

Text-to-SQL Systems: A Comprehensive Study of Evolution, Architectural Advancements, Challenges, and the Role of Large Language Models in Intelligent Query Generation

Yogesh Adhana^{1*}, Shefali Madan²

Abstract

Interacting with relational databases traditionally requires knowledge of Structured Query Language (SQL), which creates a significant barrier for users who do not have technical expertise. To address this challenge, Text-to-SQL systems have emerged as an important area of research, enabling users to express their information needs in natural language while automatically generating executable SQL queries. These systems aim to make database access more intuitive, efficient, and accessible across a wide range of domains. This paper provides a structured review of the development of Text-to-SQL systems, tracing their evolution from early rule-based and template-driven methods to more advanced machine learning, deep learning, and Large Language Model (LLM)-based approaches. It analyzes the major techniques used in these systems and highlights how recent advances in Natural Language Processing (NLP) have improved contextual understanding, semantic parsing, and the accuracy of SQL query generation. The paper also examines the benefits of modern LLM-driven architectures, particularly their ability to generalize across diverse schemas and handle more complex user requests. At the same time, several persistent challenges remain, including schema linking, ambiguity in natural language queries, domain adaptation, interpretability, and the high computational cost of large models. Emerging directions such as Retrieval-Augmented Generation (RAG), prompt-based learning, and hybrid architectures are also discussed as promising solutions. Overall, the study concludes that LLM-based Text-to-SQL systems represent a major advancement, but further research is necessary to improve their efficiency, reliability, and adaptability in real-world applications.

Keywords: Text-to-SQL, natural language processing, large language models, query generation, database systems, RAG

*Author for Correspondence

Yogesh Adhana
E-mail: yadhana12@gmail.com

¹Student, Department of Computer Science and Engineering
J.C. Bose University of Science and Technology, Faridabad,
Haryana, India.

²Associate Professor, Department of Computer Science and
Engineering J.C. Bose University of Science and Technology,
Faridabad, Haryana, India.

Received Date: April 25, 2026

Accepted Date: May 25, 2026

Published Date: June 30, 2026

Citation: Yogesh Adhana, Shefali Madan. Text-to-SQL Systems: A Comprehensive Study of Evolution, Architectural Advancements, Challenges, and the Role of Large Language Models in Intelligent Query Generation. Journal of Advanced Database Management & Systems. 2026; 13(2): 10–19p.

INTRODUCTION

With the rapid expansion of structured data across industries, there is an increasing need for intuitive methods to access and analyze this information. Traditional database interactions rely on SQL, a language that requires technical expertise and familiarity with database schemas. This creates a barrier for users who lack programming skills but still require access to data insights.

Text-to-SQL systems are designed to simplify database access by enabling users to ask questions using natural language instead of writing SQL

queries manually. The system then interprets the query and generates the corresponding SQL commands. Early solutions relied on manually crafted rules and templates, which worked well for simple tasks but lacked flexibility [1].

With the development of machine learning methods, systems have become capable of learning and adapting beyond fixed or preprogrammed patterns. However, these approaches require extensive feature engineering and struggle with complex queries. The introduction of deep learning models improved the ability to capture contextual meaning, but challenges related to schema understanding and multi-table queries remained [2].

More recently, transformer-based Large Language Models have reshaped the landscape. These models can understand context, relationships, and semantics more effectively, leading to improved SQL generation accuracy. This study examines the evolution of these systems and analyzes current trends and future directions [3].

METHODOLOGY

This study adopts a structured and systematic approach to analyze the evolution and current state of text-to-SQL systems. The methodology was designed to ensure comprehensive coverage, objective comparison, and critical evaluation of existing techniques [4].

Data Sources

Relevant literature was collected from well-established academic databases and digital libraries, including IEEE Xplore, SpringerLink, ScienceDirect, and Google Scholar. These sources were selected due to their credibility and extensive coverage of research in Natural Language Processing, database systems, and artificial intelligence [5].

Selection Criteria

To ensure relevance and uniformity throughout the study, specific criteria were considered during the selection procedure.

- Publications dated between 2018 and 2025 to capture recent advancements
- Studies specifically focused on Text-to-SQL or Natural Language Interface to Databases (NLIDB)
- Papers that include experimental validation or performance evaluation
- Research addressing model architecture, datasets, or system-level implementations.

