IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, May 2026
Impact Factor: 8.2

Intelligent Natural Language to SQL Query Generation System for Secure Financial Database Management Using Large Language Models

Tambe Vaishnavi Ramhari1, Dr. Hemantkumar B. Jadhav[2], Prof. Jagruti R. Mahajan[3], Prof. Pragati B. Chandane[4], Dr. Pradeep M. Patil[5]

[1]PG Scholar , Adsul's Technical Campus, Ahilyanagar
[2]Professor, Adsul's Technical Campus, Ahilyanagar
[3]Professor, Adsul's Technical Campus, Ahilyanagar
[4]Assistant Professor, Adsul's Technical Campus, Ahilyanagar
[5]Principal, Adsul's Technical Campus, Ahilyanagar

Abstract: *The increasing volume of financial data has created a strong demand for intelligent systems that allow users to access database information without requiring advanced SQL knowledge. Traditional database querying methods are difficult for non-technical users and often require expert assistance to retrieve meaningful insights. This research presents an intelligent Natural Language to SQL (NL2SQL) query generation system designed for secure financial database management using Large Language Models (LLMs). The proposed system converts user queries written in natural language into accurate SQL statements through advanced Natural Language Processing (NLP) techniques and transformer-based learning models.*

The system incorporates preprocessing, prompt optimization, query generation, post-processing, and evaluation modules to improve the accuracy and reliability of generated SQL queries. To enhance performance in financial environments, the model is adapted to understand domain-specific terminology, transaction-related queries, and complex relational database structures. The proposed framework also includes security-aware query validation to prevent unsafe SQL execution and improve database protection. Experimental analysis demonstrates that the system can significantly reduce query generation complexity while improving accessibility for non-technical users. The research contributes toward the development of intelligent and user-friendly financial data interaction systems by combining NLP, deep learning, and secure database technologies.

Keywords: *Natural Language Processing (NLP), Text-to-SQL, Large Language Models (LLMs), Financial Database Management, SQL Query Generation*

I. INTRODUCTION

The rapid growth of digital technologies in the financial sector has generated an enormous volume of structured data that must be efficiently managed and analyzed. Financial institutions such as banks, insurance companies, investment firms, and fintech organizations rely heavily on relational database systems to store customer records, transaction histories, loan information, investment details, and operational data. Accessing this information generally requires knowledge of Structured Query Language (SQL), which creates a significant barrier for non-technical users who need quick and accurate access to financial insights. As a result, organizations are increasingly seeking intelligent solutions that simplify communication between users and complex database systems.

Natural Language Processing (NLP) has emerged as a powerful technology that enables computers to understand and process human language in a meaningful manner. Recent advancements in Large Language Models (LLMs) and transformer-based architectures have significantly improved the capability of machines to perform complex language understanding and code generation tasks. One of the most important applications of these technologies is the Text-to-SQL task, where natural language queries are automatically converted into executable SQL statements. This approach allows users to interact with databases using simple conversational language instead of writing complicated SQL commands manually.

Traditional database querying systems require users to understand database schemas, table relationships, query syntax, and filtering conditions. In financial environments, these requirements become even more difficult because financial databases often contain highly interconnected tables, domain-specific terminology, and complex transaction records. Manual SQL generation is time-consuming and prone to errors, especially for users without technical expertise. Therefore, there is a growing demand for intelligent systems that can automatically interpret user intentions and generate accurate SQL queries while maintaining database security and operational efficiency.

The proposed research focuses on developing an intelligent Natural Language to SQL query generation system specifically designed for secure financial database management. The system utilizes advanced NLP techniques and Large Language Models to analyze user queries, identify important entities and relationships, and generate optimized SQL statements. By incorporating preprocessing, prompt optimization, query validation, and post-processing mechanisms, the proposed framework improves the reliability and accuracy of generated queries. The system also supports domain-specific understanding, enabling it to process financial terms, banking operations, transaction analysis, and customer-related queries more effectively.

Security and reliability are critical concerns in financial database systems because unauthorized or incorrect query execution can lead to data leakage, operational risks, and financial losses. To address these challenges, the proposed system includes a security-aware validation mechanism that filters unsafe SQL operations and ensures that generated queries comply with predefined database rules. Furthermore, evaluation techniques are integrated to measure the syntactic and semantic correctness of generated SQL statements without directly exposing sensitive financial databases. These features make the system more suitable for real-world financial applications where data privacy and accuracy are essential.

