

# Enhancing database query interpretation: a comparative analysis of semantic parsing models

Gunjan Keswani, Manoj B. Chandak

Department of Computer Science and Engineering, School of Computer Science and Engineering, Ramdeobaba University, Nagpur, India

## Article Info

### Article history:

Received Aug 18, 2024

Revised Dec 10, 2024

Accepted Jan 19, 2025

### Keywords:

Deployment complexity

NoSQL databases

Query accuracy

Scalability

Semantic parsing

## ABSTRACT

The rapid proliferation of NoSQL databases in various domains necessitates effective parsing models for interpreting NoSQL queries, a fundamental aspect often overlooked in database management research. This paper addresses the critical need for a comprehensive understanding of existing semantic parsing models tailored for NoSQL query interpretation. We identify inherent issues in current models, such as limitations in precision, accuracy, and scalability, alongside challenges in deployment complexity and processing delays. This review is pivotal, shedding light on the intricacies and inefficiencies of existing systems, thereby guiding future advancements in NoSQL database querying. This methodical comparison of these models across key performance metrics-precision, accuracy, recall, delay, deployment complexity, and scalability-reveals significant disparities and areas for improvement. By evaluating these models against both individual and combined parameters, we identify the most effective methods currently available. The impact of this work is far-reaching, providing a foundational framework for developing more robust, efficient, and scalable parsing models. This, in turn, has the potential to revolutionize the way NoSQL databases are queried and managed, offering significant improvements in data retrieval and analysis. Through this paper, we aim to bridge the gap between theoretical model development and practical database management, paving the way for enhanced data processing capabilities in diverse NoSQL database applications.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Gunjan Keswani

Department of Computer Science and Engineering

School of Computer Science and Engineering, Ramdeobaba University

Nagpur, India

Email: keswanigv@rknc.edu

## 1. INTRODUCTION

The advent of NoSQL databases marked a paradigm shift in data storage and retrieval, catering to the growing demand for scalability, flexibility, and speed in handling large volumes of unstructured data. However, the efficient interpretation of NoSQL queries remains a challenging frontier, largely due to the complexity and diversity of NoSQL data models. The criticality of this issue is amplified by the increasing reliance on NoSQL databases across various sectors, from web applications to big data analytics. This necessitates the development of advanced semantic parsing models that can accurately, efficiently, and effectively interpret NoSQL queries.

Existing semantic parsing models, while having made significant strides in recent years, still grapple with various limitations. These include issues related to precision and accuracy, which are paramount in

ensuring that query results are reliable and relevant. Furthermore, the recall of these models, or their ability to retrieve all relevant data points, is another area of concern, particularly in scenarios involving complex queries. Delays in query processing, stemming from inefficiencies in the parsing models, can lead to bottlenecks, adversely affecting the performance of real-time applications. Additionally, the complexity of deploying these models and their scalability in handling growing data volumes pose significant challenges.

The impact of this work extends beyond the theoretical realm, offering practical implications for database management. By pinpointing the most effective parsing models, we contribute to enhancing the efficiency and effectiveness of NoSQL databases. Overall, this paper aims to bridge the gap between current model capabilities and the evolving needs of NoSQL database management. We provide a roadmap for future research and development, with the ultimate goal of advancing the field of database query interpretation and management.

The motivation behind this research stems from the rapidly evolving landscape of database technology, where NoSQL systems have emerged as a cornerstone for managing large-scale, unstructured, and semi-structured data. Despite their growing popularity, a significant gap exists in the effective interpretation of NoSQL queries. This gap is primarily due to the diversity of NoSQL data models and the lack of specialized semantic parsing models that can accommodate their unique characteristics. The need for high precision, accuracy, and speed in querying these databases is more pressing than ever, given their extensive use in critical applications ranging from e-commerce platforms to real-time analytics in IoT devices.

## 2. LITERATURE REVIEW

An extensive summary of various models can be observed from Table 1 in Appendix. The Table 1 summarizes a diverse range of research works in the fields of databases and natural language processing [1]-[50].

## 3. METHOD

This paper uses a systematic approach toward the comparative analysis of semantic parsing models applied to the interpretation of NoSQL database queries. It evaluates the existing models against strategic performance metrics such as precision, accuracy, delay, deployment complexity, and scalability. A wide literature review was conducted to identify the state-of-art models in the area of text-to-SQL, NLIDB, and schema design for NoSQL databases. The models were chosen for how relevant they are in handling natural language queries and whether their unique features regarding NoSQL databases can be managed. For conducting a strong comparison, the study had a designed metric-based evaluation framework. Each of these models was tested based on performance parameters which came under five categories, namely: very low (VL, 0–10), low (L, 10–20), medium (M, 20–40), high (H, 40–80), and very high (VH, 80–100). Such a rating scheme allowed the results from different types of models to be normalized and combined with equal representation for comparison. Data was accumulated from experiments that were carried out in the form of previously conducted studies, which included public datasets like Spider, WikiSQL, and CoSQL. These datasets were chosen as they are widely utilized in the literature and pertain to cross-domain natural language-to-SQL translation tasks. The evaluation process took place over three phases. In the first phase, models were evaluated individually, regarding how well they performed within the domain for which they were originally envisioned; this meant collecting accuracy, precision, and recall scores directly from published papers. Cross-domain testing: This involved the performance of the models on datasets outside the domain of the main application to prove scalability and adaptability. This comprised the third phase, which included an analysis of processing delay, deployment complexity, as well as scalability based on the models' architecture and computational requirements. A qualitative assessment throughout the analysis addressed deployment constraints and real-time usability in large-scale NoSQL database environments.

## 4. RESULTS AND DISCUSSION

### 4.1. Introduction

The authors researched semantic parsing models with the intent of filling in the missing gap found in previous studies, which did not well address the issues NoSQL databases present. Actually, work previously done on structured databases and text-to-SQL systems considered no complexity that comes from the less structured and flexible schemas NoSQL introduces. This clearly depicts that there were huge gaps between the current models and the desired precision and scalability. The applications of these models are mostly very low within the NoSQL environment.

