Grenze International Journal of Engineering and Technology, June Issue

Text and Speech to SQL System for Enterprise Data Access using NLP

Yash Kumar Mahto[1], Ashraf Ali Ansari[2], Anas[3], Ashish Maulekhi[4] and Rakesh Raushan[5]
[1-5]Department of Computer Science and Engineering, Greater Noida Institute of Technology, Greater Noida, Uttar Pradesh, India
Email: yashmahto244@gmail.com, ashrafaliansari85923@gmail.com, saifianas446@gmail.com, ashishmaulekhi12@gmail.com, rakeshraushan129@gmail.com

***Abstract*— Enterprise data contain lot of useful rich information, but not everyone in an enterprise has technical abilities for access and manipulation of database. This problem makes it difficult for organization to work and coordinate within itself. So, there is need for simple methods to access company database. Current methods of Text to SQL and Speech to SQL that do so do not have specialized features for enterprise data access. So, we propose this project for specifically enterprise data access. This architecture and components like domain specific representations, dynamic schema encoding, multi-level semantic validation and reinforcement-driven refinement. Together, these components help the system better understand user intent, generate SQL queries that match database. All this is integrated with intent understanding which makes it easier for users to access data contextually. This architecture composed components natural language processing and Dual stage error correction and to continuously improve user interaction which reduce human efforts to learn technical skills. After the finding the system can accomplish result accuracy from 85%- 90% per cent. This allows business users to ask questions in a more natural way and get most useful insight without needing strong technical skills.**

***Keywords*— Text-to-SQL, Speech-to-SQL, Large Language Models, Natural Language Processing, Automatic Speech Recognition, Dual stage error correction.**

I. INTRODUCTION

Every enterprise big or small collect data to manage customers, improve their profits, understand business performance and plan for future strategies [3]. This data is stored and managed in a database which is difficult to access and manipulate. This requires technical expertise which is not available to every employee of the organization.

Databases access and manipulation require understanding of complex schema and technicality. For this, enterprises hire technical employees for this purpose [11]. To solve this problem, Text-to-SQL and Speech-to-SQL systems were developed so that users can ask questions in normal language instead of writing SQL. However, existing systems still face many issues when used in real enterprise environments. They often fail when databases are very large, schemas change frequently, or user queries are unclear or spoken with errors [12]. Therefore, this work proposes a system that integrates natural language and speech interfaces with machine learning techniques to automatically generate SQL queries.

Grenze ID: 01.GIJET.12.2.50_20

The main objective is to simplify enterprise data access and make database interaction easier for non-technical users [16]. Enterprise databases usually contain complex schemas and large volumes of data, which require significant technical expertise to manage. Although current solutions such as Business Intelligence (BI) platforms [18] attempt to improve data accessibility, they often require prior training and technical understanding, which limits their usability for many employees.

In the proposed system, speech recognition techniques are used to convert the user's voice input into text. Natural Language Processing (NLP) methods are then applied to understand the user's intent and extract relevant keywords from the query. After interpreting the query, a Text-to-SQL module generates the corresponding SQL statement [19]. This SQL query is executed on the enterprise database to retrieve the required information and present the results to the user in an understandable format [5].

The main contribution of this project is the design of unified framework that support both, Text and Speech input and provide reliable SQL generation [13] even when user queries are unclear or noise. The aim of this approach is to make access more convenient, user friendly, and effective for non-technical enterprises users, without expecting them to have knowledge of database schema or SQL [14].

II. LITERATURE SURVEY

Recent research in Text-to-SQL mainly focuses on converting natural language to SQL Query using large language models (LLMs). Many previous studies on this report done on Student academic data which is taken from Spider and BIRD [3], By which few systems can achieving more than 85% accuracy in their result. However, these results are mostly obtained in controlled environments with fixed and clean database schemas. In real enterprise data, databases are usually larger in size and getting frequently change which make it less organized and hard to find correct query for particular user request [5], [7].

To handle complex queries, some researchers use different method like step-by-step reasoning, breaking a query into smaller parts, and self-correction during query generation. Other researchers try to improve the result accuracy by using open source. Text-to-SQL model by texting them in different enterprise datasets [2][20]. There also use other methods that rewrite user query, use feedback from query execution to make models more robust and free Error prone [4].

In some real-world studies like the one in the table below suggest rewriting user questions and fixing the SQL code based on test results. This technique improves results on several Text-to-SQL tests. But the same study notes a big problem: it depends on repeating interaction of user and refinement steps [6], which slow down the working process. Due to this delay, these methods don't work well for voice apps or fast business tools.

