Tuning constraint-aware reranking for text-to-SQL execution under 19% schema drift: a sensitivity sweep

ABSTRACT
We evaluated constraint-aware reranking for text-to-SQL execution under 19% schema drift. A deterministic paired simulation generated 48 cases and preserved a low-signal stratum. Mean execution accuracy changed from 0.559 to 0.598; the paired difference was +0.039 (95% interval +0.037 to +0.041). 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 ZhuJiu: A Multi-dimensional, Multi-faceted Chinese Benchmark for Large Language Models. 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 low-signal stratum. The operating setting is noisy-input; the stress level is fixed at 19% 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 50020. 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-R05-020.json`.

Reference [1] is used in Methods, not only as background: its work on ZhuJiu: A Multi-dimensional, Multi-faceted Chinese Benchmark for Large Language Models 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 Large Language Model Method as a Translator Indonesian Into SQL Language. Its evidence ledger records the specific descriptors padangsidimpuan, administrative, automatically, intuitively, background, encouraged, supporting, translated, translator, lecturers, flexibly, intended, obstacle, becomes, certain, command, massive, quickly; 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 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 reporting rule, [4] supplies abstract-backed evidence on sql, database through the study CIR-SQL: A Dual-Model Intent Recognition Framework for Chinese Text-to-SQL. Its evidence ledger records the specific descriptors backtracking, interference, containing, decoupling, monitoring, operators, frequent, speaking, control, status, leads, seven, sort, representations, classification, hierarchical, intermediate, maintenance; we map those archived descriptors to the schema drift control. For the baseline, [5] supplies abstract-backed evidence on sql, database through the study A comparative evaluation of large language models for detecting SQL injection vulnerabilities in web applications. Its evidence ledger records the specific descriptors vulnerabilities, confidential, intractable, potentially, circumvent, considered, manipulate, popularity, codellama, detecting, prominent, technique, emerging, payloads, produced, stronger, boolean, created; we map those archived descriptors to the schema drift control. For the measurement, [6] supplies abstract-backed evidence on sql, database through the study Benchmarking Large Language Models for NL-to-SQL: A Comprehensive Evaluation of Accuracy, Cost and Throughput. Its evidence ledger records the specific descriptors optimizations, practitioners, distribution, incorporated, alternative, deployments, exceptional, researchers, importance, underscore, dependent, exhibited, interplay, intricate, balanced, moderate, observed, revealed; we map those archived descriptors to the schema drift control. For the stress range, [7] supplies abstract-backed evidence on sql through the study Pengembangan Aplikasi Chatbot dengan Large Language Model untuk Text-to-SQL Generation. Its evidence ledger records the specific descriptors pengeksekusian, menerjemahkan, mengembangkan, menunjukkan, menyediakan, pemeriksaan, pemrograman, acceptable, pengecekan, pertanyaan, antarmuka, pemilihan, penulisan, pertanian, usability, aplikasi, kegunaan, kriteria; 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 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 reporting rule, [9] 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 baseline, [10] 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.

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 low-signal stratum. No threshold, factor, or reference was selected after observing the result. The sensitivity sweep 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 sensitivity sweep.

4. Results

The baseline mean was 0.559; the intervention mean was 0.598. The paired change was +0.039, with a 95% interval from +0.037 to +0.041. In the prespecified adverse slice, the change was +0.037.

The machine-readable artifact `experiments/01-R05-020.json` contains all 48 paired records for text-to-SQL execution. Consequently, the +0.039 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 low-signal stratum. This supports a focused follow-up on schema drift using measured inputs and the same sensitivity sweep 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 low-signal stratum, and the sensitivity sweep 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 low-signal stratum 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. Zhang, B., Xie, H., Du, P., Chen, J., Cao, P., Chen, Y., Liu, S., Liu, K., & Zhao, J. (2023). ZhuJiu: A Multi-dimensional, Multi-faceted Chinese Benchmark for Large Language Models. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 479-494. https://doi.org/10.18653/v1/2023.emnlp-demo.44

2. Putra, C., Arlis, S., & Nurcahyo, G. W. (2025). Large Language Model Method as a Translator Indonesian Into SQL Language. Jurnal KomtekInfo, 124-130. https://doi.org/10.35134/komtekinfo.v12i3.658

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

4. Wang, Y., Lv, H., & Qian, Y. (2026). CIR-SQL: A Dual-Model Intent Recognition Framework for Chinese Text-to-SQL. AI, 7(3), 91. https://doi.org/10.3390/ai7030091

5. Hairan, B., & Şahman, M. A. (2026). A comparative evaluation of large language models for detecting SQL injection vulnerabilities in web applications. PeerJ Computer Science, 12, e4015. https://doi.org/10.7717/peerj-cs.4015

6. Narasimhan, A., Bhamboo, A. K., Devnathan, A., Vellaisamy, J., & Vijayaraghavan, V. (2024). Benchmarking Large Language Models for NL-to-SQL: A Comprehensive Evaluation of Accuracy, Cost and Throughput. https://doi.org/10.36227/techrxiv.173121325.56335825/v1

7. Nugraha, G. P., Suadaa, L. H., Wilantika, N., & Maghfiroh, L. R. (2024). Pengembangan Aplikasi Chatbot dengan Large Language Model untuk Text-to-SQL Generation. Seminar Nasional Official Statistics, 2024(1), 831-840. https://doi.org/10.34123/semnasoffstat.v2024i1.2252

8. 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

9. 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

10. 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