#### 4.2. Summarizing key findings

Based on the empirical analysis, some major key findings are that schema inference models, like the ones in [1], have very high precision, at times going up to 100%, making them very suitable for schema-structure interpretation in NoSQL databases with a high degree of correctness and effectiveness. Other examples are the models, such as cross-domain text-to-SQL [7] and elastic data conversion framework [22], which exhibited excellent scalability when performing well in a wide range of domains with different kinds of datasets in the evaluation, thus supporting their applicability in large-scale NoSQL systems. Many models, RDF querying [2] and ontology-based knowledge bases [3], exhibited considerable limitations concerning scalability and complexity of deployment, which indeed made them less suitable for real-time query processing in expansive NoSQL infrastructures and scenarios.

#### 4.3. Interpreting results

Such results compared to prior work found that transformer-based models like the unified framework with self-attention [31] and transformer-based seq-to-seq for text-to-SQL [45] are in a position to outperform classical parsing methods. Not only have these models allowed transformer-based models to better handle complex natural language queries through self-attention mechanisms but also ensured significant high accuracy levels of 80-100 in cross-domain tasks in both accuracy and scalability. However, the earlier versions such as SQLNet [38] and Seq2SQL [39] suffered from scalability and adaptability problems, especially when implemented within a NoSQL scenario where the rigidity of the data structure is not so rigid. These conclusions are in line with other recent research that reveals the first natural language-to-SQL methods cannot compete with messy data examples.

#### 4.4. Addressing limitations

Although promising, a few limitations were realized. The first weakness was the inherent bias of the test datasets. Although test datasets like Spider and WikiSQL are very much welcomed, they do have a considerable inclination towards structured queries and relational databases, which could not be able to simulate the complexity NoSQL environments. This could make limited applicability in generalizing findings from the study. In fact, the computational overhead introduced by transformer-based models is manageable in structured environments, but it may lead to delays in real-time querying of NoSQL databases, especially as the scale of the data increases in processing.

#### 4.5. Implications for future research

Future research needs to cross these limitations and set up more-specific models that can best cater to the demands of NoSQL databases, especially those characterized by highly flexible or dynamic schemas. New benchmark datasets that reflect the richness of NoSQL data models, such as document stores and key-value databases, would make it a more challenging undertaking in regard to the evaluation of these models. Support for more effective cross-domain functionality of models without demanding extensive retraining may make them scale better and applied in real-time.

#### 4.6. Conclusion

In conclusion, this study has evidence that while great strides have been made in structured databases, the NoSQL environment is not one in which these semantic parsing models have achieved consistent results. High precision and scalability shown by schema inference and transformer-based models indicate that these are the approaches most likely to lead to advances in NoSQL query interpretation. However, proper optimization of such models for the specific characteristics of the NoSQL systems, in that with unstructured, dynamic, and heterogeneous samples of data, is very much needed in the future developments to fully unleash their potential.

### 5. CONCLUSION

This paper provides an overview of complete analyses for a wide range of semantic parsing models while interpreting NoSQL database queries and identifies both strengths and weaknesses of the approaches in light of key performance metrics such as precision, accuracy, recall, delay, cost, and scalability.

The results indicate that while schema inference and transformer-based approaches seem to achieve high precision and scalability, existing models suffer from adapting the unstructured nature of the NoSQL database, especially the relational one. Though natural language interfaces and text-to-SQL models have greatly improved query handling in structured environments, they are inherently missing flexible and dynamic schemas that are inherently characteristic of NoSQL systems. As a result, even though the design and deployment of customized NoSQL parsing models is highly improved, there is still much improvement required in the development of the same.

**FUNDING INFORMATION**

No funding involved.

**AUTHOR CONTRIBUTIONS STATEMENT**

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Gunjan Keswani	✓	✓			✓	✓			✓	✓				
Manoj B. Chandak	✓					✓				✓		✓		

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

**CONFLICT OF INTEREST STATEMENT**

No conflict of interest.

**DATA AVAILABILITY**

Data availability is not applicable to this paper as no new data were created or analyzed in this study. All data supporting the findings of this study are available in the published literature, which has been properly cited in the references.

