Exploring Feedback-Aware Retrieval for IT Support First-Reply Generation

Domen Jeršek
domenjersek@gmail.com
Jožef Stefan Institute
Slovenia

Blaž Kažič
blaz.kazic@ijs.si
Jožef Stefan Institute
Slovenia

Klemen Kenda
klemen.kenda@ijs.si
Jožef Stefan Institute
Slovenia

Abstract

We examine *when* adding feedback signals to retrieval-augmented generation (RAG) helps the generated first reply to IT support tickets and when it hurts. Using a FAISS-based pipeline on 250 tickets under strict leave-one-out isolation, we compute a Laplace-corrected Bayesian popularity lift from 102,722 LLM-judge-scored (query, candidate) pairs drawn from an approximately 1,600-ticket knowledge base. The lift replaces the top-5 FAISS ranking (98 % of tickets change candidates, median promoted rank 17). In an exploratory analysis, we find that feedback popularity and semantic similarity are orthogonal ($r = 0.06$), and an oracle signal exists: feedback helps a weak baseline (+0.21 in the weakest decile) and hurts a strong one (-0.09; $r = -0.41$). We test cosine-based gating (it fails because the retrieval-to-answer path, $r = 0.24$, is too weak) and pool-level filtering (M1 global, M2 team, M3 type, M4 team∩type). Pool analysis reveals why some pools benefit: query diversity ($r = -0.38$) and procedural mismatch ($r = 0.21$) are candidate gating features observable before generation. A pool-based oracle gate reaches +0.017 in-sample, motivating a learned gating model as the natural next step.

Keywords

IT support, retrieval-augmented generation, LLM-judge feedback, retrieval replacement, exploratory analysis

1 Introduction

Enterprise IT-support teams answer thousands of tickets, many nearly identical to tickets answered before. RAG is appealing: retrieve the most similar historical tickets and let a language model write the first reply. The implicit assumption is that *textual* similarity tracks *procedural* similarity.

That assumption can fail. A ticket saying *"I cannot open* `.ica` *files in Citrix"* retrieves Citrix *troubleshooting* neighbours that share its vocabulary. But the correct reply is not a troubleshooting guide; it is a form redirect to a software-installation request. The right template lives among less similar neighbours (we return to this exact ticket in §5).

Feedback offers a way out: if some past replies are known to be more useful, a candidate whose reply was rated useful could be promoted even if textually dissimilar. Our prior work [1] demonstrated a RAG pipeline for this domain; here we add LLM-judge feedback as a retrieval modifier and ask a diagnostic question:

> *When does adding feedback to retrieval improve the generated first reply, and when does it hurt — and can we tell the difference before we generate anything?*

This is an exploratory paper structured around three research questions. **RQ1**: Does feedback-aware retrieval improve first-reply generation over semantic retrieval alone? We compare four feedback-pooling scopes (M1 global, M2 team, M3 type, M4 team∩type). **RQ2**: Under what conditions does feedback help or hurt? We test baseline quality, team/type membership, procedural mismatch, and within-pool query diversity as explanatory factors. **RQ3**: Can the benefit be predicted before generation? We sweep retrieval-margin thresholds, pool-identity gates, and text-derived features, finding that simple retrieval-only signals are insufficient. The central finding remains: feedback and semantic similarity are orthogonal signals ($r = 0.06$), making feedback-driven retrieval an unconditional trade-off.

2 Related Work

RAG [3] combines parametric LLM knowledge with external retrieval. Dense retrieval with sentence embeddings [5, 6] and FAISS indexing [2] provide efficient semantic search. Feedback-for-retrieval is under-explored compared to feedback-for-generation (RLHF [4]). We use structured procedural ratings from an LLM judge [7] and ask the *conditional* question of when feedback helps. Jeršek et al. [1] built the same RAG pipeline for this domain without feedback; we extend it with LLM-judge feedback signals. The finding that retrieval confidence weakly predicts answer quality aligns with the broader observation that retrieval-time similarity metrics routinely under-predict downstream accuracy.

