MULTIDISCIPLINARY RESEARCH INSTITUTE INDIA

Multidisciplinary Journal of Research in Engineering and Technology

ISSN: 2348-6953

Volume 13 Issue 2s (2026) | Pages: 21-27

Comprehensive Review on Adaptive Natural Language Interfaces for Autonomous Relational Database Schema Synthesis and Context-Aware Query Interpretation Using Advanced NLP Techniques

Dipannita Mondal[1], Trupti Dongale[2], Mukesh Vadgav[3], Om Bambale[4], Harshad Kawade[5]

[1,2,3,4,5]Dept. Artificial Intelligence & Data Science, Dr. D.Y. Patil College of Engineering & Innovation, Talegaon, India

[1]ddmondal2684@gmail.com, [2]dongaletrupti@gmail.com, [3]vadgavmukesh@gmail.com, [4]bambale.om@gmail.com, [5]harshadkawade0659@gmail.com

Peer Review Information

***Type**: Article*
***Received**: 26 March 2026*
***Revised**: 10 April 2026*
***Accepted**: 24 May 2026*
***Published**: 15 June 2026*

Abstract

In this purposed system we provide a solution to a complex problem of building a system which can auto-matically create a database, populate it with data and re-spond to user queries. The system automatically creates a database based on user queries, and does entity extrac-tion, intent identification and automatically transforms nat-ural language input to SQL queries that can be executed on the database. The system also supports conversation and provides schema aware processing, supports query val-idation, and also includes data visualization. In order to provide these functionalities, we have integrated NLP tech-niques with state-of-the-art database systems. In addition, we have worked on multiple features that increase system performance. The main focus of this work has been to make database interactions easier for the users by reducing the gap between natural language and SQL. We have created a platform that supports all of these, and allows its users to provide inputs in natural language, and get results without having to write a line of SQL to interact with the database. This provides a scalable and efficient solution to manage data in various scenarios.

Keywords: Natural Language Processing; Structured Query Language; Database; Conversational; Schema; Visualization

How to Cite This Article

Mondal, D., Dongale, T., Vadgav, M., Bambale, O., & Kawade, H. (2026). Comprehensive review on adaptive natural language interfaces for autonomous relational database schema synthesis and context-aware query interpretation using advanced NLP techniques. *Multidisciplinary Journal of Research in Engineering and Technology*, 13(2s), 21–27.

Introduction

Introducing what we have come up with a smart system that incorporates database and natural language processing to automise the database creation and querying process. We have a method for how we would approach getting the required data set, attributes and constraints from the user. This would be done by implementing an NLP pipeline that processes unstructured human language and converts it into a structured intermedi-ate format. The system would then precisely identify the key elements required for schema generation i.e. table names, at-tributes and types.

The system builds out a database structure which it then populates with data. It transforms extracted entities into rela-tional tables within the database and at the same time which via a rule-based approach and AI tools brings about full normaliza-tion of those tables. We also see the system generate real and artificial data which is put in via the best data generation meth-ods to make sure the info in the database is very much like what you would see in the real world. The system uses schema aware processing which in turn guarantees that the data in the database is consistent with the structure as defined by the schema.

To increase interactivity we put in a step which allows the user to engage with the generated database via an interactive conversational mode. We have an AI based query generation module which uses context and schema info to take user queries and prompts and turn them into appropriate SQL queries. For clarity we validate the resulting SQL, run it on the database and return the results along with the which is the generated query. This approach we find to be very polished which in turn im-proves the user experience and we see value in it as it takes out the manual work of database design and SQL query writing.

Related Work

Improving the capability of processing natural language in-puts to generate SQL queries for both simple and complex data retrieval requests, there has recently been a surge of interests in leveraging advanced Natural Language Processing (NLP) tech-niques and building automated interactive database systems. In the area of generating SQL directly from user input, Tri-SQL was proposed in Su et al. [1], a framework that utilizes schema-aware reasoning to improve the accuracy of generated SQL. In order to improve the performance of query-to-schema mapping, another study proposed a knowledge graph-based ap-proach in Su et al. [2]. For enabling humans to interact with databases more efficiently through LLM-based adaptive query systems, another study explored systems that support complex queries, and examined the important challenges and techniques of schema linking, and context-aware processing in practical scenarios in Su et al. [3]. A hybrid model of Natural Lan-guage Processing (NLP) and Artificial Intelligence (AI) by us-ing multi-stage reasoning also improved query comprehension. These methods and approaches indicated that current research efforts in supporting database interaction are shifting toward de-veloping increasingly intelligent and interactive systems.