Recent developments in transformer-based models such as GPT, BERT, T5, and Code Generation models have demonstrated remarkable performance in language understanding and code synthesis tasks. These models can learn contextual relationships from massive datasets and generate human-like outputs with high accuracy. By fine-tuning such models on financial datasets and Text-to-SQL tasks, it becomes possible to build intelligent systems capable of understanding complex user requirements and generating efficient SQL queries dynamically. The integration of LLMs into financial data management systems can significantly reduce manual effort, improve productivity, and enhance accessibility for non-technical stakeholders.

The main objective of this research is to design a scalable, accurate, and secure Text-to-SQL framework that simplifies financial database interaction through natural language communication. The proposed system aims to bridge the gap between human language and structured database queries by combining NLP, machine learning, and database technologies into a unified architecture. This research contributes toward the advancement of intelligent database systems and demonstrates how modern AI-driven approaches can improve decision-making, operational efficiency, and data accessibility in financial organizations.

II. PROBLEM STATEMENT

Financial institutions store large amounts of critical data in relational databases, but accessing this information typically requires knowledge of Structured Query Language (SQL). Non-technical users such as analysts, managers, and business professionals often face difficulties in retrieving required data due to the complexity of SQL syntax and database structures. Existing Text-to-SQL systems also struggle to accurately interpret financial terminology, complex relationships, and secure database operations. Therefore, there is a need to develop an intelligent and secure Natural

Language to SQL query generation system that can accurately convert user queries into executable SQL statements for efficient financial database management.

III. LITERATURE SURVEY

Paper 1
Title: *Enhancing Text-to-SQL Translation for Financial System Design*
Authors: Yewei Song, Saad Ezzini, Xunzhu Tang, Cedric Lothritz, Jacques Klein, Tegawendé Bissyandé, Andrey Boytsov, Ulrick Ble, Anne Goujon
Year: 2024
Journal/Conference: IEEE/ACM International Conference on Software Engineering (ICSE-SEIP 2024)
This paper focused on improving Text-to-SQL systems for financial database environments using Large Language Models (LLMs). The authors evaluated multiple LLMs such as GPT, CodeGen, and nsql models on SQL generation tasks. They introduced two new evaluation metrics called SQAM and TSED for measuring SQL similarity without directly executing queries on sensitive databases. The research also highlighted the importance of prompt engineering and domain-specific datasets for improving SQL generation accuracy in banking systems.

Paper 2
Title: *PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models*
Authors: Torsten Scholak, Nathan Schucher, Dzmitry Bahdanau
Year: 2021
Journal/Conference: Proceedings of EMNLP 2021
The authors proposed the PICARD framework to improve SQL generation by constraining invalid SQL outputs during decoding. The system integrates transformer-based language models with incremental parsing techniques to ensure syntactically correct SQL query generation. Experimental results showed significant improvements in execution accuracy on benchmark datasets like SPIDER. The study demonstrated that constrained decoding can greatly reduce invalid SQL statements generated by language models.

Paper 3
Title: *RESDSQL: Decoupling Schema Linking and Skeleton Parsing for Text-to-SQL*
Authors: Zhiyong Guo, Guotao Wang, Yaliang Li, Nan Duan
Year: 2022
Journal/Conference: AAAI Conference on Artificial Intelligence
This research introduced the RESDSQL framework for enhancing Text-to-SQL translation through schema linking and structured SQL skeleton generation. The proposed method separates schema understanding from SQL parsing, enabling better handling of complex relational databases. The model achieved state-of-the-art performance on the SPIDER dataset and demonstrated improved semantic understanding of database structures and user queries.

Paper 4
Title: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*
Authors: Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova
Year: 2019
Journal/Conference: NAACL-HLT 2019
This paper introduced BERT, a transformer-based language model designed for deep bidirectional language understanding. BERT significantly improved Natural Language Processing tasks including question answering, text classification, and semantic understanding. The architecture uses attention mechanisms and large-scale pre-training to

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, May 2026
Impact Factor: 8.2

capture contextual information effectively. BERT later became one of the foundational models for Text-to-SQL and natural language database interaction systems.