Studies that lacked technical depth or empirical results were excluded to ensure quality.

Review Process

The review process was conducted in multiple stages:

1. *Initial screening*: Titles and abstracts were examined to identify relevant studies.
2. *Detailed analysis*: Selected papers were thoroughly reviewed to understand their methodologies, contributions, and limitations.
3. *Categorization*: Papers were grouped based on their underlying approaches, such as rule-based, machine learning, deep learning, and large language model-based methods.
4. *Comparative evaluation*: Models were compared based on performance metrics, scalability, and ability to handle complex queries.

This structured approach enabled a clear understanding of trends and advancements in the field.

Evaluation Framework

To ensure consistent comparison across different systems, the following evaluation aspects were considered:

1. *Accuracy metrics*: Including exact match accuracy and execution accuracy.

2. *Query complexity handling*: Ability to process joins, nested queries, and aggregations.
3. *Schema linking capability*: Effectiveness in mapping natural language to database schema.
4. *Computational efficiency*: Resource requirements and inference time

EVOLUTION OF TEXT-TO-SQL SYSTEMS

Text-to-SQL systems have evolved considerably over the years owing to continuous improvements in Natural Language Processing and machine learning technologies. This section outlines the major stages of their evolution, highlighting the strengths and limitations of each approach [6].

Rule-Based Systems

Early Text-to-SQL systems relied on predefined rules, templates, and hand-crafted grammar to map natural language queries to SQL statements. These systems are interpretable and easy to design for restricted domains. However, they lack flexibility and are unable to handle variations in natural language or complex query structures. Consequently, their use was mainly restricted to basic applications designed for specific domains.

Machine Learning-Based Approaches

Machine learning techniques have been introduced to overcome the rigidity of rule-based systems. These methods use statistical techniques to understand and convert natural language statements into corresponding SQL queries [7]. Although they improved generalization compared to rule-based systems, they required extensive feature engineering and domain expertise. Additionally, their performance degraded when dealing with complex queries or unseen database schemas.

Deep Learning Models

The adoption of deep learning has marked a major advancement in text-to-SQL systems. Recurrent neural network models, particularly Long Short-Term Memory (LSTM) networks, improved the ability to process sequential patterns and capture contextual relationships within data. Sequence-to-sequence architecture further improves the translation of natural language into SQL queries. Despite these improvements, deep learning models often struggled with accurately capturing relationships within database schemas, particularly in multi-table queries involving joins and nested structures [8].

Transformer and LLM-Based Systems

The introduction of transformer architecture has significantly enhanced the capabilities of text-to-SQL systems. Large Language Models (LLMs), pre-trained on vast amounts of text and code data, have demonstrated strong performance in understanding context, semantics, and complex query intent. These models support zero-shot and few-shot learning, reducing the need for task-specific training.

LLM-based systems have demonstrated superior accuracy and scalability compared to earlier approaches. However, these systems face several limitations, including heavy computational demands, the possibility of generating inaccurate queries, and difficulties in providing clear explanations for their outputs. Despite these limitations, they currently represent the most advanced approach for Text-to-SQL tasks [9].

LITERATURE REVIEW

Recent progress in Natural Language Processing and deep learning techniques has played an important role in improving text-to-SQL systems. This section discusses the major research contributions by highlighting their approaches, advantages, and existing limitations.

Early Neural Approaches

Initial neural network-based models marked a shift from rule-based systems to data-driven approaches. Seq2SQL was among the first approaches to use a sequence-to-sequence framework along with reinforcement learning techniques for converting natural language questions into SQL

queries. This approach reduced reliance on handcrafted rules and demonstrated effectiveness for simple queries. However, it was primarily limited to single-table operations and struggled with complex SQL constructs, such as joins and nested queries [10].

Subsequently, SQLNet introduced a sketch-based framework that decomposed SQL query generation into independent components. This design reduced error propagation and improved training efficiency. Although SQLNet achieved higher accuracy than earlier models, it still faced challenges in handling complex queries and in understanding intricate schema relationships.

Benchmark Datasets and Their Impact

The introduction of large-scale benchmark datasets has played a crucial role in advancing Text-to-SQL research. The Spider dataset provides a diverse and complex collection of cross-domain queries spanning multiple databases. Unlike earlier datasets, Spider requires models to generalize across different schemas and query structures. While it significantly improved evaluation standards, it also increased the difficulty of the task, particularly in terms of schema linking and query composition.

