111DUT at SemEval-2025 Task 8:Hierarchical Chain-of-Thought Reasoning and Multi-Model Deliberation for Robust TableQA

Jiaqi Yao[1], Erchen Yu[1], Yicen Tian[1], Yiyang Kang[1], Jiayi Zhang[2], Hongfei Lin[1], Linlin Zong[3], Bo Xu[1]*

[1] School of Computer Science and Technology, Dalian University of Technology, China
[2] School of Control Science and Engineering, Dalian University of Technology, China
[3] School of Software, Dalian University of Technology, China
{1741917591, yuerchen0809, yicentian, kiang0920, 2791570843}@mail.dlut.edu.cn
{hflin, llzong, xubo}@dlut.edu.cn

Abstract

The proliferation of structured tabular data in domains like healthcare and finance has intensified the demand for precise table question answering, particularly for complex numerical reasoning and cross-domain generalization. Existing approaches struggle with implicit semantics and multi-step arithmetic operations. This paper presents our solution for SemEval-2025 task,including three synergistic components: (1) a Schema Profiler that extracts structural metadata via LLM-driven analysis and statistical validation, (2) a Hierarchical Chain-of-Thought module that decomposes questions into four stages—semantic anchoring, schema mapping, query synthesis, and self-correction—to ensure SQL validity, and (3) a Confidence-Accuracy Voting mechanism that resolves discrepancies across LLMs through weighted ensemble decisions. Our framework achieves scores of 81.23 on Databench and 81.99 on Databench_lite, ranking 6th and 5th respectively, demonstrating the effectiveness of structured metadata guidance and cross-model deliberation in complex TableQA scenarios.

1 Introduction

In the era of digitization, structured data represented in tabular formats is ubiquitous across domains such as finance, healthcare, and scientific research. Table Question Answering (TableQA), which aims to retrieve precise information from tables based on natural language queries, has emerged as a critical research direction. Its applications range from database querying and spreadsheet automation to extracting insights from web tables or even image-based tabular data. Despite its practical significance, the complexity of TableQA lies in effectively aligning natural language questions with the structural and semantic features of tables, especially when handling aggregation (e.g., "summarize sales by region"), comparison (e.g., "which product has the highest revenue"), and multi-hop reasoning (e.g., "find the second-largest budget department"). Traditional approaches often rely on weakly supervised table parsers to extract relevant cells and apply predefined aggregation operators, which are limited in generalizability and scalability (Pasupat and Liang, 2015).

Recent advancements in Large Language Models (LLMs) have revolutionized TableQA by enabling more flexible and context-aware reasoning. LLMs address TableQA challenges through two primary paradigms: In-Context Learning and Text-to-SQL. These approaches leverage the models' ability to process structured data alongside free-form text, opening new possibilities for handling complex tabular reasoning tasks.

The In-Context Learning paradigm integrates tabular data into carefully designed prompts, allowing models to generate answers in zero-shot or few-shot settings. For example, structured prompting strategies encode table headers, cell values, and structural metadata (e.g., row/column indices) into the input sequence, enhancing the model's ability to reason over numerical and hierarchical relationships (Lu et al., 2025). Recent work further improves robustness through reasoning-enhanced prompting, where LLMs are guided to decompose questions into step-by-step sub-tasks (e.g., filtering, sorting, and aggregating) (Qiao et al., 2023). Notably, models like TAPAS (Herzig et al., 2020) and TaBERT (Yin et al., 2020) demonstrate that pre-training on large-scale table-text pairs significantly enhances structural awareness, achieving state-of-the-art performance on benchmarks like WikiTableQuestions and WikiSQL.

The Text-to-SQL approach translates natural language questions into executable SQL queries, enabling direct database interactions. This task requires precise alignment between linguistic expressions (e.g., "senior employees") and database schemas (e.g., 'WHERE age > 60'), while ac-

*Corresponding author

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 216–221

counting for structural constraints such as primary/foreign keys and column types. Recent studies leverage LLMs' code-generation capabilities to improve SQL accuracy. For instance, DIN-SQL (Pourreza and Rafiei, 2023a) decomposes complex queries into sub-problems solved by specialized agents, while RESDSQL (Li et al., 2023) employs a retrieval-augmented framework to align questions with schema elements.

