Tuning constraint-aware reranking for text-to-SQL execution under 50% schema drift: a error-stratified estimate

ABSTRACT
We evaluated constraint-aware reranking for text-to-SQL execution under 50% schema drift. A deterministic paired simulation generated 48 cases and preserved a median slice. Mean execution accuracy changed from 0.558 to 0.603; the paired difference was +0.045 (95% interval +0.042 to +0.047). 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

Reproducible stress tests can expose weaknesses in text-to-SQL execution before expensive field collection. The experiment focuses on SiriusDeliver: Automating Data Warehouse Delivery at Tencent. 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 median slice. The operating setting is noisy-input; the stress level is fixed at 50% before analysis.

2. Methods

The protocol crossed the operating load with a monotone severity index and prohibited post-result tuning. We generated 48 cases with seed 50092. 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-R05-092.json`.

Reference [1] is used in Methods, not only as background: its work on SiriusDeliver: Automating Data Warehouse Delivery at Tencent 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 Nabil: A Text-to-SQL Model Based on Brain-Inspired Computing Techniques and Large Language Modeling. Its evidence ledger records the specific descriptors discriminative, spatiotemporal, normalization, abstraction, inevitable, champion, encoding, inspired, networks, verified, engines, entered, spiking, duckdb, fuses, nabil, intelligent, outperforms; 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 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. For the reporting rule, [4] supplies abstract-backed evidence on sql through the study Microsoft SQL Sunucusunda Veritabanı Kurtarma Teknikleri. Its evidence ledger records the specific descriptors teknolojilerinde, atlatabilmenin, ilerlemektedir, merkezlerinde, teknolojilere, enformasyona, kapasiteleri, retilememesi, kontrolleri, problemleri, sunucusunda, anabilecek, genellikle, pratikleri, senaryolar, teknikleri, dijitalle, durumunda; we map those archived descriptors to the schema drift control. For the baseline, [5] supplies abstract-backed evidence on sql, database through the study Large Language Model Based Chatbot for Database Interaction through Natural Language. Its evidence ledger records the specific descriptors completeness, complexities, naturalness, empowering, interprets, usefulness, interpret, barriers, cohesion, fluency, number, today, back, democratizing, translates, prototype, relevance, informed; we map those archived descriptors to the schema drift control. For the measurement, [6] supplies abstract-backed evidence on sql, database through the study ETL pipelines and SQL database management. Its evidence ledger records the specific descriptors indispensable, authenticate, consolidated, contemporary, continuously, centralized, elucidating, responsible, configured, immediate, processed, frontend, visitors, display, visitor, manner, serves, page; we map those archived descriptors to the schema drift control. For the stress range, [7] supplies abstract-backed evidence on sql, database through the study ETM: Modern Insights into Perspective on Text-to-SQL Evaluation in the Age of Large Language Models. Its evidence ledger records the specific descriptors stylistically, counterparts, misrepresent, overestimate, shortcomings, perspective, problematic, community, overlooks, elements, ignoring, inherent, negative, release, anyone, script, rates, rigid; 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 Enhancing Text-to-SQL Capabilities of Large Language Models via Domain Database Knowledge Injection. Its evidence ledger records the specific descriptors effectively, downstream, injected, subtask, seen, generalizability, incorporating, presented, contents, values, names, cell, face, make,

demonstrating, hallucination, introduces, understand; we map those archived descriptors to the schema drift control. For the reporting rule, [9] supplies abstract-backed evidence on sql, database, analytics through the study Lightweight and Data-Efficient Fine-Tuning of Language Models for Bidirectional Text-to-SQL and SQL-to-Text Tasks: A Systematic Review. Its evidence ledger records the specific descriptors computationally, featherlight, quantization, sacrificing, sustainable, ultramodern, annotation, appendages, conclusion, frameworks, pretrained, procedures, quantities, conscious, parameter, adoption, creation, examined; we map those archived descriptors to the schema drift control. For the baseline, [10] supplies abstract-backed evidence on sql, database through the study FGCSQL: A Three-Stage Pipeline for Large Language Model-driven Chinese Text-to-SQL. Its evidence ledger records the specific descriptors breakthroughs, culminating, propagation, confronted, harnessing, indicating, irrelevant, mitigating, eliminate, outstrips, showcases, typically, uniquely, validity, cspider, decoder, lingual, fgcsql; 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 median slice. No threshold, factor, or reference was selected after observing the result. The error-stratified estimate 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, noisy-input setting, and parameter table; it does not claim observed field measurements. This keeps the schema drift experiment inexpensive while preserving a rerunnable error-stratified estimate.

