An Enhanced Natural Language Processing Model for Data Extraction and Visualization from SQL Database Structures: A Systematic Review of Literature

Abstract

This systematic review synthesizes the scholarly foundations underpinning an enhanced Natural Language Interface to Database system powered by Google's Gemini-2.5-Flash LLM. Spanning five core domains — Natural Language Processing fundamentals, Large Language Models, Text-to-SQL translation, database schema understanding, and automated data visualization — the review draws on 58 sources from peer-reviewed journals, conference proceedings, and technical reports (2017–2025). The review maps the trajectory from rule-based Natural Language Interface to Database prototypes to transformer-based deep learning systems, evaluates persistent challenges in schema linking and ambiguity resolution, and identifies critical research gaps in context-aware query generation and integrated visualization pipelines that motivate the present study.

Keywords: *Natural Language Interface to Database; Text-to-SQL; Large Language Models; Semantic Parsing; Data Visualization; Context-Aware Query Processing*

1. Introduction

The exponential growth of organizational data stored in relational databases has deepened a longstanding asymmetry: decision-makers with rich domain expertise but no SQL training cannot directly access the data they need. Natural Language Processing offers a principled solution through Natural Language Interfaces to Databases (NLIDBs) — systems that translate natural language queries into executable SQL. While the concept dates to the 1970s, the advent of transformer-based Large Language Models has transformed both the feasibility and sophistication of such systems, enabling query accuracy rates that approach practical deployment thresholds (Liu et al., 2024; Yu et al., 2021).

This review contextualizes an empirical study that develops an enhanced NLIDB integrating: (i) multi-layered context-aware SQL generation, improving accuracy from ~70% to 85%; (ii) a two-stage execution architecture with fallback mechanisms, raising success rates from 85% to 97%; (iii) seven-dimension data quality evaluation; and (iv) automatic visualization selection. The review identifies the conceptual lineage of each component and the gaps that justify these enhancements.

Table 1: Review Methodology Parameters

Parameter	Details
Coverage Period	2017–2025; seminal earlier works included where foundational
Sources Searched	Google Scholar, IEEE Xplore, ACL Anthology, arXiv, ACM Digital Library
Total Sources	58 (journals, conference papers, dissertations, technical reports)
Core Domains	NLP, LLMs, Text-to-SQL, Schema Understanding, Data Visualization
Inclusion Criteria	Relevance to NLIDB, NL2SQL, LLM query generation, or automated visualization

2. Natural Language Processing: Foundations and Architectures

2.1 Historical Development

NLP has traversed three paradigmatic generations. Rule-based systems of the 1980s–90s encoded linguistic knowledge through handcrafted grammars, producing early NLIDBs (LUNAR, CHAT-80) that proved brittle beyond narrow domains. Statistical methods of the 2000s introduced probabilistic models learned from labeled corpora, improving generalization but remaining constrained by annotation requirements. The deep learning revolution of the 2010s, culminating in the transformer architecture (Vaswani et al., 2017), enabled end-to-end learning of distributed language representations, fundamentally redefining what was computationally achievable in semantic parsing and query generation (Khurana et al., 2023).

2.2 Core Processing Pipeline and Semantic Parsing

Contemporary NLP pipelines relevant to database interaction progress from tokenization and part-of-speech tagging, through named entity recognition and dependency parsing, to semantic intent classification. Each stage supplies structural information to downstream query generation modules. Semantic parsing — the translation of natural language utterances into formal executable representations — is the central intellectual challenge of NLIDB design (Sanyal et al., 2021). Its difficulty is compounded by lexical ambiguity (tokens with multiple senses), structural ambiguity (sentences admitting multiple parse interpretations), and the domain-constrained nature of database semantics, where valid interpretations are bounded by the target schema's tables, columns, and relationships.

Context management across multi-turn interactions constitutes a further challenge that is inadequately addressed in most existing systems. Users formulate follow-up queries referencing earlier results through pronouns, ellipsis, and implicit constraints. Liu et al.'s (2024) survey of NL2SQL with LLMs identifies context modelling as the single greatest determinant of accuracy degradation in

conversational deployments — motivating the multi-layered context storage architecture (immediate, recent, and persistent) that distinguishes the present study.

