Journal of Computer and Communications, 2024, 12, 1-13
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Large Language Model Based Semantic Parsing for Intelligent Database Query Engine

Zhizhong Wu

College of Engineering, UC Berkeley, Berkeley, USA
Email: ecthelion.w@gmail.com

How to cite this paper: Wu, Z.Z. (2024) Large Language Model Based Semantic Parsing for Intelligent Database Query Engine. *Journal of Computer and Communications*, **12**, 1-13.
https://doi.org/10.4236/jcc.2024.1210001

Received: September 9, 2024
Accepted: October 6, 2024
Published: October 9, 2024

Abstract

With the rapid development of artificial intelligence, large language models (LLMs) have demonstrated remarkable capabilities in natural language understanding and generation. These models have great potential to enhance database query systems, enabling more intuitive and semantic query mechanisms. Our model leverages LLM's deep learning architecture to interpret and process natural language queries and translate them into accurate database queries. The system integrates an LLM-powered semantic parser that translates user input into structured queries that can be understood by the database management system. First, the user query is pre-processed, the text is normalized, and the ambiguity is removed. This is followed by semantic parsing, where the LLM interprets the pre-processed text and identifies key entities and relationships. This is followed by query generation, which converts the parsed information into a structured query format and tailors it to the target database schema. Finally, there is query execution and feedback, where the resulting query is executed on the database and the results are returned to the user. The system also provides feedback mechanisms to improve and optimize future query interpretations. By using advanced LLMs for model implementation and fine-tuning on diverse datasets, the experimental results show that the proposed method significantly improves the accuracy and usability of database queries, making data retrieval easy for users without specialized knowledge.

Keywords

Semantic Query, Large Language Models, Intelligent Database, Natural Language Processing

1. Introduction

The earliest known prototype of a database natural language query system was developed in the early 1970s. Despite years of development, this type of system remains underutilised in relational database management systems, with only a limited presence in specific database management systems. There is still a considerable distance to traverse before it can be considered a mainstream technology. The advent of deep semantic parsing technology has opened the door to the construction of a contemporary database natural language query system. Firstly, it is essential to guarantee the veracity and precision of the natural language translated into database query statements. In general, since structured queries represent the standard management language for relational databases, the transition from natural language to structured queries is a universal phenomenon [1]. However, the syntactic structure of structured query language is complex, and the structured query statements output by the existing deep semantic parsing technology cannot be guaranteed to be syntactically and architecturally legitimate. That is to say, the structured query statements predicted by the model can be executed, but it cannot be guaranteed that they will be consistent with the corresponding database architecture.

In some instances, the structured query predicted by the model is highly similar to the target structured query at the character level. However, despite this similarity, the predicted query may still fail to run correctly due to the presence of a limited number of syntax errors. The accuracy of the semantic understanding of the predicted structured query statement can only be further considered if the syntax of the predicted structured query statement is legitimate. Accordingly, the objective of this research is to enhance the semantic precision of the model while maintaining the efficacy of natural language conversion into structured query statements [2].

The rapid development of information technology and the advent of the era of big data have led to a significant increase in the use of database management systems across a range of industries. Nevertheless, the conventional Structured Query Language (SQL) necessitates a certain degree of technical expertise and professional knowledge on the part of the user, which serves to restrict the popularity and application of database systems to a certain extent.

In parallel, we also see rapid development in Machine Learning [3]-[5] and Artificial Intelligence [6] gaining ground in more and more applications. Computer Vision has been widely applied in autonomous driving, medicine [7] [8], and industrial automation, where it powers tasks such as object detection [9], anomaly recognition, and quality control. Natural Language Processing [10] is applied in social media [11], customer service [12], publishing and marketing. The same applies in the fields of time series prediction [12] [13], recommender system [14], etc.

In the field of database, NLP technology has been incorporated into database query systems, allowing users to query data in their native language [15]. Nevertheless, there remain significant challenges in semantic understanding and query statement generation in existing natural language query systems.

The current deep semantic parsing technology is unable to construct a fully accurate model that translates natural language into structured query language, resulting in instances of inaccurate semantic comprehension. As previously stated, the database natural language system can enhance system stability by employing the inverse process of structured query statements for natural language generation [16]. The crucial objective is to establish an effective methodology for the validation of structured query statements, and to develop an assessment framework for evaluating multiple potential query statements.

Moreover, the utilization of reverse technology can also address another challenge inherent to the construction of a database natural language query system. In the event that a user inputs a natural language query into the database query system, the system will return the underlying database query result, which the user is unable to assess for accuracy. Even if the corresponding structured query statement is returned, the structured query statement output by the parsing model is unintelligible to users lacking the required computer or database operation background [17].

