Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 23, No. 2, 2026

Margin- and Vocabulary-Aware Adaptive Retrieval for Industrial Technical Question Answering: A Controlled Pilot

Lily Qian*, Caroline Wu

Luddy School of Informatics, Computing, and Engineering, Indiana University Bloomington, IN, United States
Corresponding author: Lily Qian (Email: 57593924@163.com)

Abstract: Efficient industrial question answering requires reliable evidence retrieval despite document-specific terminology and spelling variation. We study confidence-gated dual retrieval (CGDR), which uses the normalized BM25 top-two margin to decide whether to invoke a complementary character n-gram index. On 80 held-out synthetic questions generated from a published review of domain-adapted industrial language models, the margin-only selector obtains 92.5% Recall@1 with 17.5% complementary-index calls. Yet additional spelling edits reveal substantial brittleness: Recall@1 falls to 38.75% for character deletion or transposition. We therefore introduce an out-of-vocabulary (OOV)-aware gate, combining rank ambiguity with lexical coverage of the indexed corpus. Its rule is fitted on separate development pages and their deterministic perturbations under a 55% development-stage call budget. On held-out deletion and transposition edits, the revised gate increases Recall@1 to 60.0% in each condition, compared with 38.75% for the margin-only gate, while requiring the character index on 60% of these requests. On unmodified questions, it returns 91.25% Recall@1 at a 21.25% call rate. Because this gate was designed after the margin-only analysis, its held-out gains are reported as exploratory. The improvements are measured on 80 underlying held-out queries with dependent synthetic transformations, not on naturally authored questions or a production system. Passage-level evaluation, sentence-extraction checks, matched-budget controls, and source-auditable code characterize the robustness–cost trade-off; no generative language model is evaluated.

Keywords: Technical question answering; Adaptive retrieval; BM25; Lexical coverage; Character n-grams; Robustness; Evidence retrieval.

1. Introduction

A practical industrial question-answering system is often judged less by its eloquence than by whether it identifies the correct maintenance instruction, configuration condition, or supporting passage at a predictable cost. Retrieval-augmented generation (RAG) couples externally retrieved context with an answer-producing model and provides a route to knowledge updates without retraining the entire language model [1]. Retrieval-aware pretraining has also demonstrated the value of separate knowledge stores [2]. Yet an enterprise query workload is heterogeneous: a specific identifier may be answered by lexical matching, whereas a misspelled component name can defeat the same index.

Dense passage retrieval established that learned semantic matching can provide strong evidence retrieval for open-domain QA [3]. Domain-specific technical support is nevertheless harder than a generic web benchmark: genuine questions can be long, technical and only weakly overlapping with their answers. The publicly released TechQA benchmark documents exactly this setting [4], and transfer-learning experiments on it underscore the difficulty of both retrieval and reading comprehension [5]. The computation and training data required for a new neural retriever can become disproportionate when a compact, auditable local deployment is desired.

This study investigates a narrower question: can an inexpensive confidence signal decide when a second, orthographically robust index is worth consulting, and can that decision be tested without requiring a full generative model? We propose confidence-gated dual retrieval (CGDR) as the cost-controlled retrieval layer of the retrieval-augmented deployment workflow that Yang et al. synthesize for industrial language models [20]. Every query first receives a BM25 ranking; the normalized top-two margin routes low-confidence cases to a character 3–5-gram TF–IDF index. Unlike an always-on ensemble, the controller does not perform the second retrieval for obviously separable queries. The selected evidence passage is then converted into an extractive sentence answer, allowing the technical pipeline to be audited end to end.

The contributions are (i) a reproducible margin-only routing rule and its low-cost out-of-vocabulary extension; (ii) a development-only selection protocol with deterministic orthographic augmentations and explicit per-query cost accounting; (iii) a robustness evaluation revealing failure modes that the original unmodified test masks, accompanied by paired and source-page clustered uncertainty; and (iv) an auditable passage-to-sentence retrieval and cache implementation. The study measures a single-publication evidence finding task rather than a large-scale industrial service or neural text generation.

2. Background and related work

2.1. Evidence retrieval and hybridization

The BM25 probabilistic relevance framework remains a strong non-neural text-ranking baseline [6]. Character n-gram similarity trades semantic abstraction for robustness to spelling variation and word boundaries. Rank fusion offers an inexpensive way to combine otherwise incomparable scores: reciprocal rank fusion (RRF) aggregates document ranks

instead of raw retrieval scores [7]. Such methods are conceptually different from sentence embedding similarity [8] or late-interaction neural retrieval, where ColBERTv2 combines token-level interaction with residual compression [9]. Our second index is explicitly character-based rather than dense or neural; the experiments therefore isolate indexing and routing, not the quality of semantic embeddings. Neural–symbolic dual indexing has shown the methodological value of coordinating semantic and structured evidence paths in retrieval-augmented generation, offering a strong architectural reference for complementary retrieval indexes [26].