Advanced Deep Learning Models

More sophisticated models have been developed to address the limitations of earlier approaches. IRNet introduced an intermediate representation that bridges the gap between natural language and SQL queries. This method improves interpretability and enables better handling of complex queries, although it introduces additional computational overhead.

Another notable model, RAT-SQL, leverages transformer-based architectures with relation-aware self-attention mechanisms. By explicitly modeling relationships within the database schema, RAT-SQL achieves strong performance on complex benchmarks. However, its high computational requirements make real-time implementation difficult.

Constrained Decoding Approaches

Ensuring the syntactic correctness of generated SQL queries became a key focus of later research. PICARD addressed this issue by incorporating constrained decoding techniques that enforce SQL grammar rules during the generation process. This approach significantly reduced invalid queries and improved execution accuracy. Despite these benefits, it introduced additional processing overhead during decoding.

Large Language Model-Based Approaches

Recent developments in transformer-driven Large Language Models (LLMs) have greatly enhanced the performance of text-to-SQL systems. Through extensive pre-training on large datasets, these models can efficiently handle zero-shot and few-shot tasks without relying heavily on specialized training data. Their ability to understand context and adapt to different domains makes them highly effective in a wide range of applications.

However, LLM-based approaches are not without limitations. Issues such as hallucinations, lack of interpretability, and high computational costs remain significant challenges. Additionally, performance can vary depending on the prompt design and input representation.

Retrieval-Augmented Approaches

Retrieval-Augmented Generation (RAG) has emerged as a promising direction for improving Text-to-SQL systems. By incorporating external knowledge sources such as database schemas, documentation, and example queries, RAG-based systems enhance the quality and relevance of generated SQL. These approaches help reduce hallucination and improve domain-specific performance. However, the efficiency of these systems largely relies on the accuracy and relevance of the obtained information.

Discussion

The literature highlights a clear progression from rule-based systems to advanced LLM-driven approaches. Early methods focused on syntactic mapping, whereas later models emphasized semantic understanding and schema awareness. Transformer-based architectures have significantly improved relational reasoning, and LLMs currently represent the most advanced solution.

Despite these advancements, challenges such as ambiguity in natural language, scalability, and lack of explainability continue to limit their practical deployment. Hybrid approaches that combine retrieval mechanisms and generative models appear to be a promising direction for future research.

TAXONOMY OF TEXT-TO-SQL APPROACHES

Text-to-SQL systems can be categorized based on their underlying methodologies and architectural designs. This classification provides a structured understanding of how different approaches address the problem of translating natural language into SQL queries.

Classification Based on Methodology

Text-to-SQL systems have evolved through several methodological paradigms:

1. *Rule-based systems*: These systems rely on predefined rules, templates, and grammar structures to map natural language queries to SQL statements. They are easy to interpret and suitable for controlled environments but lack flexibility and scalability.
2. *Machine learning-based systems*: Machine learning approaches use statistical models to learn patterns from data. These systems offer improved adaptability compared to rule-based methods but require extensive feature engineering and knowledge of the domain.
3. *Deep learning-based systems*: Deep learning models, particularly sequence-to-sequence architectures, enable the automatic learning of representations from data. They improve contextual understanding but may struggle with complex schema relationships and multi-table queries.
4. *LLM-based systems*: Large Language Model-based systems leverage transformer architectures and large-scale pretraining to generate SQL queries. They demonstrate superior performance in handling complex queries and diverse schemas; however, they require significant computational resources.

Classification Based on Architecture

From an architectural perspective, Text-to-SQL systems can be grouped as follows.

1. *Encoder-decoder architectures*: These models encode natural language inputs into a latent representation and decode them into SQL queries. They are widely used in early neural and deep-learning approaches.
2. *Transformer-based architecture*: Transformer models use attention mechanisms to capture the relationships between tokens and schema elements. They provide improved performance in handling complex queries and long-range dependencies.
3. *Retrieval-augmented architectures*: These systems integrate retrieval mechanisms with generative models to incorporate external knowledge such as database schemas and query examples. This approach enhances accuracy and reduces hallucination in query generation.

