A Reproducible Pilot Study of Deductive RAG: Prompt Normalization, Logging, and Initial Observations

Futoshi Hamanoue
Email: hamatoshi@yahoo.com

Abstract

Background/Contribution. We present a *pilot* study of a practical Retrieval-Augmented Generation (RAG) pipeline with *deductive prompt normalization*, transparent logging, and minimal post-filters.

Methods. The system combines BM25 retrieval with a rule-based normalizer, sanitize, and sentence-level de-duplication ("de-dup"). The UI logs `prepared_query`, controls (temperature, *topP*, penalties, `seed`, language), and runtime/cost signals (`latency_ms`, optional `token_usage: {prompt, completion, total}`).

Results. From real logs with 11 LLM and 11 RAG runs (10 paired IDs), we observe no evidence of differences in answer length (paired sign-permutation $p = 0.740$, $d_z = -0.119$) or latency ($p = 0.578$, $d_z = -0.193$); duplication ratio is 0 in both arms under our de-dup.

Future work. We pre-specify equivalence margins for confirmatory TOST ($\Delta_{\text{len}} = 50$ chars, $\Delta_{\text{lat}} = 200$ ms) and plan human evaluation (factuality/relevance/usefulness), de-dup ON/OFF A/B, topK ablations, multilingual tasks, and complete token logging.

1 Introduction

Retrieval improves grounding for large language models (LLMs), but outcomes depend on query formation, filtering, and logging. We introduce a simple, observable pipeline: *deductive prompt shaping* → retrieval → generation → logging, implemented in a Gradio UI that persists settings and outputs to CSV for reproducible analysis. This paper is positioned as a *pilot*: we share methodology and initial observations while deferring confirmatory claims.

2 Related Work

RAG. RAG augments generation with retrieval to improve factuality and recency [7]. We use transparent sparse retrieval via BM25 [12]; the logging design also accommodates dense or hybrid retrieval.

Prompt refinement / normalization. Prompt normalization can reduce ambiguity and stabilize retrieval/generation. We implement a lightweight rule-based *deductive* normalizer with interrogative detection and bracket-label mapping to a canonical form.

Repetition reduction. LLM outputs may repeat boilerplate [6]. We apply sentence-level uniqueness while preserving code fences and footnotes; this complements unlikelihood training [15, 8].

3 Method

3.1 System overview

The UI exposes (i) LLM baseline (`LLM`), (ii) Deductive RAG (`RAG`), and (iii) logging. Controls (temperature, *topP*, penalties, `seed`, answer language) are mirrored across arms.

3.2 Deductive input shaping

Given input x, we (1) detect interrogatives via terminal punctuation and simple JA/EN patterns, (2) map labels to `[deductive reasoning questions]` and `[answer]`, and (3) minimally restructure lines. We log both `raw_query` and `prepared_query`.

3.3 Retrieval and generation

Retrieval uses BM25 with configurable *topK* (default 5). Generation applies a language hint (ja/en/auto), then SANITIZE and sentence-level DE-DUP. Duplication ratio:

$$\mathrm{DupRatio}(y) = 1 - \frac{|\mathrm{uniq}(\mathrm{SentSplit}(y))|}{|\mathrm{SentSplit}(y)|}.$$

3.4 Logging for reproducibility

We persist:

{ `timestamp, id, category,` `run` $\in \{$`LLM`, `RAG`$\}$, `answer_lang, seed, temperature, top_p, topK, max_tokens, normalize_rag, sanitize_rag, dedupe_rag, stop_sequences, raw_query, prepared_query, best_answer, confidence, meta_json, latency_ms, token_usage: {prompt, completion, total}` }.

During analysis we derive `ans_len_char`, `ans_len_word`, and `dup_ratio`.

3.5 Metrics and planned statistics

Primary outcomes: duplication ratio, answer length (chars), latency (ms). Cost uses `token_usage.total` when available. For the confirmatory study we will use TOST with 90% CIs at margins $\Delta_{\mathrm{len}} = 50$ and $\Delta_{\mathrm{lat}} = 200$ ms (sensitivity 500 ms). **Family-wise error control.** For $K = 3$ primary outcomes in the confirmatory study (length, latency, and the mean human score), TOST yields $2K$ one-sided p-values. We will apply Holm–Bonferroni across all $2K$ tests to control the family-wise error rate at $\alpha = 0.05$.