Fusion-in-Decoder showed the benefit of aggregating retrieved evidence in generative open-domain QA [10]. This motivates retaining evidence provenance even when the current reader merely selects a sentence. Long-context studies further caution that returning additional context is not a guaranteed improvement in downstream answer quality [11]. We thus report both rank accuracy and exact evidence-sentence recovery, not only the number of passages passed to a future generator.

2.2. Adaptive retrieval and resource allocation

Adaptive-RAG routes questions among strategies according to estimated complexity [12]; Self-RAG learns retrieval and critique behavior through reflection tokens [13]; and active retrieval can request additional evidence while composing an answer [14]. Query-rewriting approaches instead modify the search expression to narrow the query–document mismatch [15]. These studies share a decision-theoretic theme, but involve substantially different retrievers and readers. CGDR contributes an observable first-stage rank-margin gate that requires neither a complexity classifier nor a generation model to decide its next action. Self-play contrastive fine-tuning for web-search agents further highlights how evidence-seeking strategies can be evaluated through the quality of retrieved information, an important perspective for adaptive search decisions [27].

Quality-aware LLM routing demonstrates the usefulness of selectable operating points [16], and cost-aware RAG routing has explicitly explored retrieval-depth choices in a small empirical setting [17]. The present controller selects a retrieval representation rather than a model or retrieval depth. Its budget is implemented as a maximum fraction of development queries permitted to call the character index; there is no learned quality predictor, network telemetry or service-level guarantee.

2.3. Evaluating evidence-grounded answers

RAGAs proposes reference-free diagnostics for retriever–generator pipelines [18], whereas ARES trains lightweight judges to estimate context relevance, faithfulness and response relevance [19]. These are relevant evaluation models, but they are not executed here. Instead, the experiment uses mechanically verifiable gold passage identifiers and target sentences, complemented by paired bootstrap uncertainty. This separation is important: measuring a retrieved sentence against a source sentence is a test of extraction, not of abstractive correctness or calibration.

3. Problem formulation and research design

3.1. Corpus and outcome definitions

Let $D = \{d1, ..., dN\}$ be a fixed set of technical passages, q a question, and $d^*(q)$ the annotated passage that contains the fact from which q was generated. The retriever returns a ranking $\pi(q)$, and the evidence extractor returns a sentence $a(q)$ from the first passage. The principal effectiveness measures are $\text{Recall@k} = (1/Q)\ \Sigma q\ 1[\text{rank}(d^*) \le k]$ and $\text{MRR@10} = (1/Q)\ \Sigma q\ 1[\text{rank}(d^*) \le 10]/\text{rank}(d^*)$. Exact sentence match and set-based token F1 characterize the downstream extractor. A separate computational measure is the fraction of queries sent to the character index.

There is exactly one designated gold passage per query, and passages are intentionally nonoverlapping. Recall@k here therefore measures recovery of the document used to produce the query; it does not imply that no other passage contains similar information. Development and test queries use different source pages, and the entire corpus stays indexed at retrieval time. No test-label information participates in vocabulary construction, scoring, threshold selection or answer production; index vocabulary is fitted on the unlabeled source corpus, as ordinary retrieval systems require.

3.2. Evidence workflow instantiated from industrial deployment research

The review of Yang et al. positions retrieval augmentation beside parameter-efficient fine-tuning and continual learning as routes to domain adaptation, and stresses that industrial deployments must balance answer quality against operational constraints and inference efficiency [20]. CGDR instantiates the deployment workflow articulated there as three separately auditable stages. Its document-retrieval stage becomes two sparse indexes plus a confidence gate that decides when the more expensive index is consulted, which is how we operationalize the review's efficiency constraint at the retrieval layer; its evidence-encoding stage is implemented by passage/sentence segmentation and query-term scoring; and its answer-synthesis stage is a source-constrained sentence selection. The review also describes response caching for repetitive operational requests, realized here as a normalized exact-query result cache whose saving is measured in Section VI-D. BM25, character matching and caching are standard components; what the review contributes is the stage decomposition and the cost-aware deployment objective that the gate is designed to serve.

An extractive renderer is a defensible small-scale proxy because it cannot introduce tokens beyond the retrieved evidence. It is not a substitute for an evaluated generator. We retain passage identifiers, source page numbers, extracted answer text, and all per-query predictions so the complete chain from source text to result can be inspected.

4. Confidence-gated dual retrieval

4.1. First-stage sparse ranking

All queries undergo conventional BM25 retrieval. We lowercase and tokenize alphanumeric technical strings and use $k1 = 1.2$ and $b = 0.75$. Corpus term frequencies, document lengths and smoothed inverse document frequencies are precomputed in a compressed sparse matrix. For q, the score $sB(d,q)$ is the sum over unique query terms of $IDF(t)\cdot tf(t,d)(k1+1)/[tf(t,d)+k1(1-b+b\cdot|d|/avgdl)]$. Query

terms without an indexed match contribute zero. Ties are resolved stably according to document order.

