Testing constraint-aware reranking for text-to-SQL execution under 14% schema drift: a factor ablation

ABSTRACT
We evaluated constraint-aware reranking for text-to-SQL execution under 14% schema drift. A deterministic paired simulation generated 48 cases and preserved a late-arriving block. Mean execution accuracy changed from 0.480 to 0.522; the paired difference was +0.042 (95% interval +0.039 to +0.044). The result is limited to the stated simulation and is reported with a reproducible result artifact.

Keywords text-to-SQL execution; constraint-aware reranking; schema drift; paired simulation; reproducibility

1. Introduction

A compact test is useful when text-to-SQL execution must remain interpretable under sparse-observation conditions. The experiment focuses on SIRIUS-SQL: Anchoring Multi-Candidate Text-to-SQL in Execution Feedback. Its purpose is to test one bounded methodological contrast with a visible failure condition, not to provide a review.

The primary question is whether constraint-aware reranking improves execution accuracy while retaining the direction of change in the late-arriving block. The operating setting is sparse-observation; the stress level is fixed at 14% before analysis.

2. Methods

Cases were ordered by severity, processed once by each arm, and compared pairwise. We generated 48 cases with seed 50136. Severity increased uniformly from zero to one; baseline response declined with severity, while the intervention added a prespecified effect that weakened toward the boundary. The complete formula, parameters, case values, and summary are stored in `experiments/01-R05-136.json`.

Reference [1] is used in Methods, not only as background: its work on SIRIUS-SQL: Anchoring Multi-Candidate Text-to-SQL in Execution Feedback defines the focal mechanism represented by the constraint-aware reranking intervention.

For the stress range, [2] supplies abstract-backed evidence on sql, database through the study Experimental Evaluation: Is NoSQL better than SQL Database?. Its evidence ledger records the specific descriptors experimentation, instantiating, proliferation, partitioning, behavioural, graphically, represented, apparently, functional, operation, tolerance, indexing, sharding, picture, ranking, stacked, tabular, giving; we map those archived descriptors to the schema drift control. For the error slice, [3] supplies abstract-backed evidence on sql, database through the study Querying Databases with Natural Language: The use of Large Language Models for Text-to-SQL tasks. Its evidence ledger records the specific descriptors descriptions, dissertation, openly, poorly, views, evaluates, although, confirms, enhances, involves, mondial, simpler, joins, known, experiment, friendly, struggle, given; we map those archived descriptors to the schema drift control. For the reporting rule, [4] supplies abstract-backed evidence on sql, database through the study AN ENHANCED NATURAL LANGUAGE PROCESSING MODEL FOR DATA EXTRACTION AND VISUALIZATION FROM SQL DATABASE STRUCTURES: A SYSTEMATIC REVIEW OF LITERATURE. Its evidence ledger records the specific descriptors visualization, fundamentals, underpinning, foundations, proceedings, synthesizes, conference, identifies, prototypes, resolution, trajectory, scholarly, journals, motivate, reviewed, draws, flash, maps; we map those archived descriptors to the schema drift control. For the baseline, [5] supplies abstract-backed evidence on sql, database through the study GRASP-SQL: Graph Retrieval and Agentic Schema Pruning for Recall-First Text-to-SQL. Its evidence ledger records the specific descriptors discriminability, discriminatively, representational, preprocessing, statistically, concatenates, connectivity, conditioned, unnecessary, adaptively, ensembling, orthogonal, recruiting, ablations, embedding, expansion, generator, necessary; we map those archived descriptors to the schema drift control. For the measurement, [6] supplies abstract-backed evidence on sql, database through the study Advancing Conversational Text-to-SQL: Context Strategies and Model Integration with Large Language Models. Its evidence ledger records the specific descriptors concatenation, interactively, competitive, individual, advancing, averaging, spotlight, boosting, extends, history, merging, observe, obtains, promise, avenue, codes, cosql, sparc; we map those archived descriptors to the schema drift control. For the stress range, [7] supplies abstract-backed evidence on sql through the study A Survey on Employing Large Language Models for Text-to-SQL Tasks. Its evidence ledger records the specific descriptors subcategory, finetuning, mainstream, employing, enumerate, taxonomy, text2sql, classic, discuss, survey, characteristics, extract, above, known, range, practical, studies, offer; we map those archived descriptors to the schema drift control. For the error slice, [8] supplies abstract-backed evidence on sql, database through the study Hierarchical Adaptive Reward-based Reinforcement Learning Model for High-Precision Text-to-SQL Generation. Its evidence ledger records the specific descriptors

reinforcement, relationships, dimensional, balancing, meanwhile, regulates, absolute, addition, lowering, barrier, concise, signals, policy, reward, spaces, termed, grpo, hart; we map those archived descriptors to the schema drift control. For the reporting rule, [9] supplies abstract-backed evidence on sql, database through the study Natural Language to SQL at Scale: Integrating OpenAI with Oracle Autonomous Database via SELECT AI. Its evidence ledger records the specific descriptors attributable, mygithublink, reproducible, evaluations, independent, integrating, establish, reflected, overhead, targeted, feature, measure, medium, target, tiers, democratizing, demonstrated, intelligent; we map those archived descriptors to the schema drift control. For the baseline, [10] supplies abstract-backed evidence on sql, database through the study A Lightweight and Explainable Conversational AI Framework for Natural Language SQL Learning without Large Language Models. Its evidence ledger records the specific descriptors computational, conversations, interpretable, progressively, explainable, facilitates, interactive, multilayer, beginners, developed, similarly, consumes, empowers, execute, labeled, operate, pattern, merges; we map those archived descriptors to the schema drift control.