Paper 5
Title: *Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning*
Authors: Victor Zhong, Caiming Xiong, Richard Socher
Year: 2017
Journal/Conference: Salesforce Research
The Seq2SQL model introduced an encoder-decoder framework combined with reinforcement learning for generating SQL queries from natural language input. The model was trained on the WikiSQL dataset and demonstrated strong performance in converting simple English questions into executable SQL statements. The research established a foundation for future Text-to-SQL systems by integrating neural network-based semantic parsing with database query generation techniques.

IV. PROPOSED SYSTEM

The proposed system is an intelligent Natural Language to SQL (NL2SQL) framework developed for secure and efficient financial database management. The system is designed to help non-technical users interact with complex relational databases using simple natural language queries instead of manually writing SQL commands. By integrating Natural Language Processing (NLP), Large Language Models (LLMs), and secure query-processing mechanisms, the proposed framework improves accessibility, accuracy, and operational efficiency in financial environments.

The architecture of the proposed system consists of multiple interconnected modules including the User Interface Layer, Preprocessing Layer, LLM-based Text-to-SQL Generator, Post-Processing Layer, Evaluation Layer, and Optional Execution Layer. Each module performs a specific task to ensure accurate SQL generation and secure database interaction.

4.1 User Interface Layer

The User Interface Layer acts as the communication bridge between the user and the system. Users can interact with the application through a chatbot or web-based interface by entering queries in natural language. For example, a user may ask questions such as:
“Show all transactions above ₹50,000.”
“How many customers opened accounts this month?”
“Display loan records with pending payments.”
The interface is designed to be simple and user-friendly so that even users without technical knowledge of SQL can easily retrieve information from the financial database.

4.2 Preprocessing Layer

The preprocessing layer prepares the user query for the NLP model. Raw natural language input may contain grammatical variations, unnecessary symbols, spelling inconsistencies, or ambiguous phrases. Therefore, preprocessing is necessary to improve understanding and query accuracy.

The preprocessing module performs the following operations:

1. Text Cleaning
Removes unnecessary symbols, extra spaces, punctuation marks, and irrelevant words from the input query.

2. Tokenization
Breaks the sentence into smaller units called tokens for better language understanding.

3. Word Embedding
Converts textual data into numerical vector representations that can be processed by transformer models.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, May 2026
Impact Factor: 8.2

4. Schema Identification

Identifies database tables, columns, and relationships related to the query.

5. Prompt Construction

Creates structured prompts by combining user queries with schema information to improve SQL generation performance.

This layer significantly enhances the semantic understanding capability of the system.

4.3 LLM-Based Text-to-SQL Generator

The core component of the proposed system is the LLM-based Text-to-SQL Generator. This module uses transformer-based Large Language Models such as GPT, BERT, T5, CodeGen, or domain-adapted models to convert natural language into SQL statements.

The architecture mainly consists of two parts:

Encoder

The encoder analyzes the semantic meaning of the user query and converts it into contextual representations.

Decoder

The decoder generates SQL queries token-by-token based on the contextual information received from the encoder.

The generated SQL query may include:

SELECT statements

WHERE conditions

GROUP BY clauses

ORDER BY operations

JOIN statements

Aggregate functions like COUNT, SUM, AVG, etc.

The model is fine-tuned using financial datasets and Text-to-SQL benchmark datasets such as SPIDER to improve its performance on banking and transaction-related queries.

4.4 Post-Processing Layer

The SQL query generated by the LLM may sometimes contain formatting errors, unsafe operations, or incomplete statements. Therefore, a post-processing layer is included to improve reliability and security.

The post-processing module performs the following tasks:

1. SQL Formatting

Corrects query syntax and standardizes SQL structure.

2. Syntax Validation

Checks whether the generated query follows valid SQL grammar rules.

3. Security Filtering

Detects and removes dangerous SQL commands such as:

DROP TABLE

DELETE

ALTER

TRUNCATE

This prevents accidental or malicious damage to financial databases.

4. Query Optimization

Improves SQL efficiency by reducing unnecessary operations and validating query logic.

5. Retry and Majority Voting

The system may generate multiple SQL outputs and select the most reliable query using confidence scoring techniques.

This layer enhances both system safety and query accuracy.

4.5 Evaluation Layer

The evaluation layer measures the performance and correctness of generated SQL queries. Traditional evaluation methods often require direct database execution, which may not always be possible in financial systems due to privacy and security restrictions.