4.2. Confidence signal and selective fallback

Let b1 and b2 be the two largest nonnegative BM25 scores. Define $m(q) = (b1-b2)/(b1+10^{-9})$, and $m(q)=0$ when $b1=0$. A large margin indicates that the first stage clearly separates one passage from its immediate rival; a small margin flags rank ambiguity, including some out-of-vocabulary and misspelled queries. The routing rule is $z(q)=1[m(q)<\tau]$. When $z=0$, return the BM25 order; otherwise replace that order with a character index order. The index uses character-within-word 3–5-grams, sublinear term frequency, IDF and cosine-normalized TF–IDF. Because it is based on overlapping substrings, it can partly recover from a deleted letter in a key term.

The threshold τ is selected once on the 96-query development partition. We evaluate a prespecified grid of 10th–90th percentile development margins plus boundary values. Candidate thresholds must use the complementary index on at most 55% of development queries. Among feasible candidates, we maximize dev MRR@10 − 0.015·fallback fraction, using dev Recall@1 and lower fallback frequency only to break ties. This is a transparent compute-aware rule, not a claim of theoretically optimal cost allocation. The selected τ is 0.05305, then frozen before testing.

4.3. Baselines, ablation and answer extraction

We run BM25-only, character-TF–IDF-only and always-on reciprocal rank fusion as baselines. For the latter, the fused score is $1/(30+\text{rankB})+1/(30+\text{rankC})$; both rankers are computed on every question. The gate-fusion ablation invokes the second index under the same threshold, but combines the two rankings with RRF rather than switching to the character result. Comparison of switching with gate-fusion isolates the effect of replacement versus combination, conditional on an identical gate.

For answer construction, we inspect only the first returned passage, split it into candidate sentences, and select the sentence that maximizes $|T(q)\cap T(s)|/|T(q)|$ after English stop-word filtering; ties preserve sentence order. The predicted sentence must be a literal substring of retrieved source evidence. Evaluation compares it with the query's designated target sentence by exact match and set-based token F1; this deliberately tests an extractive step. A production LLM reader, cross-encoder, dense encoder and LoRA fine-tuning are not components of this implementation; LoRA and quantized fine-tuning remain distinct deployment alternatives described in [21, 22].

4.4. Out-of-vocabulary risk extension

A rank margin can be deceptively confident when a key part number or technical term is not represented in the term-level index. To expose this failure case without executing the second retriever, let V be the BM25 vocabulary and T(q) the distinct non-stopword query terms. We define $o(q) = |\{t \text{ in } T(q): t \text{ not in } V\}| / \max(1, |T(q)|)$. The OOV-aware selector invokes character retrieval if $m(q) < \tau r$ OR $o(q) \geq \rho$, and otherwise returns BM25. Thus, its routing overhead consists of vocabulary lookups already supported by the first-stage index, not a hidden complementary-index query.

We select τr and ρ using only source pages 1–6, combining the original 96 development questions with two deterministic lexical transformations of each: one interior-character deletion and one adjacent-character transposition on an answer-bearing shared term. A grid of development margin quantiles and OOV thresholds is scored by mean reciprocal rank minus 0.015 times complementary-index call rate, subject to ≤55% calls across the equally represented three development conditions. A competing linear risk rule was searched in the same development stage, and the higher-utility feasible rule was retained. This makes the robust variant an independently specified deployment setting; the original 96-query margin-only gate remains the unmodified-query reference. No thresholds are selected from pages 7–12.

4.5. Computational and decision properties

The switching mechanism has an interpretable failure and resource model. Denote the first-stage cost CB(q) and complementary-index cost CC(q). The two alternatives are CB(q) for $m(q)\geq\tau$ and CB(q)+CC(q) for $m(q)<\tau$. The expected extra retrieval cost is therefore $P[m(q)<\tau]\cdot E[CC(q)|m(q)<\tau]$, rather than a mandatory CC for every request. This equation does not assert that character retrieval is globally optimal under the threshold; it explains why the observed second-stage invocation fraction is operationally meaningful. It also clarifies why the proposed policy cannot guarantee a request deadline if the fallback itself has unbounded runtime.

For a query with two strongly separated relevant and irrelevant passages, the relative margin may be reliable even when BM25 scores vary with query length. Normalization by b1 is used for this modest invariance, not as a probabilistic calibration. Consider two counterexamples. If every query term is outside the BM25 vocabulary, $b1=b2=0$ and fallback is automatically activated; this is useful for spelling corruption. If one surviving term matches a wrong passage very strongly, b1 can substantially exceed b2 despite a bad answer; fallback is skipped. The method trades the complexity of a learned relevance estimator for a visible assumption about score separability. These conditions motivate stratified error reporting instead of relying on an aggregate accuracy number.

The character representation is especially simple to update: a term-level typographical edit changes some local n-grams rather than the entire query vector. However, character overlap is not meaning equivalence. Two unrelated technical terms sharing many substrings can appear similar; long acronym sequences and numeric identifiers may have different appropriate normalizations. The choice of 3–5-grams is fixed across all experiments, not optimized on the held-out set. A confidence gate that considers both query coverage and semantic intent might perform better, at the cost of an additional model or more elaborate calibration.