Discussion

The taxonomy highlights the progression of Text-to-SQL systems from rigid, rule-based methods to highly flexible and intelligent LLM-based approaches. Methodological advancements have focused on improving generalization, contextual understanding, and schema alignment.

Architectural innovations, particularly transformer-based and retrieval-augmented models, have significantly enhanced the ability of systems to handle complex cross-domain queries. However, each category presents trade-offs in terms of interpretability, computational cost, and scalability.

COMPARATIVE ANALYSIS

A comparative evaluation of different text-to-SQL approaches is essential to understand their strengths, limitations, and suitability for real-world applications. This section analyzes the major categories of approaches based on key performance and system-level criteria.

Performance Comparison

Table 1 presents a comparison of text-to-SQL approaches in terms of accuracy, complexity, and scalability.

Rule-based systems provide low accuracy owing to their dependence on predefined templates and inability to generalize. Although machine learning approaches improve adaptability, they remain limited by feature engineering requirements. Deep learning models significantly enhance performance through automated representation learning, while LLM-based systems achieve the highest accuracy by leveraging large-scale pretraining and contextual understanding.

Analysis of Key Factors

1. *Accuracy*: LLM-based systems outperform all other approaches owing to their ability to capture semantic relationships and handle diverse query structures. Deep learning models also achieve high accuracy but may struggle with complex schema relationships.
2. *Complexity*: As model sophistication increases, so does computational complexity. LLM-based systems require substantial computational resources for training and inference, making them less suitable for resource-constrained environments.
3. *Scalability*: Scalability improves with the adoption of transformer architecture. LLM-based systems demonstrate strong generalization across domains, whereas rule-based systems are limited to specific use cases.
4. *Schema understanding*: Transformer- and LLM-based approaches show improved capabilities in schema linking compared to earlier methods. However, the accurate mapping of natural language to complex database schemas remains challenging.

Discussion

The comparative analysis indicates a clear progression in the effectiveness of Text-to-SQL systems. While early approaches prioritized simplicity and interpretability, modern systems emphasize accuracy and generalization. LLM-based models represent the current state-of-the-art, offering superior performance in handling complex and cross-domain queries.

However, this performance comes at the cost of increased computational requirements and reduced interpretability of the model. In practical applications, the choice of approach depends on the trade-off between accuracy, efficiency, and deployment requirements.

CHALLENGES IN TEXT-TO-SQL SYSTEMS

Despite significant advancements in text-to-SQL systems, several challenges continue to limit their effectiveness and real-world deployment. These challenges arise from the inherent complexities of natural language, database structures, and model limitations.

Table 1. Comparison of text-to-SQL approaches.

Approach	Accuracy	Complexity	Scalability
Rule-based	Low	Low	Limited
Machine learning	Moderate	Moderate	Moderate
Deep learning	High	High	Good
LLM-based	Very high	Very high	Excellent

Schema Understanding

One of the primary challenges is the accurate mapping of natural language queries to database schemas. Users often refer to attributes and tables in an implicit or ambiguous manner, making it difficult for the models to establish correct relationships. This issue is more pronounced in databases with complex schemas involving multiple tables, foreign keys, and hierarchical structures.

Ambiguity in Natural Language

Natural language queries are inherently ambiguous and can have multiple valid interpretations. For example, a query such as “Show the top customers” may depend on context—whether “top” refers to purchase amount, frequency, or recency of purchase. Resolving such ambiguities requires contextual understanding and, in some cases, additional user interaction.

Handling Complex Queries

Generating SQL queries that involve joins, nested subqueries, aggregations, and group operations remains a difficult task. While modern models have improved in handling such structures, errors in query composition can still occur, especially when multiple operations are combined.

Hallucination in LLMs

Large Language Models may generate syntactically correct but semantically incorrect SQL queries, a phenomenon commonly referred to as hallucination. This can lead to invalid or misleading results, particularly when the model lacks sufficient grounding in the database schema or domain knowledge.

Computational Cost

Advanced models, particularly LLM-based systems, require substantial computational resources for both training and inference. This limits their deployment in real-time or resource-constrained environments and increases operational costs.

Lack of Explainability

Many modern text-to-SQL systems operate as black-box models, providing limited insight into how a query is generated. This lack of transparency reduces user trust and makes debugging difficult, especially in critical applications such as healthcare and finance.

