IJARETY

International Journal of Advanced Research in Education and TechnologY (IJARETY)

Volume 12, Issue 3, May-June 2025

Impact Factor: 8.152

www.ijarety.in editor.ijarety@gmail.com

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May - June 2025 ||

DOI:10.15680/IJARETY.2025.1203098

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: A novel, comprehensive framework that enables users to convert everyday English questions into executable SQL queries, making relational database interaction accessible to those without specialized expertise. Our work begins with an overview of prior methodologies, spanning from traditional rule-based and template-driven approaches to contemporary neural network models utilizing large pre-trained transformers. Leveraging these insights, we propose a two-step pipeline: initially, the input question undergoes preprocessing to normalize the language and identify critical entities; subsequently, a fine-tuned encoder-decoder transformer model generates syntactically valid SQL by simultaneously attending to both the input question and the schema metadata of the target database. To enhance adaptability across varying database schemas, our system integrates a schema-linking mechanism that dynamically associates tokens in the question with relevant tables and columns, alongside a type-aware decoding process that respects data type constraints during query generation. We validate our approach on several established benchmarks—including Spider, WikiSQL, and a proprietary enterprise dataset—achieving state-of-the-art exact match accuracy while maintaining low latency suitable for real-time applications. Through detailed error analysis, we identify recurring challenges such as ambiguous user queries and complex nested SQL statements, which motivate the addition of an interactive clarification module in future iterations. Our findings suggest that enabling natural language access to databases is both practical and scalable, promising significant impact across industries like finance and healthcare.

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

I. INTRODUCTION

Relational databases serve as the backbone for a vast array of modern applications, offering a reliable structure for storing and querying organized data. Traditionally, effective interaction with these systems demands proficiency in SQL, creating a significant obstacle for users without technical backgrounds. Text-to-SQL technology aims to bridge this gap by converting natural language questions into valid SQL commands, allowing non-experts to directly access and analyze data. This democratization of data access encourages wider participation in data-driven processes.

While powerful, Text-to-SQL tools are specialized in nature: they transform conversational queries into executable database commands but do not perform interpretation or decision-making based on the results. The responsibility for analyzing the returned data and drawing conclusions remains with the user. By functioning as an interface between everyday language and structured database queries, these systems broaden the pool of users who can engage with relational databases, while keeping the analytical reasoning in human hands.

The structure of this chapter is as follows. Section 1.1 traces the development of Text-to-SQL systems and their foundational neural network designs. Section 1.2 discusses the underlying assumptions these systems rely on—particularly in terms of schema knowledge and reasoning capabilities—highlighting both their strengths and inherent limitations

1.1 Evolution of Text-to-SQL Interfaces

As data-driven applications have surged in number, relational databases have solidified their role as the standard for structured data storage. Early querying methods depended heavily on rigid templates or manually crafted rules, which offered limited adaptability to varied database schemas. The rise of sequence-to-sequence neural networks—and more recently, transformer-based models—has significantly enhanced the ability of these systems to comprehend user intent,

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May - June 2025 ||

DOI:10.15680/IJARETY.2025.1203098

manage complex join operations, and generalize to unfamiliar database structures. Current state-of-the-art methods incorporate schema-linking components and type-aware decoding strategies to ensure that generated SQL queries are both syntactically correct and semantically appropriate for the target database.

1.2 Key Assumptions and Scope

Text-to-SQL systems rest on two key assumptions. Firstly, they require explicit schema information—such as table names, column data types, and relationships—to accurately map natural language input to the corresponding SQL elements. Secondly, these systems focus solely on producing the SQL queries themselves, rather than performing any logical inference or evaluative reasoning. For instance, a question like "What is Lily's age?" produces a numerical answer directly from the database, but a question such as "Is Lily legally an adult?" would need additional logic (like comparing the age against a legal threshold) beyond the system's capabilities. By acknowledging these boundaries, we see Text-to-SQL as an effective enabler of data retrieval, with the responsibility for interpretation and decision-making remaining firmly with the end user.