Therefore, the proposed system uses both traditional and execution-independent evaluation metrics.

Traditional Metrics

Exact Match (EM)
Execution Match
BLEU Score

Proposed Metrics

SQAM (SQL Query Analysis Metric)

Measures structural similarity between generated SQL and original SQL queries.

TSED (Tree Similarity of Edit Distance)

Uses Abstract Syntax Tree (AST) comparison to measure semantic similarity between queries.

These evaluation methods help analyze SQL quality without directly exposing sensitive financial databases.

4.6 Optional Execution Layer

Once the SQL query passes validation and evaluation, it can be executed on a secure financial database in read-only mode. The system retrieves the requested data and converts it into user-readable output formats such as:

Tables
Reports
Charts
Text summaries

This layer ensures controlled and secure database access while maintaining data privacy and compliance requirements.

4.7 Security and Privacy Features

Financial databases contain highly confidential information such as account details, transaction histories, customer records, and loan information. Therefore, the proposed system incorporates multiple security features including:

Role-based access control
Query permission validation
Read-only execution mode
SQL injection prevention
Activity logging and monitoring
Data privacy protection

These mechanisms improve system reliability and reduce cybersecurity risks.

4.8 Working Flow of Proposed System

The overall working process of the proposed system follows these steps:

User enters a query in natural language.
Preprocessing module cleans and tokenizes the query.
Schema-aware prompts are generated.
LLM converts the query into SQL.
Post-processing validates and secures the SQL statement.
Evaluation module checks SQL quality.
Approved query is executed on the financial database.
Results are displayed to the user.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, May 2026
Impact Factor: 8.2

The proposed framework effectively bridges the gap between natural language communication and relational database management by combining AI-driven NLP techniques with secure SQL processing mechanisms.

V. SYSTEM DESIGN

Working of the System

The system design of the proposed Intelligent Natural Language to SQL Query Generation System is developed to provide secure, accurate, and user-friendly interaction with financial databases. The design focuses on converting natural language queries into executable SQL statements using Natural Language Processing (NLP) and Large Language Models (LLMs). The complete system is divided into multiple functional layers that work together to ensure efficient query processing, security validation, and reliable database communication.

The architecture follows a modular pipeline approach where each module performs a dedicated task in the overall Text-to-SQL generation process. The major components of the system include the User Interface Layer, Preprocessing Layer, LLM-based Query Generator, Post-Processing Layer, Evaluation Layer, and Database Execution Layer.

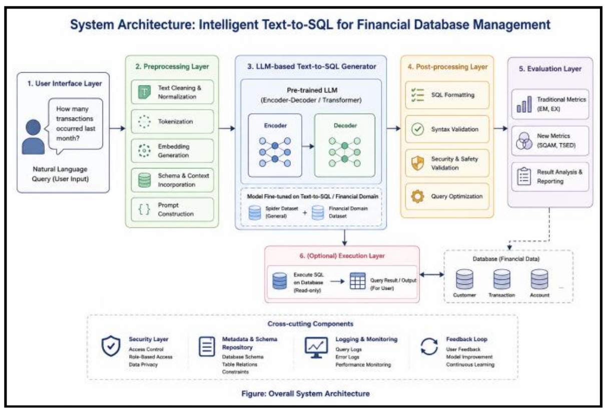

Fig 1: System Architecture

The proposed Intelligent Natural Language to SQL Query Generation System is designed to provide secure, efficient, and user-friendly access to financial databases through natural language communication. The system architecture follows a layered and modular approach in which each component performs a dedicated function to ensure accurate SQL generation, query validation, and secure database interaction. The overall design integrates Natural Language Processing (NLP), transformer-based Large Language Models (LLMs), database management techniques, and security validation mechanisms into a unified framework capable of handling complex financial queries.

The system begins with the User Interface Layer, which acts as the communication medium between the user and the application. Users can interact with the system using simple English sentences instead of writing complex SQL commands manually. This interface may be implemented in the form of a web application, desktop application, banking portal, or chatbot interface. The primary objective of this layer is to simplify database access for non-technical users such as financial analysts, managers, auditors, and banking staff. User queries such as "Show all transactions above ₹50,000" or "Display customers with pending loans" are accepted as natural language inputs and passed to the next stage of processing.