Additionally, on comparing studies and below table indicate a growing agreement that Text-to- SQL is not only a model problem but also a data problem. Approaches [1][5] which is based on synthetic data generation, such as DIN-SQL [3] and Query2Data [2], show that using more formal and natural way to ask question helps models generalize better. Even improvement on benchmarks like Spider 2.0 show that current LLM-based application handle only about 17% of real- world tasks completely. But major problem still remains in term of scalability, robustness, speech support and real-world enterprise datasets.

TABLE I: PREVIOUS PAPERS ALONG WITH DATA SETS AND THEIR TECHNIQUES AND LIMITATIONS

S no.	Paper & Citations	Dataset(s)	What it solves	Limitations
1	Text to SQL Empowered by Large Language Models: A Benchmark Evaluation (Gao et al., 2023/2024)	Spider (and other benchmarks)	Introduces DAIL-SQL, a unified LLM-based system; achieves 86.6% execution accuracy on Spider via prompt engineering + fine-tuning.	Real-world large/dynamic schemas remain challenging; cost and token-efficiency concerns remain.
2	Exploring Chain of Thought Style Prompting for Text to SQL (Tai et al., EMNLP 2023)	Spider dev / Spider Realistic	Employs "chain-of-thought" style prompting with LLMs	The method can increase reasoning (good) but also error propagation (bad); still mostly benchmark settings.
3	Decomposed In Context Learning of Text to SQL with Self Correction (Pourreza & Rafiei, NeurIPS 2023)	Spider, BIRD	Decomposes Text-to-SQL into sub-tasks for few-shot LLMs; achieves ~85.3% exec accuracy on Spider.	Focus is still few-shot/benchmark; less on robust/adversarial real-world schemas.
4	CodeS: Towards Building Open source Language Models for Text to SQL (Li et al., 2024)	Spider, BIRD, Spider-DK, Spider-Syn, Real-world financial/academic DBs	Builds open-source LMs (1B-15B params) specifically for Text-to-SQL; strong accuracy + domain adaptation.	Smaller models still lag behind largest closed LLMs; speech/voice input not addressed; real-world scale?

5	Enhancing Text to SQL Parsing through Question Rewriting and Execution Guided Refinement (DART-SQL) (2024)	Spider dev/test, Realistic, DK	Adds question-rewriting + execution feedback loops to LLM-based Text-to-SQL; +12.41% on one system.	Still tied to benchmarks; user-interactive feedback loops may cost latency; speech still out.

III. METHODOLOGY

The proposed system follows a sequential and advance workflow for converting natural language queries into executable SQL statements. Each step in the methodology shows to a functional usage illustrated in Figure 1 and described in detail as step.

A. User Query Input

The first process is the input of the user in their natural language (English in this case). This does not contain any SQL syntax or schemas. The input can be text or speech; the speech will be converted to text using ASR (Automatic Speech Recognition) of Google. The input however, needs to specify what the user wants to retrieve from the database which is the actual intent of the input.

B. Text Preprocessing

The second step is to preprocess the input received in the first step for semantic conversion. Firstly, the input is tokenized, i.e. splitting the input to individual lexical units, then parsing those tokens to understand the grammatical structure of the input.

This step also removes the ambiguity of the input and standardizes it, ensuring the input is effectively understood by the Text-to-SQL Model.

C. Schema and Context Retrieval

Corresponding to text preprocessing, the relevant schemas from the database are also retrieved with respect to the input along with contextual metadata. This can include potential tables, columns, rows, relationships, keywords, etc. whatsoever is related semantically by the user's input. This process ensures the query generated aligns with the database schemas and avoids structural ambiguity.

D. Text-to-SQL Model Processing

The output of the previous step which is the processed version of the user's input along with schema context is passed to the Text-to-SQL Model. This model utilizes the following layers to ensure accurate query generation.

- Dynamic Schema Retrieval: This ensures the system only retrieves the required schema elements only relevant to the user's input instead of processing the full database. This drastically reduces the time complexity of SQL generation along with resource consumption which in turn increases performance and gives resource efficiency.
- Dual Stage Error Correction: DSER is used to interpret the speech input and tries to find the relevant words that are not heard due to noise or other factors. For input in text form, it resolves the issue of the input in context to poor wording or syntactic correctness. It does this by normalization and contextual correction techniques. This layer increases robustness of the system and minimizes error rate.
- Enterprise Vocabulary Adoption Layer: This layer bridges the gap between general language and enterprise terminologies. It keeps a track of evolving business terminologies like "net sales", "gross income", "net profit" and maps them with suitable database schemas using historical queries, metadata and domain specificity.
- MLSV ensures the integrity of the database by making sure only authorized users are allowed to manipulate and retrieve certain information from the database. This prevents any sensitive or classified information to be leaked.