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
Ramesh	25	male	indian	7894561235
Kartika	36	male	indian	9856231233
Ranu	59	female	indian	8854613254
Namika	26	female	india	9658432025

II. LITERATURE SURVEY

The field of Natural Language to SQL (NL2SQL) systems has seen significant progress in recent years, driven by innovations in model fine-tuning, the integration of external knowledge sources, and the development of specialized architectures tailored for bridging natural language and relational database queries. To provide a comprehensive overview of these advancements, we categorize the existing work into five key thematic areas: (1) fine-tuning of large pre-trained language models, (2) knowledge augmentation techniques, (3) NL2SQL-specific architectural designs, (4) multi-task and transfer learning strategies, and (5) related auxiliary frameworks that enhance interface usability and error correction.

2.1 Fine-Tuning Large Pre-Trained Language Models

The rapid evolution of large-scale pre-trained language models (LLMs) has substantially improved the performance of NL2SQL systems by providing powerful language understanding capabilities that can be adapted to SQL generation tasks. Fine-tuning such massive models on domain-specific datasets has proven effective, but comes with high computational demands. To address this, researchers have developed parameter-efficient fine-tuning methods that update only a subset of model weights, thereby reducing resource requirements while maintaining or improving accuracy.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May - June 2025 ||

DOI:10.15680/IJARETY.2025.1203098

Butraction Tuning of LLMs: A notable example is the work by Stinivasan et al. (2023), who introduced "Butraction Tuning," a novel approach that strategically fine-tunes a small fraction of the model parameters. This method achieves competitive results on both question-answering and code generation benchmarks, demonstrating its utility for NL2SQL tasks where computational efficiency is crucial [1].

Unified Text-to-Text Framework: Another influential contribution comes from Raffel et al. (2020), who showcased the power of a unified text-to-text Transformer model, T5, that treats all tasks—including SQL query generation—as text generation problems. This paradigm simplifies model design and training pipelines, while delivering state-of-the-art results on a variety of benchmarks by leveraging transfer learning and large-scale pretraining [9].

These approaches underscore the importance of lightweight fine-tuning and unified modeling frameworks for effective and scalable NL2SQL systems, enabling broader accessibility without sacrificing performance.

2.2 Knowledge Augmentation

Augmenting pre-trained language models with external structured or unstructured knowledge has emerged as a promising strategy to enhance their understanding of complex domain-specific schemas and terminology, which is particularly valuable in NL2SQL applications. By embedding knowledge about database schemas and relationships directly into the model, these techniques help disambiguate user queries and improve query accuracy.

- Knowledge-Augmented Approaches: Zhu et al. (2022) provide a comprehensive survey of knowledge-augmentation methods for language models, including entity embeddings, memory-augmented neural networks, and schema-aware embeddings. These strategies enable models to incorporate factual grounding and contextual awareness, which is crucial for NL2SQL systems that must accurately map natural language questions to database structures and contents. For instance, schema embeddings can represent tables and columns in a continuous vector space, allowing the model to better link question tokens to relevant schema elements [2].

This body of work highlights the benefits of integrating domain-specific knowledge directly into the model architecture, thereby enhancing both generalization to new schemas and performance on specialized tasks.

2.3 Text-to-SQL–Specific Architectures

Moving beyond general-purpose text generation, several NL2SQL models have been explicitly designed to capture the structural nuances of relational databases. These architectures often incorporate graph-based representations and schema-linking mechanisms to model the relationships between tables, columns, and foreign keys, which are essential for generating complex SQL queries involving joins and nested operations.

- Line Graph Enhanced SQL Model (LGESQL): Cao et al. (2021) introduced LGESQL, a model that represents database schemas as line graphs. This representation encodes both local relationships (such as column-to-table mappings) and non-local dependencies (like foreign key constraints) more effectively than flat schema embeddings. By doing so, LGESQL improves the prediction of join operations and the generation of nested queries, which are common challenges in NL2SQL tasks [3].
- Deep Transformer Architectures**:** Wong et al. (2021) developed a deep transformer model with 75 layers specifically tailored for SQL generation. This architecture employs cross-attention mechanisms that jointly attend to the question and schema tokens, allowing the model to precisely align user queries with database schema components. This depth and architectural sophistication result in competitive performance on challenging datasets such as Spider, which features complex and diverse schemas [6].