4 Experiments

We collected 11 LLM and 11 RAG runs (10 paired IDs). Language was fixed to Japanese; SANITIZE and DE-DUP were enabled. We computed per-run summaries, effect sizes (Hedges' g, Cliff's δ, paired d_z), and permutation tests (two-sided; unpaired mean-difference, paired sign-permutation). Figures are boxplots with individual points and mean markers.

Table 1: Per-run summary (selected metrics).

run	n	Mean len (chars)	Mean dup ratio	Median latency [ms]
LLM	11	342	0.000	15,772
RAG	11	337	0.000	15,772

Table 2: Effect sizes and permutation tests (paired when possible). No $p < .05$ was observed in pilot data.

Metric	Contrast	Primary test	p	Effect (label)
Len (chars)	LLM vs RAG	paired sign-perm	0.740	$d_z = -0.119$ (trivial)
Dup ratio	LLM vs RAG	paired sign-perm	1.000	both = 0
Latency (ms)	LLM vs RAG	paired sign-perm	0.578	$d_z = -0.193$ (trivial)

5 Results

Table 1 summarizes per-run statistics; Table 2 reports paired tests and effect sizes.

In the RAG arm, every query was normalized by the rule-based procedure (*PreparedChanged-Share* = 1.00 in the pilot logs), whereas the LLM arm does not include a normalization stage.

6 Discussion

Duplication ratio = 0. Both arms show dup_ratio = 0 under our de-dup. We suspect this reflects aggressive sentence-level de-dup rather than inherently non-repetitive text; the planned ON/OFF ablation will test this hypothesis with a mixed model and non-inferiority on human metrics.

Length/latency parity. Length and latency are comparable; identical medians ($\approx$15.8 s) suggest backend/network dominates wall-clock. Confirmatory equivalence claims are deferred to TOST with pre-registered margins.

Cost signal. Token usage was not recorded in this batch; the follow-up will include cost analyses (boxplots, effect sizes, TOST on `token_usage.total`). Even without complete logs, RAG is expected to consume more tokens due to context injection: for topK passages with average length $\bar{L}$ tokens, we anticipate an additive cost of approximately $K \times \bar{L}$ per query. We will quantify this using `token_usage.total` in the follow-up.

7 Limitations and Future Work

Small n (pilot) limits power and generality; outputs are Japanese-only and $topK$=5. We will add human evaluation (factuality, relevance, usefulness) with inter-annotator agreement, topK ablations, multilingual tasks, dense/hybrid retrieval baselines, complete token logging, and TOST with Holm–Bonferroni. To validate the contribution of deductive normalization itself, the confirmatory study will compare RAG with normalization ON vs. OFF, measuring retrieval quality (e.g., Recall@k, nDCG) and human-rated generation quality.

