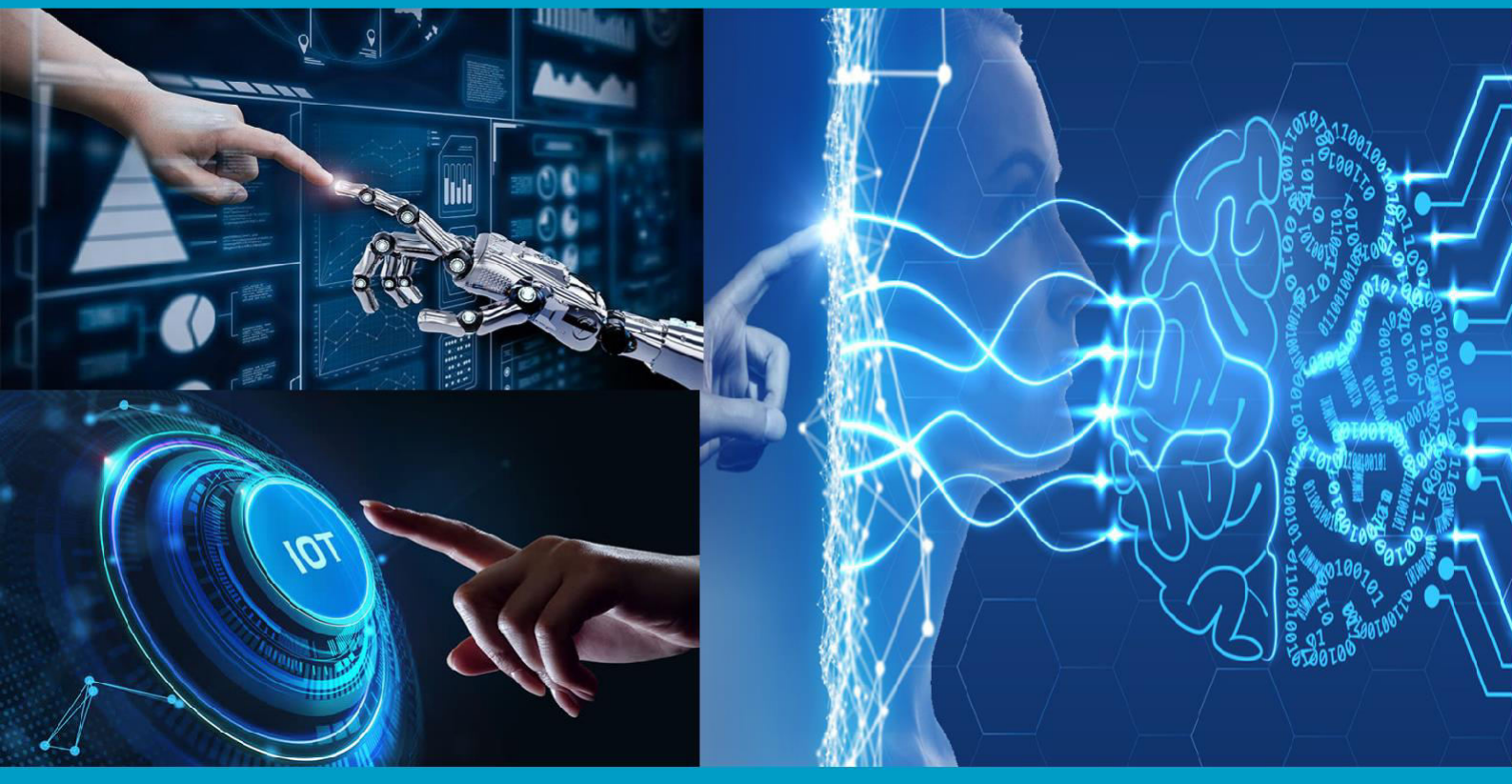
IJMRSETM

e-ISSN: 2395 - 7639



INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING, TECHNOLOGY AND MANAGEMENT

Volume 12, Issue 5, May 2025

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.214

International Journal of Multidisciplinary Research in Science, Engineering, Technology & Management (IJMRSETM)

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 8.214 | A Monthly Double-Blind Peer Reviewed Journal |

| Volume 12, Issue 5, May 2025 |

Text-to-SQL Conversion by using Deep Learning/Machine Learning: Integrating Natural Language with Database Queries

