# One Database, Three Retrievers: When Hybrid and Graph Help a Local-Knowledge Agent—and When They Hurt

**Animesh Alang**
University of California, Santa Cruz
aalang@ucsc.edu

## Abstract

Local questions mix place type, neighborhood, live hours, and "places like $X$." REGULAR is a deployed SMS/iMessage agent for Santa Cruz County whose retrieval stack lives in one PostgreSQL 17 instance: exact dense search (pgContext) over local bge-small embeddings, multi-field full-text lists fused by reciprocal rank fusion (RRF), and a property graph (pgGraph) with schema and embedding-derived edges. We ask which of these arms help on compositional local queries. On **366 simulated** queries across four shapes, equal-weight hybrid RRF *hurts* similarity-style asks (Recall@5 $0.583 \rightarrow 0.438$ on `similar_open`) but *helps* type and award asks; upweighting the dense list (weight 4 vs. 1 per lexical list) recovers much of the loss and significantly beats equal hybrid overall. Adding graph traversal on top of dense search (`dense+graph`) is the best overall configuration (Hit@1 $0.475$, MRR $0.585$), with a significant Hit@1 gain over dense-only on bare "somewhere like $X$" queries ($+0.092$, 95% CI $[+0.008, +0.175]$). We report a failure taxonomy with paraphrased examples, recorded end-to-end behavioral results (we do not re-run the agent harness because it writes throwaway users), and position the design against hybrid fusion analysis and LLM-extracted GraphRAG.

## 1 Introduction

Local assistants fail in expensive ways: wrong hours, invented venues, answers that ignore "open now" (Ji et al., 2023). Retrieval-augmented generation (Lewis et al., 2020) and tool-using agents (Yao et al., 2023; Schick et al., 2023) help only if the retriever matches the query shape. A common production pattern splits dense vectors, a search engine, and a graph database. We instead keep all three inside Postgres, next to the relational source of truth, so the same hard filters (status, neighborhood, event start, open-at) apply to every arm.

**RQ1.** Does equal-weight hybrid RRF beat dense-only on compositional local queries? **RQ2.** Does upweighting the dense list in RRF fix the failure mode of RQ1? **RQ3.** Does graph traversal over embedding-derived *similar_to* edges improve Hit@1 when added to dense search, and when added to hybrid?

Contributions: (1) a concise architecture of a deployed single-DB stack; (2) a 366-query, four-shape retriever ablation with dense, hybrid, weighted hybrid, and graph combinations, all numbers from a read-only run against production on 2026-11-05; (3) error analysis with paraphrased examples (no private rows); (4) recorded behavioral eval results, clearly separated from the RQ.

## 2 Related work

Dense passage retrieval (Karpukhin et al., 2020) and BM25-style lexical ranking (Robertson and Zaragoza, 2009) are routinely fused. Cormack et al. introduce RRF (Cormack et al., 2009); Chen et al. advocate zero-shot hybrid with RRF (Chen et al., 2022); Bruch et al. show that a tuned convex combination of normalized scores often beats untuned RRF and that RRF is parameter-sensitive (Bruch et al., 2023). Our weighted-RRF cell is a rank-space cousin of that idea: we keep RRF's scale-free form but give the dense list four votes and each lexical list one. Ma et al. argue for a single Lucene stack for dense and sparse retrieval (Ma et al., 2023); we make the same "one process" bet inside Postgres. GraphRAG builds LLM-extracted entity graphs for corpus-level sensemaking (Edge et al., 2024). Our graph is the opposite: foreign keys plus top-5 same-domain embedding neighbors, refreshed on a schedule, with no LLM extraction. BEIR-style zero-shot IR (Thakur et al., 2021) is out of scope; our gold is SQL-derived over a private local inventory.

## 3 System

**Data.** A single-transaction importer loads a private local dataset (not redistributed). At ablation time the DB held 5,679 venues, 1,433 events, 46 neighborhoods, 2,684 venues with parsed weekly hours, and 21,777 *similar_to* edges. Hours parse ∼40 textual formats; ambiguity yields *unparsed* with no rows, so `venue_open_at` returns NULL rather than a guess.

**Dense.** Documents and queries are embedded locally with bge-small-en-v1.5 (Xiao et al., 2024) (384-d, quantized, cached by content hash). pgContext 0.3.0 exact search over collections *venues*/*events*/*knowledge*; at ∼7k vectors, HNSW (Malkov and Yashunin, 2020) is unnecessary. Dense candidates beyond cosine distance 0.45 are dropped (keep ≥5).

**Lexical and fusion.** Up to eight Postgres full-text lists (name/description, kind/type/cuisine/tags, reviews, listing details, . . . ) are fused by RRF with $k$=60 (Cormack et al., 2009). Production uses equal weights; this paper also tests a dense-heavy variant.

**Graph.** pgGraph 1.2.1 projects relational tables: *in_neighborhood*, *held_at*, *awarded_to*, *storefront_of*, *about_venue*, and *similar_to* (each venue's five nearest same-domain neighbours by embedding, weight = cosine). `explore_connections` runs `graph.traverse` and `graph.shortest_path`; pg_cron refreshes every five minutes.