SemEval-2025 Task 8 tackles the challenge of answering diverse, real-world questions over large-scale tabular datasets in domains such as healthcare and finance. Existing methods face limitations in cross-domain generalization due to implicit semantics (e.g., medical jargon). They also struggle with complex numerical reasoning, including percentile calculations and multi-step arithmetic. To address these challenges, we propose a framework that combines structured schema analysis, hierarchical reasoning, and multi-model deliberation. Our method employs a three-stage architecture: (1) a Schema Profiler that automatically extracts structural metadata through guided LLM parsing and statistical verification, (2) a Hierarchical Chain-of-Thought Reasoning module that decomposes questions into semantic anchoring, schema mapping, query synthesis, and self-correction stages, and (3) a Confidence-Accuracy Voting mechanism that resolves discrepancies across three LLM agents through weighted ensemble deliberation. Our proposed method ranks 6th on the Databench dataset and 5th on the Databench_lite dataset.

2 Related Work

The rapid evolution of table question answering has been significantly propelled by advances in large language models (LLMs) and their application to Text-to-SQL tasks. Early work established the Spider benchmark (Yu et al., 2019), a cross-domain dataset that remains a cornerstone for evaluating complex SQL generation. Building on this, PICARD was introduced (Scholak et al., 2021), which integrates constrained decoding with pre-trained models like T5 to ensure syntactically valid SQL queries. The advent of powerful LLMs shifted the paradigm toward in-context learning, exemplified by DIN-SQL (Pourreza and Rafiei, 2023b), where GPT-4 iteratively decomposes questions into sub-tasks like schema linking and query refinement. Concurrently, retrieval-augmented methods like RESDSQL (Li et al., 2023) dynamically align questions with database schemas to mitigate domain shift.Meanwhile, it has been demonstrated that code-style prompts enable zero-shot SQL generation in C3 (Dong et al., 2023). Despite these innovations, challenges persist in handling implicit semantics, where domain-specific terms (e.g., medical abbreviations) require external knowledge, and context window constraints (Hao et al., 2022), which lead to truncation of large tables. Recent efforts like CoT-SQL (Wei et al., 2022) leverages chain-of-thought prompting to decompose multi-step queries.

3 System Overview

In this section, we will introduce the overall structure of our proposed system. Our proposed system comprises three core modules that synergistically enhance table-based question answering through structured reasoning and ensemble learning. Figure 1 illustrates the overall architecture of our proposed method.

Module 1: Schema Profiler: We first feed partial tabular data into a Large Language Model (moonshot-v1) to extract critical schema information. This process automatically identifies field types, value distributions, and contextual relationships within the table structure. The derived metadata establishes a semantic foundation for subsequent processing stages. **Module 2: Hierarchical Chain-of-Thought Reasoning:** We design a four-stage Chain-of-Thought (CoT) prompting strategy that combines schema metadata with task-specific instructions. This enhanced prompt is then input into three different Large Language Models to generate candidate SQL queries. **Module 3: Multi-Model Deliberation:** To ensure robustness, we implement a deliberation mechanism that directly adopts answers when all models reach consensus. When discrepancies occur, the mechanism employs cross-model voting with mutual evaluation. The voting system weights the models' confidence scores and historical accuracy to resolve conflicts, ultimately selecting the most reliable answer through ensemble decision-making.

This hierarchical architecture effectively balances schema comprehension, diverse reasoning patterns, and result verification, demonstrating strong performance on complex table QA scenarios.