3 Data

Data flow. After cleaning 1,989 raw tickets for missing data, the KB contains **1,596 tickets** (title, description, historical first reply). A FAISS index is built over this full set. **250 query tickets** are randomly selected for evaluation; a query's own ticket is excluded from its candidate pool during evaluation (standard LOO).

KB pool distribution. The KB spans 37 resolver teams and 5 first-reply intent labels (derived by co-occurrence clustering of resolver-team×Service-field-value pairs). `other` (747 tickets, 47 %) and `vpn_request` (586, 37 %) dominate; `software_request` (133, 8 %), `admin_rights` (117, 7 %), and `email_support` (13, <1 %) make up the remainder. Across 75 team×intent combinations, only 15 % are singletons — broader than the query set (38 % singletons), which later explains why M4 starves for evidence (§6.2).

Feedback database. An LLM judge (gpt-4o-mini, temperature 0) scored each (query, candidate) pair. It received the query title, description, the *ground-truth expected first reply*, and candidate tickets with their first replies, returning a score in $[0, 1]$. This LLM-judge feedback simulates expert evaluation: the judge acts as a domain expert, comparing candidate replies against the ground-truth reply written by the original resolver. **Importantly, the judge evaluated retrieval candidates (not generated answers)**: it compared each candidate ticket's historical first reply to the expected reply, producing a retrieval-level quality

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.96

score. Scores from approximately 225 batch evaluation runs were merged into 102,722 scored rows (250 queries, approximately 1,580 physical candidates, 58,318 unique query–candidate pairs). A candidate is positive (mean ≥ 0.80) or negative (mean ≤ 0.40); the middle band is ignored. Repeated scoring does not inflate n: the per-pair mean determines the label. Because both the judge and the cosine evaluation metric reference this ground truth, we validate all findings under ROUGE-L and BERTScore (§5), confirming they are not an artefact of scoring against the evaluation target.

4 Methodology

4.1 Pipeline

The system is a two-stage pipeline (Figure 1). **Retrieval:** Title and description are embedded with all-MiniLM-L6-v2 [6, 5]. The top-5 most similar tickets are retrieved from a FAISS index [2]. The *baseline* path returns these directly; the *feedback* path **replaces** the ranking using FAISS score plus a feedback lift. **Generation:** The top-5 are formatted as few-shot examples (title + first reply) and passed to gpt-5.6-luna via OpenRouter.

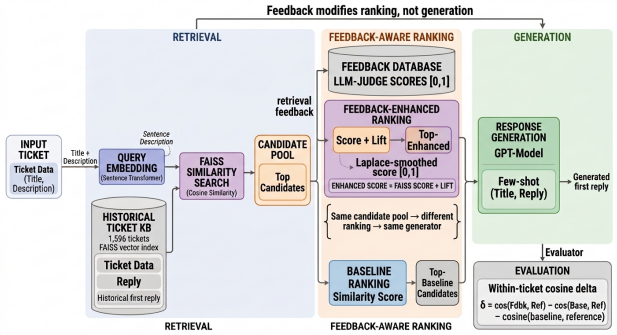


Figure 1: System architecture. Baseline and feedback differ only in the ranking of the same candidate pool.

4.2 The lift formula

For candidate c and query q, let pos and neg be positive/negative counts, $n = pos + neg$. The lift is:

$$p = \frac{pos + \alpha}{pos + neg + \alpha + \beta}, \quad \text{lift} = (p - 0.5) \cdot \min(1,\, n/2) \cdot m,$$

clamped to $\pm C, with\ \alpha = \beta = 1.0$, $m = 0.80$, $C = 0.20$. The enhanced score is $\text{FAISS}(c, q) + \text{lift}(c)$; top-5 by enhanced score become the feedback retrieval.