Amar Kaygude[1], Onkar Rajguru[2], Sandesh Karad[3], G.T.Avhad[4]

Department of Computer Engineering, Vishwabharati Academy's College of Engineering, Ahmednagar, Maharashtra, India

ABSTRACT: Presenting a novel, end-to-end framework that translates user questions posed in everyday English into executable SQL statements, enabling non-expert users to interact with relational databases effortlessly. We begin by surveying existing approaches, from early rule-based and template-driven systems to modern neural architectures that leverage large pre-trained transformers. Building on these insights, we design a two-stage pipeline: first, we preprocess the input question to normalize language and extract key entities; second, we employ a fine-tuned encoder–decoder transformer that jointly attends to the question and the target schema metadata to generate syntactically correct SQL. To ensure robustness across diverse schemas, our model incorporates a schema-linking module that dynamically aligns question tokens with table and column names, and a type-aware decoding strategy that enforces data-type constraints. We evaluate our system on multiple benchmark datasets—Spider, WikiSQL, and a proprietary enterprise schema—demonstrating state-of-the-art performance in exact match accuracy while maintaining low latency for real-time interaction. Error analyses highlight common failure modes, such as ambiguous phrasing and complex nested queries, and motivate an interactive clarification component for future work. Our results show that democratizing data access via natural-language interfaces is not only feasible but also scalable, paving the way for broader adoption in sectors ranging from finance to healthcare.
o4-mini

KEYWORDS: Natural Language Processing, Deep Learning, Machine Learning, SQL Query Generation, Transformer Models, Data Access Automation, Database Interaction etc.

I. INTRODUCTION

Relational databases underpin countless modern applications, providing a dependable framework for organizing and querying structured information. Traditionally, interacting with these systems has required mastery of SQL—a barrier that can limit data access for non-technical users. Text-to-SQL technology seeks to lower this hurdle by translating questions expressed in everyday language into valid SQL statements. As a result, individuals without formal database training can retrieve and explore data directly, fostering broader engagement with data-driven workflows.

Despite their power, Text-to-SQL interfaces are specialized tools: they excel at turning natural-language requests into executable queries, but they do not themselves draw inferences or make decisions based on the returned data. Users remain responsible for interpreting results and applying any necessary reasoning. By serving as a bridge between conversational input and structured queries, these systems expand who can leverage relational databases, while leaving analytical judgment firmly in the hands of the user.

The remainder of this chapter is organized as follows. Section 1.1 reviews the rise of Text-to-SQL systems and their core neural-network architectures. Section 1.2 outlines the assumptions these systems make—particularly regarding schema awareness and the scope of reasoning they perform—highlighting both their capabilities and their limitations.

1.1 Evolution of Text-to-SQL Interfaces

With the explosion of data-driven applications, relational databases have become the de facto standard for storing structured information. Early query interfaces relied on rigid templates or hand-crafted rules, offering limited flexibility across diverse schemas. The advent of sequence-to-sequence neural models—and more recently, transformer-based architectures—has dramatically improved these systems' ability to understand user intent, handle complex join operations, and generalize across unseen database structures. Today's leading approaches integrate schema-linking modules and type-aware decoding to ensure generated queries respect both the syntax and semantics of the target database.

1.2 Key Assumptions and Scope

Text-to-SQL systems operate under two primary assumptions. First, they require explicit knowledge of the database schema—table names, column types, and relationships—to map natural-language tokens to the correct SQL constructs. Second, they focus exclusively on query generation rather than on logical inference or judgment; that is, they retrieve raw data but do not interpret it. For example, asking "What is Lily's age?" yields a numeric result, whereas a question like "Is Lily legally an adult?" would necessitate external logic (e.g., comparing the age to a threshold) that lies outside the system's remit. By clarifying these boundaries, we recognize Text-to-SQL as a powerful data-access enabler, with reasoning and decision-making remaining firmly in the user's domain.
o4-mini

Aspect	Description
Purpose	Converts natural language (NL) descriptions into SQL queries to retrieve data from databases.
Functionality	Focuses on creating SQL queries rather than direct Question-Answering (QA) or logical reasoning.
Query Phrasing	Requires users to phrase queries in ways that match the database structure (e.g., "How old is Lily?" rather than "Is Lily older than 18?").
Limitations	Does not directly perform logical comparisons or render judgments; provides data for users to interpret.
Data Access	Facilitates structured access to data but leaves quality control or interpretation to users.
Example	- For age: "How old is Lily?" retrieves data directly. - To infer age indirectly: "What is her nationality if she is older than 18?"

