STELLAR: STAGE-SEPARATED TESTBED FOR ERROR LOCALIZATION AND ATTRIBUTION IN RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Production RAG systems are multi-stage, but a single end-to-end score cannot identify whether retrieval, reranking, or final selection failed. We introduce STELLAR, a stage-separated testbed and reusable construction pipeline for evaluating L1 whole-corpus retrieval and the L2/L3 ranking pathway separately from L1. Its fixed-pool setting supplies each query with one gold document and mined hard negatives, isolating downstream ranking from retrieval; stage-transition logs in the end-to-end cascade then identify where the gold document is lost. We instantiate STELLAR on 25,178 ServiceNow knowledge-base articles and 8,870 GPT-OSS-120B-generated queries with model-verified hard negatives, evaluating 19 model configurations under enterprise conditions that include long documents, domain terminology, low- overlap queries, and confusable negatives. Specialized rankers can outperform much larger general listwise LLMs: a 1B cross-encoder exceeds a 32B listwise LLM by 7.5 MRR points, while an 8B pointwise reranker statistically ties the 1B model. On public TechQA, BGE and Qwen reverse order and removing BGE improves the final cascade by 4.5 MRR points. Diagnostic slices expose model-specific low-overlap failures and a shared long-document weakness above 10 MRR points. By replacing an opaque aggregate score with stage attribution and query-level evidence, STELLAR supports production model selection and targeted debugging.

1 INTRODUCTION

Production RAG systems are multi-stage. A retriever first searches a knowledge base, a reranker reorders the retrieved candidates, and a final stage selects or scores the documents passed downstream. Yet these systems are commonly evaluated with one end-to-end score. When that score is low, it does not reveal whether the gold document was never retrieved, was removed by the reranker, or was misplaced during final selection (Barnett et al., 2024). This ambiguity is a practical production problem: teams cannot tell which component to replace, which cutoff to tune, or which class of queries to inspect. Even reporting one aggregate number per stage does not identify *which* queries fail or *why*.

Public IR benchmarks do not fully reproduce this setting. Open-web and Wikipedia collections (Bajaj et al., 2016; Kwiatkowski et al., 2019; Thakur et al., 2021; Muennighoff et al., 2022) generally lack the combination found in enterprise knowledge bases: long and heterogeneous documents, product-specific terminology, domain-specific queries that may have low lexical overlap with the answer, and hard negatives drawn from closely related products or procedures. Recent enterprise-focused benchmarks (Miao et al., 2026; Sun et al., 2026) address parts of the domain gap, but emphasize generation-time robustness or multi-source mockups rather than retrieval-and-reranking attribution on a real, single-product knowledge base.

We introduce STELLAR, a stage-separated testbed and reusable construction pipeline for this problem. L1 evaluates whole-corpus retrieval. L2 evaluates candidate reranking. L3 evaluates final ranking and selection. Crucially, L2 and L3 can be evaluated without L1: every query receives a fixed, shuffled pool containing one gold document and up to 20 mined hard negatives. This pool is constructed independently of L1; it is not L1 output with extra negatives appended. The intervention isolates ranking quality from retrieval quality because a reranker is never penalized for a gold

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

document that an upstream retriever failed to surface. Comparing fixed-pool, whole-corpus, and end-to-end results then attributes each failure to retrieval or downstream ranking (Fig. 1).

STELLAR is a testbed rather than a claim that one proprietary corpus is a universal dataset. We instantiate it on 25,178 real ServiceNow knowledge-base articles and 8,870 LLM-generated query–gold-document records. The same construction and evaluation interfaces can be applied to another knowledge base, while the present empirical results remain specific to this ServiceNow instance. Each record carries lexical-overlap, distractor-hardness, length, product-area, content-type, and intent metadata (§3.4), allowing scores to be sliced by the source of difficulty.

Our contributions are threefold. First, STELLAR scores L1 separately from the L2/L3 pathway and records gold-document survival at every transition in the connected cascade. Second, it provides a configurable enterprise testbed construction pipeline with controlled query generation, grounded positives, mined hard negatives, and validation gates. Third, it yields actionable findings across 19 model configurations: specialized rankers can match or outperform much larger general listwise LLMs; controlled TechQA replication reverses both model ordering and the preferred pipeline architecture; and long documents are a shared weakness, while low lexical overlap produces large, model-specific degradation. These findings show why stage-level and per-query evidence is more useful for production debugging than another single leaderboard score.

2 Related Work

Gap 1: stage-wise evaluation and attribution. Open-domain resources such as MS MARCO, Natural Questions, BEIR, and MTEB (Bajaj et al., 2016; Kwiatkowski et al., 2019; Thakur et al., 2021; Muennighoff et al., 2022) primarily provide aggregate retrieval or task scores. Diagnostic frameworks (Su et al., 2024; Pipitone & Houir Alami, 2024; Ru et al., 2024; Barnett et al., 2024) add difficulty strata, retrieval-only scoring, claim-level decomposition, or qualitative failure taxonomies, but do not jointly isolate an embedding retriever, candidate reranker, and final ranking stage on a real enterprise corpus. STELLAR combines independently scoreable fixed-pool stages with the end-to-end cascade needed to attribute production failures.

Gap 2: enterprise-specific testbeds. Public benchmarks are largely built from web, Wikipedia, or vertical-domain text and do not reproduce the document length, terminology, taxonomy, and within-product confusability of an enterprise knowledge base. Recent enterprise RAG benchmarks (Miao et al., 2026; Sun et al., 2026) target generation robustness or synthetic multi-source company settings. STELLAR instead studies retrieval and reranking over a real, single-product enterprise knowledge base. Because the underlying corpus is proprietary, we release the methodology and construction artifacts rather than the documents, following industrial precedent (Ramanath et al., 2021) (§D).

Gap 3: controlled synthetic construction for enterprise retrieval. LLM-generated query–document pairs are established as useful training data in domain-targeted retrieval; InPars is a representative example (Bonifacio et al., 2022). STELLAR uses synthetic generation for a different purpose: constructing an evaluation testbed with explicit quality, grounding, and hard-negative validation gates plus per-query diagnostics. Model-based validation also inherits the known concerns of LLM-as-judge methods, including evaluator bias (Liu et al., 2023). STELLAR therefore treats its generator and judge as replaceable components, reports an independent human audit, and explicitly limits claims from the single-judge instantiation used here.

To our knowledge, no documented benchmark jointly provides all three properties: stage-level retrieval/reranking attribution, a real enterprise knowledge base, and a controlled synthetic construction pipeline with diagnostic metadata. This claim is bounded by the possibility of undocumented industrial evaluation suites. Appendix F compares the closest precedents in detail.

3 Methods

Notation is summarized in Appendix A and introduced only where first used below. STELLAR separates three operational objectives. L1 is the retriever: it maximizes recall while searching the whole corpus. L2 is the candidate reranker: it promotes the gold document within a bounded shortlist. L3 is the final ranking and selection stage: it assigns the order and cutoff consumed downstream. L1

can be scored alone, and the combined L2/L3 pathway can be scored without L1; the connected cascade records survival across all transitions. Direct L3 isolation from L2 is not part of the current experiment matrix.

3.1 CORPUS AND DATA-GENERATION PIPELINE

The corpus $\mathcal{D}$ consists of $|\mathcal{D}| = 25{,}178$ ServiceNow-internal knowledge-base articles. For each query q, the benchmark provides a gold document $d^+(q)$ and a hard-negative set $\mathcal{N}(q)$ with $|\mathcal{N}(q)| \leq 20$, together forming the fixed candidate pool used in §3.3:

$$\text{Pool}(q) = \{d^+(q)\} \cup \mathcal{N}(q). \tag{1}$$

All triples $(q, d^+(q), \mathcal{N}(q))$ are produced by an automated pipeline. In this instantiation, construction decisions use GPT-OSS 120B or deterministic rules; no human annotator participates during construction. A separate post-hoc human audit is reported below.

STEP 1: Query and positive generation. GPT-OSS 120B generates candidate queries conditioned on a sampled source document and real style exemplars. Validation checks both query quality (realism, specificity, and resistance to trivial title matching) and query–document grounding (whether the designated positive answers the query).

STEP 2: Hard-negative mining. Negatives are mined via BM25 over GPT-OSS-120B-generated confusable queries. Relevance-margin and structural-diversity filters (Jaccard token-set similarity, Eq. 11 in Appendix G) remove candidates that are either irrelevant or template-level duplicates. A model-based correctness check then removes accidental answers. This validates that negatives are difficult without being correct, but uses the same model family as generation and is therefore not independent (§6).

STEP 3: Post-processing. Contamination and contradiction checks plus a frequency cap remove leakage and overrepresented negatives. Multi-positive augmentation writes extra records for queries with more than one correct document. The 8,870 records contain 6,013 distinct exact query strings and 2,227 distinct gold/source documents, so record-level observations are not independent (§6). Full thresholds and equations are in Appendix G.

Plug-and-play configuration. The pipeline exposes component interfaces rather than requiring the models used here. The embedding model, reranker, generator, and validation judge can be replaced independently. A build also controls its target size, prompt templates, and hard- negative count. Evaluation separately configures K_1, K_2, and K_3. Thus the same protocol can be rerun on another knowledge base without preserving this paper's models or $100 \to 21 \to 21$ widths; empirical transfer to another domain is not claimed without a corresponding run.

Figure 1 summarizes both the construction pipeline and the evaluation protocol (§3.3) end to end.

3.2 DATASET STATISTICS

Table 1 reports token-length and pool-size distributions for the evaluated corpus, queries, and candidate pools. Document length is heavy-tailed (median 326 tokens, mean 796, up to 37,496) - the direct motivation for the chunking-not-truncation policy applied throughout §3.3 and Appendix A, since naively truncating at a fixed window would silently discard content from a substantial fraction of documents rather than a rare edge case. Queries are short and low-variance by construction (median 8 tokens, p_{10}-p_{90} range 4-10), consistent with the ServiceNow KB search-query style the generation pipeline was conditioned on (§3.1). Candidate pools are 21 documents by construction (Eq. 1) and are only ever smaller (minimum 13) when some hard negatives collapse into the same document after deduplication.

Figure 4 (Appendix H) shows two related but distinct length distributions: gold-document length specifically (the n=8,870 documents that serve as an answer, rather than Table 1's full n=25,178-document corpus - hence a different median), which is heavy-tailed with a long right tail past 10^3 tokens (motivating chunking rather than truncation), and query length, narrow and roughly bell-shaped around its median of 8 tokens.