The essential function of a manual interaction application is the receipt of user feedback. The progressive feedback loop between the system and the user serves to enhance the system's semantic understanding ability, thereby strengthening the system's overall intelligence. Conversely, user feedback serves to compensate for the model's deficiency in conversion capabilities to a certain extent [18]. Accordingly, this paper proposes to address the issue of converting structured query statements into natural language by examining the inverse process of deep semantic parsing. This approach aims to enhance the feedback functionality of users and database natural language query systems.

In recent years, there have been notable advancements in the field of deep learning-based natural language processing technology, particularly with the advent of large language models (LLMs), such as GPT-4, which have exhibited impressive capabilities in language understanding and generation. This offers a novel opportunity for the enhancement of intelligent database semantic queries [19]. The combination of a large language model with a database query system has the potential to enhance the system's semantic parsing abilities and query accuracy, thereby offering users a more natural and efficient query experience.

This paper presents a methodology for enhancing the semantic capabilities of intelligent databases through the utilisation of large language models. The objective is to address the limitations of traditional database natural language query systems, namely the lack of precision in semantic understanding and the generation of non-standardised query statements. We put forth a novel system architecture that employs large language models for deep semantic parsing and reverse verification, with the objective of enhancing the system's stability and accuracy. Additionally, the user feedback mechanism enables the system to undergo further enhancements in its intelligent capabilities, thus facilitating a more effective alignment with user requirements.

2. Related Works

In the early stages of system development, techniques based on rule and template matching were commonly employed to address simple queries. However, these approaches proved inadequate when confronted with complex queries, due to their inherent limitations in semantic understanding. The principal issue with these systems is their inability to accurately comprehend the user's intent and the semantic structure of complex queries. Although these early systems demonstrated proficiency in processing basic queries, they exhibited limitations in their ability to handle diverse and complex natural language input. In order to overcome these challenges, researchers have begun to explore methods based on semantic parsing and machine learning, with the objective of improving the accuracy and reliability of natural language query systems through the application of more advanced technologies [20].

The application of deep learning [21] [22] in natural language processing (NLP) has markedly enhanced the capacity of machines to comprehend and generate human language, particularly models based on Transformer architectures, such as BERT and GPT. Devlin *et al.* [23] have achieved notable outcomes on a number of NLP tasks through a pre-training approach characterised by a bidirectional encoder. The key innovation of BERT is its pre-training phase, which employs unsupervised learning techniques to train on vast quantities of text data, followed by supervised learning on specific tasks through fine-tuning. This approach enables BERT to capture complex semantic relationships in context, thereby facilitating its exceptional performance in tasks such as question answering systems, text classification, and named entity recognition. The success of BERT has facilitated the extensive deployment of deep learning models based on Transformer [24]-[26] architecture in the domain of NLP, thereby accelerating the advancement of natural language understanding and generation technology.

In their discussion, Shaw *et al.* [27] addressed the potential of deep semantic parsing to enhance the precision of natural language queries. The study illustrates the potential of semantic parsing models in comprehending and processing intricate queries. By employing compositional semantic parsing, the model is capable of decomposing intricate queries into more fundamental semantic units, thus enhancing the overall accuracy of the parsing process. This approach not only enhances the system's capacity to comprehend intricate queries, but also fortifies its resilience in the presence of natural language variants. Furthermore, the study underscores the necessity for the model to possess robust generalisation capabilities, enabling it to effectively process an array of natural language inputs and diverse query structures.

Brown *et al.* [28] demonstrated the capacity to rapidly adapt to novel tasks with a minimal number of examples through training on a comprehensive text dataset, a phenomenon known as few-shot learning. In the context of database query systems, GPT-3 is capable of generating structured query statements through natural language, thereby providing users with the ability to query in natural language without the necessity of mastering complex SQL syntax. The implementation of GPT-3 not only streamlines the operational process of database queries, but also markedly enhances the simplicity and user experience of the query system. By combining GPT-3's generation capabilities with the requirements of the database query system, the researchers have developed a series of intelligent query tools, enabling non-expert users to efficiently obtain the information they require from the database.

This paper distinguishes itself from related works by leveraging Large Language Models (LLMs) in database-specific context, to enhance semantic parsing and database query generation, addressing limitations of earlier rule-based and template-matching approaches. While related works like Shaw *et al.* explored semantic parsing, this paper emphasizes handling complex SQL features, such as multi-table joins and nested queries, and shows superior performance in metrics like query match rate and multi-table query accuracy. Additionally, it incorporates a user interaction feedback loop, allowing the system to dynamically refine SQL queries based on real-time input. This combination of LLM-driven parsing and user feedback integration offers significant advancements in the accuracy and usability of intelligent database query systems.