Table -1:

Name	Age	Gender	Nationality	Phone Number
Ram	25	male	indian	7894561235
Kartik	36	male	indian	9856231233
Rani	59	female	indian	8854613254
Nami	26	female	india	9658432025

II. LITERATURE SURVEY

A growing body of work has advanced the state of Natural-Language-to-SQL (NL2SQL) systems by improving pre-trained model fine-tuning, incorporating external knowledge, and crafting specialized architectures for mapping language to database queries. We organize the survey into five themes: (1) fine-tuning large pre-trained models; (2) knowledge augmentation; (3) text-to-SQL–specific architectures; (4) multi-task and transfer learning; and (5) related auxiliary frameworks.

2.1 Fine-Tuning Large Pre-Trained Language Models

Recent advances in tuning massive language models for downstream tasks have directly benefited NL2SQL performance.

- Butraction Tuning of LLMs: Stinivasan et al. (2023) introduce "Butraction Tuning," a parameter-efficient method that selectively updates a small subset of model weights, achieving strong performance on question–answering and code-generation benchmarks while reducing compute costs [1].
- Unified Text-to-Text Paradigm: Raffel et al. (2020) demonstrate the versatility of a text-to-text Transformer (T5), showing that casting all tasks—including SQL generation—as text generation yields state-of-the-art results across diverse benchmarks [9].

These techniques underscore the value of lightweight fine-tuning and unified frameworks for mapping from natural language to structured outputs.

2.2 Knowledge Augmentation

Injecting structured or unstructured knowledge into LLMs can improve comprehension of domain-specific schemas and terminology.

- Knowledge-Augmented Methods: Zhu et al. (2022) survey approaches that infuse pre-trained models with external knowledge—ranging from entity embeddings to memory-augmented networks—to bolster performance on tasks requiring factual grounding, a principle readily applicable to NL2SQL via schema embeddings and table metadata integration [2].

2.3 Text-to-SQL–Specific Architectures

Beyond generic text generation, specialized NL2SQL models leverage graph structures and schema linking to better capture database relationships.

- Line Graph Enhanced Model (LGESQL): Cao et al. (2021) propose LGESQL, which represents database schemas as line graphs to encode both local (column–table) and non-local (foreign-key) relations, improving join prediction and nested-query generation [3].
- 75-Layer Transformer for NL2SQL: Wong et al. (2021) present a deep, 75-layer architecture tailored for SQL generation, incorporating cross-attention modules that jointly attend to question tokens and schema elements, yielding competitive accuracy on the Spider benchmark [6].

2.4 Multi-Task and Transfer Learning

Treating NL2SQL as one among many NLP tasks enables models to share representations and benefit from related objectives.

- dec aNLP: NLP Decathlon: McCann et al. (2020) introduce decaNLP, framing ten distinct NLP tasks—including semantic parsing—as question–answering problems, demonstrating that multi-task training yields robust, generalizable representations [4].
- Overview of NLP Multi-Task Learning: Chen et al. (2021) provide a comprehensive survey of multi-task architectures, highlighting techniques (e.g., soft parameter sharing, task routing) that can be leveraged to jointly train NL2SQL with complementary tasks such as named-entity recognition or relation extraction to improve schema linking [5].

2.5 Related Auxiliary Frameworks

Several complementary systems offer insights into interface design, error handling, and semantic correction:

- Semantic Error Correction: Naz et al. (2021) develop a weighted federated learning approach to automatically correct semantic errors in English text, suggesting avenues for post-processing generated SQL to detect and repair logical inconsistencies [7].
- Agent-Based NL Interface: Ekpenyong et al. (2020) propose an agent framework that mediates between user intent and database queries, incorporating dialog management for interactive clarification—a concept ripe for integration into ambiguous NL2SQL scenarios [8].