In order to increase generalization ability and accuracy of models, researchers in 2025 explored a wide range of ap-proaches for several natural language processing tasks [4]. For Text-to-SQL, a BERT-based encoder-decoder model was devel-oped that reaches state-of-the-art accuracy using beam search [5]. To improve the performance on Text-to-SQL, sequential consistency prompting was proposed to inculcate more correct SQL syntax and semantics into language models [6]. For multi-domain databases, Retrieval-Augmented Generation (RAG) [7] models were explored to supply necessary schema informa-tion during query generation [8]. Additionally, models that achieve better semantic understanding and query mapping us-ing transformer-based architectures were introduced [9]. In a comprehensive survey, important issues such as (deep) schema linking, domain adaptation, model interpretability and model transferability for SQL-to-JSON were presented [10]. Finally, the importance of error correction for invalid SQL generated by AI systems was discussed [11].

In order to fulfil the vision of the proposed project, the ground work was laid by earlier research in 2024 [12]. The goal of that research was to investigate the integration of Large Language Models in a processing pipeline for natural language to SQL, called Text-to-SQL [13]. Through research on prompt engineering large language models were able to generate SQL-queries from natural language input with high precision [14]. Additionally, the fine-tuning of large language models for a specific domain or dataset was investigated in some cases [15]. Moreover, further research focused on the design of an agent-based pipeline, where the generated SQL-queries are interac-tively improved by an AI-System based on user-feedback [16]. The identified challenges like natural language ambiguity, miss-ing schema information and a lack of robustness are used as a basis to design fully automated, highly convenient and user-friendly systems for natural language querying [17].

Methodology and System Architecture

- Input Acquisition and Preprocessing: The process begins with the acceptance of the user's input, from the user which may be a description of what the user's data base needs are or a direct question the user has. We then go through the same process of prepping the input as we would for any Natural Language Processing task which includes tok-enization, removal of stop words, and word normalization. The goal of this pre-processing step is to remove the non-essential words which do not contribute to an answer of the user's ques-tion and to keep only the relevant information.

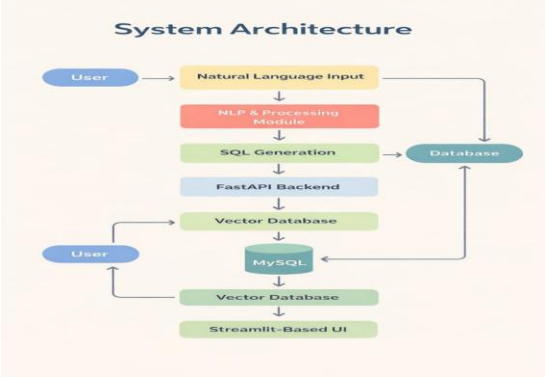

Fig. 1. *System Architecture*

- Natural Language Understanding: In this phase, the system retrieves significant information from the processed text. This we are able to do which is to identify and tag out words as table names, attributes, numbers and conditions which in turn enables us to do Named Entity Recognition (NER) and Part of Speech tagging (POS). Also, we are able to do intent detection in user input to determine if they are looking to create a database or execute one. We trans-form freeform human language into a machine-readable format which in turn we use to extract entities, relationships and con-straints and then we generate the database code.
- Schema Generation Module The extracted entities are then mapped to database compo-nents through schema generation logic. It dynamically gener-ates the relational schema from the provided table definition, its attributes, primary key and constraints. It has a feature of auto inference of data types on the basis of context, like number for price and text for name etc. It also implements normalization which helps in removing data redundancy and inconsistency. It generates a structured schema for database implementation.
- Database Creation and Data Generation: Once the schema defined, the system will then generate a corresponding database using SQL commands. The database is then provisioned by a relational database management system and populated either with synthetic data provided by external libraries (e.g., Faker library) or with actual external data sources when possible. As with the model, generated data is guaranteed to be logically consistent with both the schema and specified requirements.
- Text-to-SQL Query Generation: The system implements ability to transform user's ques-tions into SQL. To achieve, develop hybrid approach that incor-porates rule-based models and AI-powered models for question answering. Implemented schema-aware prompting mechanism, where database schema is provided as input to the prompting model. The generated SQL is guaranteed to be valid.
- Query Validation and Execution: The final output is presented to the user in a clear format, including the results of any queries executed, as well as the gen-erated SQL command to recreate the results. This system can be used to generate ER diagrams in Graphviz format as well. You can also download the generated database in various for-mats such as SQL or CSV.
- Result Generation and Visualization: The final output is presented to the user in a clear format, including the results of any queries executed, as well as the gen-erated SQL command to recreate the results. This system can be used to generate ER diagrams in Graphviz format as well. You can also download the generated database in various for-mats such as SQL or CSV.
- Conversational Interface and Context Management The system supports real-time, multi-turn conversations, enabling users to have a conversation through users' input on Gitter, rather than via API calls and queries. Gitterbot success-fully recognises follow-up questions, whether implicit, short, or context-dependent. The system's context management feature tracks the evolving interaction context, structured as a session history containing users' input, the system's responses and the inferred intent..