3. Methodologies

In this section, our proposed model uses the deep learning architecture of Large Language Models (LLMs) to interpret and process natural language queries and convert them into accurate database queries. The system integrates an LLM-powered semantic parser that translates user input into structured queries that the database management system can understand.

3.1. Preprocessing and Parsing

Preprocessing of user queries is the first step in the model, including text normalization and disambiguation. Convert input text into a uniform format, such as converting all letters to lowercase, removing punctuation, etc. The process can be expressed as Equation (1), where Q is the original query and Q' is the normalized query.

$$Q' = normaize(Q) \tag{1}$$

Eliminate ambiguity of polysemous words through contextual understanding. Use a context-based word vector representation method, such as Word2Vec, to convert input text into a vector representation. Word2vec enables vocabulary-to-vector conversion with CBOW and skip-gram models. By maximizing conditional probability and using optimization algorithms to train model parameters, Word2Vec is able to capture the semantic relationships between words and generate high-quality word vectors. These word vectors are widely used in natural language processing tasks, providing strong support for text understanding, semantic analysis, and information retrieval. The elimination is expressed as Equation (2).

$$J(\theta)=\sum_{t=1}^{T}\left[\log\sigma\left(v'_{WO}\cdot v_{WI}\right)+\sum_{i=1}^{K}\mathbb{E}_{WNi\sim P_n(W)}\left[\log\sigma\left(-v'_{WNi}\cdot v_{WI}\right)\right]\right] \tag{2}$$

where T is the total number of words in the corpus. v_{WI} is the word vector of the input word WI. v'_{WO} is the word vector of the output word WO. $\sigma(\cdot)$ is a sigmoid function and is defined as $\sigma(x)=\frac{1}{1+e^{-x}}$. K is the negative sample size. $P_n(W)$ is the noise distribution from which negative samples are used.

After preprocessing, LLMs are used for semantic parsing. LLM models use deep learning techniques to understand key entities and relationships in natural language. Coding is done using the Transformer architecture. The self-attention mechanism of the Transformer model implements the encoding of the input sequence by calculating the weighted sum between the query and the key-value pairs, which is expressed as Equation (3), where Q, K and V represent the query, key, and value matrix, respectively, and d_k is the dimension of the key vector.

$$Attention(Q,K,V)=softmax\left(\frac{QK^{T}}{\sqrt{d_k}}\right)V \tag{3}$$

Identify entities (E) and relationships (R) in queries by LLM. Transformer-based Named Entity Recognition (NER) and relational extraction models are used. The identification process is expressed as Equation (4).

$$E,R=LLM_{entity\_relation}(H) \tag{4}$$

where E represents a collection of entities, R represents a collection of relationships, and H represents a hidden layer representation after encoding.

3.2. Query Generation and Execution

The parsed information is converted into a structured query in the target database schema. The database schema is S, which contains a table (T) and a column (C). Map identified entities and relationships to tables and columns in the database schema. We assume the mapping function is map, which is expressed as Equation (5).

$$T,C=map(E,R,S) \tag{5}$$

The next step is to generate a structured query statements ($SQL = generate_{query(T,C)}$). Specifically, if a user query is "Find the products with the highest sales in 2023", the resulting SQL might include SELECT product_name, MAX(sales), FROM sales_table, WHERE year = 2023, GROUP BY product_name. Before LLM, query generation is usually done by template matching, which is implemented by a two-stage network: the first stage for searching candidate template as a classification problem, the second stage for populating the template as a masked language model task. With the help of LLM, we can directly generate a structured SQL statement given the tables and table's metadata in an end-to-end fashion. This paper utilizes the latter method.

The resulting query is executed on the database and the results are returned to the user. Execute the generated SQL statement. The database execution function is $result = execute\_SQL(SQL)$. The user gives feedback on the query result, and the system optimizes and adjusts it according to the feedback. Feedback mechanisms can be implemented through reinforcement learning. Based on user feedback, reinforcement learning is used for model optimization. The goal of reinforcement learning is to optimize strategies by maximizing cumulative rewards. Define the reward function $R(\theta)$, which represents how the model will perform for a given feedback signal F. Following Equation 6 describes the reward function, where r_t represents the instant reward obtained at time step t.

$$R(\theta) = \sum_{t=0}^{T} r_t \tag{6}$$

Above all, the proposed system implements an intelligent database semantic query enhancement through large language model, and uses deep learning technology to preprocess, semantic parsing, query generation and execution of natural language query. Through the feedback mechanism and reinforcement learning optimization, the system continuously improves the accuracy and reliability of query interpretation and generation. Through this method, it not only improves the user experience, but also provides new ideas and methods for the intelligent development of the database management system.