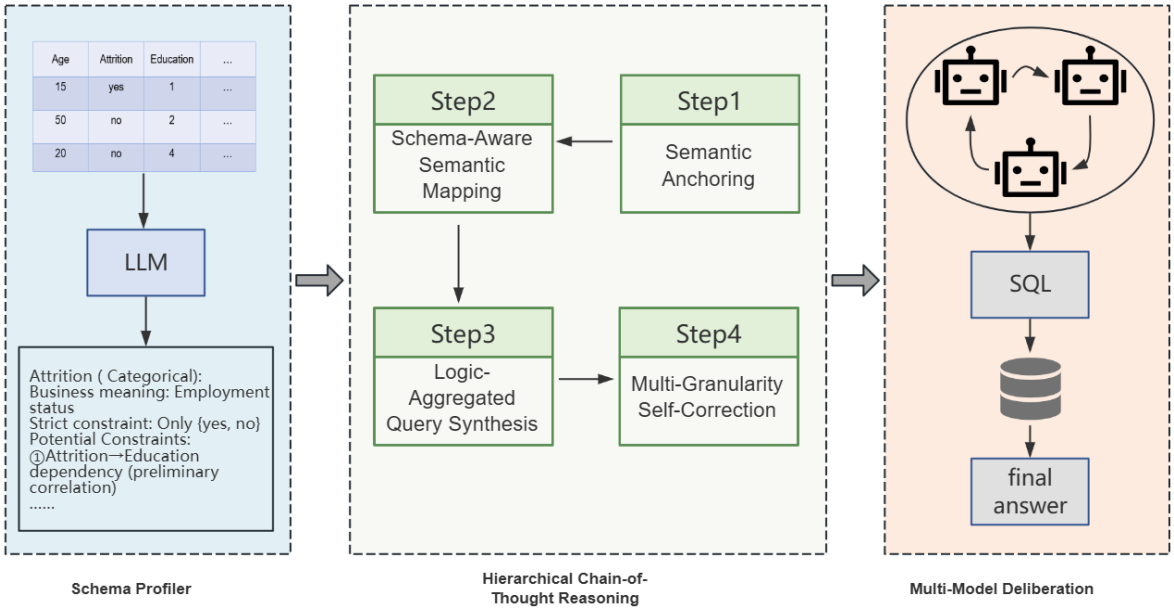

Figure 1: The overall architecture of our proposed method.

3.1 Schema Profiler

The framework initiates with structural metadata parsing to achieve a deep understanding of the tabular schema. Specifically, we input a 20-row sample from the Databench Lite dataset into moonshot-v1 and process it using a multi-turn guided prompting strategy. The primary prompt instructs the model to analyze the table structure and explicitly requires the output to include: (1) Column attributes, including data types (string/numerical/temporal) and value characteristics (units for numerical columns, frequent values for categorical columns); (2) Field semantics, which involves precisely parsing the meaning of each field to clarify its specific role in the business context and the relationships between fields; (3) Constraint discovery, which identifies implicit business rules (e.g., inventory $\leq$ warehouse capacity). This process generates a standardized JSON schema profile, thereby establishing a reliable structural foundation for downstream SQL generation.

3.2 Hierarchical Chain-of-Thought Reasoning

To address the challenges of generating accurate SQL queries from natural language questions over heterogeneous tables, we propose a hierarchical Chain-of-Thought (CoT) framework that decomposes the reasoning process into four interconnected cognitive stages. This structured approach ensures both syntactic validity and semantic alignment with the database schema.

(1)Semantic Anchoring: The initial phase is the semantic mining and classification stage, where the model is required to mine the semantics of the question and determine its type: Boolean, scalar, or list. Boolean questions are typically used for existence checks, such as determining whether a certain condition is met through trigger words like "whether," "does... exist," or "is there any...". Scalar questions involve quantitative queries and usually contain terms such as "highest," "lowest," "average," or "total," aiming to obtain a single numerical result. For example, "What is the highest price?" or "What is the average value?". List questions, on the other hand, require returning a set of entities or results, such as through expressions like "how many," "list all...," or "return the set of...," which are used to obtain multiple results or entity collections that meet specific conditions.

(2)Schema-Aware Semantic Mapping: In this stage, the structured metadata profile is utilized to map entities in the query to corresponding column names. For explicit entity linking, field names in the query are directly matched (e.g., “patient age” $\rightarrow$ age). For implicit semantic inference, potential associations are uncovered (e.g., “hospitalization duration” $\rightarrow$ discharge_date - admission_date). For value range normalization, expressions are transformed into database storage formats (e.g., “Q1” $\rightarrow$ BETWEEN ’2023-01’ AND ’2023-03’).

(3)Logic-Aggregated Query Synthesis: This phase systematically integrates parsed seman-

tic components into executable SQL structures through three operational principles. Parenthesis-encapsulated precedence rules govern multi-clause logic composition (e.g., ‘(A OR B) AND C‘), complemented by type-driven operator selection for temporal or numerical comparisons. Dynamic aggregation binding associates question intent with SQL functions—‘AVG()‘ for "average price" and ‘COUNT()‘ for "total quantity". Subquery optimization prioritizes nested structures over joins when processing comparative constraints (e.g., "books above average price"), effectively mitigating Cartesian product risks through predicate pushdown techniques.