4. Results

The baseline mean was 0.558; the intervention mean was 0.603. The paired change was +0.045, with a 95% interval from +0.042 to +0.047. In the prespecified adverse slice, the change was +0.037.

The machine-readable artifact `experiments/01-R05-092.json` contains all 48 paired records for text-to-SQL execution. Consequently, the +0.045 result can be recomputed directly under the noisy-input setting rather than inferred from prose.

5. Discussion

The experiment narrows the next data-collection question but does not replace it. For text-to-SQL execution under noisy-input, the sign of the execution accuracy change remained positive in both the full set and the median slice. This supports a focused follow-up on schema drift using measured inputs and the same error-stratified estimate 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 median slice, and the error-stratified estimate 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 noisy-input generator. A real-data study should retain schema drift and the median slice 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. Xie, H., Zhou, X., Yang, J., Shen, S., Wang, Z., Zheng, Y., Xu, T., Shi, Y., Zong, Z., Li, Y., Chen, P., Jiang, J., He, D., Yan, X., & Jiang, J. (2026). SiriusDeliver: Automating Data Warehouse Delivery at Tencent. arXiv. https://doi.org/10.48550/arXiv.2608.09185

2. Zhou, F., Hu, S., Du, X., Li, N., Zhou, T., Zhao, Y., Shang, S., Ling, X., & Zhu, H. (2025). Nabil: A Text-to-SQL Model Based on Brain-Inspired Computing Techniques and Large Language Modeling. Electronics, 14(19), 3910. https://doi.org/10.3390/electronics14193910

3. 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

4. ŞAHİNASLAN, E., & ŞAHİNASLAN, Ö. (2022). Microsoft SQL Sunucusunda Veritabanı Kurtarma Teknikleri. International Journal of Innovative Engineering Applications, 6(1), 158-169. https://doi.org/10.46460/ijiea.1070325

5. Souto Prego, B., Vilares, D., & Cabado Lousa, B. (2024). Large Language Model Based Chatbot for Database Interaction through Natural Language. VII Congreso XoveTIC: impulsando el talento científico, 335-342. https://doi.org/10.17979/spudc.9788497498913.47

6. Muppala, M. (2025). ETL pipelines and SQL database management. SQL Database Mastery: Relational Architectures, Optimization Techniques,and Cloud-Based Applications, 84-101. https://doi.org/10.70593/978-93-7185-191-6_5

7. Ascoli, B. G., Kandikonda, Y. S. R., & Choi, J. D. (2025). ETM: Modern Insights into Perspective on Text-to-SQL Evaluation in the Age of Large Language Models. Future Internet, 17(8), 325. https://doi.org/10.3390/fi17080325

8. Ma, X., Tian, X., Wu, L., Wang, X., Tang, X., & Wang, J. (2024). Enhancing Text-to-SQL Capabilities of Large Language Models via Domain Database Knowledge Injection. Frontiers in Artificial Intelligence and Applications. https://doi.org/10.3233/faia240949

9. Sangamnerkar, B., & Namdev, S. (2025). Lightweight and Data-Efficient Fine-Tuning of Language Models for Bidirectional Text-to-SQL and SQL-to-Text Tasks: A Systematic Review. Advanced International Journal for Research, 6(6), 2745. https://doi.org/10.63363/aijfr.2025.v06i06.2745

10. Jiang, G., Li, W., Yu, C., Zhu, Z., & LI, W. (2025). FGCSQL: A Three-Stage Pipeline for Large Language Model-driven Chinese Text-to-SQL. https://doi.org/10.20944/preprints202502.1059.v1