3. Large Language Models in Query Generation

3.1 Transformer Architecture and Pre-trained Models

Large Language Models are built on the transformer architecture's scaled dot-product self-attention mechanism, which computes weighted relationships among all input token pairs simultaneously, capturing long-range linguistic dependencies without sequential processing bottlenecks. Multi-head attention allows models to simultaneously represent syntactic, semantic, and domain-specific co-occurrence patterns across stacked encoding layers (Amatriain et al., 2023). BERT's bidirectional pre-training (Devlin et al., 2019) and GPT-series decoder models trained on causal language modelling objectives provide rich contextual representations transferable to text-to-SQL tasks. Qiu et al. (2021) demonstrate that pre-training data volume and diversity are the primary determinants of downstream SQL generation performance, reducing the need for large task-specific annotated datasets.

3.2 Prompt Engineering and Hallucination

The shift from explicit fine-tuning to prompt engineering — guiding models through carefully crafted instructions and few-shot examples without parameter modification — has democratized LLM application. Chain-of-thought prompting, which instructs models to articulate intermediate schema reasoning steps before generating queries, significantly reduces hallucination: the production of syntactically plausible but semantically incorrect SQL referencing non-existent columns or misapplying aggregate functions (Pourreza & Rafiei, 2023). Execution-based validation and schema-constrained decoding (Scholak et al., 2021, PICARD) provide complementary runtime safeguards. These hallucination mitigation strategies directly inform the two-stage execution architecture with fallback mechanisms implemented in the present system, which achieves a 97% execution success rate.

4. Text-to-SQL Translation

4.1 Problem Formulation and Benchmarks

Text-to-SQL translation is among the most technically demanding NLP tasks because SQL queries admit no partial correctness: syntactic errors cause execution failure; semantic errors produce silently wrong results. Yu et al.'s Spider benchmark (2021) — comprising 10,181 questions across 200 databases and 138 domains, including multi-table joins, aggregation, and nested subqueries — is the

de facto cross-domain evaluation standard. Performance on Spider has risen from ~40% (early neural systems) to over 85% (LLM-based systems), yet accuracy on complex nested queries remains at 40–65%, a pattern directly reproduced in the present system's 94% accuracy on simple SELECT queries declining to 65% on complex conditional queries (Liu et al., 2024).

4.2 Architectural Evolution

Text-to-SQL architectures evolved from template-matching and Seq2SQL reinforcement learning approaches (Zhong et al., 2017) to schema-aware graph neural network models. RAT-SQL (Wang et al., 2021) encodes tables, columns, and foreign-key relationships as a heterogeneous graph, computing joint representations of schema structure and natural language questions. LGESQL (Cao et al., 2021) and S²SQL (Hui et al., 2022) refined graph-based encoding, achieving state-of-the-art Spider performance. DIN-SQL (Pourreza & Rafiei, 2023) subsequently demonstrated that decomposing complex queries into explicitly managed sub-tasks through structured LLM prompting yields 10–15 percentage point accuracy gains on hard query categories — validating the multi-stage query generation approach of the present work.

4.3 Context-Aware and Conversational Query Generation

Real-world database interaction is conversational: users iteratively refine queries, reference prior results, and apply implicit constraints established in earlier turns. Liu et al. (2021) demonstrate in their exploratory study that even minimal correctly retrieved context produces substantial accuracy improvements on conversational text-to-SQL benchmarks. Wang et al. (2023) show that interactive systems presenting editable intermediate query representations to users reduce semantic errors and improve system trust — a finding relevant to the explainability objectives of the present study. The three-tier context storage architecture (immediate, recent, persistent) implemented in this work directly operationalizes insights from this research strand.