(4)Multi-Granularity Self-Correction: In this stage, common error patterns of Large Language Models (LLMs) are countered through syntactic, semantic, and logical validation. Syntax validation enforces schema-compliant escaping for special column names (e.g., auto-correcting malformed ‘`Price (TK)`‘ to ‘`"Price (TK)"`‘) and verifies join paths against foreign key constraints. Semantic consistency checks eliminate contradictory conditional logic (e.g., conflicting ‘`Stock_Status`‘ values) while injecting null-safety clauses (e.g., ‘`IS NOT NULL`‘) for optional fields. Output alignment ensures that Boolean queries strictly return truth values and scalar queries produce singleton aggregation results, among others.

3.3 Multi-Model Deliberation

To resolve discrepancies in SQL generation across multiple large language models (LLMs), we propose a streamlined consensus mechanism that harmonizes model confidence and empirical performance.

(1)Unanimity Prioritization: If the SQL outputs from all models yield identical answers when executed in the database, the output is directly adopted, leveraging inter-model agreement as a high-reliability indicator.

(2) Confidence-Accuracy Voting: When the three LLM agents (qwen-max, Qwen2.5-Coder-Instruct, Moonshot) generate conflicting SQL candidates, a voting protocol is triggered. For each candidate query, the system calculates its final score through a Confidence-Accuracy Voting mechanism:

$$\text{Score}_k = \underbrace{\left(\sum_{j \neq m} \text{Conf}_{j \to k} \right)}_{\text{Cross-Model Consensus}} \times \underbrace{\text{HisAcc}_m}_{\text{Model Reliability}} \quad (1)$$

where:

- **$\text{Conf}_{j \to k}$** (0–1): Model j's confidence score for candidate SQL_k. For example, if SQL_1 is generated by qwen-max, Moonshot and Qwen-Coder assess its correctness likelihood separately.
- **HisAcc_m** (0–1):Pre-computed accuracy of the model on the dev Databench set containing diverse table schemas and question types.

The candidate with the highest aggregated score is selected, ensuring both peer validation and source model competency are leveraged.

4 Experiment

4.1 Dataset

The dataset for this study is derived from the SemEval 2025 Task 8 benchmark suite, which includes two versions: DataBench and its lightweight variant DataBench Lite. The full-scale DataBench comprises 65 real-world tabular corpora spanning 3,269,975 rows and 1,615 columns, paired with 1,300 annotated questions split into training and development subsets. For streamlined evaluation, DataBench Lite provides sampled versions of these corpora, retaining 20 rows per table. The test set consists of an independent collection of 15 corpora and 522 questions to ensure rigorous evaluation.

4.2 Implementation

In our experiment, we utilized three LLMs to evaluate their performance on the given task. Specifically, we called the APIs of Qwen-max, Qwen-coder, and Moonshot. Table 1 summarizes the configuration settings used for each model during the experiment.

Table 1: Model configuration settings.

Setting	Qwen-max	Qwen-coder	Moonshot
temperature	0.7	0.7	0.3
top_p	0.8	0.8	0.8
presence_penalty	1.5	1.5	1.5
max_tokens	8,192	8,192	8192

4.3 Results

Table 2 and Table 3 show the performance of three models on the Databench and Databench_lite datasets with different modules added. The experimental results indicate that systematically introducing the Schema Profiler and hierarchical Chain of Thought (CoT) strategy significantly improves table question-answering performance. Under the full configuration (+Profiler+COT), Qwen-max achieves a score of 77.39 on the complete dataset Databench, an improvement of 8.04 over the baseline (69.35), and reaches 77.97 (+8.05) on the lightweight version Databench_lite. This validates the universal advantage of structured metadata guidance. The hierarchical CoT enhances the execution accuracy of complex queries through step-by-step parsing. The synergistic effect of the two strategies generates a superadditive improvement—the combined gain (Databench: 8.04–9.96) exceeds the sum of individual module gains, highlighting the role of metadata in directing the reasoning path.

Table 2: Comparison of Scores for three models on the test set of Databench. "Base" indicates no strategy added, "+Profiler" indicates the addition of Profiler, "+COT" indicates the addition of COT.