Results and Discussion

Dataset Generation Output

Enter a prompt in natural language. Based on your prompt, Fivetran's Data Model Generator will generate a database schema from it and populate it with the specified number of rows. You will see the prompt you entered, the number of rows in the dataset, and a toggle to generate a dataset. Once the database is generated successfully, you will see the option to generate a dataset. Picture

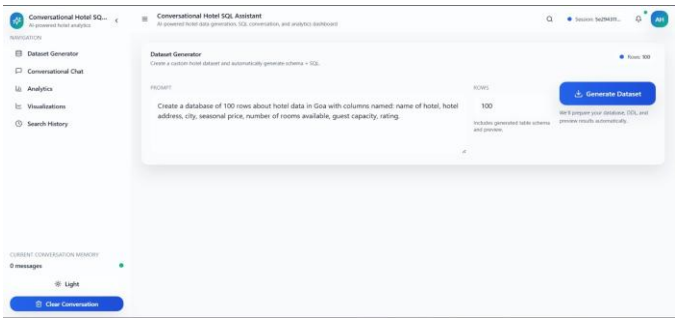

Fig. 2. *Dataset Generation*

Analytics Dashboard Output

After we generated a dataset, the system also calculated a few key performance indicators like total records, average price, lowest price, average rating and most common category. This automatic data analysis feature of the system provides the user with a quick summary of what the data actually looks like.

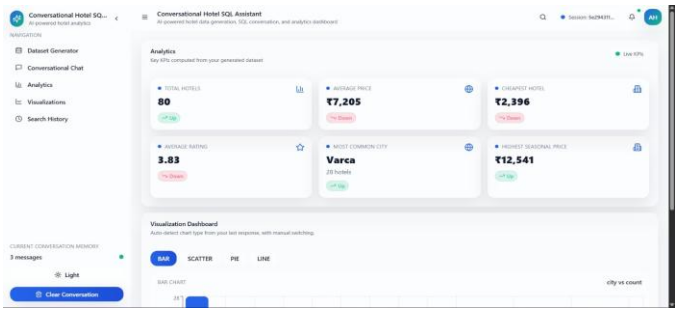

Fig. 3. *Analytical Report*

Data Visualization Output

The system provides several kinds of visualization: bar charts, pie charts, scatter plots, and line graphs. It dynamically selects visualizations based on user interaction and/or query re-sults to effectively display patterns and distributions within the data.

Fig. 4. *Data Visualization: Pie Chart.*

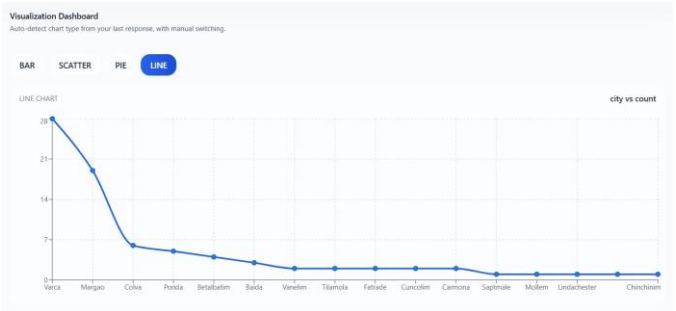

Fig. 5. Data Visualization: Line Plot

Conversational Query Output

[t] Interact with the database using natural language queries. The system translates the user input into SQL com-mands, executes the

command, displays the generated SQL and the results to the user. It supports text-to-SQL capabilities.

Search History and Context Handling

The system maintains a history of user queries and sup-ports contextual understanding. Previous queries can be revisited, and follow-up queries are handled effectively, demonstrating conversational continuity and improved user interaction.