We define **fidelity** as the Pearson correlation between the continuous mean judge score and the assigned lift, measuring how faithfully the formula preserves score ordering. We compared three candidate formulas on 24,111 lift values from 100 sampled queries (sufficient for formula selection, since the per-candidate score distribution stabilises well below 250 queries): Laplace, tanh shrinkage, and Bayesian lower confidence bound (LCB). Laplace has the widest dynamic range ($\sigma = 0.090$ vs. tanh $\sigma = 0.024$) and the highest fidelity to the underlying scores ($r = 0.85$ vs. tanh 0.72, LCB 0.68). It is weakly driven by observation count ($r = 0.19$ to n vs. tanh 0.31, LCB 0.45). The alternatives fail structurally: tanh compresses nearly all values toward zero, making candidates effectively indistinguishable; Bayesian LCB collapses 82 % of lifts to exactly zero, destroying ranking resolution. A limitation: quantization into discrete values ($0, \pm 0.067, \pm 0.133, \pm 0.200$) due to integer counts and $\min(1, n/2)$. The parameters $m = 0.80$, $C = 0.20$, and thresholds 0.80/0.40 were fixed a priori.

Filtering scopes (M1–M4). The lift can be pooled at four granularities:

- **M1 (global):** all 250 queries contribute — primary scope, largest n.
- **M2 (team-only):** only queries from the same resolver team.
- **M3 (type-only):** only queries with the same intent label.
- **M4 (team∩type):** strict intersection — smallest pools, starves for 33 % of tickets where no feedback lift exists.

M2–M4 use known team/type labels to isolate scope effects without classifier noise.

4.3 Evaluation

Evaluation is strict leave-one-out (250 folds). For each query, its own feedback rows and ticket are removed from the pool (for M2–M4 this means the query's own vote is excluded from its team or type pool). The candidate pool and generator are identical between baseline and feedback; only the ranking differs. The primary metric is within-ticket cosine delta: $\delta(t) = (fb_t, ref_t) - (base_t, ref_t)$. Cosine correlates $r > 0.85$ with ROUGE-L and BERTScore.

5 Exploratory Analysis

To answer RQ2, we first characterise how feedback changes retrieval, then examine the conditions under which the difference is beneficial.

5.1 Feedback replaces retrieval

Feedback changes at least one candidate for **98 % of tickets** (mean overlap 1.71/5, Figure 2).

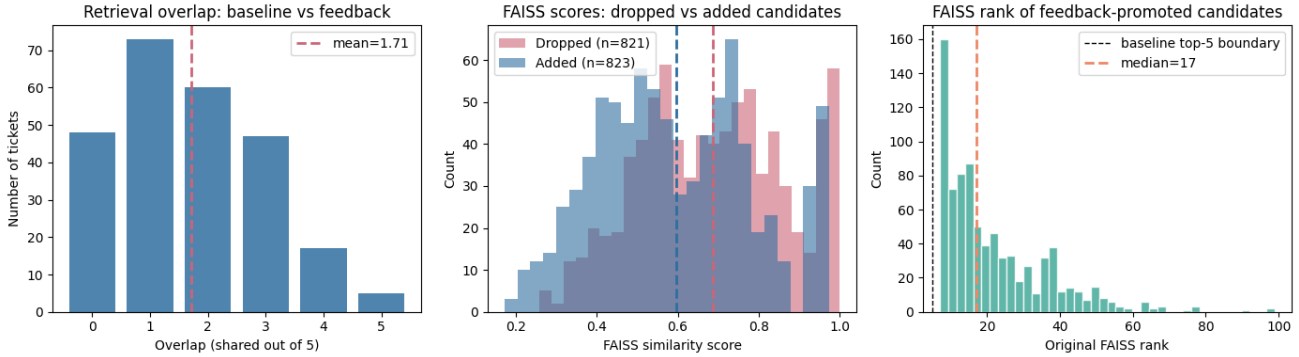


Figure 2: Feedback replaces retrieval, not reorders it. Promoted candidates have lower FAISS scores and come from deep in the FAISS ranking (median rank 17).