**REFERENCES**

- [1] P. Koupil, S. Hricko, and I. Holubová, "A universal approach for multi-model schema inference," *Journal of Big Data*, vol. 9, no. 1, p. 97, Aug. 2022, doi: 10.1186/s40537-022-00645-9.
- [2] W. Ali, M. Saleem, B. Yao, A. Hogan, and A. C. N. Ngomo, "A survey of RDF stores & SPARQL engines for querying knowledge graphs," *VLDB Journal*, vol. 31, no. 3, pp. 1–26, May 2022, doi: 10.1007/s00778-021-00711-3.
- [3] C. Lei, A. Quamar, V. Efthymiou, F. Özcan, and R. Alotaibi, "HERMES: data placement and schema optimization for enterprise knowledge bases," *VLDB Journal*, vol. 32, no. 3, pp. 549–574, May 2023, doi: 10.1007/s00778-022-00756-y.
- [4] L. Dou *et al.*, "UniSAR: a unified structure-aware autoregressive language model for text-to-SQL semantic parsing," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 12, pp. 4361–4376, Dec. 2023, doi: 10.1007/s13042-023-01898-3.
- [5] Y. Foufoulas, E. Zacharia, H. Dimitropoulos, N. Manola, and Y. Ioannidis, "DETEXA: declarative extensible text exploration and analysis through SQL," *International Journal on Digital Libraries*, vol. 25, no. 3, pp. 457–469, Sep. 2023, doi: 10.1007/s00799-023-00358-1.
- [6] C. Wei, S. Huang, and R. Li, "Enhance text-to-SQL model performance with information sharing and reweight loss," *Multimedia Tools and Applications*, vol. 81, no. 11, pp. 15205–15217, May 2022, doi: 10.1007/s11042-022-12573-0.
- [7] B. B. Naik, T. J. V. R. Reddy, K. R. V. karthik, and P. Kuila, "An SQL query generator for cross-domain human language based questions based on NLP model," *Multimedia Tools and Applications*, vol. 83, no. 4, pp. 11861–11884, 2024, doi: 10.1007/s11042-023-15731-0.
- [8] S. Swamidori, T. S. Murthy, and K. V. Sriharsha, "Translating natural language questions to SQL queries (nested queries)," *Multimedia Tools and Applications*, vol. 83, no. 15, pp. 45391–45405, Oct. 2024, doi: 10.1007/s11042-023-16987-2.
- [9] B. K. Saha, P. Gordon, and T. Gillbrand, "NLINQ: A natural language interface for querying network performance," *Applied Intelligence*, vol. 53, no. 23, pp. 28848–28864, Dec. 2023, doi: 10.1007/s10489-023-05043-z.
- [10] M. A. Jose and F. G. Cozman, "A multilingual translator to SQL with database schema pruning to improve self-attention," *International Journal of Information Technology (Singapore)*, vol. 15, no. 6, pp. 3015–3023, Aug. 2023, doi: 10.1007/s41870-023-01342-3.
- [11] A. Solanki and A. Kumar, "A system to transform natural language queries into SQL queries," *International Journal of Information Technology (Singapore)*, vol. 14, no. 1, pp. 437–446, Feb. 2022, doi: 10.1007/s41870-018-0095-2.
- [12] M. Spasić and M. V. Janičić, "Verification supported refactoring of embedded sql," *Software Quality Journal*, vol. 29, no. 3, pp. 629–665, Sep. 2021, doi: 10.1007/s11219-020-09517-y.
- [13] S. J. Qiao *et al.*, "Cardinality estimator: processing SQL with a vertical scanning convolutional neural network," *Journal of Computer Science and Technology*, vol. 36, no. 4, pp. 762–777, Jul. 2021, doi: 10.1007/s11390-021-1351-7.
- [14] A. P. Marathe, "Towards intelligent database systems using clusters of SQL transactions," *Knowledge and Information Systems*, vol. 65, no. 7, pp. 2863–2894, Jul. 2023, doi: 10.1007/s10115-023-01850-5.
- [15] R. Ma, X. Han, L. Yan, N. Khan, and Z. Ma, "Modeling and querying temporal RDF knowledge graphs with relational databases," *Journal of Intelligent Information Systems*, vol. 61, no. 2, pp. 569–609, Oct. 2023, doi: 10.1007/s10844-023-00780-6.
- [16] B. Namdeo and U. Suman, "Schema design advisor model for RDBMS to NoSQL database migration," *International Journal of Information Technology (Singapore)*, vol. 13, no. 1, pp. 277–286, Feb. 2021, doi: 10.1007/s41870-020-00515-8.
- [17] L. Liu, "Design of NoSQL database in oral English teaching based on 5G network and AI recognition," *Soft Computing*, vol. 27, no. 14, pp. 10337–10345, Jul. 2023, doi: 10.1007/s00500-023-08306-6.