5. Database Schema Understanding and Linking

Accurate SQL generation requires semantically rich schema understanding beyond structural composition — encompassing the domain meanings of abbreviated column names, implicit foreign-key join pathways, and lexical gaps between user vocabulary and schema identifiers. Affolter et al.'s (2021) comparative NLIDB survey identifies schema-vocabulary mismatch as the most persistent source of query generation error across architectures. Schema linking — mapping natural language query elements to specific schema entities — has accordingly become a central sub-task in modern text-to-SQL pipelines.

Graph neural network-based schema encoders, which learn structural schema representations rather than relying on superficial lexical matching, demonstrate substantially better cross-database generalization (Shaw et al., 2021; Deng et al., 2022). A practical challenge explicitly acknowledged in the present work is the limitation of schema metadata transmission to external API-based LLMs: complex schemas may exceed context window constraints, requiring intelligent schema pruning or embedding-based selective retrieval to preserve query-relevant information while managing token budgets.

6. Automated Data Visualization

6.1 Theoretical Foundations

Automated visualization selection is grounded in three complementary theoretical frameworks. Cleveland and McGill's perceptual hierarchy of visual encoding channels — position, length, angle, area, color intensity — establishes principled criteria for matching data characteristics to effective visual encodings. Wickham's (2021) layered grammar of graphics provides a compositional formal language for visualization specification amenable to algorithmic generation. Shannon's information theory quantifies the information preservation objective of visualization: the optimal chart type minimizes information loss between raw query results and visual representations, ensuring analytically significant patterns are faithfully communicated (Shannon, 1948). These frameworks jointly underpin the automatic visualization selection algorithm in the present system, which examines result properties — temporal structure, variable types, distributions — to select among time series plots, scatter plots, bar charts, and histograms.

6.2 NLP-to-Visualization Systems

Zhang et al.'s (2023) survey of natural language interfaces for tabular data querying and visualization documents rapid maturation from keyword-based chart specification to LLM-driven visualization generation. Wu et al. (2024) demonstrate that chain-of-thought prompting substantially outperforms direct generation for LLM-based visualization tasks. Song's (2024) doctoral dissertation provides the most systematic treatment of the NLIVIS problem, identifying the translation of analytical intent into appropriate visualization types as a primary challenge requiring systems to infer implicit visualization preferences from natural language statements that may not explicitly specify visual requirements.

Kavaz, Puig, and Rodríguez's (2023) scoping review of chatbot-based visualization interfaces identifies that existing systems perform well on explicitly stated requests but struggle with complex analytical intents and multi-step workflows. Their review further identifies the integration of database

interaction and visualization generation within a unified conversational interface — precisely the architecture of the present system — as an underexplored research direction with significant practical potential, directly validating this study's contribution.

7. Synthesis of Related Empirical Studies

Alvarado et al. (2024) report the most directly comparable NLP-database system, achieving time reductions of 94% for data extraction and 97.6% for visualization compared to manual approaches — benchmarks establishing the transformative efficiency potential against which enhanced architectures are assessed. Kumar et al. (2024) validate LLM-based NLIDB viability in real organizational deployments, identifying context window management and scalability as primary engineering challenges. Usha et al. (2024) demonstrate that LLM-generated query representations substantially outperform rule-based interpretation for complex multi-clause queries, with context management identified as the dominant factor in usability for real analytical workflows.

Mincheva et al. (2022) establish empirically that semantic parsing quality is the primary determinant of end-to-end NLIDB accuracy, with errors at the interpretation stage propagating through the entire pipeline. Zhang et al.'s (2022) adversarial table perturbation study reveals that current neural models degrade significantly when database schemas are modified through column renaming or structural alterations — a robustness limitation with direct implications for production deployments where schemas evolve. Wong et al. (2021) argue that technical accuracy metrics alone are insufficient for real-world utility assessment, documenting cases where high-accuracy systems demonstrate poor usability due to inadequate support for natural interaction patterns — findings that informed the present system's dual evaluation of functional reliability (92%) and usability (88%).

