Intelligence (BI) systems where users can submit natural language queries, AI systems may analyze the semantic structure of existing searches, simplifying complex logic and removing unnecessary pieces, making data more accessible to non-technical users, AI can also customize dashboard experiences by analyzing user behaviors, pre-optimizing commonly accessed data segments, and prioritizing relevant data prefetching, resulting in faster and more responsive dashboards tailored to specific requirements, improving user satisfaction and productivity, even for non-technical users[29].

7.5_Integration with Big Data management systems: Creative businesses manage large, complex datasets, including digital assets and multimedia information, which conventional database systems struggle to handle due to their dynamic and unstructured nature, Generative AI automates metadata tagging, optimizes content retrieval, and improves data-driven decision-making. Enterprises may streamline workflows, reduce manual participation, and increase creativity by integrating Generative AI models like Large Language Models (LLMs) and Generative Adversarial Networks (GANs) into database systems. This integration turns static data management into intelligent, self-optimizing infrastructures needed to manage creative sectors' exponential digital information growth[30].

8_Challenges and Future Directions

8.1_Challenges in Natural Language to SQL Translation Linguistic ambiguity and vagueness make translating natural language questions (NLQ) into SQL queries problematic , making user intent interpretation challenging, modern in-context learning with chain-of-thought (COT) prompting struggles with difficult tasks , while early text-to-SQL methods used expensive labeled data and supervised learning , Extended prompts may distract the model , causing missing data and poor performance , the "seesaw effect" illustrates that one prompting approach does not solve all difficulties , DEA-SQL workflows simplify difficult problems and emphasize LLM , It uses human cognitive methods to break down problems , reduce unnecessary information , and focus the LLM on specific subtasks [30].

8.2_Supporting Complex and Nested Queries Microsoft Fabric's GraphQL API handles complex, layered queries well, making it ideal for generative AI applications like RAG, GraphQL's OneLake-based consolidated query interface allows quick extraction of complicated data relationships in a single request, unlike traditional REST APIs, This is useful for generative AI applications like model training and inference that demand fast resolution of complex data structures. The system's ability to manage simultaneous queries with good response times and query optimization reduces data retrieval overhead and increases real-time analytics and AI inference application performance[31].

8.3_ Integration with multimodal interface (text , voice , visualization) Future Text-to-SQL systems will have multimodal interfaces to promote user engagement and support several query forms, Speech and diagram recognition allow users to enter information beyond text-based queries, Using voice commands to retrieve data improves system accessibility and intuitiveness, especially for people who prefer speaking to typing or are in situations where typing is impractical. Schematic recognition can let users visually identify links or select data elements, enhancing database interaction, These advancements are necessary to create more resilient and user-friendly database query interfaces that go beyond textual input and accommodate a wider range of user preferences and operational settings[32].

8.4_Development of interactive conversational NLIDB Research is needed on interactive conversational Natural Language Interface to Database (NLIDB) systems, especially those leveraging Large Language Models, These technologies eliminate sophisticated query languages and let users interface with databases in plain language. While not explicitly addressing 'Development of interactive conversational NLIDB,' the publication emphasizes important principles and applications in the context of AEC LLMs, LLMs respond like humans and perform general tasks with little training, making them suitable for conversational interfaces and question-answering systems, LLM cognitive abilities and language understanding and production determine these interfaces. Text indicates LLMs excel with conversational interfaces,This shows that interactive conversational NLIDBs that receive user queries and give relevant data from structured data sources require LLMs' natural language comprehension and generation ability[33].

9-CONCLUSIONS

This evaluation illustrates the significant breakthroughs made in Natural Language Interface to Databases (NLIDB) and Text-to-SQL systems, mostly propelled by developments in Generative AI and Large Language Models (LLMs). Although LLM-based methodologies markedly improve contextual comprehension and execution precision, they continue to encounter enduring difficulties in managing intricate and deeply nested SQL queries, cross-domain generalization, and multilingual capabilities. Real-world benchmarks like BIRD underscore the significant disparity between the performance of contemporary LLMs and human-level competence .Recent advances, including retrieval-augmented generation, schema-aware ranking, prompt-engineering procedures, and question-decomposition approaches, provide significant enhancements in efficiency and reliability. Nonetheless, challenges of interpretability, robustness, and security against malevolent natural language inputs persist as significant concerns .Future advancements will depend on the development of conversational NLIDB systems, multimodal interfaces, enhanced schema-linking mechanisms, and improved cross-lingual models. Generative AI has the capacity to democratize database access, optimize analytical operations, and enhance SQL querying accessibility for non-technical users.