- [18] R. Jemmali, F. Abdelhedi, and G. Zurfluh, "DLToDW: transferring relational and NoSQL databases from a data lake," *SN Computer Science*, vol. 3, no. 5, p. 381, Jul. 2022, doi: 10.1007/s42979-022-01287-7.
- [19] A. Maté, J. Peral, J. Trujillo, C. Blanco, D. García-Saiz, and E. Fernández-Medina, "Improving security in NoSQL document databases through model-driven modernization," *Knowledge and Information Systems*, vol. 63, no. 8, pp. 2209–2230, Aug. 2021, doi: 10.1007/s10115-021-01589-x.
- [20] S. El-Mahgary, E. Soisalon-Soininen, P. Orponen, P. Rönholm, and H. Hyyppä, "OVI-3: A NoSQL visual query system supporting efficient anti-joins," *Journal of Intelligent Information Systems*, vol. 60, no. 3, pp. 777–801, Jun. 2023, doi: 10.1007/s10844-022-00742-4.
- [21] A. Hillenbrand, U. Störl, S. Nabiyev, and M. Klettke, "Self-adapting data migration in the context of schema evolution in NoSQL databases," *Distributed and Parallel Databases*, vol. 40, no. 1, pp. 5–25, Mar. 2022, doi: 10.1007/s10619-021-07334-1.
- [22] T. K. Dang, T. M. Huy, L. H. Dang, and N. Le Hoang, "An elastic data conversion framework: a case study for MySQL and MongoDB," *SN Computer Science*, vol. 2, no. 4, p. 325, Jul. 2021, doi: 10.1007/s42979-021-00716-3.
- [23] C. Forresi, E. Gallinucci, M. Golfarelli, and H. Ben Hamadou, "A dataspace-based framework for OLAP analyses in a high-variety multistore," *VLDB Journal*, vol. 30, no. 6, pp. 1017–1040, Nov. 2021, doi: 10.1007/s00778-021-00682-5.
- [24] A. Rani, N. Goyal, and S. K. Gadia, "Big social data provenance framework for zero-information loss key-value pair (KVP) database," *International Journal of Data Science and Analytics*, vol. 14, no. 1, pp. 65–87, Jun. 2022, doi: 10.1007/s41060-021-00287-9.
- [25] R. Cao *et al.*, "A heterogeneous graph to abstract syntax tree framework for text-to-SQL," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 13796–13813, 2023, doi: 10.1109/TPAMI.2023.3298895.
- [26] R. Z. Wang, Z. H. Ling, J. B. Zhou, and Y. Hu, "A multiple-integration encoder for multi-turn text-to-SQL semantic parsing," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 29, pp. 1503–1513, 2021, doi: 10.1109/TASLP.2021.3070726.
- [27] M. Paganelli, P. Sottovia, K. Park, M. Interlandi, and F. Guerra, "Pushing ML Predictions Into DBMSs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 10295–10308, Oct. 2023, doi: 10.1109/TKDE.2023.3269592.
- [28] T. Bai, Y. Ge, S. Guo, Z. Zhang, and L. Gong, "Enhanced Natural Language Interface for Web-Based Information Retrieval," *IEEE Access*, vol. 9, pp. 4233–4241, 2021, doi: 10.1109/ACCESS.2020.3048164.
- [29] Z. Brahmia, F. Grandi, and R. Bouaziz, "rJOWL: a systematic approach to build and evolve a temporal OWL 2 ontology based on temporal JSON big data," *Big Data Mining and Analytics*, vol. 5, no. 4, pp. 271–281, Dec. 2022, doi: 10.26599/BDMA.2021.9020019.
- [30] V. Wudaru, N. Koditala, A. Reddy, and R. Mamidi, "Question answering on structured data using NLIDB approach," in *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, Mar. 2019, pp. 1–4, doi: 10.1109/ICACCS.2019.8728487.
- [31] B. Wang, R. Shin, X. Liu, O. Polozov, and M. Richardson, "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, pp. 7567–7578, doi: 10.18653/v1/2020.acl-main.677.
- [32] T. Yu, "Learning to map natural language to executable programs over learning to map natural language to executable programs over databases databases," 2021. [https://elischolar.library.yale.edu/gsas\\_dissertations](https://elischolar.library.yale.edu/gsas_dissertations) (accessed May 04, 2024).
- [33] U. Brunner and K. Stockinger, "ValueNet: a natural language-to-SQL system that learns from database information," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, Apr. 2021, pp. 2177–2182, doi: 10.1109/ICDE51399.2021.00220.
- [34] Y. Gan *et al.*, "Natural SQL: making SQL easier to infer from natural language specifications," *Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021*, 2021. .
- [35] C. J. Baik, "Maximizing user domain expertise to clarify oblique specifications of relational queries," 2020.
- [36] Y. Gan, X. Chen, and M. Purver, "Exploring underexplored limitations of cross-domain text-to-SQL generalization," *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 8926–8931, 2021, doi: 10.18653/v1/2021.emnlp-main.702.
- [37] P. Ni, R. Okhrati, S. Guan, and V. Chang, "Knowledge graph and deep learning-based text-to-GraphQL model for intelligent medical consultation chatbot," *Information Systems Frontiers*, vol. 26, no. 1, pp. 137–156, Feb. 2024, doi: 10.1007/s10796-022-10295-0.
- [38] X. Xu, C. Liu, and D. Song, "SQLNet: generating structured queries from natural language without reinforcement learning," 2017, doi: 10.48550/arxiv.1711.04436.
- [39] E. Ersoy and H. Sözer, "Using artificial neural networks to provide guidance in extending PL/SQL programs," *Software Quality Journal*, vol. 30, no. 4, pp. 885–916, Dec. 2022, doi: 10.1007/s11219-022-09586-1.
- [40] M. Johnson *et al.*, "Google's multilingual neural machine translation system: enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, Dec. 2017, doi: 10.1162/tacl\_a\_00065.
- [41] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arxiv*, 2014, [Online]. Available: <http://arxiv.org/abs/1409.2329>.
- [42] L. Dong and M. Lapata, "Language to logical form with neural attention," *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, vol. 1, pp. 33–43, 2016, doi: 10.18653/v1/p16-1004.
- [43] R. Liu, T. Wang, Y. Yang, and B. Yu, "Database development based on deep learning and cloud computing," *Mobile Information Systems*, vol. 2022, pp. 1–10, Apr. 2022, doi: 10.1155/2022/6208678.
- [44] W. Zhang *et al.*, "Deep neural network-based SQL injection detection method," *Security and Communication Networks*, vol. 2022, pp. 1–9, Mar. 2022, doi: 10.1155/2022/4836289.
- [45] K. Xu, Y. Wang, Y. Wang, Z. Wang, Z. Wen, and Y. Dong, "SeaD: end-to-end text-to-SQL generation with schema-aware denoising," *Findings of the Association for Computational Linguistics: NAACL 2022 - Findings*, pp. 1845–1853, 2022, doi: 10.18653/v1/2022.findings-naacl.141.
- [46] T. Wolfson, D. Deutch, and J. Berant, "Weakly supervised text-to-SQL parsing through question decomposition," in *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022, pp. 2528–2542, doi: 10.18653/v1/2022.findings-naacl.193.
- [47] N. Deng, Y. Chen, and Y. Zhang, "Recent advances in text-to-SQL: a survey of what we have and what we expect," *Proceedings - International Conference on Computational Linguistics, COLING*, vol. 29, no. 1, pp. 2166–2187, 2022.
- [48] T. J. Revanth, K. V. Sai, R. Ramya, R. Chava, V. Sushma, and B. S. Ramya, "NL2SQL: natural language to SQL query translator," *Lecture Notes in Electrical Engineering*, vol. 790, pp. 267–278, 2022, doi: 10.1007/978-981-16-1342-5\_21.
- [49] M. S. Geetha, R. Yashwanthika, M. S. Sri, and M. Sudiksa, "GenSQL—NLP-based SQL generation," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 93, pp. 279–288, 2022, doi: 10.1007/978-981-16-6605-6\_20.

- [50] E. Ersoy and H. Sözer, "Using artificial neural networks to provide guidance in extending PL/SQL programs," *Software Quality Journal*, vol. 30, no. 4, pp. 885–916, Dec. 2022, doi: 10.1007/s11219-022-09586-1.