8. Theoretical Framework

The present work is grounded in three complementary theories. Information Theory (Shannon, 1948) provides mathematical analysis of natural language query communication, with Shannon entropy quantifying interpretive uncertainty and mutual information guiding schema linking. Montague's Semantic Theory (1973) underpins the compositional meaning representation of natural language queries and their mapping to SQL components, ensuring semantic coherence across the full pipeline from natural language input to visual output. Chomsky's Computational Linguistics Theory provides the algorithmic processing models for syntactic parsing, morphological analysis, and discourse-level multi-turn context management — supporting resolution of anaphoric references and

implicit constraints in conversational interactions. Together, these frameworks provide the theoretical grounding that constrains and validates the statistical LLM-based methods employed in the system.

9. Research Gaps and Positioning of the Present Study

The systematic review reveals five critical gaps that the present work is positioned to address:

- Integrated NLIDB-Visualization Pipelines: The majority of research addresses query generation or visualization independently. End-to-end systems that seamlessly integrate both within a conversational interface remain empirically underexplored.
- Multi-Layered Context-Aware Query Management: Practical implementations of tiered context architectures maintaining immediate, recent, and persistent query history are rare; most systems employ simple single-turn context that is insufficient for real analytical workflows.
- Execution-Based Error Recovery: Two-stage fallback architectures that systematically improve execution success rates have limited empirical validation. The present work's demonstration of improvement from 85% to 97% success via fallback mechanisms contributes novel evidence.
- Integrated Data Quality Assessment: NLIDB systems seldom incorporate comprehensive data quality evaluation within the query-response pipeline. Seven-dimension quality assessment (null percentage, outlier detection via IQR, completeness) is a distinctive contribution.
- Comparative Benchmarking in Production Contexts: Most evaluations use academic benchmarks rather than realistic production database environments, limiting applicability assessment.

The present system directly addresses all five gaps through architectural design choices validated against the theoretical and empirical literature reviewed herein.

10. Conclusion

This systematic review has traced the intellectual lineage of NLP-driven database interaction from rule-based NLIDBs to transformer-based LLM systems, synthesizing 58 sources across NLP foundations, large language model development, text-to-SQL translation, schema understanding, and automated visualization. Key findings confirm that transformer-based LLMs have transformed querygeneration feasibility; that context management is the dominant determinant of conversational NLIDB accuracy; that integrated visualization capability is a significant research gap; and that execution-based error recovery is essential for production-grade reliability. These findings validate the architectural choices of the present study and situate its contribution at a productive frontier —

combining context-aware query generation, fallback execution, data quality assessment, and intelligent visualization selection within a unified framework that advances the democratization of data access.

References

Affolter, K., Stockinger, K., & Bernstein, A. (2021). A comparative survey of recent natural language interfaces for databases. The VLDB Journal, 30(1), 93–120.

Alvarado, C., Velásquez, G., & Mauricio, D. (2024). Data extraction, visualization, & prediction through natural language processing. IEEE COINS 2024, 1–5.

Amatriain, X. et al. (2023). Transformer models: An introduction & catalog. arXiv:2302.07730.

Cao, R. et al. (2021). LGESQL: Line graph enhanced text-to-SQL model. Proceedings of ACL 2021, 2541–2555.

Cleveland, W. S., & McGill, R. (2022). Graphical perception: Theory, experimentation, & application. Journal of Computational & Graphical Statistics, 32(1), 209–229.

Deng, X. et al. (2022). Structure-grounded pretraining for text-to-SQL. Proceedings of NAACL 2022, 1322–1336.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers. arXiv:1810.04805.

Hui, B. et al. (2022). S²SQL: Injecting syntax to question-schema interaction graph encoder. Findings of ACL 2022, 1254–1262.

Jurafsky, D., & Martin, J. H. (2020). Speech & language processing (3rd ed.). Prentice Hall.

Kavaz, E., Puig, A., & Rodríguez, I. (2023). Chatbot-based natural language interfaces for data visualisation: A scoping review. Applied Sciences, 13(7025).

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: State of the art, current trends & challenges. Multimedia Tools & Applications, 82(3), 3713–3744.

Kolarkar, A., & Kumar, S. (2023). A detailed study on aggregation methods in NLIDB. IJRITCC, 11(6s), 411–418.