Promoted candidates have *lower* semantic similarity than those they replace (added 0.60 vs. dropped 0.69) and come from deep in the FAISS ranking (median rank 17). This happens because feedback popularity — $(pos - neg)/(pos + neg + 1)$ — and FAISS similarity are uncorrelated ($r = 0.058$ across 1,250 pairs). Only 5 % of candidates are in the top quarter of both.

5.2 Oracle signal and retrieval-vs-answer relationship

We define **oracle** as any post-generation quantity that distinguishes tickets where feedback helps from those where it harms. Figure 3 (left) shows the key finding: delta exhibits a clear negative trend against baseline answer cosine ($r = -0.41$). The weakest decile gains **+0.21** (76 % improved), the strongest loses **−0.09** (16 % improved). Feedback helps when the baseline is weak and hurts when it is strong. Before testing cosine-based gating, we examine the retrieval-to-answer-to-delta chain that any gate must traverse. Figure 3 (right) shows the first link: top-1 FAISS similarity predicts baseline answer cosine at $r = 0.24$ (6 % of variance). Combined with the oracle's answer-to-delta link ($r = -0.41$, 17 % of variance), the indirect retrieval-to-delta path is $r \approx 0.24 \times (-0.41) \approx -0.10$, while the *direct* retrieval-to-delta correlation measured on the data is even weaker ($r \approx -0.03$). In variance terms, a retrieval-only gate captures roughly 1 % ($r^2 = 0.01$) of the signal the oracle captures ($r^2 = 0.17$). This is a **structural gap** — not a tuning failure, but a fundamental information bottleneck: the quantity that predicts feedback benefit (baseline answer quality) is itself only weakly predictable from pre-generation retrieval signals.

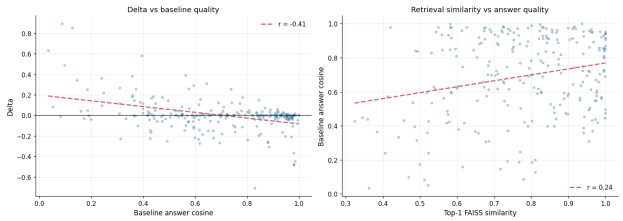


Figure 3: Left: delta vs. baseline answer quality — a clear negative trend ($r = -0.41$): feedback helps weak baselines and hurts strong ones. Right: retrieval similarity predicts answer quality only weakly ($r = 0.24$, 6 % variance), explaining why cosine-based gating fails.

Caveat. The baseline-cosine vs. delta relationship is partly mechanical (baseline cosine appears on both sides of delta). The same direction holds under ROUGE-L (D1 +0.18, D10 −0.09, $r = -0.31$) and BERTScore (D1 +0.02, D10 −0.02, $r = -0.31$).

5.3 Worked examples

These are the most extreme cases in the dataset (mean $\delta = +0.004$), chosen to illustrate the mechanism rather than to represent the typical ticket.

R-163 — the rescue ($\delta = +0.85$). This is the `.ica`-file ticket from the introduction. A user cannot open an `.ica` file in Citrix. The baseline retrieves troubleshooting neighbours and produces a generic reply (cosine 0.13). The correct form-redirect candidate was already at FAISS rank 4; feedback re-ranked within the same five, and the generator reproduced the form almost verbatim (cosine 0.98).

R-143 — the disruption ($\delta = -0.71$). A Salesforce travel-requisition request. The baseline produces a sound reply about network authorisation (cosine 0.83). Feedback promotes upvoted Salesforce candidates from deep ranks about different procedures — user permissions and onboarding. The generator switches to the wrong template (cosine 0.12). `Salesforce` is the most consistently feedback-hostile team (mean $\delta = -0.116$, §5): heterogeneous sub-procedures pooled under one label.