## APPENDIX

Table 1. Comparative analysis of existing models

Method used	Details	Advantages	Research gap
Schema Inference [1]	This work addresses the inference of a common schema for multi-model data, including local integrity constraints and inter-model references. It handles overlapping models, data redundancy, and large data efficiently.	<ul style="list-style-type: none"> <li>- Inference of a common schema for multi-model data.</li> <li>- Handling of overlapping models and data redundancy.</li> <li>- Efficient processing of significant data amounts.</li> </ul>	- Specific to multi-model data, may not be suitable for single-model scenarios.
RDF Querying [2]	This survey reviews techniques and systems for querying RDF knowledge graphs, with a focus on local (single-machine) settings. It also discusses contemporary research challenges in SPARQL query engines.	<ul style="list-style-type: none"> <li>- Comprehensive review of RDF querying techniques.</li> <li>- Emphasis on local (single-machine) settings.</li> <li>- Discussion of research challenges.</li> </ul>	- Limited to local querying, not distributed settings.
Ontology-Based KBs [3]	HERMES is introduced for querying domain-specific knowledge bases stored in multiple backends with different query languages. Challenges of data placement and schema optimization are addressed.	<ul style="list-style-type: none"> <li>- Querying domain-specific KBs with multiple backends and query languages.</li> <li>- Data placement optimization.</li> <li>- Schema optimization, including property graph schemas.</li> </ul>	- Complex to implement for large-scale KBs.
Text-to-SQL Parsing [4]	UNISAR is presented as a structure-aware autoregressive language model for text-to-SQL parsing, achieving high performance under various settings.	<ul style="list-style-type: none"> <li>- Use of an off-the-shelf language model architecture.</li> <li>- Incorporation of structure-aware extensions.</li> <li>- High performance in different text-to-SQL scenarios.</li> </ul>	- May not outperform specialized models in specific settings.
Metadata Enrichment [5]	A text analysis framework implemented in extended SQL is introduced for metadata enrichment in digital libraries. The framework offers scalability and ease of use.	<ul style="list-style-type: none"> <li>- Scalable framework for text mining in databases.</li> <li>- Declarative nature of SQL for easy workflow creation.</li> <li>- Significant speedup compared to other approaches.</li> </ul>	- Limited to metadata enrichment, may not cover all text mining tasks.
Text-to-SQL Mapping [6]	This work introduces a method for text-to-SQL mapping using multi-task learning and sharing decoders for different subtasks, reducing model complexity and improving learning.	<ul style="list-style-type: none"> <li>- Reduction of model complexity.</li> <li>- Better learning of dependencies between subtasks.</li> <li>- Accuracy improvement on the WikiSQL dataset.</li> </ul>	- May not cover all possible subtask dependencies.
Cross-Domain Text-to-SQL [7]	An approach for improving text-to-SQL conversion for cross-domain datasets is presented, emphasizing linguistic dependencies between queries. Evaluation is conducted on Sparc, Spider, and CoSQL datasets.	<ul style="list-style-type: none"> <li>- Utilization of linguistic dependencies between queries.</li> <li>- Evaluation on various cross-domain datasets.</li> <li>- Comparison with existing algorithms.</li> </ul>	- May not perform well on highly domain-specific datasets.
Nested SQL Queries [8]	This study proposes an improved IRNet framework for translating natural language queries to nested SQL queries, addressing the challenge of complex queries. Data oversampling and a novel loss function are introduced.	<ul style="list-style-type: none"> <li>- Data oversampling for representation improvement.</li> <li>- Novel loss function considering SQL complexity.</li> <li>- 5% improvement on hard queries in Spider dataset.</li> </ul>	- Specific to nested SQL queries, may not cover other query types.
Natural Language-Based Querying [9]	Investigates natural language-based querying of network performance databases for Wireless Mesh Networks (WMNs), including semantic column names, domain-specific corrections, and real-time querying.	<ul style="list-style-type: none"> <li>- Translation of natural language to SQL with high accuracy.</li> <li>- Semantic column names generation.</li> <li>- Suitable for real-time querying in WMNs.</li> </ul>	- Limited to WMN context, may not generalize to other domains.
Text-to-SQL with Transformers [10]	This work presents techniques to improve text-to-SQL results with transformers, including handling long-text sequences and multilingual fine-tuning. It enhances accuracy in NL2SQL tasks.	<ul style="list-style-type: none"> <li>- Handling of long-text sequences by transformers.</li> <li>- Multilingual fine-tuning for improved accuracy.</li> <li>- Increase in exact set match accuracy.</li> </ul>	- Improvement techniques may not apply to all text-to-SQL models.

Table 1. Comparative analysis of existing models (*Continued*)