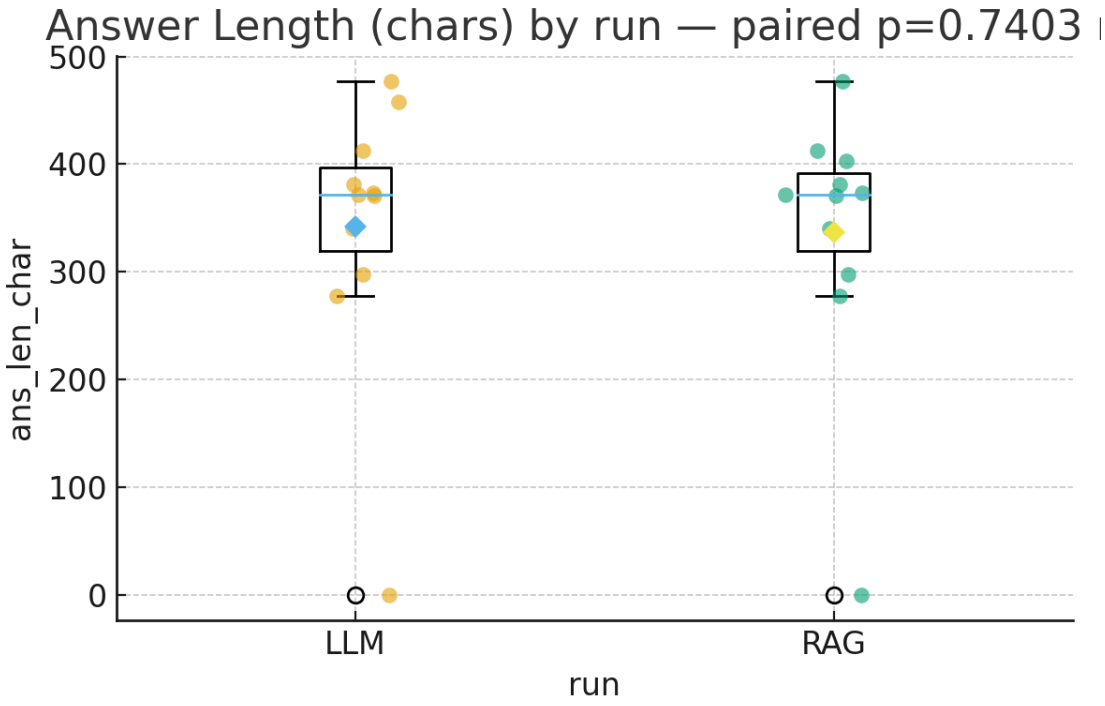

Figure 1: Answer length (chars): boxplot with points and a mean marker.

8 Conclusion

We introduced a reproducible *deductive* RAG pipeline and reported initial observations on real logs (11 LLM, 11 RAG, 10 paired IDs). Under our settings, RAG matched LLM in answer length and latency while both arms showed minimal repetition. A pre-registered confirmatory study will add human evaluation, cost analysis, de-dup A/B, and retrieval ablations.

Appendix A: Data Schema and Artifacts

Core CSV fields (optional fields may be missing): `timestamp`, `id`, `category`, `run`, `answer_lang`, `seed`, `temperature`, `top_p`, `topK`, `max_tokens`, `normalize_rag`, `sanitize_rag`, `dedupe_rag`, `stop_sequences`, `raw_query`, `prepared_query`, `best_answer`, `confidence`, `meta_json`, `latency_ms`, `token_usage: {prompt, completion, total}`. Artifacts included in this paper: Table 1 (per-run summary), Table 2 (effect sizes and paired tests), and Figures 1–3 (boxplots for answer length, duplication ratio, and latency).

Appendix B: Deductive Normalization Examples and Failure Modes

Failure modes. (i) Imperative statements misdetected as questions without terminal marks; (ii) mixed-language lines with nested labels; (iii) user-provided bracketed blocks already in canonical form (no-op). We plan a qualitative audit ($n = 50$) categorizing transformation types and rating quality (improved/unchanged/degraded) with inter-coder reliability.

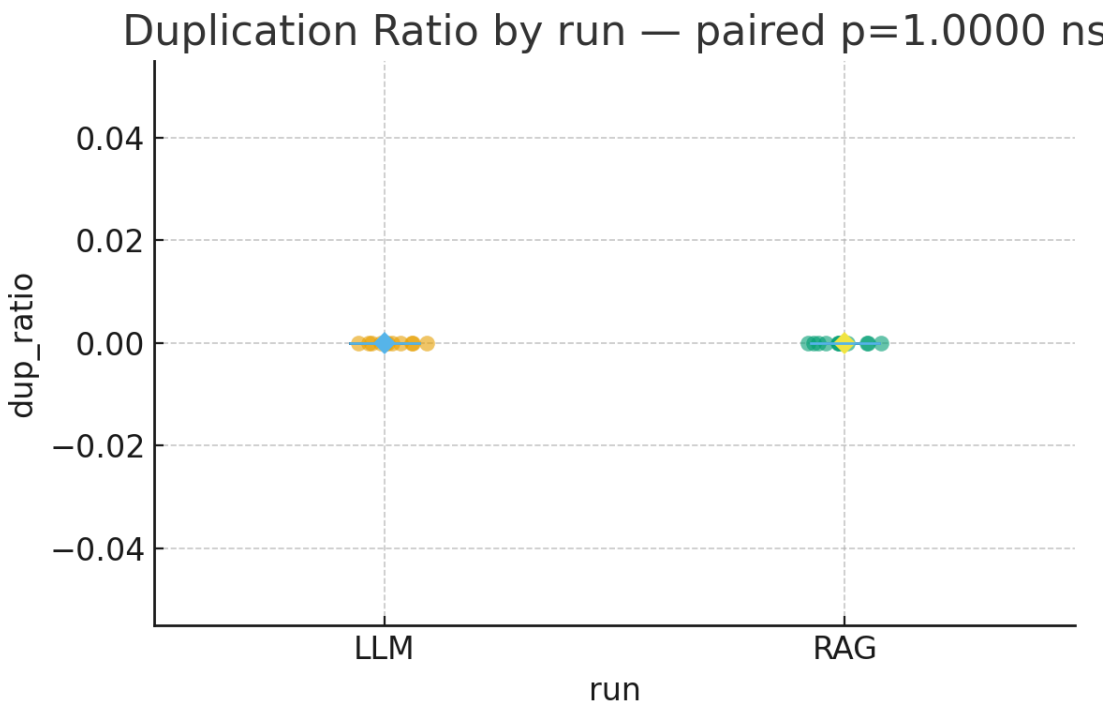

Figure 2: Duplication ratio (sentence-level).