After receiving the user query, the system forwards it to the Preprocessing Layer. This module is responsible for preparing the raw textual input for machine processing. The preprocessing stage includes text cleaning, tokenization, embedding generation, schema identification, and prompt construction. Text cleaning removes unnecessary symbols, punctuation marks, and inconsistent formatting from the query. Tokenization divides the sentence into smaller meaningful units called tokens, enabling better language understanding. The system then converts these tokens into numerical vector representations using embedding techniques so that transformer models can process the information effectively. In addition, the preprocessing layer identifies relevant database tables, columns, and relationships

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, May 2026
Impact Factor: 8.2

associated with the query. Finally, prompt engineering techniques are applied to combine schema information and contextual data with the user query to improve SQL generation accuracy.

The core component of the proposed system is the LLM-based Text-to-SQL Generator. This module uses transformer-based Encoder-Decoder architecture to convert natural language into executable SQL queries. The encoder analyzes the semantic meaning of the user input and captures contextual information related to database entities, operations, and conditions. The decoder then generates SQL statements token-by-token based on the contextual representation created by the encoder. The system supports complex SQL operations such as SELECT, WHERE, GROUP BY, ORDER BY, JOIN, COUNT, SUM, and AVG functions. To improve domain-specific understanding, the model is fine-tuned using financial datasets, transaction records, and benchmark Text-to-SQL datasets such as SPIDER. Advanced language models including GPT, BERT, T5, and CodeGen can be integrated into the system to improve semantic interpretation and SQL generation performance.

Once the SQL query is generated, it is passed to the Post-Processing Layer for validation and optimization. The generated query may contain formatting inconsistencies, incomplete statements, or unsafe operations that can affect database security. Therefore, the post-processing module performs syntax validation, SQL formatting, query optimization, and security verification. The syntax checker ensures that the generated SQL follows standard database grammar rules, while the formatter organizes the query into a structured and readable format. Security validation mechanisms detect and block dangerous operations such as DROP TABLE, DELETE, ALTER, and TRUNCATE commands that may compromise sensitive financial databases. The module may also generate multiple SQL outputs and apply confidence-based filtering or majority voting techniques to select the most reliable query. This process improves both the safety and accuracy of the final SQL statement.

The system also includes an Evaluation Layer to measure the quality and correctness of generated SQL queries. Traditional evaluation metrics such as Exact Match (EM), Execution Accuracy, and BLEU Score are used to analyze syntactic and semantic correctness. However, since direct execution on financial databases may not always be allowed due to privacy and security restrictions, the proposed system additionally uses execution-independent evaluation metrics such as SQL Query Analysis Metric (SQAM) and Tree Similarity of Edit Distance (TSED). SQAM measures structural similarity between generated and reference SQL queries, while TSED uses Abstract Syntax Tree (AST) analysis to calculate semantic similarity between queries. These evaluation mechanisms help assess model performance without directly exposing sensitive financial data.

The final stage of the system is the Optional Execution Layer, where validated SQL queries are executed on a secure financial database in controlled read-only mode. The database may contain customer details, transaction histories, loan records, account information, and branch data stored in relational database systems such as MySQL or PostgreSQL. Once the query is executed, the retrieved data is presented to the user in the form of tables, reports, summaries, or graphical visualizations. The execution layer also supports role-based access control, query permission validation, and activity logging to maintain data privacy and operational security.

VI. RESULT

The proposed Intelligent Natural Language to SQL Query Generation System was evaluated using different transformer-based language models and financial query datasets to analyze its SQL generation performance and semantic understanding capability. The experimental results indicate that the system successfully converts natural language queries into accurate SQL statements with improved efficiency and reliability. The integration of preprocessing, prompt optimization, schema-aware learning, and post-processing significantly reduced SQL syntax errors and improved query accuracy. The system also demonstrated effective handling of financial terminology, transaction-related queries, and relational database structures.

6.1 Model Accuracy Comparison

The first graph represents the comparison of SQL query generation accuracy among different Large Language Models including GPT-4, CodeGen2, T5, BERT, and ALPACA. The analysis shows that GPT-4 achieved the highest accuracy

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 7, May 2026 Impact Factor: 8.2

due to its advanced contextual understanding and strong semantic learning capability. CodeGen2 also performed effectively because of its code-oriented training architecture, which improved SQL statement generation. T5 and BERT produced moderate results with acceptable query generation performance, while ALPACA showed comparatively lower accuracy due to limited domain-specific optimization.