Method used	Details	Advantages	Research gap
Natural Language to SQL Conversion [11]	A three-tier system using pattern matching and semantic matching techniques to transform natural language into SQL queries.	<ul style="list-style-type: none"> <li>- Enables non-expert users to interact with databases using natural language.</li> <li>- Better recall value, accuracy, and precision compared to existing systems.</li> </ul>	<ul style="list-style-type: none"> <li>- May require predefined data dictionary and complex transformation steps.</li> </ul>
Automated Code Equivalence Verification [12]	Focuses on verifying code equivalence in embedded SQL programming, including simultaneous changes in SQL and host language code. Uses first-order logic modeling and SMT solvers for verification.	<ul style="list-style-type: none"> <li>- Automated verification of code equivalence.</li> <li>- Addresses simultaneous changes in SQL and host language code.</li> <li>- Publicly available framework (SQLAV).</li> </ul>	<ul style="list-style-type: none"> <li>- Requires knowledge of first-order logic and SMT solvers.</li> </ul>
Learning-Based Cardinality Estimation [13]	Proposes a vertical scanning convolutional neural network (VSCNN) to estimate cardinalities of complex SQL queries using deep neural networks. Includes semantic information and negative sampling.	<ul style="list-style-type: none"> <li>- Improved cardinality estimation for complex join operations.</li> <li>- Utilizes deep neural networks and semantic information.</li> <li>- Reduces q-error in estimation.</li> </ul>	<ul style="list-style-type: none"> <li>- May not outperform traditional methods in all scenarios.</li> </ul>
Transaction Classification and Clustering [14]	Introduces transaction classification and clustering in database systems for monitoring and troubleshooting. Utilizes DBSCAN algorithm and server-side feature extraction.	<ul style="list-style-type: none"> <li>- Automates transaction clustering for troubleshooting.</li> <li>- Identifies root causes of performance problems.</li> <li>- Cluster count remains stable regardless of system load.</li> </ul>	<ul style="list-style-type: none"> <li>- Requires DBMS modification for implementation.</li> </ul>
Temporal RDF Model and Query Language [15]	Presents tRDF, a temporal RDF model, and a temporal query language for managing temporal RDF data in relational databases. Transformation from tRDF query language to SQL.	<ul style="list-style-type: none"> <li>- Addresses temporal semantics in RDF data.</li> <li>- Utilizes relational databases for storage.</li> <li>- Provides temporal query language.</li> </ul>	<ul style="list-style-type: none"> <li>- May require specific SQL support for temporal data.</li> </ul>
Schema Design Advisor for NoSQL [16]	Proposes a schema design advisor model for NoSQL databases, using existing SQL queries as input to recommend efficient schemas. Includes a cost model.	<ul style="list-style-type: none"> <li>- Automates schema design recommendations.</li> <li>- Considers cost-effectiveness of schemas.</li> <li>- Applicable to Old RDBMS to new NoSQL transitions.</li> </ul>	<ul style="list-style-type: none"> <li>- Limited to MongoDB in the prototype.</li> </ul>
Database Technology for Network Applications [17]	Proposes a business architecture for storing data on the network and transmitting spoken language resources for education. Focuses on AI and intelligent technology in teaching.	<ul style="list-style-type: none"> <li>- Utilizes network storage for education resources.</li> <li>- Applies AI technology in teaching.</li> <li>- Supports spoken language resources.</li> </ul>	<ul style="list-style-type: none"> <li>- Specific to education and spoken language applications.</li> </ul>
Modernizing Security in NoSQL Databases [18]	Introduces an approach for modernizing security in NoSQL databases, focusing on access control. Utilizes domain ontology and automated security issue detection.	<ul style="list-style-type: none"> <li>- Incorporates security mechanisms into existing NoSQL solutions.</li> <li>- Automated analysis of security issues.</li> <li>- Reduces modernization effort and cost.</li> </ul>	<ul style="list-style-type: none"> <li>- Requires domain ontology and customization for different NoSQL technologies.</li> </ul>
Security in NoSQL Databases [19]	Addresses security issues in NoSQL databases, particularly access control, using a domain ontology-based approach. Proposes automated solutions for identified security issues.	<ul style="list-style-type: none"> <li>- Detects and addresses security issues in existing NoSQL solutions.</li> <li>- Uses domain ontology for context-aware analysis.</li> <li>- Reduces the risk of data breaches.</li> </ul>	<ul style="list-style-type: none"> <li>- May require adjustments for different NoSQL technologies.</li> </ul>
NoSQL Visual Query System [20]	Presents OVI-3, a NoSQL visual query system based on incremental querying and directory-based indexing for complex joins. Demonstrates improved speed for certain queries.	<ul style="list-style-type: none"> <li>- Enables fast ad-hoc queries with complex joins.</li> <li>- Utilizes directory-based indexing for optimization.</li> <li>- Demonstrates speed improvement over SQL queries for specific scenarios.</li> </ul>	<ul style="list-style-type: none"> <li>- Limited to specific types of complex joins.</li> </ul>
Self-Adapting Data Migration [21]	Proposes a methodology for self-adapting data migration that automatically adjusts migration strategies based on migration scenario and service-level agreements. Evaluates and compares migration strategies using metrics.	<ul style="list-style-type: none"> <li>- Self-adapting migration for agile development.</li> <li>- Considers migration costs, latency, precision, and recall.</li> <li>- Matches migration strategy to specific scenarios.</li> </ul>	<ul style="list-style-type: none"> <li>- Requires appropriate metrics and understanding of service-level agreements.</li> </ul>

Table 1. Comparative analysis of existing models (*Continued*)

Method used	Details	Advantages	Research gap
Elastic Data Conversion Framework [22]	Introduces an elastic data conversion framework for data integration systems, aiming to link and merge different data resources into a unified data store. Evaluates the model using MySQL and MongoDB.	<ul style="list-style-type: none"> <li>- Supports data integration across various formats and types.</li> <li>- Addresses data conversion challenges.</li> <li>- Includes experimental evaluation.</li> </ul>	- Specific to data integration and data conversion.
Data Analysis in High-Variety Multistore [23]	Proposes an approach for data analysis within a high-variety multistore with heterogeneous schemas and overlapping records. Supports multiple data models and schema integration through a dataspace layer.	<ul style="list-style-type: none"> <li>- Enables OLAP analyses in heterogeneous schemas.</li> <li>- Handles schema and data model heterogeneity.</li> <li>- Supports various data models.</li> </ul>	- Specific to data analysis and heterogeneous schemas.
Big Social Data Provenance Framework [24]	Presents a Big Social Data Provenance (BSDP) Framework for key-value pair (KVP) databases using the concept of Zero-Information Loss Database (ZILD). Captures, stores, and queries provenance information for different query sets.	<ul style="list-style-type: none"> <li>- Captures provenance information for social data.</li> <li>- Supports various query types, including select, aggregate, and data update queries.</li> <li>- Provides a query-driven approach.</li> </ul>	- Specific to social data and KVP databases.
Heterogeneous Graph to AST Framework [25]	Introduces HG2AST, a framework for converting heterogeneous graph representations to abstract syntax trees (AST) for Text-to-SQL tasks. Incorporates structure knowledge and adaptive node expansion.	<ul style="list-style-type: none"> <li>- Addresses multi-turn text-to-SQL generation.</li> <li>- Utilizes heterogeneous graph encoding.</li> <li>- Improves structure knowledge incorporation.</li> </ul>	- Specific to text-to-SQL tasks and AST construction.
Information Integration Encoder for Text-to-SQL [26]	Proposes an encoder for multi-turn text-to-SQL generation, addressing challenges in multi-turn interaction and cross-domain evaluation. Uses a multiple-integration encoder with three modules for information integration.	<ul style="list-style-type: none"> <li>- Improves accuracy of multi-turn text-to-SQL generation.</li> <li>- Handles multi-turn interaction and cross-domain scenarios.</li> <li>- Utilizes lightweight multi-head attention.</li> </ul>	- Specific to text-to-SQL generation and multi-turn interaction.
ML Pipeline Translation to SQL Queries [27]	Translates trained ML pipelines containing featurizers and models into SQL queries for prediction serving within a DBMS. Compares in-DBMS performance with popular ML frameworks.	<ul style="list-style-type: none"> <li>- Enables ML inference within a DBMS.</li> <li>- Supports efficient prediction serving.</li> <li>- Reduces data movement and optimizes performance.</li> </ul>	- Specific to ML prediction serving and DBMS integration.
Neural Model for NL Query of Databases [28]	Introduces an improved neural model based on IRNet for natural language queries of databases, using Gated Graph Neural Network (GGNN) and schema information.	<ul style="list-style-type: none"> <li>- Enables NL query of databases using neural models.</li> <li>- Incorporates schema information and entity linking.</li> <li>- Provides a graph-based approach.</li> </ul>	- Specific to NL query of databases and neural models.
Temporal OWL 2 Ontology from JSON Data [29]	Proposes an approach ( $\tau$ JOWL) to automatically build a temporal OWL 2 ontology of data from temporal JSON-based big data. Manages incremental maintenance for evolving data.	<ul style="list-style-type: none"> <li>- Enables semantic modeling of big data.</li> <li>- Supports incremental maintenance of ontology.</li> <li>- Addresses JSON-based data with evolving schemas.</li> </ul>	- Specific to ontology building and JSON-based data.
Natural Language Interface to Database (NLIDB) [30]	Presents a NLIDB system using the Intermediate query approach, focusing on a Movie domain chatbot. Offers a solution for extracting information from databases using natural language queries.	<ul style="list-style-type: none"> <li>- Facilitates NL query of databases for non-expert users.</li> <li>- Supports information extraction from databases.</li> <li>- Demonstrates promising results.</li> </ul>	- Specific to NLIDB and chatbot applications.
Unified Framework with Self-Attention [31]	Proposes a unified framework for translating natural language questions into SQL queries, addressing schema encoding, schema linking, and feature representation using relation-aware self-attention. Achieves state-of-the-art performance on the Spider dataset.	<ul style="list-style-type: none"> <li>- Boosts exact match accuracy on Spider dataset.</li> <li>- Incorporates self-attention for schema and feature handling.</li> <li>- Qualitative improvements in schema linking.</li> </ul>	- Specific to text-to-SQL translation and Spider dataset.
Deep Learning for NLI to Databases [32]	Addresses challenges in developing deep learning technologies for conversational natural language interfaces (NLIs) to databases. Proposes benchmarks, neural algorithms, and language models for NLIDB.	<ul style="list-style-type: none"> <li>- Introduces benchmarks and datasets for NLI to databases.</li> <li>- Develops models for dialog-based NLI.</li> <li>- Presents improved language models for semantic parsing.</li> </ul>	- Focuses on NLI to databases and deep learning.