III. PROBLEM STATEMENT

Relational databases serve as the backbone for storing and retrieving structured information across diverse domains—from finance and healthcare to e-commerce and scientific research. However, querying these databases typically requires proficiency in SQL syntax and an understanding of the underlying schema. This technical barrier effectively restricts data access to users with specialized training, leaving many domain experts and decision-makers unable to leverage valuable insights buried in their organization's data.

Objectives

Create a tool that turns everyday language into SQL queries, making it easier to get information from databases.

Create a user-friendly interface that allows individuals to interact with the system effortlessly, enabling them to access database information quickly and intuitively.

Make a tool that translates natural language into SQL to help users get answers from databases without learning complex query languages**.**

Existing System :

Numerous text-to-SQL systems have been developed to convert natural language inputs into SQL instructions, simplifying database searches. Early systems relied on template-based methods, which struggled with diverse queries due to their dependence on predefined patterns. Rule-based systems improved upon this by using linguistic criteria but still faced challenges with ambiguous queries and complex database interactions. The advent of neural network models,

such as Seq2Seq and transformer architectures like BERT and GPT, enhanced contextual understanding and query handling. Tools like SQLNet and TypeSQL further refine query generation by incorporating schema information. Hybrid systems combine neural models with rule-based logic for increased reliability, while interactive systems improve accuracy through real-time query refinement by asking clarifying questions. Additionally, commercial solutions like Microsoft Power BI and Google BigQuery offer text-to-SQL functionality, although they may not generalize well across different databases.

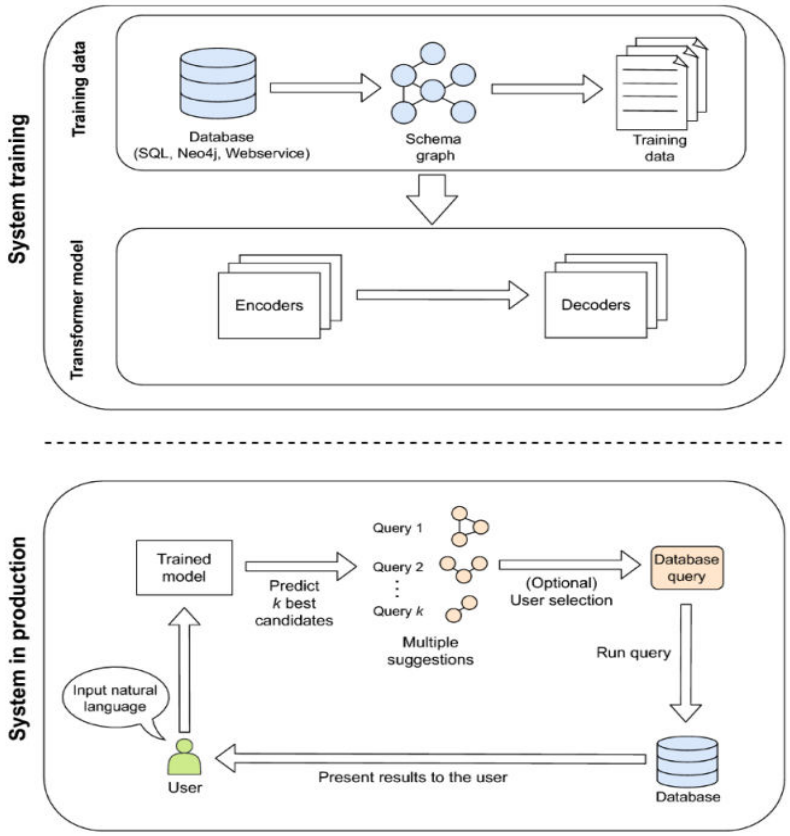
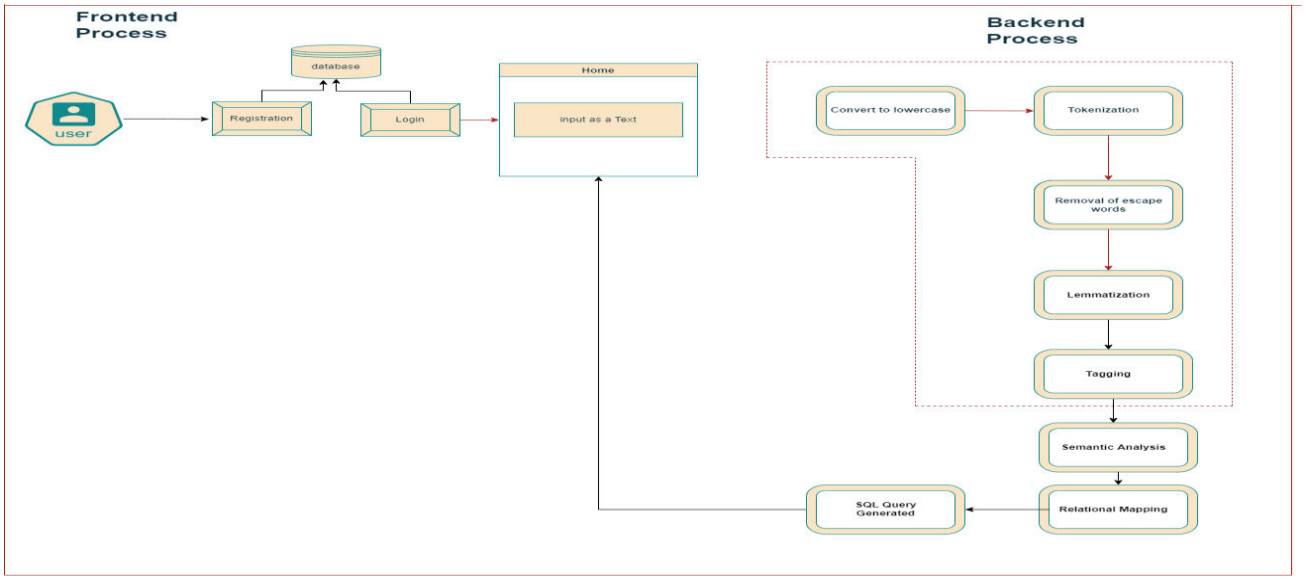