Discussion

The challenges identified highlight the gap between research advancements and the practical deployment of Text-to-SQL systems. Although LLM-based approaches have significantly improved performance, issues such as ambiguity, schema alignment, and computational efficiency remain unresolved. Addressing these challenges will require a combination of improved model architecture, better integration of domain knowledge, and the development of explainable AI techniques.

APPLICATIONS OF TEXT-TO-SQL SYSTEMS

Text-to-SQL systems have gained significant attention because of their ability to simplify database interactions for non-technical users. By enabling natural language querying, these systems are increasingly integrated into real-world applications across various domains.

Business Intelligence and Analytics

Text-to-SQL systems are used in business environments to support data-driven decision-making. They allow users to query large datasets without requiring knowledge of SQL, thereby enabling faster access to insights. These systems are often integrated into dashboards and reporting tools, where users can retrieve information such as sales trends, customer behavior, and performance metrics using natural language queries.

Healthcare Data Management

Healthcare systems generate large volumes of structured data, including patient records, treatment histories, and diagnostic information. Text-to-SQL systems facilitate efficient data retrieval by

allowing healthcare professionals to query databases in natural language. This improves accessibility and can assist in clinical decision-making, research, and patient care management in the future.

E-Commerce and Retail Analytics

In e-commerce platforms, Text-to-SQL systems are used to analyze customer interactions, purchase patterns, and product performance. Business users can generate queries related to sales, inventory, and customer segmentation without relying on technical teams. This enables faster insights and supports personalized marketing strategies.

Financial Data Analysis

Financial institutions rely heavily on structured data for reporting, auditing, and risk analysis. Text-to-SQL systems allow analysts to efficiently retrieve financial information by converting natural language queries into SQL. This reduces the dependency on database specialists and improves productivity in tasks such as transaction analysis and regulatory reporting.

Educational and Research Systems

Text-to-SQL systems are also used in educational platforms to help students and researchers interact with data sets. They provide an intuitive interface for querying academic databases, enabling users to explore data without requiring advanced technical expertise. This is particularly useful for teaching database concepts and conducting exploratory research.

Discussion

The growing adoption of Text-to-SQL systems across multiple domains highlights their practical significance. These systems improve accessibility, reduce reliance on technical expertise, and accelerate data-driven workflows. However, their effectiveness in real-world applications depends on their ability to handle domain-specific queries, ensure accuracy, and remain reliable.

FUTURE DIRECTIONS

Although significant progress has been made in text-to-SQL systems, several areas remain open for further research and improvement. Future work is expected to focus on enhancing the accuracy, efficiency, and adaptability of the model while addressing its existing limitations.

Integration of Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a promising approach for improving the reliability of text-to-SQL systems. By incorporating external knowledge sources such as database schemas, documentation, and query examples, RAG-based systems can generate more accurate and context-aware SQL queries. Future research should focus on optimizing retrieval mechanisms to ensure that relevant and high-quality information is used during query generation.

Domain-Specific Model Adaptation

Current LLM-based systems are typically trained on general-purpose datasets, which may limit their effectiveness in specialized fields. Fine-tuning models on domain-specific data, such as healthcare, finance, and e-commerce datasets, can significantly improve performance. Developing efficient and scalable methods for domain adaptation remains an important research direction.

Hybrid Symbolic-Neural Approaches

Combining symbolic reasoning with neural network models offers a potential solution for improving both accuracy and interpretability. Hybrid approaches can leverage the strengths of rule-based systems, such as logical consistency, and the flexibility of neural models. This integration may lead to more robust and reliable text-to-SQL systems.

Explainability and Interpretability

Improving the transparency of text-to-SQL systems is critical for building user trust. Future systems should provide explanations for the generated SQL queries, enabling users to understand how

their natural language input is interpreted. Research in explainable artificial intelligence can play a key role in addressing this challenge.

Efficiency and Real-Time Deployment

The high computational cost of LLM-based systems remains a significant barrier to their real-world deployment. Future work should focus on model optimization techniques, such as parameter reduction, knowledge distillation, and efficient inference strategies, to enable real-time performance in resource-constrained environments.

Interactive and Conversational Query Systems

Next-generation text-to-SQL systems are expected to move beyond single-query generation toward interactive and conversational interfaces. These systems allow users to refine queries iteratively, clarify ambiguities, and engage in multi-turn interactions with databases. This direction aligns with the broader goal of creating intelligent data assistant systems.