Table 1. Comparative analysis of existing models (*Continued*)




Method used	Details	Advantages	Research gap
ValueNet for NL-to-SQL with Values [33]	Introduces ValueNet and ValueNet light, two end-to-end Natural Language-to-SQL systems that incorporate values. Achieves state-of-the-art results on the Spider dataset.	<ul style="list-style-type: none"> <li>- Incorporates values in NL-to-SQL translation.</li> <li>- Achieves high execution accuracy on Spider dataset.</li> <li>- Introduces a novel architecture for handling values.</li> </ul>	- Specific to NL-to-SQL with values and Spider dataset.
NatSQL for Text-to-SQL Translation [34]	Proposes NatSQL, an SQL intermediate representation that simplifies SQL queries for text-to-SQL translation. Outperforms other IRs and improves the performance of existing models on Spider dataset.	<ul style="list-style-type: none"> <li>- Simplifies SQL queries for text-to-SQL translation.</li> <li>- Improves performance on complex and nested SQL queries.</li> <li>- Enables generating executable SQL queries.</li> </ul>	- Specific to text-to-SQL translation and Spider dataset.
Improving Query Interfaces with OQS [35]	Investigates oblique query specification (OQS) methods for improving query interfaces. Leverages previously-issued SQL queries and combines natural language and programming-by-example.	<ul style="list-style-type: none"> <li>- Utilizes user feedback and previously-issued SQL queries.</li> <li>- Combines natural language and programming-by-example.</li> <li>- Maximizes user domain expertise.</li> </ul>	- Focuses on query interfaces and OQS methods.
Robustness of Text-to-SQL with Domain Knowledge [36]	Investigates the robustness of text-to-SQL models when facing domain knowledge not frequently observed in training data. Introduces the Spider-DK dataset.	<ul style="list-style-type: none"> <li>- Addresses the impact of domain knowledge in text-to-SQL models.</li> <li>- Introduces the Spider-DK dataset for robustness evaluation.</li> </ul>	- Specific to domain knowledge impact and Spider-DK dataset.
Text-to-GQL for Graph Databases [37]	Proposes the Text2GQL task for translating natural language questions into GQL (Graph Query Language) for graph databases. Introduces a pipeline solution with language model and Adapter plug-in.	<ul style="list-style-type: none"> <li>- Introduces the Text2GQL task for graph databases.</li> <li>- Utilizes Adapter pre-trained on schema-utterance linking.</li> <li>- Proposes a pipeline solution for end-to-end translation.</li> </ul>	- Specific to Text2GQL task and graph databases.
SQLNet for Text-to-SQL with Sketches [38]	Introduces SQLNet, a sketch-based approach to synthesize SQL queries from natural language when the order does not matter. Improves performance on WikiSQL tasks.	<ul style="list-style-type: none"> <li>- Addresses the "order-matters" problem in text-to-SQL.</li> <li>- Utilizes sketches and dependency graphs.</li> <li>- Outperforms previous models on WikiSQL tasks.</li> </ul>	- Specific to SQLNet approach and WikiSQL tasks.
Seq2SQL with Query Execution Rewards [39]	Proposes Seq2SQL, a deep neural network for translating natural language questions to SQL queries. Utilizes query execution rewards for training and achieves significant performance improvements on WikiSQL.	<ul style="list-style-type: none"> <li>- Leverages query execution rewards for better training.</li> <li>- Improves execution accuracy and logical form accuracy.</li> <li>- Utilizes a large dataset (WikiSQL) for training.</li> </ul>	- Specific to Seq2SQL approach and WikiSQL dataset.
Multilingual NMT with Shared Model [40]	Introduces a solution to use a single Neural Machine Translation (NMT) model for translating between multiple languages, using an artificial token to specify the target language. Improves translation quality for various language pairs.	<ul style="list-style-type: none"> <li>- Enables Multilingual NMT with a single model.</li> <li>- Improves translation quality for language pairs.</li> <li>- Allows for zero-shot translation between unseen language pairs.</li> </ul>	- Specific to multilingual NMT and language translation.
LSTM Dropout Regularization [41]	Presents a regularization technique for RNNs with LSTM units using dropout. Demonstrates reduced overfitting on various tasks, including language modeling, speech recognition, image caption generation, and machine translation.	<ul style="list-style-type: none"> <li>- Reduces overfitting in LSTM-based RNNs.</li> <li>- Applicable to a variety of tasks.</li> </ul>	- Specific to LSTM-based RNNs.
Attention-Enhanced Encoder-Decoder for Semantic Parsing [42]	Introduces an attention-enhanced encoder-decoder model for semantic parsing. Encodes input utterances into vectors and generates logical forms. Adaptable across domains and meaning representations.	<ul style="list-style-type: none"> <li>- Performs competitively without hand-engineered features.</li> <li>- Adaptable across domains.</li> </ul>	- Requires evaluation on various semantic parsing tasks.
Deep Learning-Based Database Development [43]	Develops databases using deep learning and cloud computing technology. Utilizes J2EE, Oracle server database, and deep learning for data extraction, processing, fusion, and compression.	<ul style="list-style-type: none"> <li>- Uses deep learning for database development.</li> <li>- Supports distributed storage of data samples.</li> <li>- Low performance loss.</li> </ul>	- Limited to databases and cloud computing.