**Agent.** A tool-calling LLM via OpenRouter (default `deepseek/deepseek-v4.1-flash`) chooses among ten tools. Deterministic guards strip ungrounded names, phones, and unsourced claims. Replies go out as SMS/iMessage via SendBlue.

## 4 Experiments

### 4.1 Behavioral suite (recorded; not re-run)

`scripts/eval.ts` drives hand-written cases through the live agent and checks tool choice, required/forbidden content, honesty, memory, and grounding. The suite now has 68 cases (v2); the runs we have on disk are from 2026-10-20 on a 34-case subset. We **did not re-run** the harness for this paper: it inserts and deletes throwaway `eval-*` users, which would modify production state. Table 1 therefore reports those recorded runs only. Only one case (*places like Verve?*) expects `explore_connections`, so these numbers do not answer RQ1–3.

| Model (OpenRouter) | Pass | p50 s | p95 s |
|---|---|---|---|
| deepseek-v4.1-flash (first) | 82.4% | 7.0 | 40.1 |
| deepseek-v4.1-flash (latest) | 94.1% | 3.2 | 10.4 |
| claude-haiku-5.5 | 97.1% | 4.7 | 9.2 |
| qwen3.8-flash | 94.1% | 11.9 | 25.5 |
| qwen3.7-flash | 67.6% | 20.9 | 34.2 |

Table 1: Recorded end-to-end results, 34-case behavioral suite, 2026-10-20. Latest deepseek run cost $0.069. Adversarial: 6/6 on latest deepseek and haiku.

### 4.2 Retriever ablation (simulated queries)

We isolate retrieval (no LLM). All numbers below come from a read-only SQL run against production on 2026-11-05 (`SET default_transaction_read_only=on`). **Queries are simulated**; gold is SQL, not human judgments.

**Shapes (366 queries).**

1. `similar_open` (120): "somewhere like [SEED] in [hood] that's open saturday around 7pm." Gold: other same domain+subtype+hood venues open Sat 19:00. Mean |gold| = 1.79.
2. `similar_bare` (120): "somewhere like [SEED]." Gold: county-wide same subtype. Mean |gold| = 30.2 (so Recall@5 is low by construction; prefer Hit@1/MRR).
3. `type_hood_open` (66): "[subtype] place in [hood] that's open saturday around 7pm." Gold: all such open venues. Mean |gold| = 2.42.
4. `award_hood` (60): "award-winning [category] in [hood]." Gold: venues holding that award in that hood. Mean |gold| = 2.33.

Seeds for (1)–(2) were sampled with a fixed hash order from food/nightlife venues that have a subtype, a neighborhood, and ≥1 open peer; (3)–(4) from groups with ≥2 gold venues.

**Configurations.** Hard filters (domain, optional neighborhood, not closed/hidden, optional open-at) are identical across arms; seed venues are excluded from candidates on similarity shapes.

- **A** dense (distance-capped).

| Config | R@5 | P@5 | Hit@1 | MRR |
|---|---|---|---|---|
| A dense | 0.435 | 0.290 | 0.456 | 0.553 |
| B hybrid = | 0.416 | 0.254 | 0.352 | 0.461 |
| C hybrid+graph = | 0.432 | 0.257 | 0.393 | 0.497 |
| **DG** dense+graph | **0.449** | 0.287 | **0.475** | **0.585** |
| W weighted hybrid | 0.440 | 0.282 | 0.402 | 0.516 |
| WG weighted+graph | 0.441 | 0.282 | 0.421 | 0.536 |

Table 2: Overall retrieval ablation, 366 simulated queries.

| Shape | Config | R@5 | Hit@1 | MRR |
|---|---|---|---|---|
| similar_open ($n$=120) | A | **0.583** | 0.408 | 0.499 |
| | B | 0.438 | 0.233 | 0.340 |
| | C | 0.466 | 0.292 | 0.387 |
| | DG | **0.633** | **0.425** | **0.543** |
| | W | 0.492 | 0.308 | 0.411 |
| | WG | 0.506 | 0.325 | 0.433 |
| similar_bare ($n$=120) | A | 0.088 | 0.433 | 0.529 |
| | B | 0.094 | 0.317 | 0.395 |
| | C | 0.109 | 0.383 | 0.461 |
| | DG | **0.128** | **0.525** | **0.621** |
| | W | 0.100 | 0.367 | 0.470 |
| | WG | 0.115 | 0.433 | 0.531 |
| type_hood_open ($n$=66) | A | 0.858 | **0.773** | **0.849** |
| | B | **0.902** | 0.712 | 0.815 |
| | W | 0.901 | 0.667 | 0.801 |
| | DG | 0.793 | 0.742 | 0.825 |
| award_hood ($n$=60) | A | 0.368 | 0.250 | 0.384 |
| | B | 0.481 | 0.267 | 0.444 |
| | W | **0.508** | **0.367** | **0.505** |
| | DG | 0.343 | 0.183 | 0.333 |

Table 3: Per-shape results (selected rows). Full six-config tables in the ablation directory.