10-REFERENCES

[1] J. Kossmann, T. Papenbrock, and F. Naumann, “Data dependencies for query optimization: a survey,” *VLDB J.*, vol. 31, no. 1, pp. 1–22, 2022, doi: 10.1007/s00778-021-00676-3.

[2] N. Rajkumar, R. Li, and D. Bahdanau, “Evaluating the Text-to-SQL Capabilities of Large Language Models,” 2022, [Online]. Available: http://arxiv.org/abs/2204.00498

[3] N. Nihalani, S. Silakari, and M. Motwani, “Natural language Interface for Database: A Brief review,” *Ijcsi*, vol. 8, no. 2, pp. 600–608, 2011, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.402.9531&rep=rep1&type=pdf#page=621

[4] D. Gao *et al.*, “Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation,” *Proc. VLDB Endow.*, vol. 17, no. 5, pp. 1132–1145, 2024, doi: 10.14778/3641204.3641221.

[5] C. J. F. Candel, D. Sevilla Ruiz, and J. J. García-Molina, “A unified metamodel for NoSQL and relational databases,” *Inf. Syst.*, vol. 104, no. January 2021, p. 101898, 2022, doi: 10.1016/j.is.2021.101898.

[6] Chaitanya Bharat Dadi, “Natural Language Interfaces for Database Management: Bridging the Gap Between Users and Data through Conversational AI,” *J. Comput. Sci. Technol. Stud.*, vol. 7, no. 3, pp. 927–933, 2025, doi: 10.32996/jcsts.2025.7.3.103.

[7] R. Ofori-Boateng, M. Aceves-Martins, N. Wiratunga, and C. F. Moreno-Garcia, *Towards the automation of systematic reviews using natural language processing, machine learning, and deep learning: a comprehensive review*, vol. 57, no. 8. Springer Netherlands, 2024. doi: 10.1007/s10462-024-10844-w.

[8] K. Yalova, K. Yashyna, and A. B. M. Salem, “Natural Language Interface to Database Approach in the Task of Relational Databases Design,” *CEUR Workshop Proc.*, vol. 3373, pp. 320–331, 2023.

[9] P. Vougiouklis *et al.*, “FastRAT: Fast and Efficient Cross-lingual Text-to-SQL Semantic Parsing,” vol. 1, pp. 564–576, 2024, doi: 10.18653/v1/2023.ijcnlp-main.38.

[10] L. Shi, Z. Tang, N. Zhang, X. Zhang, and Z. Yang, “A Survey on Employing Large Language Models for Text-to-SQL Tasks,” *ACM Comput. Surv.*, vol. 1, no. 1, pp. 1–36, 2025, doi: 10.1145/3737873.

[11] N. Deng, Y. Chen, and Y. Zhang, “Recent Advances in Text-to-SQL: A Survey of What We Have and What We Expect,” *Proc. - Int. Conf. Comput. Linguist. COLING*, vol. 29, no. 1, pp. 2166–2187, 2022.

[12] C. Y. Tai, Z. Chen, T. Zhang, X. Deng, and H. Sun, “Exploring Chain of Thought Style Prompting for Text-to-SQL,” *EMNLP 2023 - 2023 Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 5376–5393, 2023, doi: 10.18653/v1/2023.emnlp-main.327.

[13] Z. Shen, P. Vougiouklis, C. Diao, K. Vyas, Y. Ji, and J. Z. Pan, “Improving Retrieval-augmented Text-to-SQL with AST-based Ranking and Schema Pruning,” *EMNLP 2024 - 2024 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 7865–7879, 2024, doi: 10.18653/v1/2024.emnlp-main.449.

[14] W. Zhang *et al.*, “Natural Language Interfaces for Tabular Data Querying and Visualization: A Survey,” *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 6699–6718, 2024, doi: 10.1109/TKDE.2024.3400824.

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v2.i2.a12

[15] A. Jha, N. Anand, and H. Karthikeyan, "Conversion of natural language text to SQL queries using generative AI," *Hybrid Adv. Technol.*, pp. 25–32, 2025, doi: 10.1201/9781003559139-3.

[16] H. Kim, B. H. So, W. S. Han, and H. Lee, "Natural language to SQL: Where are we today?," *Proc. VLDB Endow.*, vol. 13, no. 10, pp. 1737–1750, 2020, doi: 10.14778/3401960.3401970.