Beyond length, every query carries a coarse product-area tag, ServiceNow content-type label, and 7-way LLM-tagged intent (§3.4); all three are long-tailed or skewed (Appendix Fig. 3). We treat every

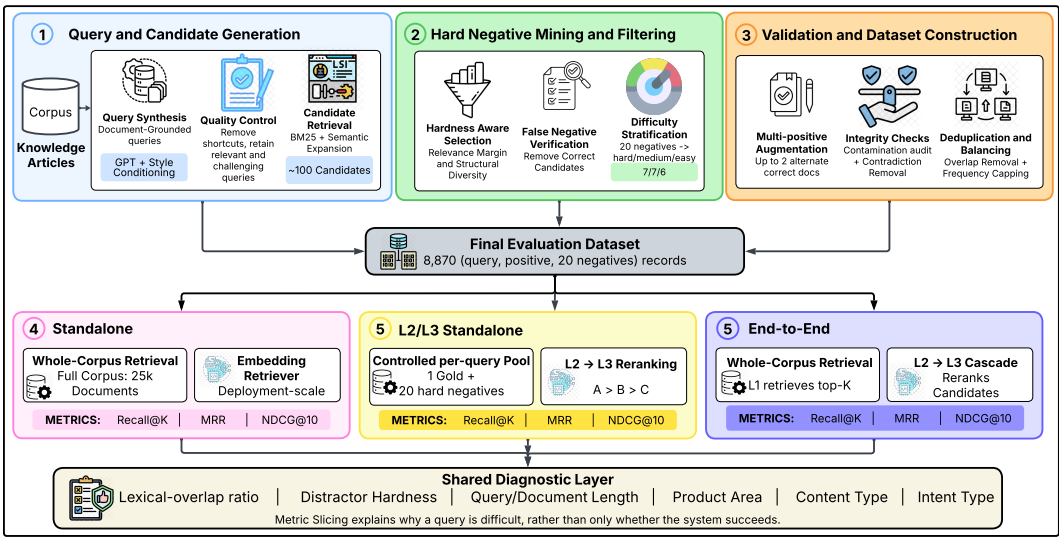

Figure 1: STELLAR construction and evaluation. Construction (top) validates query quality, query–document grounding, and hard-negative correctness before writing the testbed. Evaluation (bottom) scores L1 whole-corpus retrieval, L2/L3 fixed-pool ranking independently of L1, and the connected end-to-end cascade. A shared diagnostic layer slices every metric by *why* a query is difficult rather than only *whether* the system succeeded.

Table 1: Token-length and pool-size statistics. Lengths use whitespace tokens; p_{10}/p_{90} are the 10th/90th percentiles.

Quantity	n	min	p_{10}	median	mean	p_{90}	max
Document length (tokens)	25,178	2	119	326	796.1	1,429	37,496
Document title (tokens)	25,178	1	2	5	6.4	12	51
Query length (tokens)	8,870	2	4	8	7.6	10	15
Query length (characters)	8,870	11	27	43	44.1	62	93
Candidate pool size	8,870	13	21	21	21.0	21	21

category with $n<30$ as descriptive only and make no comparative claim from it; intent categories in the low-reliability triangle are also excluded from fine-grained claims unless independently relabeled. Diagnostic quantities computed per query - lexical-overlap ratio $\rho(q)$ (Eq. 8) and distractor hardness $h(q)$ (Eq. 9) - are shown in Figure 5 (Appendix H) and discussed in §3.4.

Public replication. We rerun the protocol on TechQA (Castelli et al., 2020): 610 real technical-support questions and 28,481 IBM technical documents. Chunked EmbeddingGemma retrieves top-100 candidates; each isolated pool contains the gold plus its top-20 non-gold candidates. We then evaluate the same bge-v2-m3 and Qwen3-32B configurations alone and in cascade. This public build uses embedding-mined rather than GPT-OSS-mined negatives, testing transfer under a distinct, fully reproducible pool construction.

3.3 Evaluation protocol

The evaluation artifacts use a BEIR-style (Thakur et al., 2021) corpus/query/qrels layout, extended with a fixed-pool file (Eq. 1, one line per query) and a per-query diagnostic file (§3.4). The proprietary corpus file is withheld; the queries, qrels identifiers, pools, diagnostics, construction code, and evaluation code are the releasable artifacts. The protocol mirrors the three-stage cascade and evaluation modes of our existing internal reranking-evaluation framework, so it requires no bespoke scorer; metrics are computed with *pytrec_eval*. A retriever or reranker at stage i maps a query and an input candidate list to an output ranking $R_i(q)$, truncated to width K_i:

$$R_1(q) = \mathrm{Retrieve}(q,\ \mathcal{C}_1), \quad |R_1(q)| = K_1, \tag{2}$$

$$R_2(q) = \mathrm{Rerank}(q,\ R_1(q)), \quad |R_2(q)| = K_2, \tag{3}$$

$$R_3(q) = \mathrm{SelectRank}(q,\ R_2(q)), \quad |R_3(q)| = K_3, \tag{4}$$

with $K_1 \geq K_2 \geq K_3$ (this paper's instantiation: $100 \to 21 \to 21$; the P1 width sweep, Appendix H.5, deliberately sets $K_2{=}K_3$ as a controlled exception to compare cascade width directly). Documents exceeding a model's input window are split into windows and assigned the maximum chunk score rather than truncated. This preserves coverage but can still split evidence across windows or dilute a relevant passage among many chunks (§6). Equations 2- 4 are evaluated in three configurations that differ only in the input candidate list $\mathcal{C}_1$ given to Stage 1:

Mode	$\mathcal{C}_1$	What is scored
Whole-corpus (WC)	$\mathcal{D}$ (all 25,178 docs)	Stage 1 alone, against *qrels.tsv*.
Per-query (per-q)	$\mathrm{Pool}(q)$ (Eq. 1, $\lvert R_1(q)\rvert \leq 21$)	Stages 2-3 re-rank only this fixed, shuffled (seed=42) pool, never the full corpus - isolates reranking quality from retrieval quality; a reranker cannot be penalized for a document Stage 1 never surfaced, since Stage 1 plays no role here.
End-to-end	$\mathcal{D}$, as in WC	$R_1(q)$ is passed into Stages 2-3 rather than scored alone.

All three configurations are scored with the same three metrics, given the 1-indexed rank of the gold document in a ranking R, $\mathrm{rank}(d^+(q); R)$ (defined as ∞ if $d^+(q) \notin R$), and its binary relevance $\mathrm{rel}_i = \mathbb{1}[R_i = d^+(q)]$:

$$\mathrm{Recall@}K(q) = \mathbb{1}\big[\mathrm{rank}(d^+(q); R) \leq K\big], \tag{5}$$

$$\mathrm{MRR}(q) = 1/\mathrm{rank}(d^+(q); R), \tag{6}$$

$$\mathrm{NDCG@}K(q) = \frac{1}{\mathrm{IDCG@}K} \sum_{i=1}^{K} \frac{\mathrm{rel}_i}{\log_2(i+1)}. \tag{7}$$

Because every query has exactly one relevant document, $\mathrm{IDCG@}K = 1$ for $K \geq 1$ and Eq. 7 reduces to a single term at $i = \mathrm{rank}(d^+(q); R)$ when that rank is $\leq K$, and 0 otherwise. All three metrics are averaged over queries.

Why three metrics. All tables report all three; prose and figures foreground whichever is most informative for that comparison. Full reasoning for which metric is foregrounded where, and why, is in Appendix G.

Attribution. Comparing the end-to-end score against WC and per-q directly localizes a low end-to-end result: if $d^+(q) \notin R_1(q)$ under WC, no reranker can recover it (a retrieval-recall ceiling,

independent of Stages 2-3); if $d^+(q) \in R_1(q)$ but is not promoted after Eqs. 3-4, the failure is attributable to reranking.

Stage 1 can additionally be scored under per-q. This is a diagnostic manipulation, not a deployment condition - Stage 1 never receives a pre-vetted shortlist in production. Its purpose: shrinking the candidate list to $\mathrm{Pool}(q)$ (Eq. 1) separates (i) the embedding space failing to discriminate $d^+(q)$ from its hardest confusables, from (ii) unrelated documents elsewhere in $\mathcal{D}$ outscoring it due to corpus scale - a distinction WC Recall@K alone conflates, and one that also enables a controlled comparison against Stages 2-3's per-q scores (§3.4).

3.4 Diagnostic metadata

Every query carries metadata enabling Eqs. 5-7 to be aggregated over a slice rather than only over all queries. Let $T(x)$ denote the content-token set of text x (lowercased, alphanumeric, stopwords and single characters removed). The *lexical-overlap ratio*

$$\rho(q) = \frac{|T(q) \cap T(d^+(q))|}{|T(q)|} \tag{8}$$

separates queries a lexical retriever could plausibly solve ($\rho(q)$ near 1) from queries requiring genuine semantic matching ($\rho(q)$ near 0); across the benchmark, ρ has mean 0.775 and median 0.800 (Figure 5, left - note the mode at $\rho = 1$: a sizeable minority of queries are near-verbatim lexical matches of their gold document). Let $\varphi(x)$ denote the TF-IDF vector of document x under a vectorizer fit on $\mathcal{D}$. *Distractor hardness* is the mean cosine similarity between the gold document and its hard negatives,

$$h(q) = \frac{1}{|\mathcal{N}(q)|} \sum_{n \in \mathcal{N}(q)} \frac{\varphi(d^+(q)) \cdot \varphi(n)}{\|\varphi(d^+(q))\| \, \|\varphi(n)\|}, \tag{9}$$

which distinguishes genuinely confusable pools ($h(q)$ large) from weakly-mined ones ($h(q)$ near 0); h has mean 0.264 across the benchmark, right-skewed with most pools in the 0.1-0.4 range (Figure 5, right). Query and document token counts are further bucketed into terciles (short/medium/long) of their respective empirical distributions, and each query is tagged with a coarse product area (from its document's breadcrumb path), the raw ServiceNow content type, a flag indicating whether the gold article has an exact-text duplicate elsewhere in the corpus (0.5% of queries), and a 7-category query-intent label produced by a single Claude Sonnet 5 annotation pass over all 8,870 queries.

Reliability of the intent labels. A test-retest check (second independent annotation pass, Cohen's κ) gives $\kappa = 0.785$ overall, with one low-agreement triangle among the 7 categories; full protocol, the κ equation, and per-category agreement are in Appendix B.

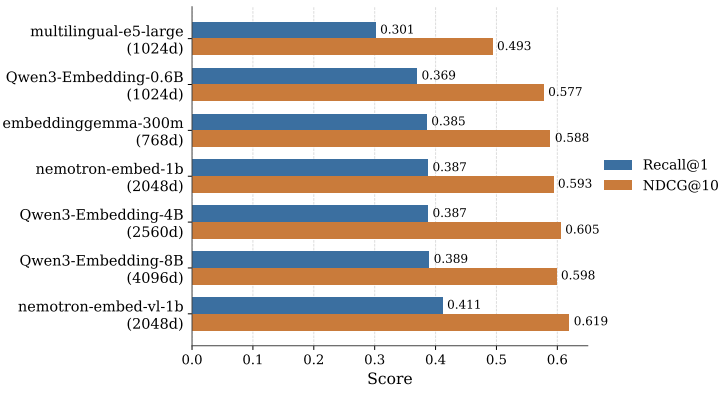

Figure 2: L1 whole-corpus retrieval performance across embedding models. Colors denote model families (E5, Qwen3-Embedding, EmbeddingGemma, and NVIDIA Nemotron variants). Nemotron-Embed-VL-1B achieves the highest Recall@1 and NDCG@10. The Qwen3-Embedding family shows non-monotonic scaling between 4B and 8B on NDCG@10.

Independent label check. One human reviewer audited a random 200 queries, finding 196 (98.0%) golds correct, 3 incorrect, and 1 unsure (95% Wilson interval 95.0–99.2%). This is a single-human spot check, not inter-annotator validation; protocol and item-level caveats are in Appendix C.

Data and artifact availability is stated in Appendix D; the benchmark's limitations are stated in §6.