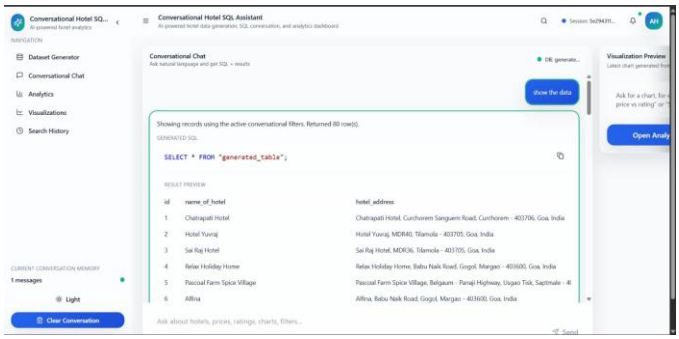

Fig. 6. *Generated Database*

Applications

- Business Data Management and Analysis: The system allows businesses to rapidly create and manage databases, with-out ever needing to understand the behind-the-scenes code. With Natural Language Processing, using the platform, users can simply tell the system what they need, and it will gen-erate the correct database structure for analysis in a matter of seconds. Natural allows businesses to make quicker, more in-formed decisions, identify trends, and most importantly save time and money by not having to rely on a team of highly trained database engineers.
- Educational and Learning Support: This system can be used interactively as an educational tool during lessons on Database Systems and SQL for students to better understand database creation, designing a schema and executing SQL queries.
- Rapid Prototyping for Startups: The data generator can be useful to a startup that needs a populated database to test their new ideas, or to a developer who wants to see how their application would work with some data. The automated schema generation simplifies the process of building a sample dataset, reducing the time it takes to test new application ideas.
- Data Analytics and Visualization: Data comes to life with built-in analytics and visualization to provide clear and insightful charts and metrics. Power Analytics also makes analysis easier by providing simple insights such as averages, counts and trends on large datasets of information.
- Conversational Data Query Systems: The system can be integrated in any application where users need to retrieve data by typing natural language queries. Unlike current systems that require users to write SQL statements to extract relevant information, users can simply chat with the system and ask natural language questions, which would be internally converted and sent to MySQL for query processing.

Limitations and Future Work

Current Limitations

- Limited Understanding of Complex Queries:Natural Lan-guage Processing is used, the system has limitations and complex natural language queries that are ambiguous or which include advanced conditions (e.g. nested condi-tions), joins, or unclear intent will not be supported
- Dependence on Schema Context:The accuracy of the SQL that is generated depends on the accuracy of the database schema as it is interpreted by the XQuery code generator. Incomplete or incorrect schema interpretation leads to gen-erated SQL statements that are incorrect or suboptimal.
- Synthetic Data Quality: The system can work best in struc-tured domains, such as the hotel domain for the provided sample dataset. It may require additional customization and/or training data for other domains like healthcare and finance.
- Scalability Constraints: Performance may be affected by large datasets or multiple users accessing at the same time, although the tool is currently designed to handle moderate amounts of data and handles generating and querying at moderate speeds.

Future Directions

The focus of future work will be on finishing the system robustness, scalability and strictness.

- Improved Query Understanding: Enhance exploration into improved query handling, we examine the possibilities for handling more complex and multi-step queries by developing new Natural Language Processing techniques. We also inves-tigate the use

of joins and nested queries, and discuss possible approaches to handling ambiguity in user input.

- Domain Adaptability: Extend the current system to sup-port the multiple domains such as healthcare, finance, e-commerce etc. by utilizing domain specific schema templates and their corresponding data generation rules.
- Scalability and Performance Optimization:Enable your application to handle large amounts of data simultaneous users by optimizing database queries and internal processing within MySQL and your backend.
- Deployment as a SaaS Platform: Develop recommenda-tion system that is to implement the solution as a Cloud Based Application allowing users access to the database creation and analytics tools from anywhere.

Conclusion

The Proposed is a model which puts forth a solution for the automation of database creation and query processing via the use of Natural Language Processing and Database Systems. We present a method which makes database management a more user-friendly task by which the user may put in data in a nat-ural way which in turn is easily processed without the require-ment of the user to design the database schema out himself and also write SQL queries. The method we present includes in schema generation, data population, and conversational query-ing which we have integrated into one system to ensure smooth operation. The system is able to adapt to a variety of user inputs and put out the appropriate response with a minimum amount of intervention. Also included in this system are features of query validation, error handling, and result transparency which in turn improve system reliability and usability. In conclusion, this project has developed scalable and intelligent system for handling problems with your data. Although we still have a few issues to iron out, such as correctly interpreting vague user queries and handling them in real time, the base for future im-provements to this system has been established and continue to work on issues like these to further develop an automatic data system.