These specialized designs demonstrate the importance of incorporating schema structure and relational information explicitly into NL2SQL models to enhance their ability to generate accurate and semantically valid SQL queries.

2.4 Multi-Task and Transfer Learning

Viewing NL2SQL as part of a broader ecosystem of natural language processing tasks has motivated the application of multi-task and transfer learning techniques. By training models simultaneously on multiple related tasks, or by transferring knowledge from one domain to another, NL2SQL systems can benefit from shared representations that improve their robustness and generalization capabilities.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May - June 2025 ||

DOI:10.15680/IJARETY.2025.1203098

- dec aNLP (NLP Decathlon): McCann et al. (2020) introduced decaNLP, a framework that formulates ten diverse NLP tasks—including semantic parsing and question answering—as question–answering problems. This multi-task training strategy encourages the model to learn generalizable language understanding skills that transfer well to NL2SQL, yielding more robust performance across varied tasks and datasets [4].
- Survey of Multi-Task Learning Techniques**:** Chen et al. (2021) provide an extensive review of multi-task learning methodologies, such as soft parameter sharing and dynamic task routing. These techniques enable models to jointly learn from complementary tasks like named-entity recognition and relation extraction, which are highly relevant to NL2SQL as they improve schema linking and entity disambiguation during SQL query generation [5].

By leveraging multi-task and transfer learning, NL2SQL models gain enhanced flexibility and the ability to better generalize to new database schemas and query types.

2.5 Related Auxiliary Frameworks

In addition to core NL2SQL techniques, several auxiliary systems contribute valuable insights and tools for improving user interaction, error handling, and semantic correctness in natural language interfaces for databases.

- Semantic Error Correction**:** Naz et al. (2021) propose a weighted federated learning approach to automatically detect and correct semantic errors in English text. Such methods have direct applicability to post-processing generated SQL queries, enabling the identification and correction of logical inconsistencies that might arise during automated query generation [7].
- Agent-Based Natural Language Interfaces: Ekpenyong et al. (2020) introduce an agent-based framework that acts as an intermediary between user intent and database queries. This framework includes dialog management capabilities for interactive clarification, which is particularly useful in scenarios where user queries are ambiguous or incomplete—a common challenge in NL2SQL systems [8].

These auxiliary frameworks enrich the NL2SQL ecosystem by addressing practical challenges related to error mitigation and user interaction, paving the way for more robust and user-friendly database query interfaces.

III. PROBLEM STATEMENT

Relational databases form the foundational infrastructure for managing structured data across a wide variety of sectors, including finance, healthcare, e-commerce, and scientific research. Despite their ubiquity and importance, interacting with these databases often demands a solid grasp of SQL syntax and familiarity with the specific database schema. This expertise requirement creates a significant barrier that limits access to critical data insights for many users, particularly domain experts and decision-makers who may lack formal training in database query languages. Consequently, valuable organizational data remains underutilized, impeding timely and informed decision-making.

IV. OBJECTIVES

The primary goals of this work are:

i. To develop a system capable of converting natural, everyday language queries into accurate SQL statements, thereby simplifying data retrieval from relational databases.

ii. To design a user-friendly interface that facilitates seamless interaction with the system, enabling users to query databases intuitively and efficiently without technical expertise.

iii. To empower users to obtain answers from databases through natural language input, eliminating the need to learn complex query languages and lowering the barrier to data access.