Table 2: Stage 2/3, per-query mode: fixed 21-document pool, isolated from retrieval quality. R@K denotes Recall@K; N@K denotes NDCG@K. Rows sorted by MRR. *jinaai/jina-reranker-v2-base-multilingual* and *castorini/first-mistral* were originally planned for this model inventory but were dropped after confirming neither is vLLM-hostable.

Model	Type	MRR	R@1	R@10	N@5	N@10
Qwen3-Reranker-8B (Qwen Team, 2025)	pointwise LLM (logit)	**0.868**	**0.805**	**0.983**	**0.885**	**0.895**
nemotron-rerank-1b-v2[1]	cross-encoder	0.863	0.800	0.977	0.879	0.890
Qwen3-Reranker-4B (Qwen Team, 2025)	pointwise LLM (logit)	0.839	0.764	0.978	0.858	0.872
bge-reranker-v2-m3 (Chen et al., 2024)	cross-encoder	0.839	0.766	0.977	0.857	0.871
bge-reranker-large (Chen et al., 2024)	cross-encoder	0.829	0.751	0.976	0.849	0.863
Qwen3-32B (pointwise, same weights)	pointwise LLM (logit)	0.816	0.724	**0.980**	0.840	0.855
Qwen3-32B (listwise, with-CoT)	listwise LLM	0.799	0.714	0.955	0.820	0.835
Qwen3-Reranker-0.6B (Qwen Team, 2025)	pointwise LLM (logit)	0.796	0.704	0.973	0.821	0.838
Qwen3-32B (Qwen Team, 2025) (listwise, no-CoT)	listwise LLM	0.789	0.703	0.951	0.808	0.825
bge-reranker-base (Chen et al., 2024)	cross-encoder	0.785	0.692	0.969	0.808	0.828
Qwen3-8B (Qwen Team, 2025) (listwise, no-CoT)	listwise LLM	0.731	0.615	0.941	0.760	0.779
Qwen3-8B (pointwise, same weights)	pointwise LLM (logit)	0.693	0.553	0.963	0.728	0.756

4 RESULTS

4.1 STAGE 1 (L1): WHOLE-CORPUS RETRIEVAL

Full per-model numbers are in Appendix Table 9; Fig. 2 summarizes the key comparison.

Finding. *nemotron-embed-vl-1b-v2* (text-only use of a VL model) is best on every cutoff. *Qwen3-Embedding*'s 0.6B→4B→8B curve (Fig. 2, orange) is non-monotonic: 0.6B→4B helps on every metric (Recall@1 $0.369 \rightarrow 0.387$, NDCG@10 $0.577 \rightarrow 0.605$), but 4B→8B only ticks Recall@1 up marginally ($\rightarrow 0.389$) while NDCG@10 *regresses* ($\rightarrow 0.598$) despite 8B's larger embedding (4096d vs. 2560d) - mirroring the L2/L3 finding (§4.2) that parameter count does not reliably predict quality here, now shown at retrieval too.

4.2 STAGE 2/3 (L2/L3): PER-QUERY RERANKING, ISOLATED FROM RETRIEVAL

Finding: specialized rankers can beat much larger general listwise LLMs. *Qwen3-Reranker-8B* leads every column but is *statistically tied* with *nemotron-rerank-1b-v2* (1B, cross-encoder) on MRR (0.868 vs. 0.863; gold-document-clustered 95% CI for the difference $[-0.004, 0.013]$, Holm-adjusted p=0.60) despite being $\sim 8\times$ larger. Same-family scaling is now fully monotonic for Qwen3-Reranker ($0.6\text{B} \rightarrow 4\text{B} \rightarrow 8\text{B}$: $0.796 \rightarrow 0.839 \rightarrow 0.868$, every step significant) and helps bge too ($0.785 \rightarrow 0.829 \rightarrow 0.839$), but a CoT toggle on Qwen3-32B only closes part of the gap ($0.789 \rightarrow 0.799$), and the same-weights pointwise-vs-listwise test cuts both ways (helps the 32B checkpoint, hurts the 8B one). These cross-family results are descriptive rather than a controlled effect of formulation: architecture, training, and objective all differ. The matched-weight pointwise-vs-listwise result also changes sign by checkpoint, so formulation requires per-checkpoint measurement. Full breakdown and all significance tests (paired cluster bootstrap over 2,227 gold/source documents, 20,000 resamples) in Appendix H.4 and H.3.

4.3 END-TO-END

Finding: the attribution the whole methodology is named for. Chaining all three stages - Stage 1 whole-corpus retrieval (§4.1, top-100) → Stage 2 bge-reranker-v2-m3 (top-21) → Stage 3 Qwen3-32B listwise, no-CoT - on the full 8,870-query set loses queries at two distinct points, not one: Stage 1 loses 372 outright (Recall@100 $= 0.958$, matching Table 9's nemotron-embed-vl-1b-v2 exactly - these queries never reach Stage 2 at all), and a further 373 are lost when bge's own reranking

truncates its output to top-K_2 (the gold document was retrieved within Stage 1's top-100 but did not survive bge's re-ordering into its own top-21) - a Stage-2 loss, not a retrieval-recall loss. On all 8,870 queries, assigning zero to upstream losses, Stage 3 improves MRR $0.581 \rightarrow 0.610$ (Δ=+0.028, gold-document-clustered 95% CI $[0.022, 0.035]$, Holm-adjusted p<0.001). Conditional on the 8,125 L2 survivors, the corresponding values are $0.635 \rightarrow 0.666$.

Table 3: Candidate-set bridge for bge-v2-m3. "Forced" starts from L1 ranks 1–21 and, when the gold is absent, replaces rank 21 with it. BGE's already-computed top-100 scores are restricted to these 21 documents, holding width and query subset fixed while changing composition.

Candidate condition	n	MRR	R@10
Mined pool (all)	8,870	0.839	0.977
Mined pool (L1 survivors)	8,498	0.854	0.982
L1 top-21, gold forced in	8,498	0.638	0.947
Full cascade through L2 top-21	8,870	0.581	0.861

The matched-width comparison shows that candidate composition, not only missing gold documents, drives the isolated-to-end-to-end gap: MRR falls from 0.854 on mined pools to 0.638 on L1-derived pools even when every pool contains the gold. The identified stage losses remain the transition counts above: 372 at L1 and 373 at L2.

4.4 Cascade: Stage 2 → Stage 3 marginal value

Table 4: Direct and cascaded L2/L3 evaluation on the fixed pools. All rows use all 8,870 queries; queries not forwarded into the top-10 receive zero.

Configuration	MRR	R@1	N@5	N@10
Qwen3-32B direct on 21	0.789	0.703	0.808	0.825
bge direct on 21	0.839	0.766	0.857	0.871
bge top-10 only	0.837	0.766	0.857	0.871
bge top-10 → Qwen3-32B	**0.886**	**0.837**	**0.899**	**0.908**

Finding: L3's value is input-dependent. Qwen3-32B is weaker than bge when each ranks the 21-document pool from scratch (MRR 0.789 vs. 0.839), yet improves bge's top-10 order from 0.837 to 0.886 MRR over all queries. Thus standalone L3 quality does not predict its marginal cascade value; STELLAR measures both interventions rather than selecting only L2 survivors.

4.5 Public replication reverses model and pipeline choices

TechQA supplies a controlled external test: the same checkpoints, fixed-pool intervention, and metrics on real IBM support questions. Table 5 shows two reversals. BGE beats direct Qwen by 5.0 MRR points on ServiceNow but trails it by 12.9 on TechQA. Consequently, BGE→Qwen is preferred on ServiceNow, whereas direct Qwen is 2.9 points better on TechQA because BGE removes 113 golds before L3 (gold-document-clustered 95% CI $[0.001, 0.057]$, Holm p=0.040).

Table 5: Controlled public replication. All MRR values use the full query population; upstream losses receive zero. Bold marks the better of the two direct models or two deployment choices.

Configuration	ServiceNow	TechQA
bge direct, fixed pool	**0.839**	0.587
Qwen3-32B direct, fixed pool	0.789	**0.716**
bge→Qwen, fixed pool	**0.886**	0.687
Qwen direct, fixed pool	0.789	**0.716**

Qwen's 12.9-point direct-model advantage is robust (95% CI $[0.097, 0.163]$, Holm $p<0.001$; 496 gold-document clusters, 20,000 resamples). The full TechQA cascade localizes 47 L1 top-100 misses and 81 subsequent L2 top-21 losses. Qwen raises all-query MRR $0.547 \rightarrow 0.655$ (CI $[0.085, 0.133]$, $p<0.001$), yet direct L1-top-21→Qwen reaches 0.700 (CI for its advantage $[0.019, 0.072]$, $p=0.0016$). Thus omitting BGE is the better public-domain pipeline. This is measured transfer failure under one controlled protocol, not a comparison of incompatible published scores.

4.6 Where models fail: diagnostic-slice ablations

A1-A9 re-aggregate Table 2's 7 per-query reranker outputs, sliced by *query_diagnostics.jsonl* (§3.4); post-hoc, no new model calls. Stage 1 is not sliced by this metadata, so these findings concern L2/L3 only. Three results stand out. First, long gold documents are the only universal large failure mode: every reported model loses 10–14 MRR points from the short to long bucket. Second, low lexical overlap is strongly model-dependent; for example, bge-v2-m3 falls from 0.874 MRR at high overlap to 0.594 at low overlap, whereas Nemotron falls from 0.871 to 0.791. Third, hard distractors reduce MRR from the mid to hard bucket for every model. Query length is *not* a failure mode in this build: every model improves mildly on the long-query bucket. Product-area scores range from 0.72 to 0.95 MRR, while duplicate-query and duplicate-gold slices expose no large scoring artifact. Full numbers are in Appendix H.8 (Table 14).

5 Conclusion

STELLAR replaces a single enterprise RAG score with independently scoreable L1 retrieval, L2 reranking, and L3 final selection, then reconnects them for end-to-end attribution. Across 19 configurations, specialized rankers can outperform much larger general listwise LLMs. A 1B cross-encoder significantly outperforms a 32B listwise LLM, while the largest pointwise reranker only statistically ties the 1B model. This cross-family comparison is descriptive, not a controlled causal effect of formulation. Controlled TechQA replication reverses both BGE/Qwen ordering and the preferred use of BGE in the cascade, showing that one enterprise-style domain does not determine model or architecture choice in another.

The diagnostic slices make the attribution actionable. Long documents cause a shared 10–14 MRR point loss, low lexical overlap creates large but model-specific degradation, and harder negatives reduce every model's score. In contrast, long queries improve mildly in this build and should not be presented as a failure mode. The end-to-end cascade further shows why stages must be separated: retrieval and L2 truncation lose distinct query sets before L3 is evaluated.

We release the construction pipeline, evaluation protocol, prompts, and diagnostic scripts so the testbed can be reproduced or extended. A jina-/RankLLM-style reranker was omitted because it was not vLLM-hostable (§4.2).

Future work. The next validation priorities are to run the full protocol on multiple enterprise and public knowledge bases, test domain transfer, and replace the coupled generator– judge setup with cross-model validation. Larger multi-annotator audits should measure agreement and cover hard negatives as well as positives. Long-context work should compare adaptive chunking, overlapping windows, and native long-context rankers, while future construction runs should test whether alternative generators better reproduce long and compositionally difficult user queries.