- **B** hybrid equal RRF: A + four lexical lists (name, kind, reviews, details).
- **C** B + graph list.
- **DG** dense + graph.
- **W** weighted hybrid: dense weight 4, each lexical list weight 1.
- **WG** W + graph weight 2.

Graph start: oracle seed id on similarity shapes (live agent instead fuzzy-matches the name); dense top-1 cascade on type/award shapes. Queries were embedded on a separate machine with the same quantized bge-small model. Metrics: Recall@5, P@5, Hit@1, MRR. Significance: paired bootstrap, 10k resamples, 95% CI; win/tie/loss counts on the metric.

**Significance (selected).** Overall, W−B Recall@5 $+0.024$ $[+0.007, +0.042]$ (61/288/17); C−B Hit@1 $+0.041$ $[+0.008, +0.074]$. On `similar_bare`, DG−A Hit@1 $+0.092$ $[+0.008, +0.175]$ (18/95/7). On `similar_open`, W−B Recall@5 $+0.054$ $[+0.018, +0.096]$, but W−A Recall@5 $-0.090$ $[-0.165, -0.014]$: weighting helps vs. equal hybrid, not enough to beat dense. On `award_hood`, W−A Recall@5 $+0.140$ $[+0.043, +0.242]$. Overall DG−A Hit@1 $+0.019$ $[-0.022, +0.060]$ is not significant; the gain is concentrated on similarity shapes.

### 4.3 Error analysis

We paraphrase queries and never print private venue rows.

**Lexical name leak (`similar_open`).** Equal hybrid loses Recall@5 to dense on 36/120 queries. OR-matching the seed's own name plus stop-ish tokens ("saturday", "open") promotes unrelated venues; four lexical lists outvote dense. Example pattern: "somewhere like [SEED] in Watsonville that's open saturday around 7pm" (subtype *convenience_store*): A Recall@5 $= 1.0$, B $= 0.0$, W $= 1.0$ (weighting restores dense). Another: pizza seed in West Side: A 0.67, B 0.0, W 0.0 (weighting insufficient).

**Lexical helps when the query *is* a type.** On `type_hood_open`, B beats A on Recall@5 for 9 queries and loses on 6. Examples: "pizza place in Freedom…" A $0.0 \rightarrow$B 1.0; "burger place in East Side…" A $0.0 \rightarrow$B 1.0; "italian place in Downtown Santa Cruz…" A $0.5 \rightarrow$B 1.0. Award queries show the same direction: W is best overall on that shape.

**Graph helps bare similarity.** On `similar_bare`, DG raises Hit@1 on 18 queries and lowers it on 7 vs. A. On `similar_open`, DG helps Hit@1 on 9 and hurts on 7; several wins are re-rankings where gold was already in the top 5 (Hit@1 flips, Recall@5 flat). Dense-top-1 graph cascade on type/award shapes rarely helps and sometimes dilutes: those queries want keyword precision, not neighbor expansion.

**Takeaway for the agent.** Gate lexical lists by query class: keep them for type/cuisine/award asks; downweight or drop them when the query names a reference venue. Prefer dense+graph for "places like $X$"; keep hybrid (preferably weighted) for typed lookup.

## 5 Discussion and limitations

Keeping three retrievers in one database makes filters identical across arms and removes cross-service sync; the cost is dependence on young extensions. Gold "same subtype" is a proxy for similarity, not human judgments; similarity shapes use oracle entity linking; type/award graph uses dense-top-1, not a learned router. Lexical lists here are four of the production set. Behavioral evidence is a recorded 34-case slice of a 68-case suite and does not test the RQ. Single county, one embedder, simulated queries. We do not claim workshop-maintrack strength until a small real-user study exists.

## Ethics Statement

The local dataset is privately owned and not released. Ablation SQL was read-only. Eval users in recorded runs were throwaway accounts; no real user messages appear here. Paraphrased examples omit venue names.

## Acknowledgments

Thanks to collaborators on the local data pipeline. Errors are mine.

## References

Sebastian Bruch, Siyu Gai, and Amir Ingber. 2023. An analysis of fusion functions for hybrid retrieval. *ACM Transactions on Information Systems*, 42(1):1–35.

Tao Chen, Mingyang Zhang, Jing Lu, Michael Bendersky, and Marc Najork. 2022. Out-of-domain semantics to the rescue! zero-shot hybrid retrieval models. In *Proceedings of ECIR*, pages 95–110.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of SIGIR*, pages 758–759.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12).

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, pages 6769–6781.

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. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*.

Xueguang Ma, Tommaso Teofili, and Jimmy Lin. 2023. Anserini gets dense retrieval: Integration of Lucene's HNSW indexes. In *Proceedings of CIKM*.

Yu A. Malkov and Dmitry A. Yashunin. 2020. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the NeurIPS Datasets and Benchmarks Track*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed resources for general Chinese embeddings. In *Proceedings of SIGIR*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of ICLR*.