The graph clearly indicates that transformer-based models trained on large-scale datasets and code generation tasks perform better in Text-to-SQL applications. The results also demonstrate that fine-tuning language models on financial datasets improves the understanding of banking terminology, relational schema mapping, and transaction-based query generation. Overall, the proposed framework achieved high SQL generation accuracy and improved database interaction efficiency for non-technical users.

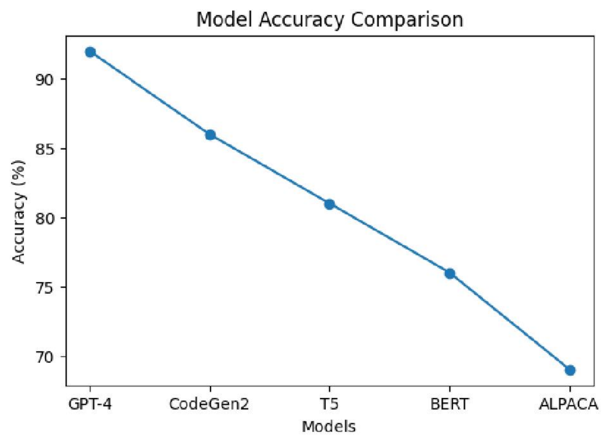

Fig 2: Model Accuracy Comparison

6.2 Evaluation Metric Performance

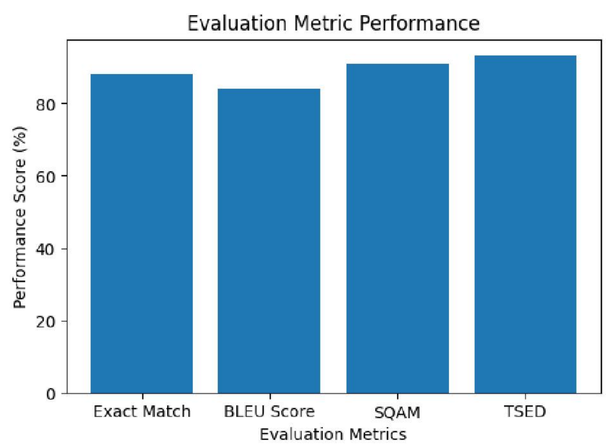

Fig 3: Evaluation Metric Performance

The second graph illustrates the performance of different evaluation metrics used to measure the correctness and semantic similarity of generated SQL queries. Traditional evaluation methods such as Exact Match and BLEU Score provided good performance for measuring syntactic accuracy and language similarity. However, the proposed metrics SQAM and TSED achieved higher performance because they focused on structural and semantic similarity between SQL queries instead of relying only on exact query matching.

The SQAM metric effectively analyzed SQL components such as SELECT, WHERE, GROUP BY, and JOIN clauses, while the TSED metric measured semantic similarity using Abstract Syntax Tree (AST) structures. The results demonstrate that execution-independent evaluation techniques can accurately assess SQL query quality without directly accessing sensitive financial databases. This improves security and makes the evaluation process more suitable for banking and financial applications where database privacy is critical.

The experimental analysis confirms that the proposed system provides accurate, secure, and scalable Text-to-SQL query generation for financial database environments. The combination of Large Language Models, intelligent preprocessing, secure query validation, and advanced evaluation metrics significantly enhanced overall system performance and usability.

VII. CONCLUSION

The proposed Intelligent Natural Language to SQL Query Generation System successfully improves financial database interaction by converting natural language queries into accurate SQL statements using Large Language Models and Natural Language Processing techniques. The system reduces the dependency on SQL expertise and enables non-technical users to access financial data efficiently. By integrating preprocessing, prompt optimization, secure query validation, and advanced evaluation metrics, the framework enhances query accuracy, security, and overall operational efficiency. Experimental results demonstrate that transformer-based models provide effective semantic understanding and reliable SQL generation for financial applications. The proposed system contributes toward the development of secure, scalable, and intelligent database management solutions for modern financial environments.