5.4 Pool delta distributions

Team-level mean deltas (Figure 4) span a wide range: `Group` benefits (+0.048) while `Salesforce` is harmed (−0.116). The team delta rank-order is cross-scope stable ($r > 0.55$ across M1–M4), meaning the pool-level pattern is not an artefact of the global scope. Type-level deltas show the same structure: `admin_rights` (+0.024) and `software_request` (+0.028) consist of reusable procedure templates where popularity tracks correctness; `email_support` (−0.074) is feedback-hostile.

These stable deltas suggest a causal hypothesis: same-team feedback excludes procedurally-unrelated votes, making it more relevant than the global pool.

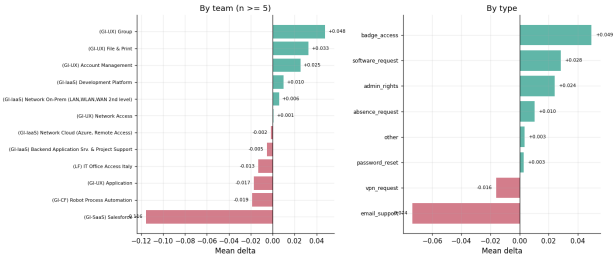


Figure 4: Mean feedback delta by team (left, teams with ≥ 5 tickets) and by ticket type (right). Teal: feedback benefits; red: feedback harms.

6 Modeling

6.1 Cosine-based gating

To answer RQ3 — whether the benefit can be predicted before generation — we test whether deployable retrieval-score proxies can approximate the oracle. An oracle gate — applying feedback only to tickets below a baseline-quality threshold — concentrates the benefit: the best threshold (0.6) routes feedback to 88/250 tickets for a mean delta of **+0.026**, versus +0.004 always-on.

However, the deployable proxies fail. A 5-fold cross-validated sweep over top-1 FAISS, top-5 FAISS, retrieval margin, and top5-minus-top1 thresholds finds no signal distinguishable from always-on feedback: the best signal (top5-minus-top1) reaches +0.006 versus +0.004 always-on, within cross-validation noise (±0.03). As shown in §5, the retrieval-to-delta signal captures only ∼1 % of the oracle's variance. Rescues ($\delta > +0.02$) and disruptions ($\delta < -0.02$) have near-identical top-1 FAISS scores (0.76 vs. 0.75). Simple similarity-threshold routing fails.

6.2 Pool filtering: M2, M3, M4

To answer RQ3 for pool-scoped evidence, we narrow the *feedback pool*. Team deltas being wide and stable suggests that feedback computed from team-only or type-only evidence might outperform global feedback. We test three additional scopes: M2 (team-only), M3 (type-only), and M4 (team∩type). M4 yields the largest

mean delta at +0.010 (vs. M1's +0.004), but starves for 33 % of tickets where no nonzero-lift candidates exist (§3). M2 and M3 are intermediate: +0.003 and −0.002 respectively.

6.3 Pool analysis: why some pools benefit

Returning to the pool delta distributions from §5, we test three post-hoc hypotheses. Structural score does not predict delta ($r = 0.06$). **Procedural mismatch** — a problem-worded query whose answer is a form redirect — is the strongest ticket-level signal: mismatch tickets gain +0.082 vs. −0.011 ($r = 0.21$). At the pool level this correlation reverses ($r \approx -0.59$, $n = 12$ teams): no single pattern explains both levels, motivating a learned gate. **Query diversity** — mean pairwise Jaccard similarity of ticket descriptions within a pool — predicts pool-level delta ($r = -0.38$, $n = 12$ teams): diverse pools (Group: VPN, passwords) are where FAISS retrieves the wrong procedure and feedback rescues; homogeneous pools (permission-change templates) are where FAISS is already right and feedback disrupts.

A naive in-sample team-identity gate reaches +0.017 (154/250 tickets). **Critical disclaimer:** these in-sample oracle numbers are circular; they motivate learning generalisable pool-level features, not a deployable rule.