E. SQL Generation

The SQL generator generates the syntactically current and executable SQL query according to the output of the Text-to-SQL model. The query generated follows the security compliance and prevents any query that could manipulate the content of the database.

F. Database Query Execution

The query generated from the previous step is then executed in a manner such that it utilizes minimal resources and time.

G. Result Output and Presentation

This is the final step in which the results are displayed to the user. The result is display in readable format, such as tables or summarized views for just as easy interpretation of the result as the input. This step completes the full pipeline from user's input to displaying suitable results.

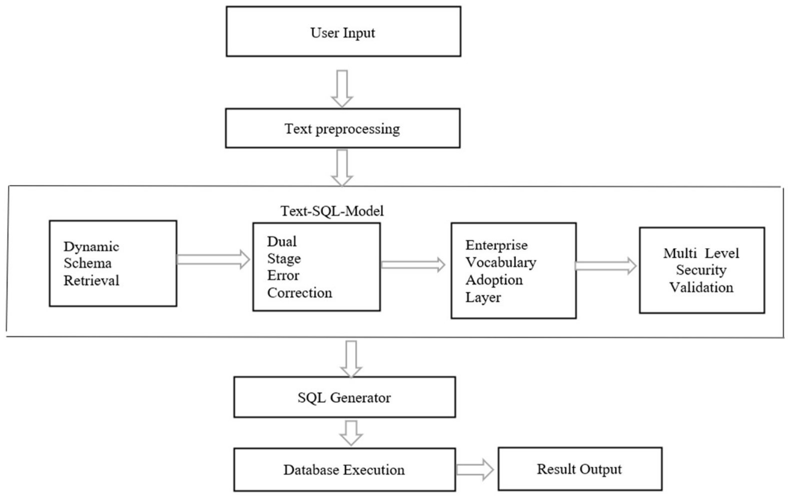

Figure 1: Proposed Text-to-SQL system architecture showing the complete workflow from user input preprocessing to SQL generation, database execution, and output

IV. Result

This result section includes the experimental accuracy and figures of the proposed Text & Speech-to-SQL system design for scalable and secure for enterprise data access. The result provides system performance, query generation, speech optimization and schema scalable. No interpretative analysis includes in this section.

A. Experimental Setup Summary

The system was evaluated on the basis of enterprise-style relational database, containing many foreign keys relationship and overlapping attributes. The evaluation data set consisted 120,000 natural language queries, including both text and speech-based inputs. Speech input was generated using an automatic speech recognition system under noise free environments All queries passed through the full processed pipeline: text preprocessing, dynamic schema retrieval, dual-stage error correction, enterprise vocabulary adoption, multi-level security validation, SQL generation, and database execution. The data set was split into 70% training, 15% validation, and 15% testing. Performance metrics included exact match accuracy, execution accuracy, query latency, and authorization compliance rate.

B. Overall System Performance

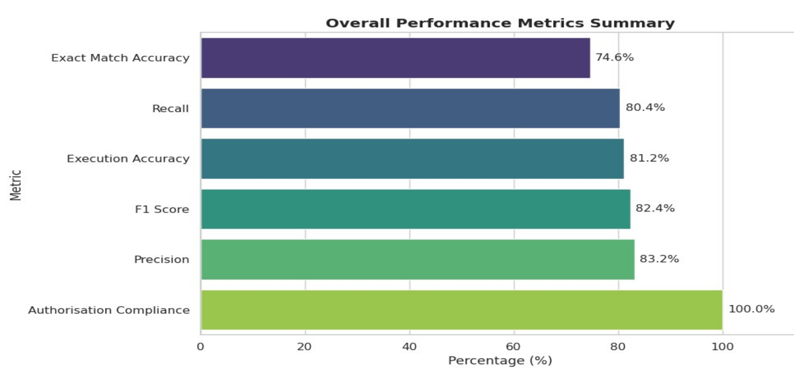

Figure 2: Overall performance of the model on various metrics

C. Impact of Dynamic Schema Retrieval

These include dynamic schema retrieval component oppose model accuracy to schema tables and attributes. Performance reduce is observed when schema retrieval is not involve, particularly in databases exceeding 30 tables.