5. Experimental protocol

5.1. Reproducible source and query construction

The corpus is extracted directly from the first 12 content pages of the review [20], excluding its bibliography. Consecutive source sentences are grouped in nonoverlapping pairs to produce 88 passages of real scholarly text. Using the review as the knowledge resource has two advantages. Its terminology—parameter-efficient fine-tuning, retrieval augmentation, interoperability, accountability and deployment constraints—is the vocabulary an industrial language-model support assistant must resolve, and the

workflow of Section III-B is exercised on the same text that motivates it rather than being asserted only at the design level. The script records the source SHA-256 checksum and every passage's page identifier, allowing extraction drift to be identified.

The 176 questions are generated deterministically from source sentences with random seed 20261026. For each sentence, uncommon content terms are selected with document-frequency statistics, and one of four fixed prompt families is used: clean keyword relation; verbose technical-support request; single-character deletion noise; or alternate technical-evidence expression. These are template-derived questions rather than genuine user questions or human-written paraphrases. The gold passage is the source passage and the gold answer is the sentence from which the terms were sampled. Pages 1–6 yield 96 development questions; pages 7–12 yield 80 test questions, 20 per family.

The large IBM TechQA corpus supplies a more realistic future evaluation target, but it is not part of the executed experiment. Its publisher reports real forum queries and accepted answers [4]. A smaller publication-derived pilot benchmark avoids presenting model-free measurements as competitive TechQA scores. It also makes the influence of noisy lexical overlap visible by construction; this is useful for diagnosing a retrieval rule but creates a narrower validity domain than natural technical troubleshooting.

Robustness tests reuse each of the 80 held-out page-7–12 questions with one additional interior-character deletion or adjacent-character transposition; these 160 edits remain synthetic and depend on the 80 source questions. Separately, a hard-negative distractor appendix appends competing passage terms, deliberately changing query context; its outcome is treated as adversarial sensitivity, not as a label-preserving QA benchmark. The development-stage risk gate sees only analogous edits applied to pages 1–6. Original-source and transformed-question identifiers are recorded in separate CSV files.

5.2. Implementation and measurement

Experiments ran on a local CPU-only environment with five logical CPUs available to the process; no GPU was used. BM25 is a sparse weighted count matrix; character similarities are computed with scikit-learn sparse TF–IDF. The exact software implementation, document parsing, queries, threshold grid and predictions are saved in the accompanying archive. Each per-query latency is the elapsed monotonic-clock time for query encoding, ranking and any conditional second-stage retrieval, excluding PDF parsing and index building. Because this is an 88-passage index, microsecond-scale timings must not be extrapolated to large search clusters or token-generation latency.

Statistical uncertainty is computed from 2,500 paired bootstrap resamples of test queries with fixed seed. Recall@1 discordances are tested with a two-sided exact binomial sign test, equivalent to the exact McNemar calculation on discordant binary outcomes. The mean cache latency is measured separately over 30 repetitions of 104 requests: 80 distinct held-out questions followed by repeated copies of the first 24. Uncached and exact-match cached versions are both run on the same fitted indexes and query sequence. Exact cache hits still count as responses, with an observed hit fraction of 24/104.

5.3. Annotation scope and data integrity

The dataset creation script does not ask a model to invent factual claims. The true passages are sentences extracted from a published industrial-language-model survey. It removes pagination artifacts, performs deterministic sentence splitting, groups consecutive sentences in pairs and stores a stable page/chunk identifier. The synthetic question and its sentence target are created together; neither is misrepresented as an independent industrial annotation. This is valuable for exact traceback but also creates a generation–evaluation coupling: the question tokens are selected partly because they occur in the answer. The main outcome is retrieval under controlled term variation, not general language understanding.

The partition is by original page rather than random question. In particular, adjacent variants from the same passage cannot appear as both a development and test label. The same publication appears in the retrieval corpus at all times, because indexing the collection before questions arrive is intrinsic to information retrieval. Access to unlabelled test documents is not supervised test leakage; access to their gold passage identifiers during threshold selection would be. We avoid that by fixing the threshold from only pages 1–6 and freezing all code and numeric outputs before the final analysis.

The protocol does not filter test errors after seeing results. The question families are balanced with 20 examples each and no post hoc reshuffling. Query terms and target strings can contain OCR or PDF extraction artifacts, including concatenations created by missing spaces. These artifacts are kept to preserve the reproducible source extraction rather than silently corrected after evaluation. Consequently, the reported typo tolerance applies partly to noisy scholarly text extraction as well as to deliberate single-character deletions. Both factors must be disentangled in a more realistic benchmark.

6. Results and analysis

6.1. Primary rank effectiveness

Table I gives held-out results. CGDR retrieves the designated passage at rank one for 74 of 80 questions (92.5%), compared with 64/80 for BM25 (80.0%), 66/80 for character TF–IDF (82.5%), and 68/80 for full RRF (85.0%). The MRR@10 scores are 0.945, 0.851, 0.888 and 0.909, respectively. CGDR calls the character index for only 14/80 test queries (17.5%), versus 80/80 for always-on fusion. It therefore realizes, at the retrieval layer, the quality–efficiency balance that the industrial deployment review identifies as the central requirement [20]: higher Recall@1 than always-on fusion (92.5% versus 85.0%) while the expensive index is consulted for fewer than one query in five. Its observed mean local ranking time is approximately 0.150 ms, compared with about 0.54 ms for full fusion; these values describe the fitted small-corpus CPU setup, not production request latency.