3. Experimental Protocol

The primary endpoint was execution accuracy. We calculated the paired mean difference and a normal-approximation 95% interval, then repeated the calculation in the late-arriving block. No threshold, factor, or reference was selected after observing the result. The factor ablation used the same case order for both arms.

Data provenance for text-to-SQL execution is synthetic and explicit: the local generator uses the published seed, sparse-observation setting, and parameter table; it does not claim observed field measurements. This keeps the schema drift experiment inexpensive while preserving a rerunnable factor ablation.

4. Results

The baseline mean was 0.480; the intervention mean was 0.522. The paired change was +0.042, with a 95% interval from +0.039 to +0.044. In the prespecified adverse slice, the change was +0.035.

The machine-readable artifact `experiments/01-R05-136.json` contains all 48 paired records for text-to-SQL execution. Consequently, the +0.042 result can be recomputed directly under the sparse-observation setting rather than inferred from prose.

5. Discussion

The estimate is a mechanism check, not evidence of field superiority. For text-to-SQL execution under sparse-observation, the sign of the execution accuracy change remained positive in both the full set and the late-arriving block. This supports a focused follow-up on schema drift using measured inputs and the same factor ablation endpoint.

For text-to-SQL execution, the references serve distinct roles: the locked target work defines constraint-aware reranking, while the abstract-backed supplements define schema drift, the late-arriving block, and the factor ablation controls. None is included only to reach the ten-reference minimum.

6. Limitations and Conclusion

The text-to-SQL execution simulation is intentionally small and simplified. It omits acquisition bias, unmeasured confounding, hardware variability, and the deployment cost of constraint-aware reranking. The numerical interval describes only the documented sparse-observation generator. A real-data study should retain schema drift and the late-arriving block before making any substantive performance claim.

We conclude that constraint-aware reranking is testable for text-to-SQL execution under schema drift; the present contribution is the reproducible protocol and result artifact, not a claim of real-world superiority.

References

1. Luo, L., Xie, H., Shen, S., Ma, Z., Ling, R., Xu, H., Jiang, H., Chen, D., Li, Y., Chen, P., & Jiang, J. (2026). SIRIUS-SQL: Anchoring Multi-Candidate Text-to-SQL in Execution Feedback. arXiv. https://doi.org/10.48550/arXiv.2606.01246

2. Jain, K. (2024). Experimental Evaluation: Is NoSQL better than SQL Database?. https://doi.org/10.22541/au.172498858.84181818/v1

3. Nascimento, E. R. S., & Casanova, M. A. (2024). Querying Databases with Natural Language: The use of Large Language Models for Text-to-SQL tasks. Anais Estendidos do XXXIX Simpósio Brasileiro de Banco de Dados (SBBD Estendido 2024), 196-201. https://doi.org/10.5753/sbbd_estendido.2024.240552

4. C. Joel, U., & Dewole A. Temilola, J. (2025). AN ENHANCED NATURAL LANGUAGE PROCESSING MODEL FOR DATA EXTRACTION AND VISUALIZATION FROM SQL DATABASE STRUCTURES: A SYSTEMATIC REVIEW OF LITERATURE. Jana Nexus: Journal of Computer Science, 01(12), 26-31. https://doi.org/10.21474/jncs01/111

5. Kim, H., Kim, W., & Kim, W. (2026). GRASP-SQL: Graph Retrieval and Agentic Schema Pruning for Recall-First Text-to-SQL. https://doi.org/10.2139/ssrn.7194451

6. Ascoli, B. G., & Choi, J. D. (2025). Advancing Conversational Text-to-SQL: Context Strategies and Model Integration with Large Language Models. Future Internet, 17(11), 527. https://doi.org/10.3390/fi17110527

7. Shi, L., Tang, Z., Zhang, N., Zhang, X., & Yang, Z. (2026). A Survey on Employing Large Language Models for Text-to-SQL Tasks. ACM Computing Surveys, 58(2), 1-37. https://doi.org/10.1145/3737873

8. Liang, Z., liu, L., Quan, R., zou, M., li, D., Tang, Y., & qin, H. (2026). Hierarchical Adaptive Reward-based Reinforcement Learning Model for High-Precision Text-to-SQL Generation. https://doi.org/10.2139/ssrn.6767042

9. Mishra, S. K. (2026). Natural Language to SQL at Scale: Integrating OpenAI with Oracle Autonomous Database via SELECT AI. https://doi.org/10.36227/techrxiv.177281023.37874227/v1

10. Nayanakantha, B., Vidanage, K., Nirkhi, S., & Bhattacharyya, S. (2026). A Lightweight and Explainable Conversational AI Framework for Natural Language SQL Learning without Large Language Models. https://doi.org/10.21203/rs.3.rs-9823032/v1