7 Results

Table 1 consolidates all methods with bootstrap 95 % confidence intervals. All always-on methods (M1–M4 and the cosine gate) have CIs that cross zero; only the oracle gate (baseline answer quality < 0.6, 88/250 tickets) sits clearly above zero. The pool-based oracle gate (team-identity, in-sample) reaches +0.017 but is expressly circular.

Table 1: Method comparison (gpt-5.6-luna, $n = 250$, 10,000 bootstrap resamples). All CIs cross zero except the oracle gate. Pool-based oracle gate (§6.3) is in-sample only.

Method	Mean δ	95 % CI
Baseline (FAISS only)	0.000	—
M1 global	+0.004	[−0.016, +0.025]
Cosine gate*	+0.006	[−0.012, +0.026]
M2 team-only	+0.003	[−0.019, +0.026]
M3 type-only	−0.002	[−0.021, +0.018]
M4 team∩type	+0.010	[−0.006, +0.029]
Oracle gate†	+0.026	[+0.011, +0.043]
Pool-based oracle‡	+0.017	in-sample only

*Best 5-fold CV threshold (top5 minus top1 FAISS); within CV noise of always-on. †Baseline answer quality < 0.6 (88/250 tickets). ‡Team-identity: route feedback only to teams with positive mean delta, chosen on the same 250 tickets — **circular**, for directional intuition only.

8 Discussion

Limitations. Single enterprise, single evaluation run, 250 query tickets. LLM-judge feedback is a stronger signal than deployable human ratings: the judge sees the ground-truth reply and the evaluation metric measures similarity to it. Cosine similarity to a single reference reply is a narrow measure of usefulness; a ticket may admit more than one valid reply. Pool hypotheses are post-hoc and require held-out validation. Parameters were fixed a priori but sensitivity to m and C is left to future work. Team and type gating priors may not transfer to other organisations. Multiple-comparison corrections would not change the findings (the only CI above zero is the oracle gate).

Future work. Given the structural gap between retrieval signals and feedback benefit (§5), a learned gating model is the natural next step. Such a model could combine pool-level query diversity, procedural mismatch indicators, retrieval margin, and feedback-evidence strength to predict *which* filtering scope (M1–M4, or an adaptive pool) will perform best for each incoming ticket, using only pre-generation features. This would test whether the ∼0.017 pool-based oracle ceiling generalises beyond the in-sample analysis presented here.

9 Conclusion

Adding LLM-judge feedback to retrieval helps when the baseline retrieval is procedurally wrong and hurts when it is already right — the two signals are orthogonal. Simple retrieval-score proxies do not usefully gate this effect; the retrieval-to-answer-to-delta chain is too indirect. Narrowing the feedback pool by team or type (M2–M4) modestly improves the best case but starves for evidence. However, pool-level properties — query diversity ($r = -0.38$) and procedural mismatch ($r = 0.21$) — are observable before generation and predict benefit. The natural next step is a learned gating model that combines these pool-level features with evidence strength to decide *which* feedback pool to use *for which* ticket.

10 Acknowledgements

This work was supported by the European Union-funded project HumAIne [grant number 101120218]. The authors acknowledge the use of LLMs for language optimisation and feedback scoring. All conceptual frameworks, analyses, and interpretations remain the sole responsibility of the authors.

References

[1] Domen Jeršek, Klemen Kenda, Rok Klančič, and Matteo Frattini. 2025. Automated first-reply generation for it support tickets using retrieval-augmented generation and multi-modal response synthesis. In *Proceedings of the 28th International Multiconference Information Society (IS 2025)*. doi:10.70314/is.2025.sikdd.19.

[2] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7, 3, 535–547.

[3] Patrick Lewis et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474.

[4] Long Ouyang et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 27730–27744.

[5] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP-IJCNLP*, 3982–3992.

[6] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*. Vol. 33, 5776–5788.

[7] Lianmin Zheng et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*. Vol. 36.