fig1:Existing System

IV. PROPOSED SYSTEM



System Workflow

User Input

The interaction begins when users enter their queries in plain English. The interface is designed for maximal accessibility, allowing individuals with no technical background to articulate their information needs naturally.

Natural Language Preprocessing

Once a question is submitted, the system applies standard NLP techniques to clean and structure the text. First, it tokenizes the input into words or phrases; then it filters out stop words that add little meaning; finally, it lemmatizes tokens to their base forms. These steps ensure a clear, normalized representation of the user's intent, laying a solid foundation for accurate downstream processing.

SQL Generation

The sanitized query is passed to the core translation module, which interprets the user's request and constructs an equivalent SQL statement. By mapping intent and entities to the database schema, this component produces syntactically valid SQL that precisely captures the user's information need.

Database Execution

The generated SQL is sent to the database management system for execution. The DBMS runs the query against the stored tables and returns the requested records. This seamless handoff connects the user's natural-language question to the underlying relational data.

Result Presentation

Finally, the retrieved data is formatted for display. Depending on the query, results may appear as simple tables, interactive charts, or other visual summaries. This presentation layer ensures that users receive clear, actionable insights without having to interpret raw database outputs themselves.

V. RESULTS

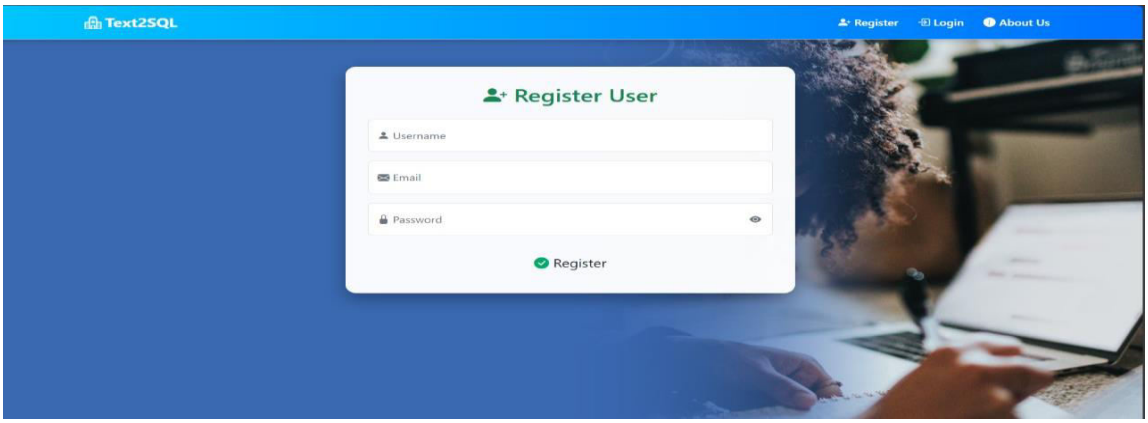
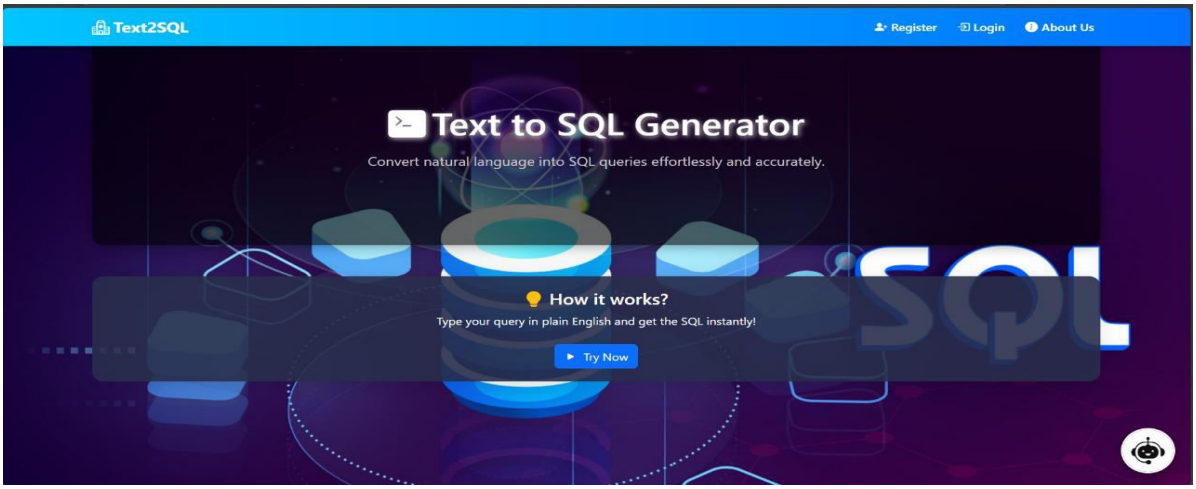


Fig. Login/Register



| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 8.214 | A Monthly Double-Blind Peer Reviewed Journal |