Discussion

The future of Text-to-SQL systems lies in developing solutions that balance accuracy, efficiency, and usability of the system. While LLM-based approaches have significantly advanced the field, integrating retrieval mechanisms, domain knowledge, and explainability features will be essential for practical adoption. Continued research in these areas is expected to enable more robust and user-friendly systems capable of supporting complex real-world applications.

CONCLUSION

This study examined the development of Text-to-SQL systems, highlighting their progression from rule-based and machine learning approaches to advanced deep learning and Large Language Model (LLM)-based techniques. Early systems, while interpretable, were limited in flexibility and scalability. Subsequent machine learning and neural network models improved adaptability and contextual understanding but faced challenges in handling complex queries and schema relationships.

The development of transformer-based and large language models (LLMs) has greatly improved the efficiency and accuracy of text-to-SQL systems. These models demonstrate strong capabilities in understanding natural language semantics, enabling the accurate generation of SQL queries across diverse and complex scenarios. In particular, improvements in schema linking, contextual reasoning, and generalization have contributed to their effectiveness.

Despite these advancements, several challenges remain. Issues such as ambiguity in natural language, difficulty in schema alignment, high computational requirements, and lack of explainability continue to limit the reliability and deployment of these systems in real-world environments. Additionally, the risk of generating incorrect or misleading queries highlights the need for improved validation.

Future research directions, including the integration of retrieval-based methods, domain-specific model adaptation, and hybrid symbolic-neural approaches, offer promising solutions to these challenges. Enhancing efficiency and developing explainable systems will also be critical for broader adoption.

In conclusion, Text-to-SQL systems have made substantial progress and are becoming increasingly capable of bridging the gap between human language and structured data. Continued research and innovation are expected to further improve their robustness, usability, and scalability, enabling more intuitive and intelligent interaction with databases across various domains.

REFERENCES

1. Zhong V, Xiong C, Socher R. Seq2SQL: generating structured queries from natural language using reinforcement learning [Preprint]. arXiv:1709.00103; 2017.

2. Xu X, Liu C, Song D. SQLNet: generating structured queries from natural language without reinforcement learning [Preprint]. arXiv:1711.04436; 2017.
3. Yu T, Zhang R, Yang K, Yasunaga M, Wang D, Li Z, et al. Spider: a large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing; 2018 Oct-Nov; Brussels, Belgium. Stroudsburg (PA): Association for Computational Linguistics; 2018. p. 3911-3921. doi:10.18653/v1/D18-1425.
4. Guo J, Zhan Z, Gao Y, Xiao Y, Lou JG, Liu T, et al. Towards complex text-to-SQL in cross-domain database with intermediate representation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; 2019 Jul; Florence, Italy. Stroudsburg (PA): Association for Computational Linguistics; 2019. p. 4524-4535. doi:10.18653/v1/P19-1444.
5. Wang B, Shin R, Liu X, Polozov O, Richardson M. RAT-SQL: relation-aware schema encoding and linking for text-to-SQL parsers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020 Jul; Online. Stroudsburg (PA): Association for Computational Linguistics; 2020. p. 7567-7578. doi:10.18653/v1/2020.acl-main.677.
6. Scholak T, Schucher N, Bahdanau D. PICARD: parsing incrementally for constrained autoregressive decoding from language models. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; 2021 Nov; Online and Punta Cana, Dominican Republic. Stroudsburg (PA): Association for Computational Linguistics; 2021. p. 9895-9901. doi:10.18653/v1/2021.emnlp-main.779.
7. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020;21(140):1-67.
8. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. In: Advances in Neural Information Processing Systems. 2020;33:1877-1901.
9. Li H, Zhang J, Li C, Chen H. RESDSQL: decoupling schema linking and skeleton parsing for text-to-SQL. Proc AAAI Conf Artif Intell. 2023;37(11):13067-13075. doi:10.1609/aaai.v37i11.26535.
10. Guo C, Tian Z, Tang J, Li S, Wang T. Multi-pattern retrieval-augmented framework for Text-to-SQL with Poincaré-Skeleton retrieval and meta-instruction reasoning. Inf Process Manag. 2025;62(3):103978. doi:10.1016/j.ipm.2024.103978.