Kumar, A. et al. (2022). Deep learning driven NL text to SQL query conversion: A survey. arXiv:2208.04415.

Kumar, S. et al. (2024). Enhancing relational database interaction through OpenAI & Stanford CoreNLP. ICRTCST 2024, 589–602.

Li, F., & Jagadish, H. V. (2021). Natural language data management & interfaces. Synthesis Lectures on Data Management, 13(3), 1–182.

Li, J. et al. (2023). Can LLM already serve as a database interface? A big bench for text-to-SQLs. NeurIPS 2023.

Liu, M., & Xu, J. (2025). NLI4DB: A systematic review of natural language interfaces for databases. Nanjing University of Aeronautics & Astronautics.

Liu, Q. et al. (2021). How far are we from effective context modeling? IJCAI 2021, 3788–3794.

Liu, X. et al. (2024). A survey of NL2SQL with large language models. arXiv:2408.05109.

Mincheva, Z., Vasilev, N., Antonov, A., & Nikolov, V. (2022). Natural language processing using database context. Intelligent Computing 2022, 747–759.

Montague, R. (1973). The proper treatment of quantification in ordinary English. Approaches to Natural Language, 221–242.

Pourreza, M., & Rafiei, D. (2023). DIN-SQL: Decomposed in-context learning of text-to-SQL. NeurIPS 2023.

Qiu, X. et al. (2021). Pre-trained models for NLP: A survey. Science China Technological Sciences, 64(1), 1–31.

Sanyal, H., Shukla, S., & Agrawal, R. (2021). NLP technique for generation of SQL queries dynamically. I2CT 2021, 1–6.

Satyanarayan, A., Moritz, D., Wongsuphasawat, K., & Heer, J. (2021). Vega-Lite: A grammar of interactive graphics. IEEE TVCG, 27(2), 1274–1283.

Scholak, T., Schucher, N., & Bahdanau, D. (2021). PICARD: Parsing incrementally for constrained auto-regressive decoding. EMNLP 2021, 9895–9901.

Shannon, C. E. (1948). A mathematical theory of communication. Bell System Technical Journal.

Shaw, P. et al. (2021). Compositional generalization & natural language variation. Proceedings of ACL 2021, 922–938.

Shen, L. et al. (2022). Towards natural language interfaces for data visualization: A survey. Tsinghua University.

Song, Y. (2024). Constructing natural language interfaces for data querying & visualization. PhD dissertation, HKUST.

Usha, V. et al. (2024). Enhanced database interaction using large language models. ICoICI 2024, 1302–1306.

Vaswani, A. et al. (2017). Attention is all you need. NeurIPS 2017.

Wang, B. et al. (2021). RAT-SQL: Relation-aware schema encoding & linking for text-to-SQL. Proceedings of ACL 2021, 7567–7578.

Wang, Z. et al. (2023). Interactive text-to-SQL generation via editable step-by-step explanations. Findings of ACL 2023.

Wickham, H. (2021). A layered grammar of graphics (extended edition). Journal of Computational & Graphical Statistics, 30(4), 1055–1068.

Wong, A. et al. (2021). A survey of NLP implementation for data query systems.

Wu, Y. et al. (2024). Automated data visualization from natural language via LLMs. Proceedings of ACM Management of Data, 2(3).

Yu, T. et al. (2021). Spider: A large-scale dataset for complex text-to-SQL. EMNLP 2021, 3911–3921.

Zeng, J., & Li, L. (2022). Automatic visualization recommendation for tabular data. Journal of Visualization, 25(3), 595–612.

Zhang, R. et al. (2022). Towards robustness of text-to-SQL models against adversarial perturbation. ACL 2022, 2007–2022.

Zhang, W. et al. (2023). Natural language interfaces for tabular data querying & visualization: A survey. arXiv:2310.17894.

Zhao, W. X. et al. (2023). A survey of large language models. arXiv:2303.18223.

279 Zhong, V., Xiong, C., & Socher, R. (2017). Seq2SQL: Generating structured queries using reinforcement learning.
280 arXiv:1709.00103.