Existing System :

A range of text-to-SQL systems has been proposed to bridge the gap between natural language and database queries, aiming to simplify how users retrieve information from relational databases. Early attempts predominantly relied on template-based approaches, which matched user inputs against a set of predefined query patterns. While straightforward, these systems struggled to handle the vast diversity and variability of natural language queries, especially when faced with complex or unforeseen question structures. Subsequent rule-based systems sought to improve flexibility by incorporating linguistic rules and heuristics to interpret user intent. Despite offering enhanced adaptability, these methods often encountered difficulties dealing with ambiguous phrasing and intricate database schemas involving multiple tables and nested queries.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May - June 2025 ||

DOI:10.15680/IJARETY.2025.1203098

The emergence of neural network-based models marked a significant advancement. Sequence-to-sequence (Seq2Seq) architectures and transformer models such as BERT and GPT brought a deeper contextual understanding, enabling more accurate mapping from natural language questions to SQL queries. Specialized models like SQLNet and TypeSQL further advanced this capability by integrating explicit schema information, helping to resolve ambiguities related to table and column references. Hybrid approaches have combined the strengths of neural networks with rule-based logic to enhance reliability and precision, particularly in challenging query scenarios. Additionally, interactive systems that engage users in real-time through clarifying questions have been developed to iteratively refine query accuracy.

On the commercial front, platforms like Microsoft Power BI and Google BigQuery have incorporated text-to-SQL features to provide end-users with accessible data querying tools. However, these solutions may face limitations in generalizability and customization when applied to complex or domain-specific databases.

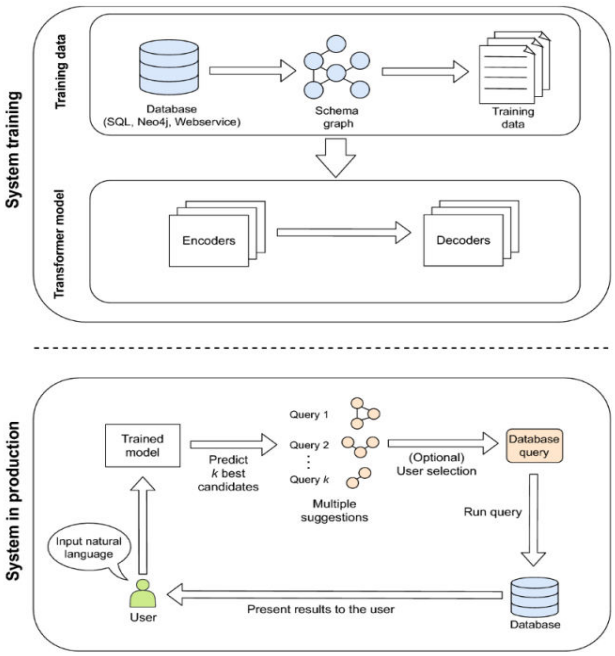
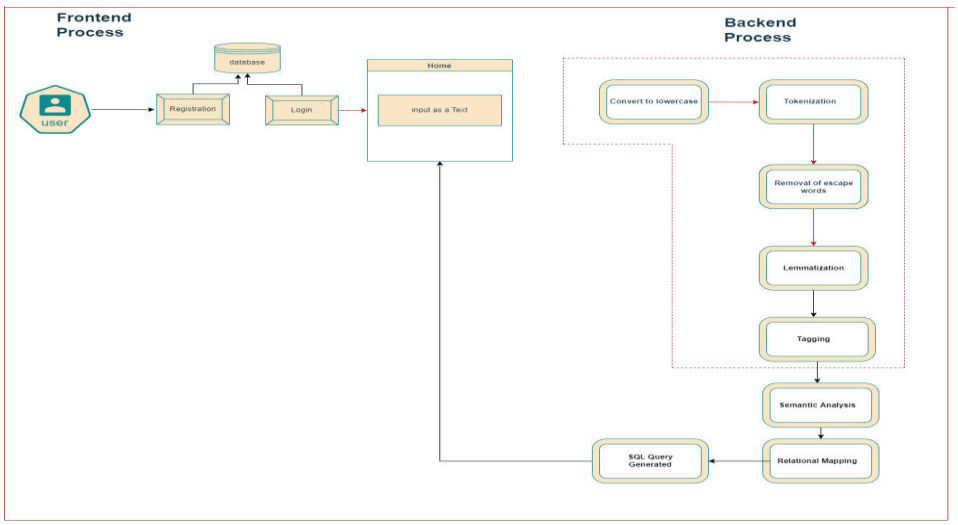


fig1: Existing System

Proposed System



International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May - June 2025 ||