Future work will focus on improving generalisation across subjects, extending to multi-class scenarios, and incorporating adaptive learning mechanisms to enhance real-world applicabil-ity.

Acknowledgment

The authors thank the BCI Competition IV organisers for providing the publicly available EEG datasets used in this work. This work was conducted at the Department of Artificial Intelli-gence and Data Science, D. Y. Patil College of Engineering and Innovation, Pune, India.

Reference

1. Su, X., Gu, Y., Wang, P., Gu, W., Qi, L., & He, J. (2026). A robust natural language text-to-SQL generation framework with dynamic strategies based on LLMs. Scientific Reports.
2. Nascimento, E. R., Avila, C. V. S., Izquierdo, Y. T., García, G. M., Andrade, L. F. L., Silva, M. O., & Casanova, M. A. (2026). A Text-to-SQL strategy based on large language models and knowledge graphs for real-world databases. Data & Knowledge Engineering, 164, 102580.
3. Rajaput, A., Sangolli, V., Bellerimath, A., Abbigeri, A., Kulkarni, U., & Hegde, S. Text to SQL Generation Using Beam Search and an Enhanced BERT Model.
4. Zhang, Z., Chen, Y., Guo, C., Song, J., & He, G. (2025). Enhancing Text-to-SQL generation with language sequential consistency. Neurocomputing, 131721.
5. Baig, M. S., Sher, T., Rehman, A., & Sheikh, S. (2025). A Systematic Literature Review of Text-to-SQL: Performance, Challenges, and Limitations. ICCK Transactions on Advanced Computing and Systems, 2(1), 1–24.
6. Singh, A., Shetty, A., Ehtesham, A., Kumar, S., & Khoei, T. T. (2025, January). A survey of large language model-based generative AI for text-to-SQL: Benchmarks, applications, use cases, and challenges. In 2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 15–21). IEEE.
7. Belefqih, S., Barchane, M., Zellou, A., & Benlahmar, E. H. (2025). A novel framework for RDF schema extraction in NoSQL databases using Sentence-BERT. IEEE Access.
8. Madduru, P. AI-Based Schema Mapping Using NLP in Large Data Integration Projects.
9. Vadgav, M., Bambale, O., & Narwade, D. (2025). Fraud detection in banking using ML. International Journal on Advanced Computer Theory and Engineering, 14(1), 600–606.
10. Narwade, D., Bambale, O., & Vadgav, M. (2025). Rise of AI in healthcare engineering: Opportunities and ethical challenges. International Journal on Advanced Computer Theory and Engineering, 14(1), 607–613.
11. Yang, Y., Peng, Z., Zhou, F., Yao, X., & Zhang, Y. (2025, October). Conversational Text-to-SQL: A Comprehensive Survey of Paradigms, Challenges, and Future Directions. In Workshop on Agents and Robots for Reliable Engineered Autonomy (pp. 1–19). Springer Nature Switzerland.
12. Wang, Q., Li, Y., Feng, Y., Chen, S., Li, Z., Zhang, P., ... Jin, W. (2025). Text2Schema: Filling the Gap in Designing Database Table Structures Based on Natural Language. arXiv preprint arXiv:2503.23886.

13. Zhu, X., Li, Q., Cui, L., & Liu, Y. (2024). Large language model enhanced text-to-SQL generation: A survey. arXiv preprint arXiv:2410.06011.
14. Mohammadjafari, A., Maida, A. S., & Gottumukkala, R. (2024). From natural language to SQL: Review of LLM-based text-to-SQL systems. arXiv preprint arXiv:2410.01066.
15. Kanburoğlu, A. B., & Tek, F. B. (2024). Text-to-SQL: A methodical review of challenges and models. Turkish Journal of Electrical Engineering and Computer Sciences, 32(3), 403–419.
16. Orrenius, A. (2024). Enhancing Text-to-SQL Applications with Retrieval Augmented Generation: How Academic Advancements in Text-to-SQL Translate to Industry Usage.
17. Gao, D., Wang, H., Li, Y., Sun, X., Qian, Y., Ding, B., & Zhou, J. (2023). Text-to-SQL empowered by large language models: A benchmark evaluation. arXiv preprint arXiv:2308.15363.