4. Experiments

4.1. Experiment Setup

We experimentally verify the performance of the intelligent database semantic query enhancement system based on pre-trained large language models GPT-3.5, namely gpt-3.5-turbo-0125, as the foundational model and fine-tuned on semantic parsing task-specific dataset through OpenAI provided API. With the initial learning rate of 1e−4, a batch size of 32, and a training round of 10. The experimental dataset uses WikiSQL (80,654 natural language-query pairs) and Spider (10,181 natural language-query pairs), and the dataset is divided into 70% training set, 15% validation set, and 15% test set. Through detailed model training, evaluation and user feedback mechanism, the results show that the system has excellent performance in semantic parsing accuracy, query execution success rate and user satisfaction, which proves its significant advantages and generalization ability in dealing with complex natural language queries.

4.2. Experiment Analysis

The complex query match rate is a metric used to evaluate how well the SQL query generated by the system matches the target SQL query when processing a complex query. Complex queries often involve several advanced SQL features, including nested queries, subqueries, aggregate functions, and multi-table joins. These features make the structure and logic of queries more complex, and put forward higher requirements for the semantic understanding and query generation capabilities of the system. The high complex query match rate indicates that the system can accurately parse and generate SQL queries that meet expectations, reflecting the accuracy and robustness of the system when processing complex natural language queries. This indicator is of great significance for evaluating the effectiveness of the system in practical application scenarios. Following Figure 1 shows the complex query match rate comparison results.

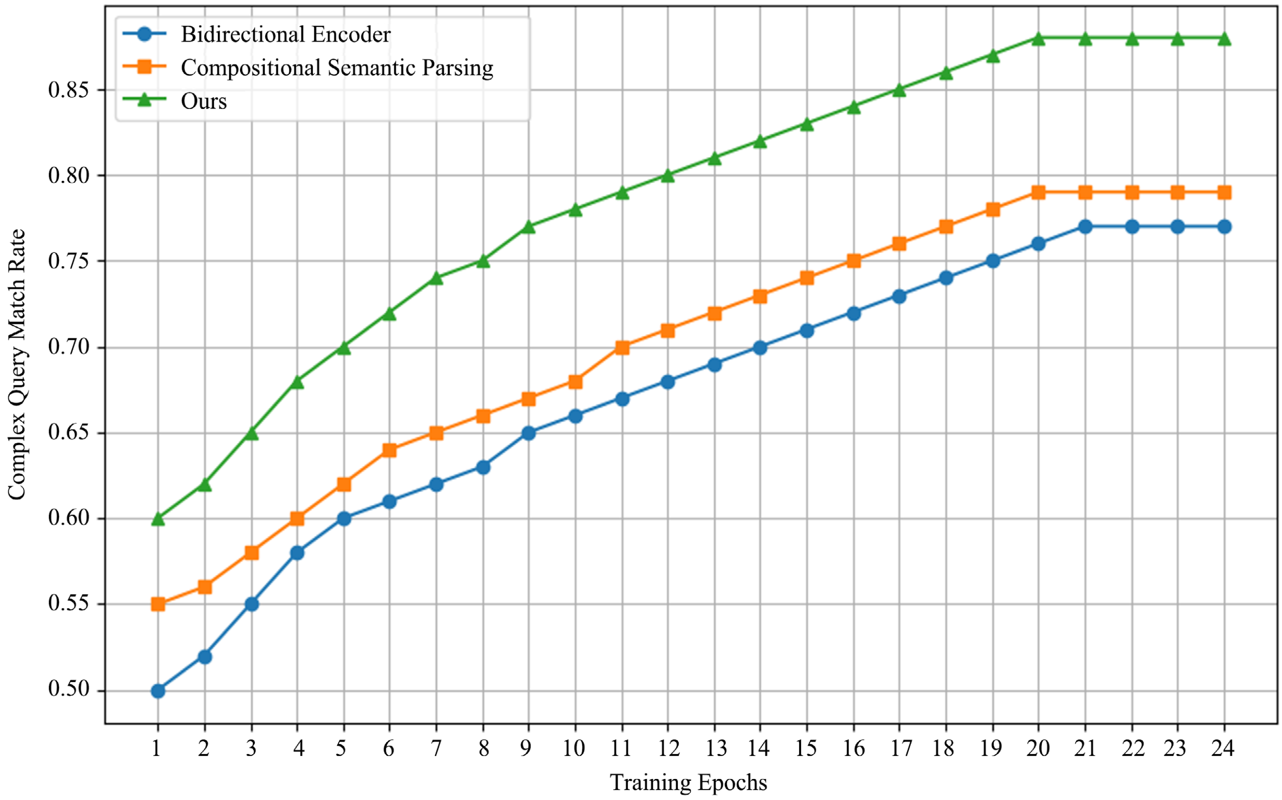

Figure 1. Complex query match rate over training epochs.