[17] K. Shen and M. Kejriwal, "SelECT-SQL: Self-correcting ensemble Chain-of-Thought for Text-to-SQL," pp. 1–15, 2024, [Online]. Available: http://arxiv.org/abs/2409.10007

[18] S. Chafik, S. Ezzini, and I. Berrada, "Enhancing security in text-to-SQL systems: A novel dataset and agent-based framework," *Nat. Lang. Process.*, pp. 1–24, 2025, doi: 10.1017/nlp.2025.10008.

[19] A. B. Kanburoğlu and F. B. Tek, "Text-to-SQL: A methodical review of challenges and models," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 32, no. 3, pp. 403–419, 2024, doi: 10.55730/1300-0632.4077.

[20] G. S. N. Murthy, K. Anshu, T. Srilakshmi, C. Sumanth, and M. N. Mounika, "Ai Powered Translator: Transforming Natural Language To Database Queries," *Int. J. Eng. Appl. Sci. Technol.*, vol. 09, no. 11, pp. 39–43, 2025, doi: 10.33564/ijeast.2025.v09i11.006.

[21] C. Kombade, "Natural Language Processing with some Abbreviation to SQL," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 5, pp. 1046–1048, 2020, doi: 10.22214/ijraset.2020.5166.

[22] G. Jeong *et al.*, "Improving Text-to-SQL with a Hybrid Decoding Method," *Entropy*, vol. 25, no. 3, pp. 1–20, 2023, doi: 10.3390/e25030513.

[23] S. Chauhan, "Knowledge Discovery in Databases Utilizing Large Language Models," *Int. J. Sci. Res.*, vol. 13, no. 10, pp. 1886–1894, 2024, doi: 10.21275/ms241026170018.

[24] G. Katsogiannis-Meimarakis and G. Koutrika, "A survey on deep learning approaches for text-to-SQL," *VLDB J.*, vol. 32, no. 4, pp. 905–936, 2023, doi: 10.1007/s00778-022-00776-8.

[25] A. Sawant, R. Raina, A. Patil, and A. Pardeshi, "AI Model to Generate SQL Queries from Natural Language Instructions through Voice," *J. Phys. Conf. Ser.*, vol. 2273, no. 1, 2022, doi: 10.1088/1742-6596/2273/1/012014.

[26] H. Gadde, "Integrating AI into SQL Query Processing: Challenges and Opportunities," *Int. J. Adv. Eng. Technol. Innov.*, vol. 01, no. 03, p. 3, 2022.

[27] D. Chandarana, M. Mathkar, A. Patil, and ..., "Natural Language Sentence to SQL Query Converter," *Int. J. ...*, vol. 9, no. Query date: 2024-09-01 20:09:48, pp. 467–472, 2021, [Online]. Available: https://vcet.edu.in/NAAC/3/3.3.2/3.3.2_219.pdf

[28] S. Sahibzada, F. S. Malik, S. Nasir, and S. K. Lodhi, "Generative AI Driven Aerodynamic Shape Optimization: A Neural Network-Based Framework for Enhancing Performance and Efficiency," *Int. J. Innov. Res. Comput. Sci. Technol.*, vol. 13, no. 1, pp. 98–105, 2025, doi: 10.55524/ijircst.2025.13.1.15.

[29] F. Chad, "AI-Driven Query Optimization for Dashboard Upgrades AUTHOR : Felix Chad DATE : 25 th April 2025 Abstract :," no. April, 2025.

[30] O. Oloruntoba, "Generative AI for Creative Data Management: Optimizing Database Systems in the Creative Industry," *Iconic Res. Eng. Journals*, vol. 7, no. 7, pp. 588–597, 2024.

[31] T. Studies, "Advancing Generative AI with GraphQL API: Unified Data Access in Microsoft Fabric Ecosystem," no. June 2025, pp. 438–450, doi: 10.32996/jcsts.

[32] A. Kaygude, O. Rajguru, S. Karad, and G. T. Avhad, "Education and TechnologY (IJARETY) Text-to-SQL Conversion by using Deep Learning / Machine Learning : Integrating Natural Language with Database Queries," vol. 12, no. 3, 2025, doi: 10.15680/IJARETY.2025.1203098.

[33] D. Kampelopoulos, A. Tsanousa, S. Vrochidis, and I. Kompatsiaris, "A review of LLMs and their applications in the architecture, engineering and construction industry," *Artif. Intell. Rev.*, vol. 58, no. 8, 2025, doi: 10.1007/s10462-025-11241-7.