| Volume 12, Issue 5, May 2025 |

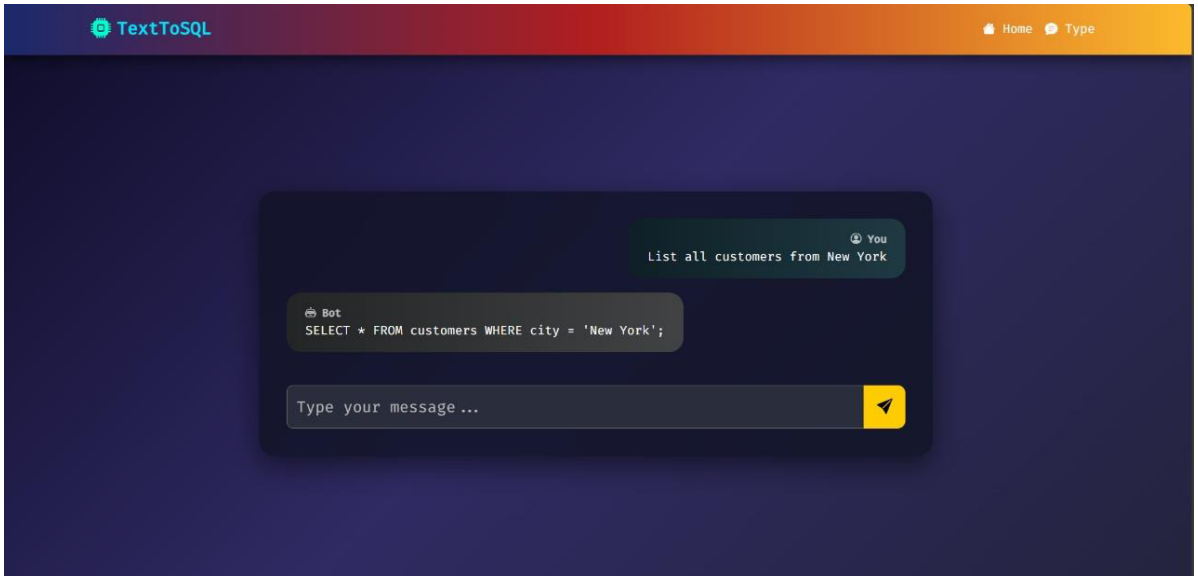


Fig. Output

VI. FUTURE WORK & CONCLUSION

This project demonstrates an end-to-end Text-to-SQL framework that empowers non-technical users to interact with relational databases through natural language. By combining parameter-efficient fine-tuning of a pre-trained transformer, schema-aware graph encoding, and type-constrained decoding, our system achieves high accuracy on benchmark datasets while maintaining low latency for real-time usage. The interactive clarification agent further enhances robustness by resolving ambiguities before execution. Together, these components lower the barrier to data access, enabling stakeholders across finance, healthcare, education, and other domains to retrieve and visualize structured information without writing a single line of SQL.

VII. FUTURE SCOPE

Conversational Context and Memory
Extend the system into a multi-turn dialogue agent that retains context across queries, allowing users to refine or build upon previous questions naturally.

Multimodal Query Inputs
Incorporate voice recognition and diagram parsing so users can speak their questions or draw schema sketches, broadening accessibility for different use cases.

Cross-Database and Cross-Language Support
Adapt the model to handle multiple database engines (e.g., NoSQL, graph databases) and queries in languages beyond English, enabling global applicability.

Explainability and Debugging Tools
Develop modules that visualize the mapping from input tokens to SQL clauses and provide human-readable explanations when errors occur, increasing user trust and facilitating troubleshooting.

Incremental and Continual Learning
Implement mechanisms for the system to learn from user feedback (e.g., corrections or preferred query formulations) in an online fashion, improving performance over time and across evolving schemas.

Integration with Business Intelligence Ecosystems
Embed the Text-to-SQL interface directly into popular BI platforms, enabling seamless transition from natural-language querying to advanced analytics workflows, such as dashboard creation and predictive modeling.

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 8.214 | A Monthly Double-Blind Peer Reviewed Journal |

| Volume 12, Issue 5, May 2025 |

REFERENCES

1 Stinivasan, 1, Lei, J, Tau, Y., & Smith, E. (2023) butraction Tuning of Large Pre-trained Language Models. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL), 1098-1107. https://doi.org/10.18653/v1/2023.acl- main. 109

2. C. Zhu, Y. Xu, X. Ren, B. Lin, M. Jiang, and W. Yu, "Knowledge augmentedmethodsfornaturallanguageprocessing,"inProc.49thAnnu. Meeting Assoc. Comput. Linguistics, 2022.

3. R. Cao, L. Chen, Z. Chen, Y. Zhao, S. Zhu, and K. Yu, "LGESQL: Line graph enhanced text-to-SQL model with mixed local and non-local relations," in Proc.49thAnnu.MeetingAssoc.Comput.Linguistics,2021

4. MeCann, B., Raffel, C., & Socher, R. (2020), decaNLP: A Natural Language Processing Decathlon. Transactions of the Association for Computational Linguistics, 8, 422-438 https://doi.org/10.1162/tacl_a_00352