AI use statement

Generative AI (Claude, Anthropic; GPT-OSS 120B) was used for: **benchmark construction** - GPT-OSS 120B generated queries, positives, and hard negatives via an existing pipeline (§3.1); no human authored or reviewed individual items during construction itself, though a post-hoc human audit of a 200-query sample was separately conducted (Appendix C); **diagnostic annotation** - Claude Sonnet 5 produced the 7-way intent label (§3.4), with reliability measured via test-retest ($\kappa = 0.785$), not inter-annotator agreement; **writing and code** - paper text, scoring scripts, and figures were drafted with Claude's assistance under author supervision, then independently verified (e.g., correcting an initial 0%-resolution bug and a hard-negative-mining mischaracterization). All statistics are com-

puted by scripts against raw model outputs rather than asserted by the AI tool. The query–gold associations and negative-validity decisions are model-assisted relevance judgments; a post-hoc human audit covers the stated sample only. We take responsibility for all content, including AI-assisted text and artifacts.

ETHICS STATEMENT

The corpus is ServiceNow-internal KB articles with no end-user personal data; queries are synthetic, not drawn from real search logs. The corpus is not released publicly; we release the construction pipeline and protocol for reproduction on comparable, non-confidential data. No human subjects were involved: all queries, labels, and diagnostic tags are produced by automated model calls (§3.1, §3.4).

REPRODUCIBILITY STATEMENT

The evaluation protocol (§3.3) is specified precisely enough to re-implement independently and runs unmodified on our internal reranking-evaluation framework; exact invocation commands are in the companion *EXPERIMENT_PLAN.md*. The data-generation pipeline (§3.1) is fully specified, including all filtering thresholds and every GPT-OSS-120B prompt template verbatim (Appendix G.1), and the script itself is released. Diagnostic statistics (§3.4) are computed by scripts operating on the releasable *query_diagnostics.jsonl*, *qrels.tsv*, and *rerank_pools.jsonl* files. Corpus-dependent internal scores cannot be independently rerun without the private corpus; TechQA provides the executable public replication. The exact release matrix and measured compute budget are in Appendices D and H.7.

6 LIMITATIONS

The study centers on one proprietary ServiceNow knowledge base with synthetic queries and one public TechQA replication; broader enterprise transfer remains untested. The same model also participates in generation and validation for the internal build. Relevance is binary and incomplete, and the 200-query audit uses one human; stronger validation requires independent judges and multi-human adjudication. The 8,870 records share only 5,972 normalized query strings and 2,227 gold/source documents, so all headline inference resamples whole dependency clusters rather than records. Finally, chunking avoids truncation but does not eliminate long-document degradation. Appendix E expands each point.

REFERENCES

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. Seven failure points when engineering a retrieval augmented generation system. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering – Software Engineering for AI (CAIN)*, 2024.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. InPars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2387–2392, 2022.

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Michael McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avi Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. The TechQA dataset. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1269–1278, 2020.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, 2023.

Huiqi Miao, Xinbao Sun, Bo Wang, Fanyu Meng, Lijun Mei, Na Wu, Di Jin, Chao Deng, and Junlan Feng. EnterpriseRAG: Benchmarking llm instruction adherence and robustness under non-ideal enterprise retrieval. *arXiv preprint arXiv:2608.11584*, 2026.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022.

Nicholas Pipitone and Ghita Houir Alami. LegalBench-RAG: A benchmark for retrieval-augmented generation in the legal domain. *arXiv preprint arXiv:2408.10343*, 2024.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Rohan Ramanath, Hakan Inan, Gungor Polatkan, Bo Hu, Qi Guo, Cagri Ozcaglar, Xianren Wu, Krishnaram Kenthapadi, and Sahin Cem Geyik. Towards deep learning-assisted search: Embedding-based retrieval in LinkedIn people search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021.

Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiwei Wang, Pengfei Liu, Yue Zhang, and Zheng Zhao. RAGChecker: A fine-grained framework for diagnosing retrieval-augmented generation. *arXiv preprint arXiv:2408.08067*, 2024.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S. Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O. Arik, Danqi Chen, and Tao Yu. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. *arXiv preprint arXiv:2407.12883*, 2024.

Yuhong Sun, Joachim Rahmfeld, Chris Weaver, Weijia Chen, Roshan Desai, Wenxi Huang, and Mark H. Butler. EnterpriseRAG-Bench: A rag benchmark for company internal knowledge. *arXiv preprint arXiv:2605.05253*, 2026.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual E5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.

A Notation

Table 6 collects every symbol used in §3, so that the main text can introduce each quantity with a single equation rather than a repeated prose gloss.

Table 6: Notation used throughout §3.

Symbol	Meaning
q	a query
d	a document
$\mathcal{D}$	the corpus, $\lvert\mathcal{D}\rvert = 25{,}178$
$d^+(q)$	the gold (relevant) document for query q
$\mathcal{N}(q)$	the hard-negative set for q, $\lvert\mathcal{N}(q)\rvert \le 20$
$\mathrm{Pool}(q)$	fixed candidate pool, $\{d^+(q)\} \cup \mathcal{N}(q)$ (Eq. 1)
$T(x)$	content-token set of text x (lowercased, alphanumeric, stopwords removed)
$\varphi(x)$	TF-IDF vector of document x under a corpus-fit vectorizer
$J(A, B)$	Jaccard similarity, $\lvert A \cap B\rvert / \lvert A \cup B\rvert$ (Eq. 11)
$s(q, c)$	GPT-OSS-120B relevance score of candidate c for query q, $s(q, c) \in [0, 10]$
K_1, K_2, K_3	Stage-1/2/3 cascade widths, $K_1 > K_2 > K_3$
$\mathcal{C}_1$	candidate list given to Stage 1: $\mathcal{D}$ (WC/end-to-end) or $\mathrm{Pool}(q)$ (per-q)
$R_i(q)$	ranked output list of Stage i for query q (Eqs. 2-4)
$\mathrm{rank}(d^+(q); R)$	1-indexed position of $d^+(q)$ in ranking R (∞ if absent)
rel_i	binary relevance of the i-th ranked item, $\mathbb{1}[R_i = d^+(q)]$
$\mathrm{Recall@}K$, MRR, $\mathrm{NDCG@}K$	ranking metrics (Eqs. 5-7)
$\rho(q)$	lexical-overlap ratio (Eq. 8)
$h(q)$	distractor hardness (Eq. 9)
$y_i^{(1)}$, $y_i^{(2)}$	first-/second-pass intent label on item i
$\mathcal{I}$	intent-label category set, $\lvert\mathcal{I}\rvert = 7$
p_o, p_e, κ	observed agreement, chance agreement, Cohen's kappa (Eq. 10)

B Reliability of the intent labels

We drew a stratified sample of 436 queries (up to 70 per category) and re-annotated them in a second, independent Claude Sonnet 5 pass with an identical rubric, blind to the first pass's labels. Writing $y_i^{(1)}, y_i^{(2)}$ for the two passes' labels on item i and $\mathcal{I}$ for the 7-category label set, agreement is measured by observed agreement p_o, chance agreement p_e under the two passes' marginal label frequencies, and Cohen's kappa:

$$p_o = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}[y_i^{(1)} = y_i^{(2)}], \qquad p_e = \sum_{c \in \mathcal{I}} \frac{n_c^{(1)}}{n} \cdot \frac{n_c^{(2)}}{n}, \qquad \kappa = \frac{p_o - p_e}{1 - p_e}. \tag{10}$$

Because both passes use the same model, κ measures *test-retest* reliability - the model's self-consistency across independent runs - rather than inter-annotator agreement between distinct judges. We obtain $\kappa = 0.785$ (substantial; (Landis & Koch, 1977)), with per-category agreement ranging from 58.6% (*other*) to 97.1% (*troubleshooting_diagnostic*); disagreement concentrates in one triangle (*other* $\leftrightarrow$ *factual_lookup* $\leftrightarrow$ *configuration_reference*), while the remaining four categories each exceed 85% agreement. Findings resting on the low-agreement triangle are reported as lower-confidence; findings based on the other four categories, or on coarser groupings, are not.

C Human audit of gold-document correctness

The construction pipeline's own verification (§3.1, STEP 2) is model-based: the same model that generates a candidate negative also judges whether it is an accidental correct answer. To independently check pipeline output, one of the authors manually audited a random sample of 200 of the 8,870 queries (2.3%), using a purpose-built review interface presenting, for each sampled query, the query text, the gold document $d^+(q)$ in full, and its hard-negative candidates, with a binary correct/incorrect/unsure judgment recorded per document. The audit asks the same question the automated pipeline's own STEP-1 quality gate and STEP-2 verification are meant to answer, but by an independent human reviewer rather than the generating model.

Gold-document correctness: of the 200 sampled gold documents, 196 (98.0%) were judged to genuinely answer their query, 3 (1.5%) were judged incorrect, and 1 (0.5%) was marked unsure. **Hard-negative correctness:** a subset of the sample's hard negatives were additionally reviewed for accidental relevance (i.e., whether a "negative" actually answers the query); no clear reversals were found, with one borderline case flagged unsure on both of its negatives (the same query whose gold document was itself judged incorrect).

This is a single-annotator spot-check on a 2.3% sample, not full-coverage review, and - unlike the intent-label check (Appendix B) - involves only one rater, so no inter-annotator agreement statistic applies; the 98.0% figure should be read as the observed rate on this sample, not a population-level guarantee for the remaining 97.7% of queries. Within that scope, it is the benchmark's only independent (non-model) check on label quality, and finds the automated pipeline's positives to be correct at a high rate.

D Data and artifact availability

Table 7 states the submission artifact boundary explicitly. Document text is proprietary; releasing identifiers and derived records does not make corpus-dependent scores independently reproducible. The public-corpus replication is therefore the independently runnable validation of the protocol.

Table 7: Artifact release matrix. "Included" means included in the anonymous submission artifact.

Artifact	Status	Rationale
Corpus text (*corpus.jsonl*)	Withheld	Proprietary KB content
Queries, qrels IDs, fixed pools	Included	Testbed structure and labels
Diagnostics and human-audit sample	Included	Recompute slices; inspect audited items
Per-query model ranks and run logs	Included	Recompute all aggregate results
Construction/evaluation/analysis code	Included	Rebuild on another corpus
Prompts and pinned run configuration	Included	Audit model-assisted steps

E Detailed limitations

Limited domain breadth. The ServiceNow corpus cannot be released. TechQA provides a fully reproducible technical-support replication, but two domains do not establish transfer to other enterprises, languages, or knowledge-base structures.

Synthetic queries and coupled validation. GPT-OSS-120B generates the queries and also participates in query-quality and hard-negative validation. This can preserve model-specific preferences and miss correlated errors; the current internal build has no cross-generator result.

Limited human annotation. One author audited 200 randomly sampled positives: 196 were correct, three incorrect, and one unsure (98.0%; 95% Wilson interval 95.0–99.2%). This bounded spot check covers 2.3% of records, has no inter-annotator statistic, and reviews only an unspecified subset of negatives.

Incomplete and dependent relevance records. Each record marks one document relevant while other documents are unlabeled. Multi-positive augmentation causes 4,896 records to share query text, and only 2,227 gold/source documents underlie the 8,870 rows. Clustered inference handles dependence but cannot remove false-negative relevance risk.

Long context and diagnostics. Fixed-window chunking with max aggregation can split or dilute evidence, as reflected by the long-document slice. Intent labels come from one model; their test–retest analysis identifies the *other/factual_lookup/configuration_reference* boundary as the least stable region.