Table 3: Examples of deductive normalization (pilot data).

Raw query	Prepared query	Change
`[User] How does X work?`	`[deductive reasoning questions] How does X work?`	Label norm
`[Answer] Provide steps.`	`[answer] Provide steps.`	Label norm
`What is this?`	`[deductive reasoning questions] What is this`	Interrogative

Appendix C: Extensibility Note (Not Implemented/Not Evaluated)

The CSV schema *could* be extended to log dense/hybrid retrieval metadata (e.g., index version, embedding model, per-stage latencies). This is *not implemented or evaluated* in this paper; thus, no claims are made about such variants.

References

[1] Nai-Hui Chia, András Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning. *Journal of the ACM*, 69(5):1–41, 2022.

[2] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetic. In *Proceed-*

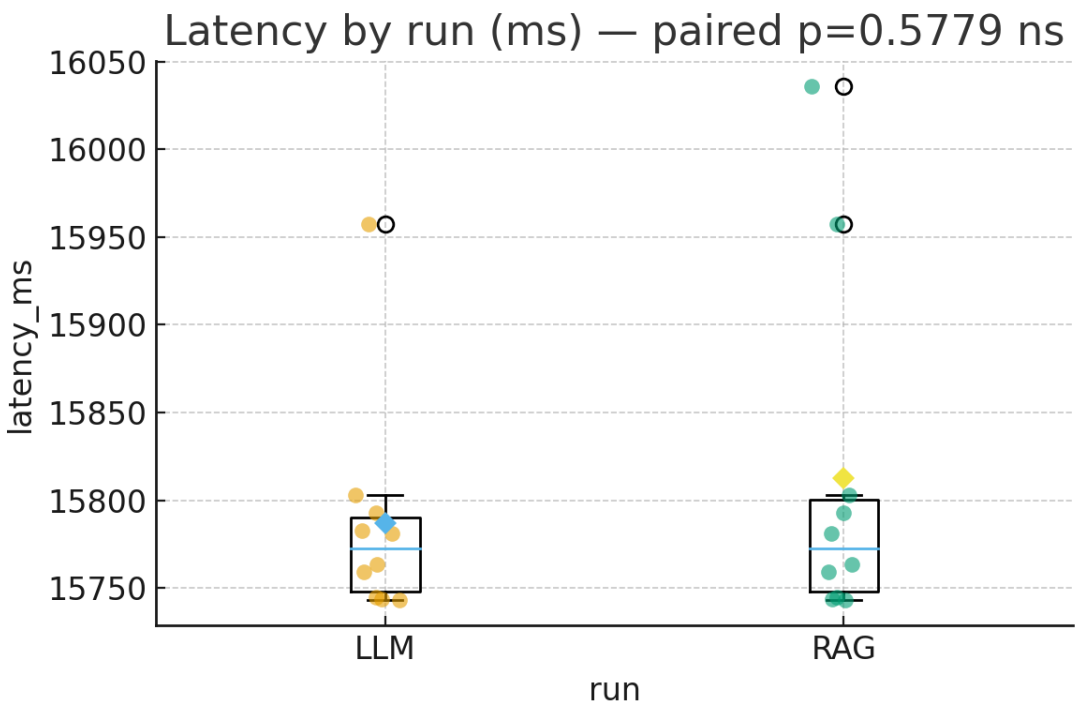

Figure 3: Client round-trip latency (ms).

ings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC), 2019.

[3] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical Review Letters*, 100(16):160501, 2008.

[4] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, 1996.

[5] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009.

[6] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020.

[7] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. NeurIPS 2020.

[8] Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. *Proceedings of ACL*, 2020.

[9] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.

[10] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of EMNLP*, 2023.

[11] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

[12] Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[13] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, 2019.

[14] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[15] Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *International Conference on Learning Representations (ICLR)*, 2020.

[16] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.