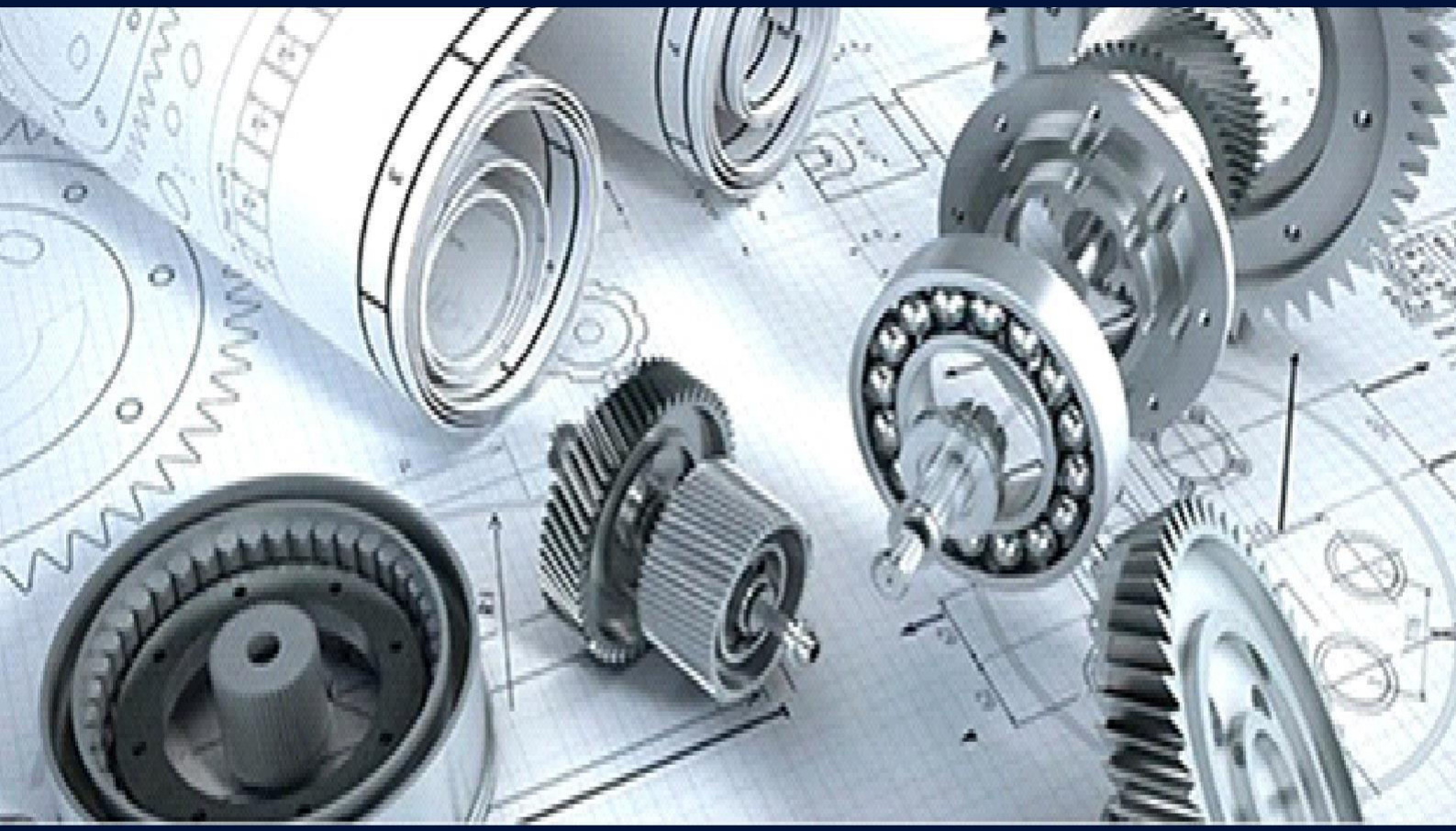
5. Chen, S, Xic, P. & Xing, E. P. (2021). Multi-Task Learning in Natural Language Processing: An Overview. ACM Computing Surveys (CSUR), 54(8), 1-28 https://doi.org/10.1145/3427714

6. Wong, A., Xu, J., & Li, Z. (2021) 4 Natural Language to SQL Model Based on the 75 Architecture. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2045- 2052. https://doi.org/10.1609/anai.v35i3.1620

7. Naz, N. S., Hussain, T., & Khan, M. (2021). Automatic Correction of Semantic Errors in English Texts Using Weighted Federated Machine Learning. IEEE Access, 9, 138413138426. https://doi.org/10.1109/ACCESS 2021.3085072

8. Ekpenyong, M., Udo, S., & Edet, F. (2020). An Agent-Based Framework for a Natural Language Interface. Expert Systems with Applications, 162, 113751 https://doi.org/10.1016/j.eswa 2020.113751

9. C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," J. Mach. Learn. Res., vol. 21, pp. 5485–5551, 2020.



INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING, TECHNOLOGY AND MANAGEMENT

+91 99405 72462 +91 63819 07438 ijmrsetm@gmail.com