F Full related-work comparison

Compares every paper considered with the properties summarized in §2.

Table 8: Full related-work comparison.

Paper	Description	Relation to STELLAR
MS MARCO (Bajaj et al., 2016)	Large-scale question-answering and passage-ranking dataset built from Bing search logs and web passages.	Open-web queries and passages; no enterprise-KB structure and no stage-level or diagnostic evaluation.
Natural Questions (Kwiatkowski et al., 2019)	Real Google search queries paired with long/short answers extracted from Wikipedia.	Wikipedia-sourced, single-stage QA evaluation; no retrieval/reranking separation or per-query diagnostics.
BEIR (Thakur et al., 2021)	Heterogeneous zero-shot retrieval benchmark aggregating multiple open-domain datasets under one corpus/query/qrels format.	Retrieval-only, aggregate scoring; STELLAR adopts its file format but adds reranker isolation, end-to-end attribution, and diagnostic slicing on an enterprise KB.
MTEB (Muennighoff et al., 2022)	Massive multi-task embedding-model leaderboard spanning retrieval, reranking, classification, and further tasks on open-domain data.	Aggregate per-task leaderboard; no stage-to-stage attribution or per-query failure diagnosis, and not enterprise-domain.
BRIGHT (Su et al., 2024)	Reasoning-intensive retrieval benchmark stratifying difficulty into lexical, paraphrase, and reasoning tiers across web, science, and code domains.	Introduces the difficulty-stratification principle that STELLAR's lexical-overlap-ratio diagnostic (Eq. 8) generalizes to an enterprise KB; not enterprise-domain, no reranker-stage isolation.
LegalBench-RAG (Pipitone & Houir Alami, 2024)	Retrieval-only, human-annotated benchmark for the legal domain, isolating retrieval quality from generation.	Closest existing precedent for domain-specific, retrieval-only evaluation; single-stage (no reranker isolation) and a different vertical (legal, not enterprise IT).
RAGChecker (Ru et al., 2024)	Claim-level diagnostic framework decomposing RAG evaluation into retriever metrics (claim recall, context precision) and generator metrics (hallucination, noise sensitivity).	Diagnostic in the generation-evaluation sense; does not isolate or attribute failure between a retriever and a downstream reranker.
Seven Failure Points (Barnett et al., 2024)	Qualitative taxonomy of seven recurring RAG failure modes drawn from production case studies.	Motivates stage-specific failure attribution qualitatively; STELLAR operationalizes the same distinction quantitatively via fixed candidate pools (§3.3).
InPars (Bonifacio et al., 2022)	LLM-generated query–document pairs used as synthetic training data for information retrieval.	Precedent for domain-targeted synthetic query generation; STELLAR uses generation to construct a controlled evaluation testbed with validation gates and hard negatives.
G-Eval (Liu et al., 2023)	LLM-based evaluator with strong human alignment that also documents evaluator bias.	Motivates explicit treatment of model-based validation as replaceable and fallible rather than as ground truth; STELLAR additionally reports a human spot check.
EnterpriseRAG (Miao et al., 2026)	Enterprise RAG benchmark simulating retrieval noise, knowledge gaps, and factual conflicts to evaluate instruction adherence.	Targets generation-time robustness under simulated failure injection; not scoped to retrieval-and-reranking stage attribution.
EnterpriseRAG-Bench (Sun et al., 2026)	Approximately 500,000-document, nine-source-type (Slack, Gmail, Jira, Confluence, and others) enterprise benchmark with ten multi-document reasoning query categories.	Synthetic multi-source enterprise mockup at large scale; not a single real production knowledge base, and not scoped to retrieval/reranking isolation.
LinkedIn embedding retrieval (Ramanath et al., 2021)	Industrial paper detailing LinkedIn's embedding-based people-search retrieval system, architecture, and evaluation protocol.	Precedent for the disclosure policy adopted here (methodology released, proprietary corpus withheld, §D); not itself a benchmark paper.

G FULL DATA-GENERATION PIPELINE DETAIL

Expands §3.1's summary with exact thresholds and equations.

Query and positive generation. For each of a sampled subset of source documents ($\min(2.5 \times$ target, $|\mathcal{D}|$) documents, where *target* is the number of source documents sampled for one construction batch, not the final released query count - 2,227 unique documents were sampled for this build), GPT-OSS 120B generates four candidate queries conditioned on the document and on 25 real queries shown as style exemplars. A candidate is discarded if it shares $\geq 50\%$ of its tokens with the document title (trivially keyword-solvable), or if a subsequent GPT-OSS 120B quality gate judges it unrealistic, irrelevant, or answerable by keyword match alone.

Hard-negative mining. Rather than nearest-neighbors of q itself, negatives are mined from five GPT-OSS-120B-generated *confusable queries* - same topic as q, deliberately different vocabulary - each issued against a BM25 index over $\mathcal{D}$ (top-20 results each), supplemented by a wider BM25 pass (top-200) on q with GPT-OSS 120B selecting semantically confusable matches, and by explicit injection of thin-content documents (≤ 60 tokens) sharing topic tokens with q. This $\sim$100-candidate pool is filtered by (i) a margin filter, keeping candidate c only if its GPT-OSS-120B relevance score $s(q, c) \in [0, 10]$ satisfies $2 \leq s(q, c) \leq 8$ (discarding both off-topic and near-correct candidates), and (ii) a structural-diversity filter using the Jaccard similarity of two token sets,

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{11}$$

which removes any candidate whose section headers satisfy $J \geq 0.8$ against $d^+(q)$'s headers (blocking document-template shortcuts). GPT-OSS 120B then selects and orders the 20 hardest survivors; a dedicated pass re-reads each and discards (replacing from the reserve pool) any that in fact answer q. This is the pipeline's only verification step, and it is model-based, not human-based.

Post-processing. Equation 11 is reused for two further guards: a contamination check discards a generated query if $J \geq 0.6$ against any held-out test query (or $J \geq 0.35$ against a style exemplar), and a contradiction check discards a negative $n \in \mathcal{N}(q)$ if $n = d^+(q')$ for some other query q' with $J(q, q') \geq 0.4$. A frequency cap removes negatives appearing in $> 2\%$ of records; records left with < 20 negatives after either check are dropped at this stage. (Table 1's reported minimum pool size of 13 does not contradict this: it arises later, when hard negatives that separately validated records still hold collapse into the same document during final pool deduplication - a finalization step that happens *after*, not instead of, the < 20 drop check above.) Because a query may correctly have more than one correct document, a multi-positive-augmentation step additionally writes, for up to 2 alternate correct documents, a further record $(q, d^{+\prime}(q), \mathcal{N}(q))$ with the same query and negative set.

G.1 DATA-GENERATION PROMPTS, VERBATIM

The GPT-OSS-120B prompt template used at each pipeline stage, exactly as sent (variable placeholders shown in *italics*). STELLAR's queries are short and infrequently use a *Product — Issue* separator (mean 7.6 tokens, 13.7% containing —), which the pipeline's own style auto-detector classifies as the *short_desc* style; the *short_desc* prompt variant is shown below for the two style-dependent stages (query generation, quality gate). Two further styles (*conversational*, *catalog*) share the same structure with domain-appropriate rule text and are released in the same script but omitted here. Two more prompts exist purely as fallbacks for over-length inputs (a per-document 0–10 hardness scorer, and a flat hardest-first rerank) and are omitted for space; they are released in full in the pipeline script.

STAGE 1 — QUERY GENERATION (*short_desc* style)

You are simulating a ServiceNow enterprise employee searching the internal knowledge base or filing an IT support ticket.

TASK: Given the knowledge base article below, write exactly {*n_queries*} realistic search queries that a real employee would type to find this article.

STYLE EXAMPLES — match the format and brevity, do NOT copy or rephrase these:
{*style_examples*}

RULES: 1. Each query must be 3–12 words. Short and direct. 2. Most queries should be plain phrases or short sentences WITHOUT a pipe. Use "Product | Issue" pipe format only when the style examples clearly use it. 3. The query should describe the USER'S PROBLEM OR SYMPTOM — not the solution. Good: "okta mfa not working" Bad: "how to resolve okta MFA token expiry issue" 4. The query must require reading the article to be answered — avoid queries where the answer is obvious from the query text alone. 5. Reflect real user intent: errors, access problems, how-to, hardware/software. 6. All {*n_queries*} queries MUST be answered by the article. Minor typos are fine. 7. No numbered lists, bullets, or headings — queries only. 8. Vary phrasing across queries — each should approach the article from a different angle or symptom so they are not near-paraphrases of each other.

ARTICLE:
{*doc_text*}

OUTPUT: exactly {*n_queries*} queries, one per line, nothing else.

STAGE 2 — QUALITY GATE (*short_desc* style)

You are a strict quality evaluator for a reranker training dataset.

Evaluate the following search query against the knowledge base article. Answer three yes/no questions:

1. REALISTIC — Would a real ServiceNow enterprise employee actually type this query in a knowledge base search? Requirements: 3–12 words, natural language, non-academic, enterprise-appropriate. Reject if it is too long (>15 words), sounds like documentation, or uses formal academic phrasing no real user would type.

2. RELEVANT — Does this article directly and substantively answer the query? Keyword overlap alone is NOT enough. The article must provide the specific information the user needs.

3. HARD — Does answering this query require reading and understanding the article, rather than just matching keywords? Reject if the query text already contains the full topic/solution so that any keyword-based system would trivially retrieve this article without semantic understanding.

Article:
{*doc_text*}
Query: {*query*}

Evaluate each field INDEPENDENTLY and respond with JSON only. Use boolean true/false (not strings). Examples:
{"realistic": true, "relevant": true, "hard": true, "reason": "..."}
{"realistic": true, "relevant": true, "hard": false, "reason": "query reveals the answer"}
{"realistic": false, "relevant": false, "hard": false, "reason": "too long / academic"}

STAGE 3A — CONFUSABLE-QUERY GENERATION

You are building a hard-negative mining dataset for a neural reranker.

TASK: Given a search query and the document that correctly answers it, generate exactly {n} CONFUSABLE QUERIES — queries that are semantically similar to the original but would be BEST answered by a DIFFERENT document in the same corpus.

A good confusable query: shares the same broad topic, product area, or domain as the original; targets a DIFFERENT specific issue, error code, feature, product, version, or scenario — not just a rephrasing of what the document below describes; would plausibly confuse a NEURAL retrieval system into returning a related-but-wrong doc; is NOT answered by the document below; matches the style and length of the original query; MUST use DIFFERENT vocabulary from the original query — e.g. if the original says "okta mfa token", try "authenticator app not working" or "two-factor login error" — same domain, zero shared keywords.

Use the document content to understand the SPECIFIC topic so you can diverge precisely.

Original query:
{*query*}
Document that answers it (read carefully to diverge from it):
{*pos_doc_text*}

Output exactly {n} confusable queries, one per line. No numbering, no bullets, no explanations — queries only.

STAGE 3D — SEMANTIC-DISTANCE FILTER

You are building a hard-negative mining dataset for a neural reranker.

A NEURAL retrieval system finds documents by semantic meaning, not keyword overlap. Below are candidate documents retrieved by keyword search for the query. Identify which ones a neural retrieval system would likely CONFUSE with a correct answer — i.e. they are semantically related to the query topic but do NOT actually answer it.