As can be seen in Figure 1, our method (Ours) performs well in the match rate of complex queries, and the match rate continues to increase as the number of training rounds increases, and is higher than the other two methods in all training rounds. This shows that our approach has significant advantages in handling complex natural language queries.

Multi-table query accuracy is an important metric to evaluate the accuracy of the system when processing multi-table join queries. It reflects the system's ability to parse relationships, generate JOIN conditions, and optimize queries, *i.e.*, whether the system can correctly parse table relationships in natural language queries and generate accurate SQL statements to represent these relationships. The high accuracy of multi-table queries indicates that the system can effectively execute complex data association query tasks and ensure that the generated SQL queries accurately reflect user intent. This metric is especially important for practical applications that need to deal with complex data relationships. Following Figure 2 shows the multi-table query accuracy comparison results.

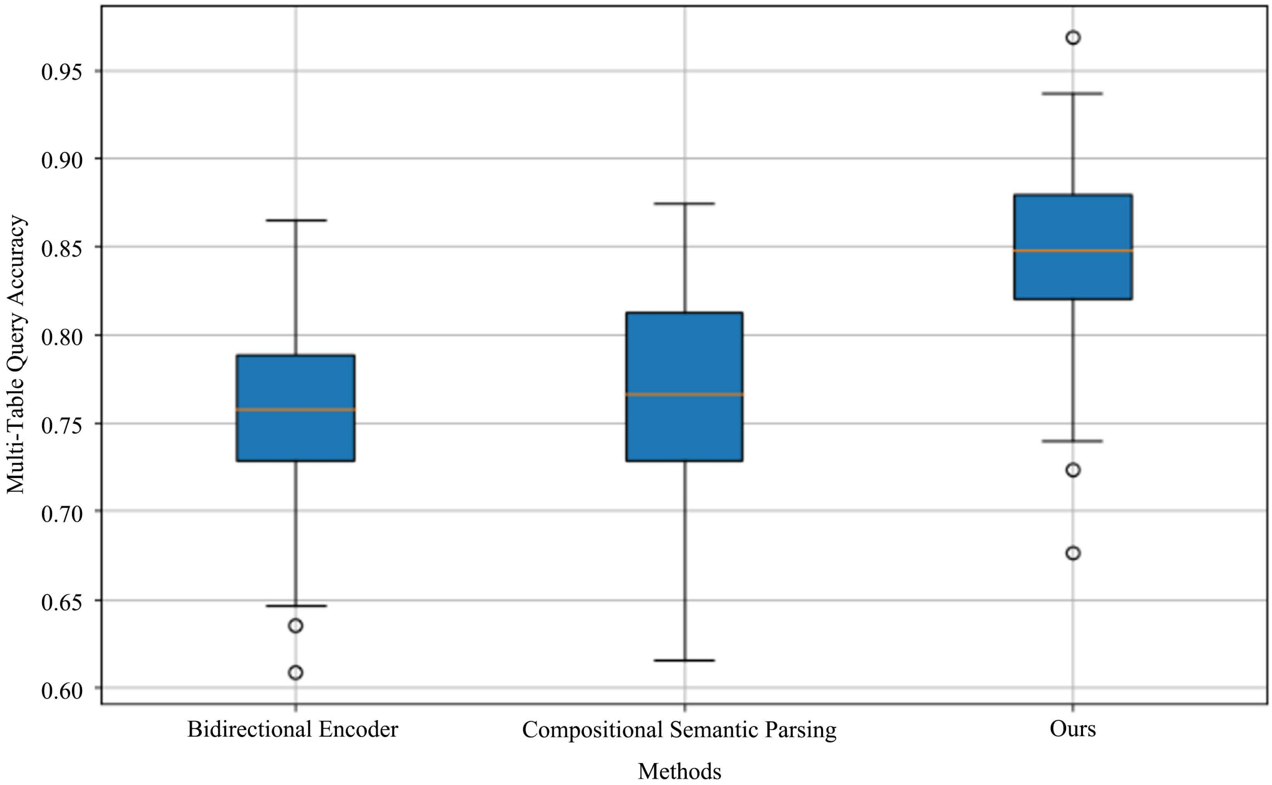

Figure 2. Comparison of multi-table query accuracy across different methods.