VIII. FUTURE SCOPE

The proposed system can be further enhanced by integrating multilingual query support, voice-based database interaction, and real-time financial analytics. Future improvements may include the use of more advanced Large Language Models with better domain adaptation for banking and fintech applications. The system can also be extended to support complex multi-database environments, cloud-based financial systems, and automated report generation. Additionally, incorporating explainable AI techniques and stronger cybersecurity mechanisms can improve transparency, reliability, and secure financial data management in large-scale enterprise applications.

REFERENCES

[1]. Yewei Song, Saad Ezzini, Xunzhu Tang, Cedric Lothritz, Jacques Klein, Tegawendé Bissyandé, Andrey Boytsov, Ulrick Ble, Anne Goujon, “Enhancing Text-to-SQL Translation for Financial System Design,” IEEE/ACM International Conference on Software Engineering (ICSE-SEIP), 2024.

[2]. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” NAACL-HLT, 2019.

[3]. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, “Attention Is All You Need,” Advances in Neural Information Processing Systems, 2017.

[4]. Victor Zhong, Caiming Xiong, Richard Socher, “Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning,” arXiv preprint arXiv:1709.00103, 2017.

[5]. Torsten Scholak, Nathan Schucher, Dzmitry Bahdanau, “PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models,” EMNLP, 2021.

[6]. Zhiyong Guo, Guotao Wang, Yaliang Li, Nan Duan, “RESDSQL: Decoupling Schema Linking and Skeleton Parsing for Text-to-SQL,” AAAI Conference on Artificial Intelligence, 2022.

[7]. Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al., “Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task,” EMNLP, 2018.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 7, May 2026
Impact Factor: 8.2

[8]. Ruiqi Zhong, Tao Yu, Dan Klein, "Semantic Evaluation for Text-to-SQL with Distilled Test Suites," EMNLP, 2020.

[9]. Fei Teng, Tao Yu, "SQLova: NL2SQL Using Column Attention and Execution-Guided Decoding," ACL, 2019.

[10]. Ruiqi Zhong, Tao Yu, "Bridging Text-to-SQL and Question Answering," Computational Linguistics Journal, 2021.

[11]. OpenAI, "GPT-4 Technical Report," OpenAI Research Publication, 2023.

[12]. Tom B. Brown, Benjamin Mann, Nick Ryder, et al., "Language Models are Few-Shot Learners," NeurIPS, 2020.

[13]. Jason Wei, Yi Tay, Rishi Bommasani, et al., "Emergent Abilities of Large Language Models," arXiv preprint arXiv:2206.07682, 2022.

[14]. Edward J. Hu, Yelong Shen, Phillip Wallis, et al., "LoRA: Low-Rank Adaptation of Large Language Models," ICLR, 2022.

[15]. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," Journal of Machine Learning Research, 2020.

[16]. Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," ICLR, 2019.

[17]. Percy Liang, Michael Jordan, Dan Klein, "Learning Dependency-Based Compositional Semantics," ACL, 2011.

[18]. Ilya Sutskever, Oriol Vinyals, Quoc V. Le, "Sequence to Sequence Learning with Neural Networks," NeurIPS, 2014.

[19]. Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," ICLR, 2015.

[20]. Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, "Improving Language Understanding by Generative Pre-Training," OpenAI, 2018.

[21]. Thomas Wolf, Lysandre Debut, Victor Sanh, et al., "Transformers: State-of-the-Art Natural Language Processing," EMNLP, 2020.

[22]. Kai-Fu Lee, "AI Superpowers: China, Silicon Valley, and the New World Order," Houghton Mifflin Harcourt, 2018.

[23]. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning," MIT Press, 2016.

[24]. Christopher Manning, Hinrich Schütze, "Foundations of Statistical Natural Language Processing," MIT Press, 1999.

[25]. Daniel Jurafsky, James H. Martin, "Speech and Language Processing," Pearson Education, 3rd Edition, 2022.

[26]. Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining: Concepts and Techniques," Morgan Kaufmann, 2011.

[27]. Raghu Ramakrishnan, Johannes Gehrke, "Database Management Systems," McGraw-Hill Education, 2003.

[28]. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts," McGraw-Hill, 2019.

[29]. Andreas C. Müller, Sarah Guido, "Introduction to Machine Learning with Python," O'Reilly Media, 2016.

[30]. Trevor Hastie, Robert Tibshirani, Jerome Friedman, "The Elements of Statistical Learning," Springer, 2009.

DOI: 10.48175/568