Focus on candidates that use DIFFERENT vocabulary from the query but cover an adjacent concept, domain, or scenario that could confuse a semantic model.

Query: {*query*}
Candidate documents (index: first ~{*snippet_chars*} chars):
{*candidates*}

Return the indices of the top {*n_select*} most semantically confusable candidates. Prefer candidates with LOW lexical overlap with the query.

Respond with JSON only: {"semantic_candidates": [*integer indices*]}

MARGIN FILTER — RELEVANCE SCORING

You are scoring candidate documents for relevance to a query on a 0–10 scale.

SCORING GUIDE: 0–1 Completely off-topic — no connection to the query domain. 2–3 Superficially related — shares some keywords or domain but clearly wrong. 4–6 Topically related — covers the same area but does NOT fully answer the query. A retrieval system might return this; a careful reader would reject it. 7–8 Closely related — could plausibly be a correct answer; requires careful reading to determine it does not answer the query. 9–10 Directly answers the query — this IS the correct answer or very close to it.

Query: {*query*}
Documents (index: first ~{*snippet_chars*} chars):
{*candidates*}

Return a JSON object mapping each index to its score: {"scores": {"0": 4, "1": 7, "2": 2, ...}}

STAGE 4 — HARD-NEGATIVE SELECTION

You are curating a reranker training dataset. Identify and RANK HARD NEGATIVES.

DEFINITION of a hard negative: topically or lexically related to the query — a retrieval system might return it; does NOT actually answer or resolve the query.

Query: {*query*}
Candidate documents (index: first ~{*snippet_chars*} chars):
{*candidates*}

Select exactly {*num_negs*} hard negatives from the {*n*} candidates above. Order them HARDEST-FIRST: position 0 = most confusingly relevant (looks like it answers the query but does NOT); position {*last_neg*} = least confusing of the selected {*num_negs*}.

Respond with JSON only — key "hard_negatives", exactly {*num_negs*} distinct integer indices from 0 to {*max_idx*}.

STAGE 4.5 — TIER BUCKETING (hard/medium/easy)

You are curating a reranker training dataset. Classify and rank the following confirmed negative documents into three difficulty tiers for the given query.

All {*n*} documents below are CONFIRMED NEGATIVES — they do NOT answer the query.

TIER DEFINITIONS: HARD — semantically very close to the query, would likely be returned by a retrieval system, requires careful reading to distinguish from a correct answer. MEDIUM — topically related but distinguishable with moderate effort. EASY — only superficially related or clearly off-topic, easy to rule out.

Query: {*query*}
Documents (full text):
{*docs*}

Assign ALL {*n*} documents to tiers. Within each tier, order by difficulty (hardest first). Target sizes: HARD={*hard_n*}, MEDIUM={*medium_n*}, EASY={*easy_n*}.

Respond with JSON only: {"hard": [*indices, hardest first*], "medium": [*indices*], "easy": [*indices, easiest last*]}

STAGE 5 — FALSE-NEGATIVE REMOVAL

You are a strict quality checker for a reranker training dataset.

Identify which of the following documents DIRECTLY and SUBSTANTIVELY answer the query below.

Rules: a document answers the query if a user reading it would get the specific information or resolution they need — topical similarity alone does NOT count, the document must provide the actual answer. When uncertain, err toward marking it as answering — a false negative kept in the training set is more harmful than a borderline case being removed.

Query: {*query*}
Documents (index: first ~{*snippet_chars*} chars):
{*docs*}

Return JSON — ONLY the indices of documents that directly answer the query: {"answers_query": [*integer indices*]}. If none answer the query, return {"answers_query": []}

MULTI-POSITIVE AUGMENTATION — ALTERNATE-ANSWER VERIFICATION

You are verifying whether documents directly answer a search query, for the purpose of building a reranker training dataset.

A document answers the query if a user reading it would get the specific information or resolution they need — topical similarity alone is NOT enough.

Query: {*query*}
Documents (index: first ~{*snippet_chars*} chars):
{*docs*}

Return the indices of documents that DIRECTLY and SUBSTANTIVELY answer the query. If none qualify, return an empty list.

Respond with JSON only: {"answers": [*integer indices*]}

Tier labels (*neg_tiers*) are produced by the pipeline for training-oriented downstream use (loss weighting when fine-tuning a reranker on this data) and are not consumed by STELLAR's own evaluation protocol, which treats all 20 negatives in $\mathcal{N}(q)$ uniformly; the tier prompt is included above for completeness of the released pipeline.

WHY THREE METRICS, AND WHEN EACH IS THE HEADLINE NUMBER

All tables report Recall@K, MRR, and NDCG@K; prose and figures foreground one where it is the more informative scalar for that comparison. **Recall@K** answers a yes/no serving-relevant question ("is $d^+(q)$ in the top K a downstream consumer would see?") and is the natural choice for the Attribution logic (§3.3), where $K{=}K_1$ literally gates whether Stages 2-3 have a chance. **MRR** is a strictly monotonic function of $\mathrm{rank}(d^+(q); R)$ with no cutoff to choose, so it is the single scalar used to rank *models against each other* in per-q mode (Table 2 and every ablation derived from it) - with only one relevant document per query, MRR already *is* the query's reciprocal-rank contribution, unlike Recall@K (binary, cutoff-dependent) or NDCG@K (redundant with MRR up to a $\log_2$ rescaling under single-relevant-document scoring, Eq. 7's reduction). **NDCG@K** is reported alongside MRR wherever a cutoff-sensitive view matters in addition (e.g. comparing WC retrieval, where K corresponds to an actual candidate-list truncation downstream, §4.1).

H ADDITIONAL RESULTS

H.1 DATASET-STATISTICS FIGURES

Referenced from §3.2. These figures show category balance, raw lengths, and the diagnostic-score distributions ($\rho(q)$, $h(q)$).

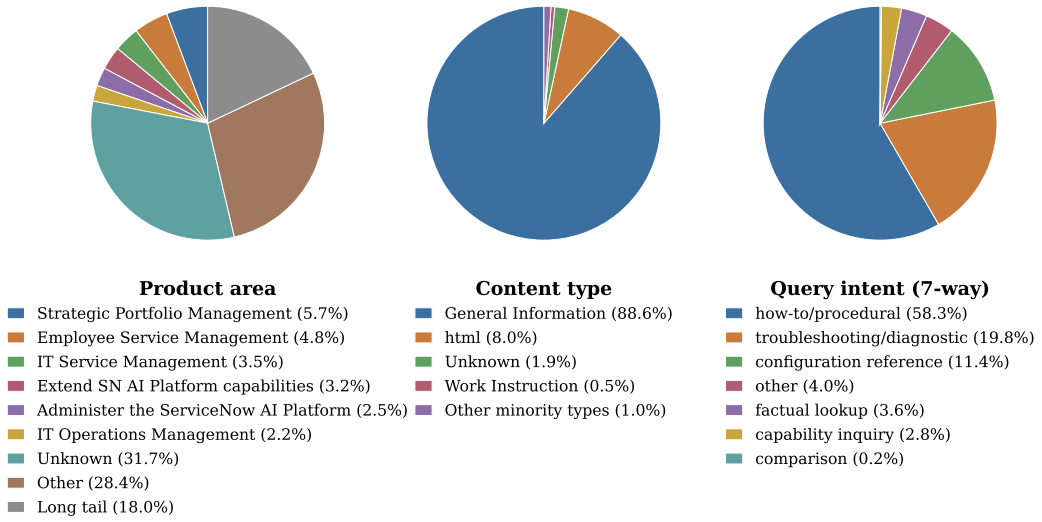

Figure 3: Product area, content type, and query intent distributions, all 8,870 queries.

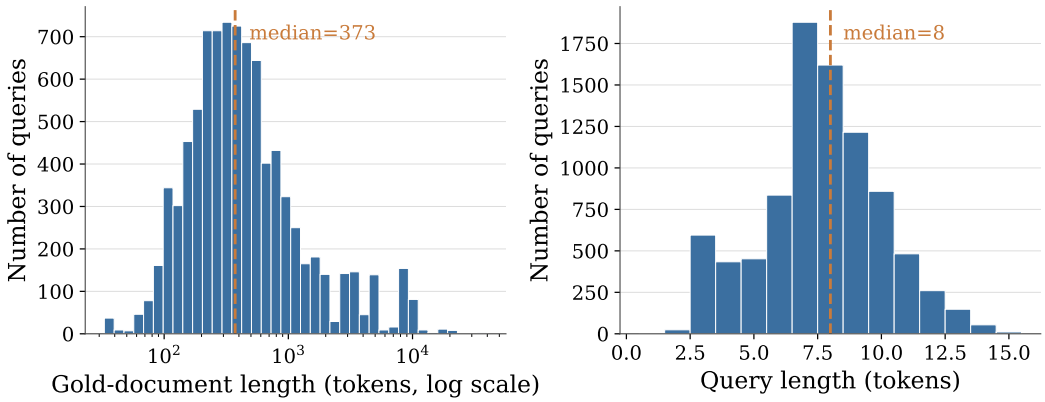

Figure 4: Gold-document length (log-scale x-axis; note the heavy right tail past 10^3 tokens) and query length (narrow, low-variance) across all 8,870 queries. Dashed line: median.

H.2 STAGE 1 (L1), WHOLE-CORPUS MODE: FULL NUMBERS

Table 9 gives the full per-model, per-cutoff numbers underlying Fig. 2 and §4.1's Finding.

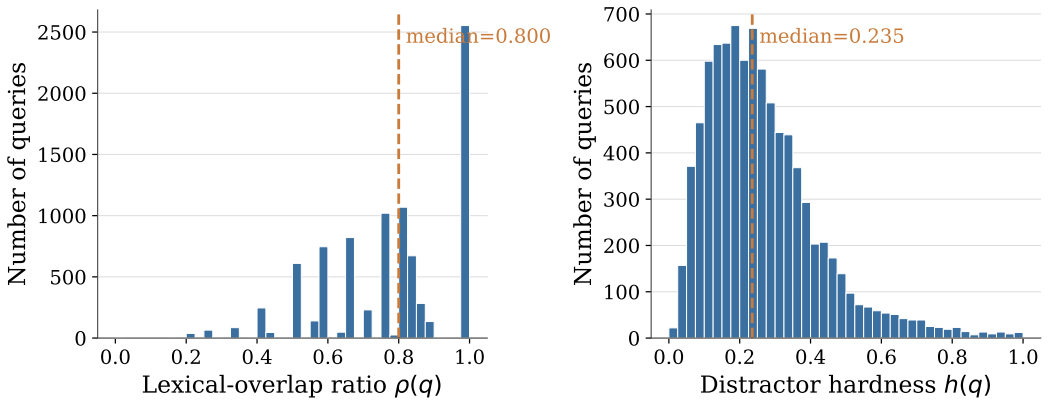

Figure 5: Per-query lexical-overlap ratio $\rho(q)$ (Eq. 8) and distractor hardness $h(q)$ (Eq. 9) across all 8,870 queries. Dashed line: median.