Method	Qwen-max	Qwen-coder	Moonshot
Base	69.35	68.2	65.9
+Profiler	71.83	70.88	69.92
+COT	74.71	73.18	72.22
+Profiler+COT	**77.39**	**76.44**	**75.86**

Table 3: Comparison of Scores for three models on the test set of Databench_lite. "Base" indicates no strategy added, "+Profiler" indicates the addition of Profiler, "+COT" indicates the addition of COT.

Method	Qwen-max	Qwen-coder	Moonshot
Base	69.92	68.0	66.28
+Profiler	72.22	71.26	70.11
+COT	75.47	74.32	72.8
+Profiler+COT	**77.97**	**76.63**	**76.05**

Table 4 compares the performance of three review strategies on the complex scenario dataset Databench and its lightweight version Databench_lite. The experimental results show that the multi-model collaborative decision-making mechanism significantly improves the accuracy of the table question-answering system. The single-model baseline (Qwen-max) achieves scores of 77.39 on Databench and 77.97 on Databench_lite without enabling review. After introducing dual-model cross-validation (Qwen-max + Moonshot), the scores increase by 2.3 and 2.11, respectively. The full review strategy integrating three models (All) further raises the accuracy to 81.23 and 81.99, achieving absolute improvements of 4.84 and 4.02 over the baseline. This progress validates the effectiveness of cross-model verification in eliminating individual biases—through a two-stage consensus mechanism (consensus adoption and weighted voting), the robustness of semantic understanding under complex table structures is enhanced. It is particularly noteworthy that dual-model review can cover approximately 75% of the potential error correction needs, providing an efficient balance between precision and computational cost for scenarios with limited resources.

Table 4: Comparison of Scores for Different Deliberation Strategies on the Databench and Databench_lite Datasets

Model	Score	$\text{Score}_{\text{lite}}$
Qwen-max	77.39	77.97
Qwen-max+moonshot	79.69	80.08
all	**81.23**	**81.99**

5 Conclusion

This paper presents our solution for SemEval-2025 Task 8 on Table Question Answering. We propose a three-stage framework integrating schema analysis, hierarchical reasoning, and multi-model deliberation. Our approach leverages: (1) a Schema Profiler that extracts structural metadata via guided LLM parsing, (2) a Hierarchical Chain-of-Thought module decomposing questions into four reasoning stages (semantic anchoring, schema mapping, query synthesis, self-correction), and (3) a Confidence-Accuracy Voting mechanism harmonizing outputs from three LLM agents through weighted ensemble decisions. Our method achieves scores of 81.23 on Databench and 81.99 on Databench_lite, ranking 6th and 5th respectively. Future work will focus on: (1) enhancing schema profiling with dynamic domain adaptation, (2) refining CoT stages for multi-table joins, and (3) extending the deliberation mechanism to hybrid LLM-Symbolic architectures.

Acknowledgments

We thank our anonymous reviewers for their helpful comments. This work was supported by the Fundamental Research Funds for the Central Universities (No. DUT24MS003) and the Liaoning Provincial Natural Science Foundation Joint Fund Program (No. 2023-MSBA-003).

References

Xuemei Dong, Chao Zhang, Yuhang Ge, Yuren Mao, Yunjun Gao, lu Chen, Jinshu Lin, and Dongfang Lou. 2023. C3: Zero-shot text-to-sql with chatgpt. *Preprint*, arXiv:2307.07306.

Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. 2022. Structured prompting: Scaling in-context learning to 1,000 examples. *Preprint*, arXiv:2212.06713.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Haoyang Li, Jing Zhang, Cuiping Li, and Hong Chen. 2023. Resdsql: Decoupling schema linking and skeleton parsing for text-to-sql. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37:13067–13075.

Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. 2025. Large language model for table processing: a survey. *Frontiers of Computer Science*, 19(2).

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Mohammadreza Pourreza and Davood Rafiei. 2023a. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. In *Advances in Neural Information Processing Systems*, volume 36, pages 36339–36348. Curran Associates, Inc.

Mohammadreza Pourreza and Davood Rafiei. 2023b. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. *Preprint*, arXiv:2304.11015.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5368–5393, Toronto, Canada. Association for Computational Linguistics.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2019. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *Preprint*, arXiv:1809.08887.