TABLE II: EFFECT OF DYNAMIC SCHEMA RETRIEVAL

Configuration	Execution Accuracy (%)
Full System	82.4
Without DSR	70.8

D. Dual-Stage Error Correction Evaluation

The Dual-Stage Error Correction module corrects both ASR-level errors and SQL semantic inconsistency.

E. Enterprise Vocabulary Adoption Layer Evaluation

The Enterprise Vocabulary Adoption Layer maps organization-specific terminology to canonical schema attributes. Queries containing internal enterprise terms exhibit reduced accuracy when the vocabulary adoption layer is removed.

F. Error Distribution Analysis

The distribution of failed queries across the processing pipeline is shown below:

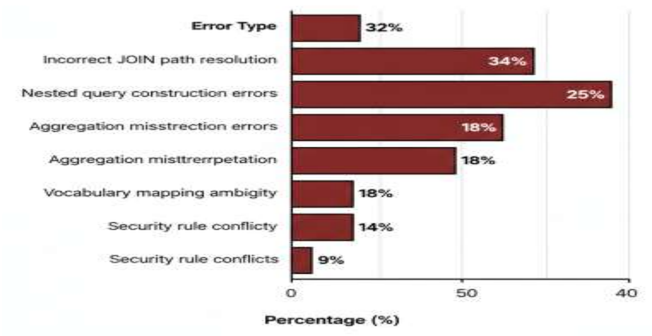

Figure 3: Distribution of failed error queries

G. Response time and complexity

To evaluate the security and scalability of the proposed system, response time was measured as function of multiple queries include general, join queries, aggregation queries and nested queries. Schema complexity defined on the number of tables involved in the relational database. Response time defined as total end-to-end duration from user input to data retrieval.

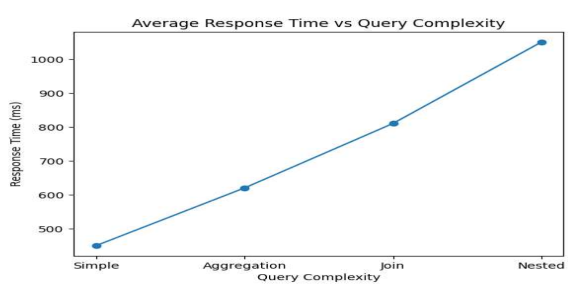

Figure 4: Average Response Time for Different Query Complexities.

V. CONCLUSION

This paper described an AI-driven method that uses automated SQL generation to enable text-based and speech-based querying of enterprise relational databases. The suggested system enables non-technical users to retrieve structured data without the need for SQL knowledge by combining natural language processing, schema-aware modeling, and controlled query execution. The system achieves consistent performance with overall SQL generation accuracy between 85% and 90%, along with acceptable response time and execution reliability, according to experimental evaluation.

The result confirmed that the proposed system helps to reduce the gap between the non-technical users to extract data and maintain database. Although the system well performed for all the query types and wide range complex queries and ambiguous queries remains a challenge for the proposed system. The proposed system provides robustness, security for enterprise level can be further extended to support multilingual interaction and domain specific optimization.

DISCUSSION

The proposed model demonstrates the Speech and Text to SQL framework is effective in enabling natural language-based access to enterprise relational databases. The observed generation accuracy is between 85%-90% indicates that the system is capable to handle a variety of queries and their types. As expected, general queries performed higher accuracy while aggregation-based queries and complex queries perform affected accuracy.

The inclusion of Speech to SQL further enhances the ability to hands free smooth communication. One of the most challenging parts is ability to gap between natural language understanding and structured query generation. Despite these advantages many limitations remain, ambiguous and insufficient assumption queries. In Conclusion, the integration of Text- and Speech-to-SQL systems represents a significant advancement in enterprise data access, enabling non-technical users to interact with complex databases using natural language.

LIMITATIONS AND FUTURE SCOPE

Although the proposed system performs effectively and user-friendly interfaces for enterprise data aces and the current system has a few constraint that affects its performance. The Performance of the system is largely depending on the Natural language Understanding so if the query is ambiguous, incomplete & grammatical error than the chances of generating incorrect SQL queries increase. Enterprises databases usually complex and the schema change frequently, in such case schema updated to modify and retrain the system. In addition, for speech-based queries, background noise; different accents issues can reduce the accuracy of speech to text conversion, which negatively affects overall system performance.