Table 9: Stage 1, whole-corpus mode: full 8,870 queries against the full 25,178-doc corpus, chunked (not truncated). R@K denotes Recall@K; N@K denotes NDCG@K. Bold = best per column.[3]

Model	R@1	R@5	R@10	R@100	N@1	N@5	N@10	N@100
multilingual-e5-large (Wang et al., 2024) (1024d)	0.301	0.600	0.699	0.907	0.301	0.461	0.493	0.537
Qwen3-Embedding-0.6B (Qwen Team, 2025) (1024d)	0.369	0.698	0.789	0.939	0.369	0.548	0.577	0.610
embeddinggemma-300m (768d)	0.385	0.709	0.794	0.947	0.385	0.560	0.588	0.621
nemotron-embed-1b-v2 (2048d)	0.387	0.718	0.798	0.950	0.387	0.566	0.593	0.625
Qwen3-Embedding-4B (Qwen Team, 2025) (2560d)	0.387	0.741	0.823	0.956	0.387	0.578	0.605	0.633
Qwen3-Embedding-8B (Qwen Team, 2025) (4096d)	0.389	0.722	0.807	0.953	0.389	0.570	0.598	0.629
nemotron-embed-vl-1b-v2 (2048d)	**0.411**	**0.745**	**0.825**	**0.958**	**0.411**	**0.593**	**0.619**	**0.648**

H.3 Clustered statistical inference

The 8,870 records contain 5,972 normalized query strings and 2,227 gold/source documents; record-wise resampling would therefore understate dependence. We use a paired cluster bootstrap (20,000 resamples, seed 20260921), resampling all records belonging to a cluster together while preserving the record-level MRR estimand. Gold/source document is the primary, more conservative cluster; normalized query text is a sensitivity analysis. Two-sided bootstrap p-values are Holm-adjusted across the 12 planned comparisons within each analysis.

Table 10: Cluster-robust headline comparisons. CIs and adjusted p-values use gold/source-document clusters; the query-text analysis gives the same accept/reject conclusions.

Comparison	ΔMRR	95% CI	Holm p
Qwen3-Reranker-8B vs. nemotron	+0.0046	[−0.0040, 0.0131]	0.600
Qwen3-Reranker-8B vs. 4B	+0.0286	[0.0215, 0.0360]	< 0.001
Qwen3-Reranker-4B vs. 0.6B	+0.0426	[0.0326, 0.0527]	< 0.001
nemotron vs. Qwen3-32B listwise	+0.0746	[0.0645, 0.0847]	< 0.001
Qwen3-Reranker-4B vs. bge-v2-m3	+0.0001	[−0.0097, 0.0099]	0.975
Qwen3-32B CoT vs. no-CoT	+0.0108	[0.0059, 0.0160]	< 0.001
Qwen3-32B pointwise vs. listwise	+0.0271	[0.0158, 0.0384]	< 0.001
Qwen3-8B pointwise vs. listwise	−0.0381	[−0.0518, −0.0247]	< 0.001
Qwen3-Reranker-0.6B vs. Qwen3-32B listwise	+0.0078	[−0.0040, 0.0198]	0.590

The central descriptive results survive the dependency correction: nemotron exceeds the 32B listwise model by 7.5 MRR points, Qwen3-Reranker scales monotonically, and the same-weight pointwise/listwise effect changes sign between 8B and 32B. Two earlier borderline claims do not: Qwen3-Reranker-8B remains tied with nemotron, and the 0.6B reranker does not significantly beat the 32B listwise model after clustering and multiplicity correction.

A2 non-monotonicity, tested. The weak-vs-mid distractor-hardness dip noted in §4.6 is statistically real for two of three models checked (nemotron: $\Delta{=}-0.019$, CI $[-0.038,-0.001]$, $p{=}0.044$; Qwen3-Reranker-4B: $\Delta{=}-0.026$, CI $[-0.046,-0.006]$, $p{=}0.009$) but not for bge-v2-m3 ($\Delta{=}-0.017$, CI $[-0.037,0.003]$, $p{=}0.094$) - a real but model-inconsistent effect, consistent with treating it as a minor footnote rather than a general pattern.

Scaling-curve caveat. Both same-family curves now have three checkpoints (Qwen3-Embedding and Qwen3-Reranker, each 0.6B/4B/8B), but they disagree in *shape*: Embedding is non-monotonic (4B→8B regresses on NDCG@10 despite the larger embedding dimension, §4.1), while Reranker is monotonic and every step is individually significant (0.6B→4B→8B: $0.796 \rightarrow 0.839 \rightarrow 0.868$, both steps $p{<}10^{-4}$). Three points is still a thin basis for generalizing the *shape* of either curve beyond these specific checkpoints - we report these as measured, significant point-comparisons rather than as claims about the general scaling law of either family, and would want more checkpoints per family (particularly between 4B and 8B, where both curves bend) before treating either curve shape as established. Note also that Reranker's monotonic gain does not translate into a decisive win at the top of Table 2: at 8B it only reaches a statistical tie with the 8×-smaller nemotron-rerank-1b-v2 cross-encoder (clustered, Holm-adjusted $p{=}0.60$, Table 10) - scaling this family closes the gap to cross-encoders but does not clearly surpass them.

H.4 Table 3 scaling/formulation comparisons, in full

Expands §4.2's Finding paragraph with the full three-way breakdown.

Table 11: Controlled comparisons isolating (i) scale, (ii) CoT prompting, and (iii) task formulation using matched configurations from Table 2.

Setting	Observation	Takeaway
Scale (same family)	Qwen3-Reranker 0.6B→4B→8B improves monotonically (0.796→0.839→0.868, both steps significant); bge base→large→v2-m3 also improves (0.785→0.829→0.839). Qwen3-4B ties bge-v2-m3 despite ∼8× more parameters, and even at 8B (now ∼8× nemotron-rerank-1b-v2's size) Qwen3-Reranker only reaches a statistical tie with it (0.868 vs. 0.863, clustered Holm-adjusted $p{=}0.60$), never a clean win.	Scaling helps within-family and can be fully monotonic, but gains saturate across architectures - an order of magnitude more parameters buys a tie with a much smaller cross-encoder, not a win.
CoT (same model)	Qwen3-32B listwise improves slightly with CoT (0.789→0.799), but remains below best cross-encoders (0.863) and Qwen3-4B (0.839).	CoT provides marginal gains but does not close cross-family performance gaps.
Task formulation (cross-family)	Qwen3-0.6B pointwise (0.796) outperforms Qwen3-32B listwise (0.789). In controlled same-weights tests, Qwen3-32B pointwise (0.816) exceeds its listwise variants, while Qwen3-8B pointwise (0.693) underperforms its listwise setup (0.731).	Formulation effects are checkpoint-dependent: pointwise scoring helps at 32B but hurts at 8B, so no general pointwise advantage is supported.

H.5 Pipeline ablations

Sweeping $K_2{=}K_3 \in \{5, 10, 21\}$ on the same 1,500 queries yields a stable Stage-3 marginal MRR gain ($+0.044$ to $+0.045$) while absolute MRR falls with wider, harder pools. Re-shuffling the fixed pools and rescoring two rerankers changes MRR by less than 0.0002, ruling out a positional artifact. CoT and matched-weight formulation results appear in Table 11; latency appears below.

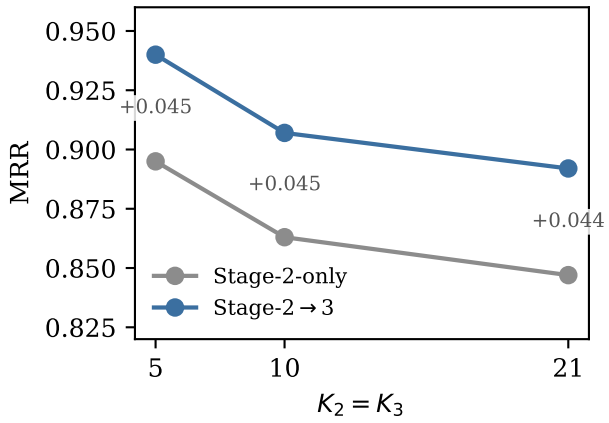

Figure 6: P1: cascade width sensitivity, 1,500-query subset.

H.6 Real per-query latency

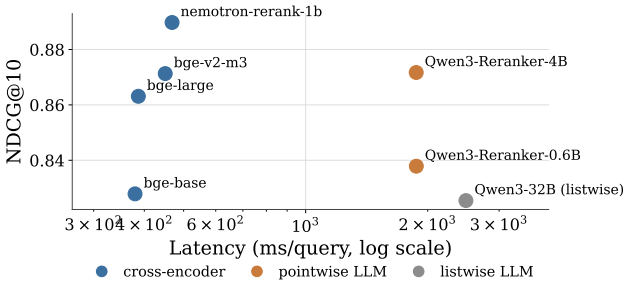

Figure 7: NDCG@10 versus latency for configurations with clean single-host measurements. Cross-encoders define the measured quality–latency frontier; Qwen3-Reranker-8B is omitted because its run crossed a host outage and has no comparable latency.

Latency is real logged wall-clock time (total scoring wall time divided by queries scored), read directly from each model's own run log, not estimated or read off pixels of any figure.

Table 12: Per-query wall-clock latency, the 7 models plotted in Fig. 7. Qwen3-Reranker-8B (Table 2's top scorer) is excluded: its scoring run was fragmented across a host outage, a host move, and a final resume, so its cumulative wall time is not a fair single-host measurement and is not reported here.

Model	ms/query
bge-reranker-base	380
bge-reranker-large	387
bge-reranker-v2-m3	450
nemotron-rerank-1b-v2	468
Qwen3-Reranker-4B	1,874
Qwen3-Reranker-0.6B	1,875
Qwen3-32B (listwise)	2,485

The four cross-encoders average ∼420ms/query; the three LLM-based rerankers average ∼2,080ms/query - a ∼5× gap, with the per-pair range spanning $1{,}874/468 \approx 4.0\times$ to $2{,}485/380 \approx 6.5\times$, consistent with the "4-7×" figure quoted in the main text.

H.7 COMPUTE AND REPLICATION COST

Table 13 reports measured model-scoring wall time, not GPU energy or a hardware-normalized cost. All models were served from local or internal endpoints, so no commercial API charge was incurred. Setup, model loading, and failed/outage intervals are excluded.

Table 13: Measured compute accounting. ServiceNow totals cover the seven configurations with clean single-host logs; five additional configurations lack comparable timing. Public runs used an H100 80GB for embedding and the stated internal model endpoints for reranking.

Experiment group	Queries	Wall time	Scope
ServiceNow fixed-pool, 7 timed configs	$7 \times 8{,}870$	19.51 h	scoring only
TechQA L1 + chunk correction	610	0.09 h	28,481 docs
TechQA BGE fixed + end-to-end	2×610	0.04 h	2 runs
TechQA Qwen direct/cascade	4×610	1.10 h	4 runs
TechQA total measured	–	1.23 h	all public runs

The public replication is therefore feasible in roughly 1.2 measured accelerator/endpoint hours once three model services are available. The full 19-configuration internal matrix is materially more expensive; Table 12 permits readers to budget a chosen subset without implying that all configurations are necessary to reuse the protocol.

H.8 DIAGNOSTIC-SLICE ABLATIONS: FULL FINDINGS AND NUMBERS

Expands §4.6's two-finding summary with all nine ablations.

Table 14: Diagnostic-slice ablations (A1-A9), computed over Table 2's 7 reranker models. Full per-bucket, per-model numbers follow below.