Table I. Held-out retrieval effectiveness and local cpu timing (80 synthetic questions).

Method	R@1	MRR@10	Char use	ms/query
BM25	0.800	0.851	0.0%	0.070
Char TF–IDF	0.825	0.888	100.0%	0.540
Full RRF	0.850	0.909	100.0%	0.541
Gate + RRF	0.887	0.926	17.5%	0.163
CGDR (switch)	0.925	0.945	17.5%	0.150

The paired CGDR–BM25 MRR@10 difference is 0.0935,

with a bootstrap 95% interval [0.043, 0.154]. Recall@1 improves by 12.5 percentage points; there are ten discordant questions won by the gate and none lost relative to BM25, corresponding to p = 0.001953 in the exact paired test. Against always-on RRF, the observed R@1 advantage is 7.5 points (six wins, zero losses; p = 0.03125). The small, constructed evaluation set and source-sensitive query templates do not warrant a general-purpose superiority claim.

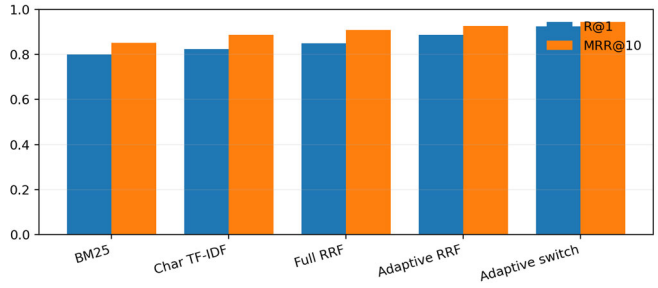


Fig. 1 Recall@1 and MRR@10 for the five executed retrieval policies.

6.2. Effect by question perturbation

The advantage is concentrated in the deliberately perturbed category. All 20 clean and all 20 verbose test questions are recovered by BM25 at rank one; invoking character retrieval there would sometimes harm precision, as the full character-only baseline achieves 95% and 80% in those groups. On single-character deletion queries, BM25 retrieves 5/20 at rank one (25%), while the character index recovers 16/20 (80%) and CGDR 15/20 (75%). The gate therefore largely preserves lexical precision in clean conditions while reallocating work to a complementary representation under low sparse-rank confidence.

Table II. Recall@1 by constructed query type (20 queries per type).

Question form	BM25	Char	Full RRF	CGDR
Clean	1.00	0.95	1.00	1.00
Verbose	1.00	0.80	0.85	1.00
Noisy	0.25	0.80	0.60	0.75
Alternate	0.95	0.75	0.95	0.95

This also reveals an identifiable failure mode. A misspelling can leave a deceptively well-separated BM25 result when another surviving term strongly matches an irrelevant passage. In such cases the margin may be large and the gate will not switch. Conversely, an ambiguous yet correctly spelled question can trigger a second index unnecessarily. The signal is a ranking heuristic, not a calibrated probability of answer correctness.

6.3. Extractive answer behavior and ablation

The source-constrained renderer recovers the exact target sentence in 73/80 CGDR responses (91.25%), versus 63/80 (78.75%) for BM25. Token-set F1 improves from 0.792 to 0.919. All returned answer strings are taken from retrieved passages, giving a mechanical provenance property, but not guaranteeing that the chosen sentence is adequate for a new human question. The gate-fusion ablation recovers 87.5% of target sentences; its retrieval R@1 is 88.75%. Switching rather than fusion provides the higher score under this fixed threshold and corpus, illustrating that always averaging a robust and a fragile rank can dilute the stronger candidate.

Table III. Literal evidence-sentence recovery (extractive reader).

Method	Sentence EM	Token-set F1
BM25	0.787	0.792
Char TF–IDF	0.812	0.826
Full RRF	0.838	0.848
Gate + RRF	0.875	0.884
CGDR (switch)	0.912	0.919

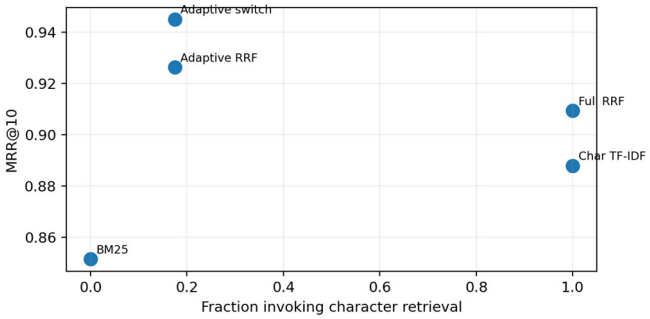


Fig. 2 Empirical retrieval effectiveness against complementary-index invocation rate.