As can be seen in **Figure 2**, our method (Ours) performs well in the accuracy of multi-table queries, and the median and upper and lower quartiles of the accuracy are higher than those of the other two methods. This shows that our method has higher accuracy and stability when handling multi-table join queries.

Evaluating the success rate of user interaction with the system, that is, the correctness of the SQL query generated by the system after adjusting according to user feedback, is an important indicator to measure the flexibility and adaptability of the system in practical applications [29] [30]. The success rate of user feedback interaction reflects whether the system can effectively understand the feedback and make corresponding adjustments after receiving the user's feedback or correction suggestions, so as to generate the correct SQL query that meets the user's intent. This indicator not only examines the system's initial parsing and query generation capabilities, but also emphasizes the system's self-correction and continuous learning capabilities in a dynamic environment. The experimental results is demonstrated as **Figure 3**.

As can be seen in **Figure 3**, our method (Ours) performs well in terms of user interaction success rate, with higher median and upper and lower quartile success rates than the other two methods. This shows that our method can more effectively understand and adjust the generated SQL query after receiving user feedback, so as to more accurately reflect the user's intent, and has strong adaptability and learning ability.

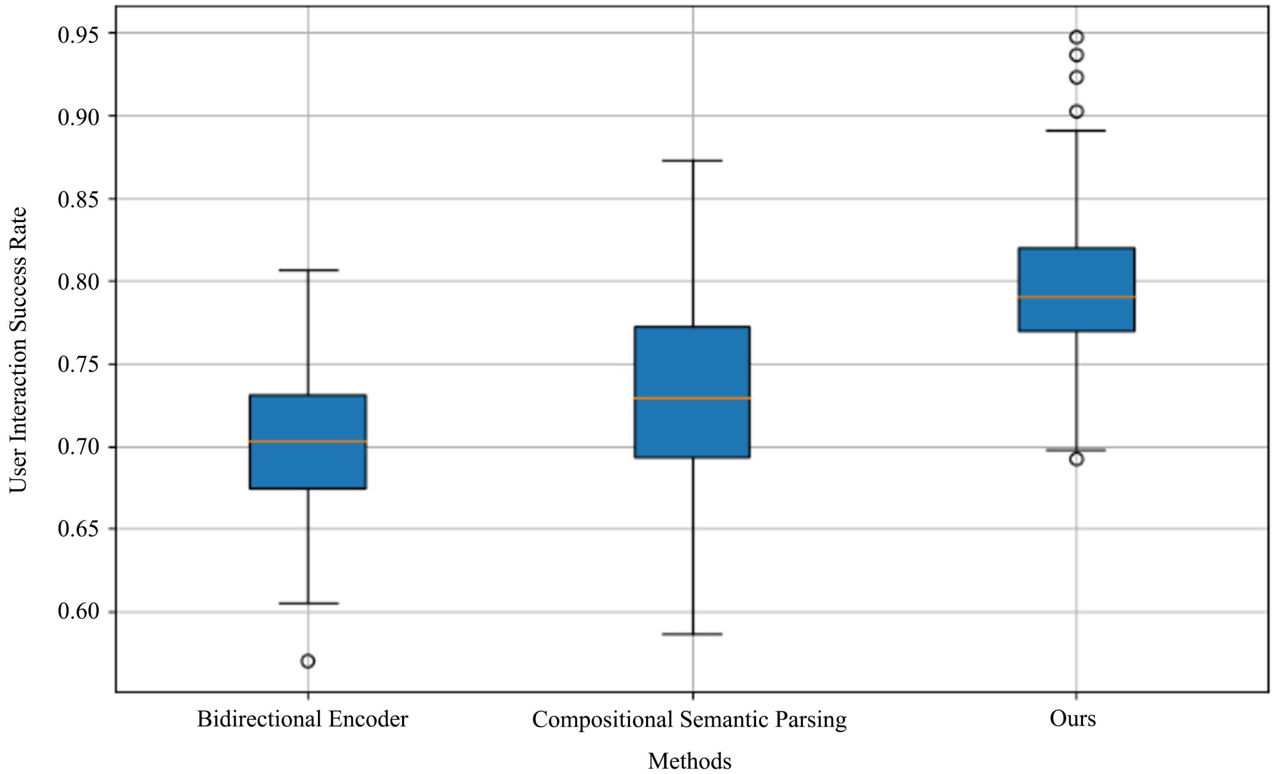

Figure 3. Comparison of user interaction success rate across different methods.