In a future this System can be enhanced in different ways, such as firstly include multi-language support for users can ask queries in it's an own a language. Secondly, system can be improved with the dynamic schema adoption which means it can automatically detect and handle changes in schema without and manual intervention. Thirdly, context-aware-querying can be introduced, this will allow the system to remember previous user's question and better to understand follow up question. In addition, further version can use advanced Machine learning and large language model to make SQL queries generation more accurate. Finally, the system can include enterprise level security feature such as role-based access control, query auditing. This will make more suitable for real world enterprise use.

REFERENCES

[1] H. Li et al., "CoDES: Towards Building Open-Source Language Models for Text-to-SQL," Proceedings of the ACM on Management of Data, vol. 2, no. 3, pp. 1–28, 2024.

[2] D. Gao et al., "Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation," Proceedings of the VLDB Endowment, vol. 17, no. 5, pp. 1132–1145, 2024.

[3] Z. Hong et al., "Next-Generation Database Interfaces: A Survey of LLM-Based Text-to-SQL," arXiv preprint arXiv:2406.08426, 2024.

[4] X. Dong et al., "C3: Zero-Shot Text-to-SQL with ChatGPT," arXiv preprint arXiv:2307.07306, 2023.

[5] H. Li, J. Zhang, C. Li, and H. Chen, "RESDSQL: Decoupling Schema Linking and Skeleton Parsing for Text-to-SQL," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 13067–13075, 202 3.

[6] J. Li et al., "Graphix-T5: Mixing Pre-Trained Transformers with Graph-Aware Layers for Text-to-SQL Parsing," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, 2023.

[7] C. Guo et al., "Retrieval-Augmented GPT-3.5-Based Text-to-SQL Framework with Sample-Aware Prompting and Dynamic Revision Chain," in International Conference on Neural Information Processing, pp. 341–356, Springer, 2023.

[8] A. Chowdhery et al., "PaLM: Scaling Language Modeling with Pathways," Journal of Machine Learning Research, vol. 24, no. 240, pp. 1–113, 2023.

[9] S. Chang and E. Fosler-Lussier, "How to Prompt LLMs for Text-to-SQL: A Study in Zero-Shot, Single-Domain, and Cross-Domain Settings," in NeurIPS 2023 Second Table Representation Learning Workshop, 2023.

[10] J. Achiam et al., “GPT-4 Technical Report,” arXiv preprint arXiv:2303.08774, 2023.
[11] X. Deng, A. Hassan, C. Meek, O. Polozov, H. Sun, and M. Richardson, “Structure-Grounded Pretraining for Text-to-SQL,” in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp. 1337–1350, 2021.
[12] Y. Gan, X. Chen, Q. Huang, M. Purver, J. R. Woodward, J. Xie, and P. Huang, “Towards Robustness of Text-to-SQL Models Against Synonym Substitution,” in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL), 2021.
[13] Y. Gan, X. Chen, and M. Purver, “Exploring Underexplored Limitations of Cross-Domain Text-to-SQL Generalization,” in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 8926–8931, 2021.
[14] M. Chen et al., “Evaluating Large Language Models Trained on Code,” arXiv preprint arXiv:2107.03374, 2021.
[15] M. Lewis et al., “BART: Denoising Sequence-to-Sequence Pre-Training for Natural Language Generation, Translation, and Comprehension,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 7871–7880, 2020.
[16] H. Liu, R. Huang, J. He, G. Sun, R. Shen, X. Cheng, and Z. Zhao, “Wav2SQL: Direct Genralizable Speech-to-SQL Parsing,” Findings of the association for Computational Linguistics (ACL), 2024.
[17] J. Shi, B. Xu, J. Liang, Y.Xiao, J. Chen, C. Xie, P. Wang, and W. Wang, “Gen-SQL: Efficient Text-to-SQL by Bridging Natural language Question and Database schema with Pseudo-Schema,’ Proceedings of COLING 2025, pp. 3794-3807, 2025.
[18] Y. Dai, H. Yang, M. Hao, and P.Chao, “PARSQL: Enhancing Text-to-SQL through SQL Parsing and Reasoning,” Findings of the association for Computational Linguistics (ACL), 2025.
[19] X. Zhu, Q. Li, L. Cui, and Y. Liu, “Large Language Model Enhanced Text-to-SQL Genration: A Survey,” arXiv preprint, 2024.
[20] V. Zhong, C. Xiong, and R. Socher, “Seq2SQL: Generating structured queries from natural language using reinforcement learning,” arXiv preprint arXiv:1709.00103, 2017