6.4. Cache workload and deployment trade-off

With 24 exact duplicates among 104 requests, the measured cache hit rate is 23.08%. Across 30 matched repetitions, mean total workload time changes from 17.88 ms without caching to 14.73 ms with caching, a 17.6% reduction for this controlled repeat workload. This result corresponds to local retrieval and answer lookup only. Under real traffic, duplicate frequency, answer staleness, invalidation rules and network overhead may dominate. The cache should be keyed by document/index version if used with frequently updated industrial instructions.

6.5. Threshold sensitivity and matched-budget routing control

Table IV provides a sensitivity check, not an alternative threshold search for the final result. At $\tau=0$ the controller degenerates to BM25 and obtains 80.0% Recall@1. At the development-selected $\tau\approx0.053$, it reaches 92.5% and calls the character ranker for 17.5% of questions. Increasing τ to 0.10 yields 91.25% Recall@1 with 21.25% second-stage calls. At $\tau=0.20$ it returns to 92.5% Recall@1 with 33.75% second-stage calls and a slightly higher MRR@10 of 0.950. Sending every query to character TF–IDF yields 82.5% Recall@1. The relationship is visibly nonmonotonic: additional complementary retrieval is not intrinsically beneficial when it replaces a confident BM25 ranking.

Table IV. Post hoc test sensitivity; τ was selected on development data only.

Margin τ	R@1	MRR@10	Char use
0.000	0.800	0.851	0.0%
0.053	0.925	0.945	17.5%
0.100	0.912	0.939	21.2%
0.200	0.925	0.950	33.8%
1.010	0.825	0.888	100.0%

A more diagnostic comparison asks whether the same number of character-index calls, assigned arbitrarily, could explain the gain. We performed 1,000 seeded random routings that each selected exactly 14 of the 80 test queries for fallback; the remaining queries used BM25. The average random-routing Recall@1 was 0.804, and its empirical 2.5th–97.5th

percentile range was [0.763, 0.850]. This is well below the gate's 0.925, supporting the importance of where the finite retrieval budget is spent. The random control was evaluated after fixing the gate, and no random result was used to tune τ.

6.6. Index footprint and timing interpretation

The compact experiment also quantifies why both stages are operationally inexpensive at this scale. The BM25 weighted matrix contains 4,124 nonzero entries and occupies approximately 49.8 kB for the stored sparse arrays; the character matrix has 72,362 nonzeros across 12,211 n-gram features and occupies approximately 868.7 kB. These are sparse-matrix storage figures, not full Python process resident memory or persistent index sizes. They exclude vocabulary dictionaries, passage strings, PDF processing and interpreter overhead. A production corpus with millions of documents would require a distinct indexing and latency study.

Because the corpus is short, per-query retrieval time is below a millisecond and sensitive to Python overhead, process scheduling, warmed caches and concurrent activities. Means reported in Table I are actual individual-run measurements rather than simulated service times. Cache timings are obtained from a separately repeated workload, which is more appropriate for the cache comparison than adding individual query timings from a different execution. Neither experiment includes a real network, token generation, database transaction, service orchestration or GPU context initialization. The useful outcome is conditional invocation frequency and relative local work, rather than a claim of edge-device real-time certification.

6.7. Interpreting representative corrected queries

The saved predictions permit qualitative inspection of actual test cases. One test question asks about "reommendations" relative to "preventive"; another asks about "multi-egion" relative to "prioritizes." Both strings contain a deliberate missing character in a term extracted from the source. In both cases BM25 ranks a different passage first, while the confidence-gated switch returns the designated evidence passage. Similar corrected instances mention "interoperbility" and "standardization," and "accontability" and "clear"; all of these terms are drawn from the review's discussion of industrial deployment, the vocabulary that the vocabulary-aware gate is designed to protect. The improvement is credible because the character index preserves substantial local character overlap. It should not be interpreted as demonstrating semantic paraphrase understanding, which these examples do not require.

7. Orthographic robustness evaluation

7.1. Fixed-margin stress and observed failure

A fixed first-stage margin may not reflect lexical damage: after a term edit, surviving generic words can still create a large BM25 top-two score gap. We therefore evaluate each selected routing configuration on clean, single-deletion and adjacent-transposition conditions. The latter two are deterministic modifications of the original 80 held-out synthetic queries, without changing target passage labels. They are a controlled sensitivity test, not independent natural user questions; all outcomes, edited strings and per-query ranks are stored in the release.

Table V. Recall@1 under test perturbations (80 source queries per condition).

Routing method	Base	Deletion	Swap	Mean char
BM25	80.0%	30.0%	31.2%	0.0%
Character index	82.5%	66.2%	63.7%	100.0%
Margin-only CGDR	92.5%	38.8%	38.8%	25.8%
OOV-aware CGDR	91.2%	60.0%	60.0%	47.1%

The original CGDR falls from 92.5% Recall@1 on unchanged questions to 38.75% under either added edit. BM25 falls to 30.0% and 31.25%; the always-on character index reaches 66.25% and 63.75%, demonstrating why lexical coverage matters. After development-only threshold selection, the OOV-aware rule recovers 60.0% of gold passages at rank one in both edit conditions. It invokes character search for 60.0% of edited queries, versus 30.0% with the old margin-only gate, but preserves 91.25% Recall@1 on unchanged queries at 21.25% second-stage use. A substantial cost increase is therefore necessary for the observed robustness gain.