5. Conclusions

In conclusion, our research on Intelligent Database Semantic Query Enhancement based on Large Language Models (LLMs) demonstrates significant improvements in handling complex natural language queries. Using LLM and datasets like WikiSQL and Spider, our method outperformed others in metrics such as Complex Query Match Rate, Multi-Table Query Accuracy, and User Interaction Success Rate. The system's ability to accurately parse and generate SQL queries, effectively incorporate user feedback, and adapt dynamically highlights its robustness and practical applicability.

Limitations of this method including the inference speed of LLM may not satisfy the stringent query latency SLO of some of the most time-sensitive applications, such as financial transaction system or online advertising recommendation system. In addition, the resource consumption of LLM makes it unsuitable to be deployed in edge computing environment or low-resource machines. These limitations may be addressed with the development of LLM foundational models or improvement on the methods proposed in this paper.

Overall, this integration of LLMs into database query systems enhances semantic understanding and user experience, making advanced data querying more accessible and efficient. The integration of LLMs into database query systems has the potential to revolutionize both AI research and industry by making complex data querying more accessible and intuitive. This approach lowers the technical barrier for non-experts, enabling more users across sectors like finance, healthcare, and retail to leverage advanced data insights. By enhancing user experience and enabling continuous system improvements through feedback, this technology can drive efficiency, innovation, and data-driven decision-making across industries.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Sharma, D.K., Pamula, R. and Chauhan, D.S. (2021) Semantic Approaches for Query Expansion. *Evolutionary Intelligence*, **14**, 1101-1116. https://doi.org/10.1007/s12065-020-00554-x

[2] Aggoune, A. (2022) Intelligent Data Integration from Heterogeneous Relational Databases Containing Incomplete and Uncertain Information. *Intelligent Data Analysis*, **26**, 75-99. https://doi.org/10.3233/ida-205535

[3] Hu, T., Zhu, W. and Yan, Y. (2023) Artificial Intelligence Aspect of Transportation Analysis Using Large Scale Systems. 2023 6*th Artificial Intelligence and Cloud Computing Conference* (*AICCC*), Kyoto, 16-18 December 2023, 54-59. https://doi.org/10.1145/3639592.3639600

[4] Zhu, W. (2022) Optimizing Distributed Networking with Big Data Scheduling and Cloud Computing. *International Conference on Cloud Computing*, *Internet of Things*, *and Computer Applications* (*CICA* 2022), Luoyang, 22-24 April 2022, 23-28. https://doi.org/10.1117/12.2642577

[5] Yan, Y. (2022) Influencing Factors of Housing Price in New York-Analysis: Based on Excel Multi-Regression Model. *Proceedings of the International Conference on Big Data Economy and Digital Management*, Volume 1, 1005-1009. https://doi.org/10.5220/0011362000003440

[6] Liu, X., Qiu, H., Li, M., Yu, Z., Yang, Y. and Yan, Y. (2024) Application of Multimodal Fusion Deep Learning Model in Disease Recognition.

[7] Liu, X., Yu, Z. and Tan, L. (2024) Deep Learning for Lung Disease Classification Using Transfer Learning and a Customized CNN Architecture with Attention.

[8] Liu, X. and Wang, Z. (2024) Deep Learning in Medical Image Classification from MRI-Based Brain Tumor Images.

[9] Liu, X., Yu, Z., Tan, L., Yan, Y. and Shi, G. (2024) Enhancing Skin Lesion Diagnosis with Ensemble Learning.

[10] Pan, W., Liu, T., Zhang, W. and Sun, Y. (2021) Learning New Word Semantics with Conceptual Text. 2021 *International Joint Conference on Neural Networks* (*IJCNN*), Shenzhen, 18-22 July 2021, 1-8. https://doi.org/10.1109/ijcnn52387.2021.9533503

[11] Zhu, W. and Hu, T. (2021) Twitter Sentiment Analysis of Covid Vaccines. 2021 5*th International Conference on Artificial Intelligence and Virtual Reality* (*AIVR*), Kumamoto, 23-25 July 2021, 118-122. https://doi.org/10.1145/3480433.3480442

[12] Sun, Y., Duan, Y., Gong, H. and Wang, M. (2019) Learning Low-Dimensional State Embeddings and Metastable Clusters from Time Series Data. 33*rd Conference on Neural Information Processing Systems* (*NeurIPS* 2019), Vancouver, 8-14 December 2019.