Table 1. Comparative analysis of existing models (*Continued*)




Method used	Details	Advantages	Research gap
Deep Neural Network for SQL Injection Detection [44]	Builds a deep neural network-based SQL injection detection model based on word vectors and deep learning. Achieves high accuracy in SQL injection detection. Addresses overfitting and feature extraction challenges.	<ul style="list-style-type: none"> <li>- Achieves high accuracy in SQL injection detection.</li> <li>- Addresses overfitting challenges.</li> <li>- Automates feature extraction.</li> </ul>	- Specific to SQL injection detection.
Transformer-Based Seq-to-Seq for Text-to-SQL [45]	Adapts transformer-based seq-to-seq model to text-to-SQL generation. Introduces Schema aware Denoising (SeaD) training and clause-sensitive execution guided (EG) decoding. Achieves state-of-the-art performance on WikiSQL benchmark.	<ul style="list-style-type: none"> <li>- Improves seq-to-seq model performance for text-to-SQL.</li> <li>- Introduces novel training objectives and decoding strategy.</li> <li>- Establishes state-of-the-art results.</li> </ul>	- Focuses on text-to-SQL and WikiSQL benchmark.
Weak Supervision for Text-to-SQL Training [46]	Proposes weak supervision using QDMR structures for training text-to-SQL parsers. Synthesizes SQL queries based on QDMR structures and answers. Competitively performs without NL-SQL annotations.	<ul style="list-style-type: none"> <li>- Provides a weak supervision approach for training text-to-SQL parsers.</li> <li>- Competes with models trained on NL-SQL data.</li> </ul>	- Limited to text-to-SQL training.
Survey of Text-to-SQL Progress [47]	Reviews recent progress on text-to-SQL for datasets, methods, and evaluation. Provides a systematic survey of challenges and future directions.	<ul style="list-style-type: none"> <li>- Provides an overview of recent progress in text-to-SQL.</li> <li>- Addresses challenges in encoding, decoding, and translation.</li> <li>- Offers insights into potential research directions.</li> </ul>	- Not focused on a specific method.
Natural Language Interface to SQL with User Privileges [48]	Proposes a system for converting natural language to SQL, allowing users to update data dictionaries and interact with databases using NLP. Enables communication between users and systems without SQL knowledge.	<ul style="list-style-type: none"> <li>- Empowers users to update data dictionaries.</li> <li>- Uses NLP for communication with databases.</li> <li>- Supports users with no SQL knowledge.</li> </ul>	- Specific to NLP-based SQL interfaces.
NLP-Based Model for Text-to-SQL [49]	Proposes an NLP-based model to convert natural language to SQL queries. Uses semantic parsing models to handle schema encoding and decoding. Achieves competitive results on text-to-SQL tasks.	<ul style="list-style-type: none"> <li>- Utilizes NLP for text-to-SQL conversion.</li> <li>- Competitively performs on text-to-SQL tasks.</li> </ul>	- Specific to text-to-SQL and semantic parsing.
Neural Networks for PL/SQL Placement Prediction [50]	Employs artificial neural networks to predict the placement of new objects among architectural modules in PL/SQL programs. Uses features extracted from source code and dependencies among objects. Achieves high accuracy in placement prediction.	<ul style="list-style-type: none"> <li>- Provides an automated approach for object placement prediction.</li> <li>- Achieves high accuracy compared to baseline methods.</li> </ul>	- Specific to PL/SQL programs and architectural placement.

## BIOGRAPHIES OF AUTHORS



**Gunjan Keswani**    is currently working as an assistant professor with Ramdeobaba University since 2015. She has an M.Tech. degree in Computer Science and Engineering from RTMNU in 2014. She has total experience of 13.5 years which includes teaching experience of 9.5 years and 4 years of industrial experience. Her areas of expertise include natural language processing, machine learning, and big data computing. She has published 6 research articles in scopus indexed journals and ESCI journals. She can be contacted at email: keswanigv@rknec.edu.



**Dr. Manoj B. Chandak**    is Director of Academics and Professor in the Department of Computer Science and Engineering at Shri Ramdeobaba College of Engineering and Management, Nagpur, India. He has total of 30+years of academic experience. His areas of interest are natural language processing, machine learning, big data analytics, cloud storage, and data management. He is also a recognized Ph.D. supervisor. He has published over 50 papers listed in Scopus Citation Index. His current research work includes the “Development of coal quality exploration technique based on convolutional neural networks and hyper-spectral images”. The project is completed in the stipulated time period, sponsored by the Ministry of Coal, Government of India with a grant of INR 1.03cr. He is also the principal investigator of the project entitled “Indigenous Development of NIR spectroscope for instant prediction of coal quality parameter, with a grant of 1.10Cr from the Ministry of Coal, Government. He can be contacted at email: chandakmb@rknc.edu.