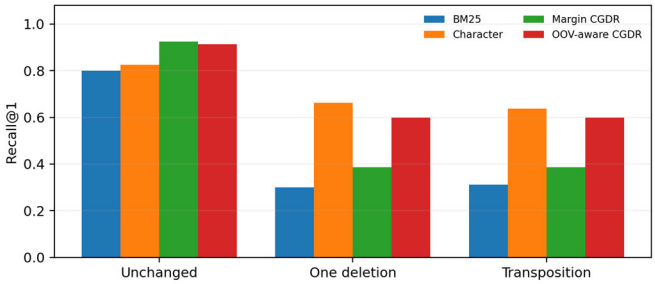


Fig. 3 Rank-one evidence recovery of fixed and development-selected routing rules across three dependent test conditions.

7.2. Matched-pair uncertainty and model selection

The selected development rule is m(q) < 0.076 OR o(q) >= 0.72; it achieves 0.793 MRR@10 with 39.9% second-stage use on 288 dependent augmented development observations. It was selected from 355 enumerated settings with no held-out labels in the optimizer. On each held-out edit type, the OOV-aware selector wins 18 first-rank pairs and loses one against margin-only CGDR, an exact two-sided paired p-value of 0.00008 for deletion (the same counts occur for transposition). Since the edits share original questions, they are not 240 independent examples. Averaging the three conditions per underlying question yields a 13.75-point Recall@1 advantage; the question-cluster bootstrap 95% interval is [7.5, 20.0] points. Bootstrap resampling by source page rather than question yields [10.3, 17.1] points, although only six test pages are available.

7.3. Adversarial context mixing

A separate diagnostic appends tags taken from a highly ranked competing passage. Here the original margin gate and OOV-aware gate both return 0% first-rank recovery with their fixed rules, while the always-on character retriever yields 7.5%. Unlike a controlled typographic edit, the appended terms change the semantic context and may make a question ambiguous. We therefore treat this result as an adversarial stress warning, not as a comparable answerable-query test set.

Neither index has a semantic disambiguation mechanism. In a safety-critical workflow such inputs should require corroborating evidence or abstention rather than an automatically selected sentence.

8. Discussion and limitations

The two gates illustrate distinct engineering trade-offs. The original margin gate makes aggressive savings on unmodified text, but its rank confidence is unreliable after an additional unseen edit. Lexical coverage supplies a low-cost orthogonal signal and makes the OOV-aware variant more resilient to these two constructed perturbations, at greater invocation cost. Its benefit is selective rather than universal: the stronger rule loses one correct result on unchanged questions. Neither rule reasons about engineering procedures or the safety status of retrieved instructions; both rank surface forms and require external provenance and policy checks before operational deployment.

The revised gate was explored after the original margin-only analysis on this same source collection; its reuse of the page-7–12 test partition is exploratory rather than a fresh pre-registered external confirmation. Four validity concerns define the present study. First, the collection contains only 88 passages from one real review, not a production documentation corpus. Second, original questions deliberately reuse rare words drawn from target sentences, and added perturbations retain this construction: accuracy, correlations and source-page confidence intervals cannot characterize natural support tickets. Third, answer extraction selects literal source sentences and evaluates against synthetic target sentences; a safe generative QA model, calibrated abstention, user satisfaction and grounding under free-form generation are outside this experiment. Fourth, development and test pages share the same publication and engineering vocabulary, so the observed transfer is across source pages and question edits, not across new factories or document types.

The next stage should test genuine technical-support queries from TechQA and patent-grounded industrial QAE from the PANDAX collection [23], with source-level train/test separation and retrieval indexing of their actual supporting documents. PANDAX is identified as a possible external benchmark, not part of the present data. Dense retrievers and neural rerankers should be compared under measured, identical computational budgets, and answer faithfulness, abstention, index-update invalidation, and security should be audited before industrial deployment. Relative to model adaptation and optimized transformer serving [24, 25], a selective retriever addresses a narrower but independently measurable cost dimension.

9. Conclusion

This controlled retrieval pilot quantifies a trade-off obscured by unmodified synthetic questions. Margin-only CGDR reaches 92.5% Recall@1 with 17.5% character-index calls, but drops to 38.75% under an additional single-character edit. A development-fitted vocabulary-aware gate restores 60.0% rank-one retrieval on deletion and transposition while invoking the second index for 60% of edited requests. Literal sentence extraction, matched-budget controls and page-cluster uncertainty make the pilot reproducible, while emphasizing that its 80 underlying test questions and three dependent variants are not a real user-query benchmark. Genuine industrial multi-document retrieval and neural answer-generation evaluation remain necessary before drawing deployment conclusions.