[13] Gong, H. and Wang, M. (2020) A Duality Approach for Regret Minimization in Average-Award Ergodic Markov Decision Processes. *Learning for Dynamics and Control*,

Berkeley, 10-11 June 2020, 862-883.

[14] Xie, Y., Sun, Y. and Bertino, E. (2021) Learning Domain Semantics and Cross-Domain Correlations for Paper Recommendation. *Proceedings of the* 44*th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 11-15 July 2021, 706-715. https://doi.org/10.1145/3404835.3462975

[15] Ma, C. and Molnár, B. (2021) Ontology Learning from Relational Database: Opportunities for Semantic Information Integration. *Vietnam Journal of Computer Science*, **9**, 31-57. https://doi.org/10.1142/s219688882150024x

[16] Hamroun, M., Lajmi, S., Jallouli, M. and Souid, A. (2023) Efficient Text-Based Query Based on Multi-Level and Deep-Semantic Multimedia Indexing and Retrieval. *Multimedia Tools and Applications*, **83**, 55811-55850. https://doi.org/10.1007/s11042-023-17256-y

[17] Rahman, M.M., Islam, S., Kamruzzaman, M. and Joy, Z.H. (2024) Advanced Query Optimization in SQL Databases for Real-Time Big Data Analytics. *Academic Journal on Business Administration*, *Innovation & Sustainability*, **4**, 1-14. https://doi.org/10.69593/ajbais.v4i3.77

[18] Abgaz, Y., Rocha Souza, R., Methuku, J., Koch, G. and Dorn, A. (2021) A Methodology for Semantic Enrichment of Cultural Heritage Images Using Artificial Intelligence Technologies. *Journal of Imaging*, **7**, Article No. 121. https://doi.org/10.3390/jimaging7080121

[19] Li, L., Shu, Z., Yu, Z. and Wu, X. (2024) Robust Online Hashing with Label Semantic Enhancement for Cross-Modal Retrieval. *Pattern Recognition*, **145**, Article ID: 109972. https://doi.org/10.1016/j.patcog.2023.109972

[20] Androutsopoulos, I., Ritchie, G.D. and Thanisch, P. (1995) Natural Language Interfaces to Databases—An Introduction. *Natural Language Engineering*, **1**, 29-81. https://doi.org/10.1017/s135132490000005x

[21] Dan, H., Lu, B. and Li, M. (2024) Evaluation of Asphalt Pavement Texture Using Multiview Stereo Reconstruction Based on Deep Learning. *Construction and Building Materials*, **412**, Article ID: 134837. https://doi.org/10.1016/j.conbuildmat.2023.134837

[22] Dan, H., Yan, P., Tan, J., Zhou, Y. and Lu, B. (2024) Multiple Distresses Detection for Asphalt Pavement Using Improved You Only Look Once Algorithm Based on Convolutional Neural Network. *International Journal of Pavement Engineering*, **25**, Article ID: 2308169. https://doi.org/10.1080/10298436.2024.2308169

[23] Kenton, J.D.M.W.C. and Toutanova, L.K. (2019) Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT*, Vol. 1, 2.

[24] Ni, H., Meng, S., Geng, X., Li, P., Li, Z., Chen, X. and Zhang, S. (2024) Time Series Modeling for Heart Rate Prediction: From ARIMA to Transformers.

[25] Ni, H., Meng, S., Chen, X., Zhao, Z., Chen, A., Li, P. and Chan, Y. (2024) Harnessing Earnings Reports for Stock Predictions: A QLoRA-Enhanced LLM Approach.

[26] Zhou, Y., Zeng, Z., Chen, A., Zhou, X., Ni, H., Zhang, S. and Chen, X. (2024) Evaluating Modern Approaches in 3D Scene Reconstruction: NeRF vs Gaussian-Based Methods.

[27] Shaw, P., Chang, M., Pasupat, P. and Toutanova, K. (2021) Compositional Generalization and Natural Language Variation: Can a Semantic Parsing Approach Handle Both? *Proceedings of the* 59*th Annual Meeting of the Association for Computational Linguistics and the* 11*th International Joint Conference on Natural Language*

Processing, Volume 1, 922–938. https://doi.org/10.18653/v1/2021.acl-long.75

[28] Brown, T.B. (2020) Language Models Are Few-Shot Learners.

[29] Tan, L., Liu, S., Gao, J., Liu, X., Chu, L. and Jiang, H. (2024) Enhanced Self-Checkout System for Retail Based on Improved YOLOv10.

[30] Eyal, B., Bachar, A., Haroche, O. and Elhadad, M. (2023) Semantic Parsing for Complex Data Retrieval: Targeting Query Plans vs. SQL for No-Code Access to Relational Databases.