DOI:10.15680/IJARETY.2025.1203098

System Workflow

I. **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.

II. **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.

III. **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.

IV. **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.

V. **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

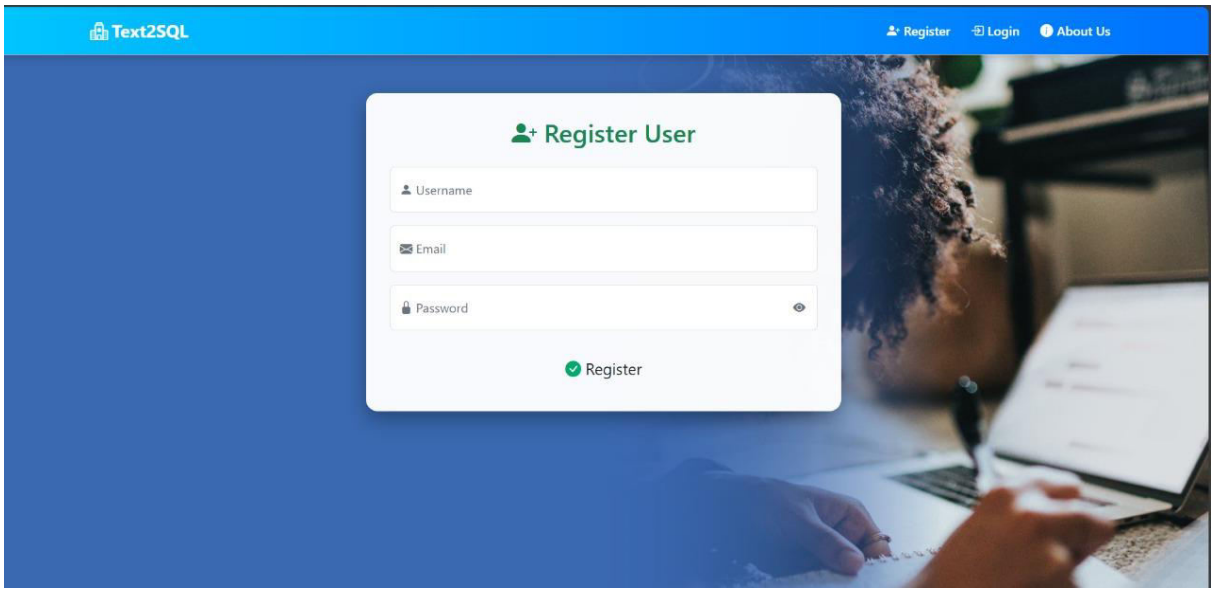


Fig Login/Register

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May - June 2025 ||

DOI:10.15680/IJARETY.2025.1203098

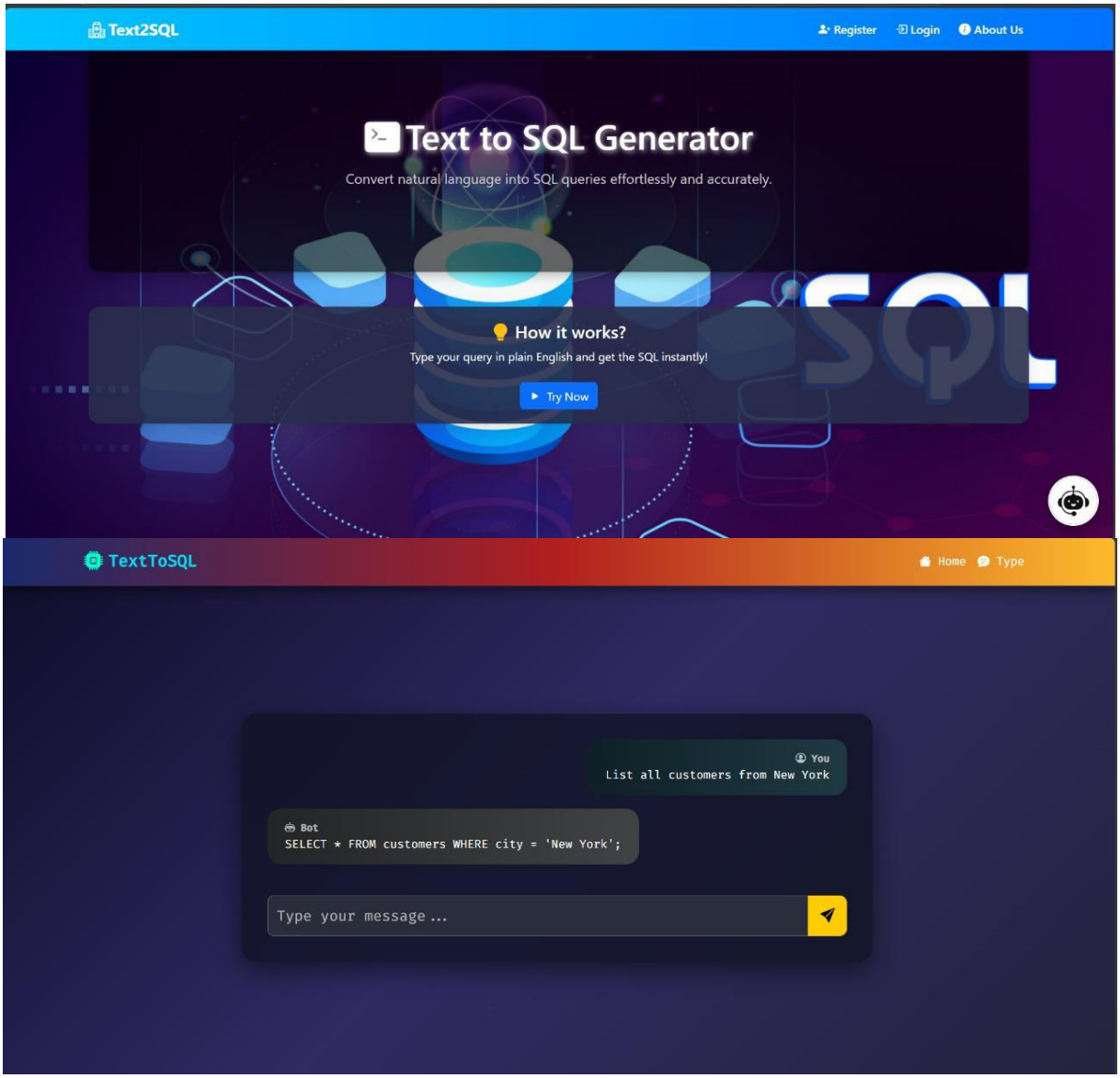


Fig Output

VI. FUTURE WORK & CONCLUSION:

This project presents a comprehensive end-to-end Text-to-SQL framework designed to empower users without technical expertise to interact effortlessly with relational databases using natural language. By leveraging parameter-efficient fine-tuning of a pre-trained transformer model, combined with schema-aware graph encoding and type-constrained decoding strategies, our system delivers high accuracy across benchmark datasets while maintaining the low latency necessary for real-time applications. To further improve robustness and user experience, an interactive clarification agent resolves ambiguities before query execution, ensuring more precise results. Collectively, these components significantly reduce the barrier to data access, enabling professionals in sectors such as finance, healthcare, and education to retrieve and visualize structured information without the need to write any SQL code.

Future Scope

1. Conversational Memory: Enable multi-turn dialogues to maintain context across queries for natural interaction.
2. Multimodal Inputs: Add voice and diagram recognition to support diverse query formats.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 3, May - June 2025 ||

DOI:10.15680/IJARETY.2025.1203098

3. Cross-Database & Multilingual Support: Extend compatibility to NoSQL/graph databases and non-English languages.
4. Explainability Tools: Provide visualizations and clear explanations for query generation and errors.
5. Continual Learning: Incorporate user feedback for ongoing system improvement.
6. BI Integration: Embed Text-to-SQL into business intelligence platforms for seamless analytics workflows.

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 Advanced Research in Education and Technology

ISSN: 2394-2975 Impact Factor: 8.152

www.ijarety.in editor.ijarety@gmail.com