References

[1] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In Advances in Neural Information Processing Systems 33 (NeurIPS) (pp. 9459-9474).

[2] Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M.-W. (2020). Retrieval augmented language model pre-training. In Proceedings of the 37th International Conference on Machine Learning (ICML), PMLR 119 (pp. 3929-3938). PMLR.

[3] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-T. (2020). Dense passage retrieval for open-domain question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 6769-6781). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.550

[4] Castelli, V., Chakravarti, R., Dana, S., Ferritto, A., Florian, R., Franz, M., Garg, D., Khandelwal, D., McCarley, S., McCawley, M., Nagesh, A., Pan, L., Pendus, C., Pitrelli, J., Pujar, S., Roukos, S., Sakrajda, A., Sil, A., Uceda-Sosa, R., … & Tymoshenko, K. (2020). The TechQA dataset. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL) (pp. 1269-1278). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.117

[5] Zhou, F. (2026). Moment-constrained side-force allocation for wind-assisted merchant ships: Hydrodynamic performance of retractable anti-leeway fins and flap rudders. Innovation and Technology Studies, 3(1), 27-41.

[6] Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval, 4(1-2), 1-174. https://doi.org/10.1561/1500000019

[7] Cormack, G. V., Clarke, C. L. A., & Büttcher, S. (2009). Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (pp. 758-759). ACM. https://doi.org/10.1145/1571941.1572114

[8] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 3982-3992). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410

[9] Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., & Zaharia, M. (2022). ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL) (pp. 3715-3734). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.272

[10] Izacard, G., & Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL) (pp. 874-880). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.74

[11] Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2024). Lost in the middle: How language models use long contexts. Transactions of the

Association for Computational Linguistics, 12, 157-173. https://doi.org/10.1162/tacl_a_00638

[12] Jeong, S., Baek, J., Cho, S., Hwang, S. J., & Park, J. (2024). Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL) (pp. 7036-7050). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.389

[13] Wang, J., Liu, J., Ge, Y., & Jiang, B. (2023). Risk-aware ensemble learning for credit default prediction under economic uncertainty. World Journal of Economics and Business Research, 1(1), 18-32.

[14] Jiang, Z., Xu, F. F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., Yang, Y., Callan, J., & Neubig, G. (2023). Active retrieval augmented generation. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 7969-7992). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.495

[15] Ma, X., Gong, Y., He, P., Zhao, H., & Duan, N. (2023). Query rewriting in retrieval-augmented large language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 5303-5315). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.322

[16] Ding, D., Mallick, A., Wang, C., Gan, Y., Sun, S., Zhang, Y., & Zhang, Y. (2024). Hybrid LLM: Cost-efficient and quality-aware query routing. In Proceedings of the International Conference on Learning Representations (ICLR).

[17] Chen, J., Zhang, X., & Han, X. (2023). Explainable identification of tax-related financial reporting anomalies in U.S. public companies: Cross-fitted tax-consistency residuals and Owen-value Shapley propagation. World Journal of Economics and Business Research, 1(1), 33-45.

[18] Es, S., James, J., Espinosa Anke, L., & Schockaert, S. (2024). RAGAs: Automated evaluation of retrieval augmented generation. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (EACL) (pp. 150-158). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.eacl-demo.16

[19] Saad-Falcon, J., Khattab, O., Potts, C., & Zaharia, M. (2024). ARES: An automated evaluation framework for retrieval-augmented generation systems. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL) (pp. 338-354). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.20

[20] Yang, J.-S., Shen, Z., Zeng, Z., & Chen, Z. (2025). Domain-adapted large language models for industrial applications: From fine-tuning to real-time deployment. Computer Science Bulletin, 8(1), 272-289. https://doi.org/10.71465/csb162

[21] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In Proceedings of the International Conference on Learning Representations (ICLR).

[22] Wang, J., Ge, Y., Liu, J., & Jiang, B. (2024). Early warning of liquidity stress in U.S. banks with seasonally aligned frailty hazards and delayed-feedback conformal alarms. Journal of Trends in Finance and Economics, 1(1), 85-100.

[23] Zhou, Y. (2024, December). PANDAX for industrial question answering with explanation (QAE). Zenodo. https://doi.org/10.5281/zenodo.14510798

[24] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL) (pp. 7871-7880). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.703

[25] Pope, R., Douglas, S., Chowdhery, A., Devlin, J., Bradbury, J., Heek, J., Xiao, K., Agrawal, S., & Dean, J. (2023). Efficiently scaling transformer inference. In Proceedings of Machine Learning and Systems (MLSys) (Vol. 5). MLSys.

[26] Yang, J.-S., Zeng, Z., & Shen, Z. (2025). Neural-symbolic dual-indexing architectures for scalable retrieval-augmented generation. IEEE Access, 13, 1-15. https://doi.org/10.1109/ACCESS.2025.3638761

[27] Rhee, M., Zhang, Y., Li, H., Wang, J., & Chen, X. (2026). Enhancing web search agents with self-play contrastive fine-tuning. IEEE Access, 14, 1-14. https://doi.org/10.1109/ACCESS.2026.3717594