#	Finding
A1	**Model-dependent.** At high lexical overlap (ρ>0.7, n=5,996) bge-v2-m3 and nemotron are statistically tied (MRR 0.874 vs. 0.871); at low overlap (ρ<0.3, n=120, genuine semantic matching required) nemotron leads by a wide margin (MRR 0.791 vs. bge-v2-m3's 0.594) and even the listwise Qwen3-32B beats bge-v2-m3 there (0.738 vs. 0.594). bge-v2-m3 rides lexical overlap more than nemotron does (Fig. 8).
A2	Distractor hardness degrades every model monotonically from mid to hard buckets (e.g. nemotron MRR 0.893→0.810), confirming $h(q)$ tracks genuine difficulty; weak-bucket scores are slightly below mid for six of seven models (all but Qwen3-32B listwise, where weak $0.820 >$ mid 0.814), a mild non-monotonicity worth a footnote, not a red flag.
A3	Query length is *not* a difficulty signal here - MRR rises mildly from short to long for every model (e.g. nemotron 0.851→0.884), the opposite of the naive "longer = harder" assumption; longer queries in this benchmark tend to be more specific, not more ambiguous.
A4	The single largest, most consistent effect of any slice: every model loses 10-14 MRR points from short to long gold documents (nemotron 0.927→0.791, bge-v2-m3 0.888→0.791) - direct evidence for the embedding/reranking-dilution effect motivating the chunking methodology itself (§3.3).
A5	Per-product-area MRR ranges from ∼0.72 (IT Operations Management, weakest for Qwen3-Reranker models) to ∼0.95 (Strategic Portfolio Management, n=507) - a genuine enterprise-specific finding: reranker choice should account for target product-area mix, not just aggregate benchmark score.
A6	Dominated by *General Information* (n=7,858). Only html (711), Unknown (169), and Work Instruction (46) among the minority types meet the $n\geq$30 support rule; all smaller buckets are descriptive only.
A7	The four reliable intent categories broadly mirror aggregate ordering. We make no fine-grained comparative claim for *other*, *factual_lookup*, or *configuration_reference*, regardless of bucket size, because they form the low-agreement triangle in the test–retest audit.
A8	**Corrected figure (see note below): 4,896/8,870 rows (55.2%), not 2,033/8,870, share their query text with ≥1 other row.** Despite the much larger affected fraction than previously documented, per-model MRR is nearly identical between unique and duplicated-text rows for every model (largest gap: bge-reranker-base 0.767 unique vs. 0.800 dup, a 3.3-point difference; bge-v2-m3's own gap is only 1.4 points). Aggregate model ordering is similar across these slices, but the duplicated records invalidate record-wise independence; all headline inference therefore uses query- and gold-document-clustered resampling (Appendix H.3).
A9	The 44 queries whose gold article has an exact-text duplicate elsewhere in the corpus score at least as well as the remaining queries. This checks duplicate corpus content; it is not a count of articles reused as golds across generated queries.

Note on A8. An earlier internal draft of this documentation reported that 2,033 of 8,870 rows share their query text with another row. Recomputing directly from the released query file during this ablation found 2,039 *distinct texts* are duplicated, but the number of *rows* carrying one of those texts is 4,896, not 2,033 - the earlier figure conflated "duplicated texts" with "rows affected." We flag this correction explicitly because it is exactly the kind of undetected labeling error the diagnostic-ablation methodology is meant to catch.

Full per-bucket, per-model MRR/NDCG@10 tables follow. Model abbreviations: Nemo-1B = NVIDIA llama-nemotron-rerank-1b-v2, qw-rr-4B/0.6B = Qwen3-Reranker-4B/0.6B, bge-v2m3/lg/bs = bge-reranker-v2-m3/large/base, qw32B-lw = Qwen3-32B listwise.

H.9 A1: LEXICAL-OVERLAP RATIO $\rho(q)$

Buckets $\rho(q)$ (Eq. 8) into low/mid/high lexical overlap with the gold document. Tests whether a model is riding lexical overlap rather than doing genuine semantic matching; summarized in the main text as Finding A1 (Table 14).

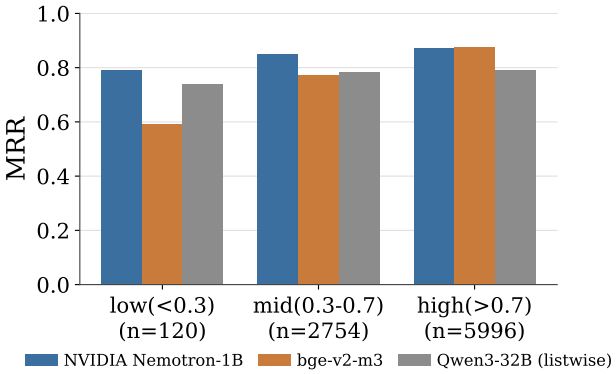

Figure 8: A1: MRR by lexical-overlap-ratio bucket, three representative models. bge-v2-m3 (cross-encoder) is strongest when $\rho(q)$ is high but weakest when it is low; nemotron (cross-encoder, different training) and Qwen3-32B (listwise) degrade far less at low $\rho(q)$.

Table 15: A1: lexical-overlap ratio $\rho(q)$ - MRR (top) / NDCG@10 (bottom) per bucket, per model. n = queries in bucket.

Bucket (n)	Nemo-1B	qw-rr-4B	bge-v2m3	bge-lg	qw-rr-0.6B	qw32B-lw	bge-bs
low($<$0.3) (120)	0.791	0.784	0.594	0.636	0.710	0.738	0.505
NDCG@10	0.837	0.821	0.659	0.681	0.765	0.785	0.575
mid(0.3-0.7) (2754)	0.850	0.828	0.773	0.784	0.790	0.784	0.708
NDCG@10	0.879	0.864	0.816	0.827	0.832	0.821	0.764
high($>$0.7) (5996)	0.871	0.845	0.874	0.853	0.801	0.792	0.826
NDCG@10	0.896	0.876	0.901	0.884	0.842	0.828	0.862

H.10 A2: DISTRACTOR HARDNESS $h(q)$

Buckets $h(q)$ (Eq. 9) into weak/mid/hard distractor pools. A sanity check on the hardness metric itself: scores should degrade monotonically as pools get harder to confuse with the gold document.

Table 16: A2: distractor hardness $h(q)$ - MRR (top) / NDCG@10 (bottom) per bucket, per model. n = queries in bucket.

Bucket (n)	Nemo-1B	qw-rr-4B	bge-v2m3	bge-lg	qw-rr-0.6B	qw32B-lw	bge-bs
weak($<$0.1) (1015)	0.874	0.848	0.846	0.855	0.816	0.820	0.803
NDCG@10	0.895	0.874	0.873	0.882	0.847	0.847	0.839
mid(0.1-0.3) (4902)	0.893	0.874	0.863	0.862	0.829	0.814	0.817
NDCG@10	0.916	0.901	0.891	0.891	0.865	0.847	0.855
hard($>$0.3) (2953)	0.810	0.778	0.798	0.764	0.735	0.736	0.727
NDCG@10	0.845	0.822	0.839	0.810	0.790	0.782	0.779

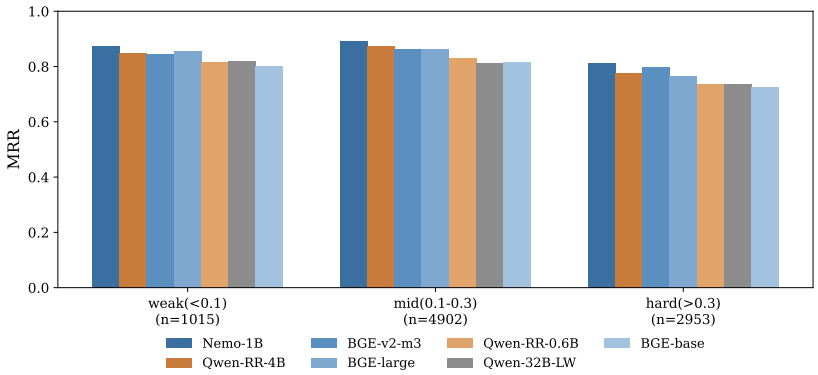

Figure 9: A2: MRR by distractor-hardness bucket, all 7 models, same data as Table 16. The mid→hard degradation is consistent across every model; weak→mid is mildly non-monotonic (§H.10).

H.11 A3: QUERY-LENGTH BUCKET

Buckets queries into terciles of token length. Tests whether longer, more complex-looking queries are actually harder to rank correctly.

Table 17: A3: query-length bucket - MRR (top) / NDCG@10 (bottom) per bucket, per model. n = queries in bucket.

Bucket (n)	Nemo-1B	qw-rr-4B	bge-v2m3	bge-lg	qw-rr-0.6B	qw32B-lw	bge-bs
short (4221)	0.851	0.830	0.832	0.815	0.790	0.763	0.772
NDCG@10	0.880	0.863	0.866	0.852	0.831	0.804	0.816
medium (2835)	0.869	0.836	0.837	0.829	0.799	0.799	0.783
NDCG@10	0.894	0.870	0.869	0.863	0.841	0.833	0.827
long (1814)	0.884	0.865	0.858	0.860	0.808	0.832	0.820
NDCG@10	0.906	0.893	0.888	0.889	0.849	0.863	0.856

H.12 A4: GOLD-DOCUMENT-LENGTH BUCKET

Buckets the gold document into terciles of token length. Tests the embedding/reranking-dilution effect that motivates chunking (§3.3) - does a longer source document make its own correct retrieval harder?

Table 18: A4: gold-document-length bucket - MRR (top) / NDCG@10 (bottom) per bucket, per model. n = queries in bucket.

Bucket (n)	Nemo-1B	qw-rr-4B	bge-v2m3	bge-lg	qw-rr-0.6B	qw32B-lw	bge-bs
short (2517)	0.927	0.904	0.888	0.891	0.899	0.831	0.844
NDCG@10	0.944	0.926	0.910	0.914	0.922	0.861	0.876
medium (3069)	0.888	0.875	0.850	0.858	0.807	0.822	0.809
NDCG@10	0.912	0.901	0.881	0.888	0.847	0.854	0.848
long (3284)	0.791	0.756	0.791	0.753	0.708	0.725	0.719
NDCG@10	0.827	0.802	0.832	0.800	0.764	0.771	0.772

H.13 A5: PRODUCT AREA

Slices by the query's ServiceNow product-area tag (top 8 named areas, plus the *Unknown/Other* catch-alls and remaining long tail). An enterprise-specific slice with no open-domain-IR analogue.

Table 19: A5: product area - MRR (top) / NDCG@10 (bottom) per bucket, per model. n = queries in bucket.

Bucket (n)	Nemo-1B	qw-rr-4B	bge-v2m3	bge-lg	qw-rr-0.6B	qw32B-lw	bge-bs
Administer the ServiceNow AI Platform (226)	0.885	0.890	0.870	0.851	0.851	0.789	0.829
NDCG@10	0.912	0.911	0.896	0.878	0.877	0.825	0.857
Building applications (145)	0.875	0.719	0.849	0.836	0.722	0.763	0.784
NDCG@10	0.904	0.783	0.880	0.874	0.787	0.809	0.828
Employee Service Management (423)	0.896	0.827	0.847	0.823	0.808	0.749	0.789
NDCG@10	0.919	0.869	0.879	0.864	0.851	0.796	0.833
Extend ServiceNow AI Platform capabilities (287)	0.890	0.856	0.896	0.886	0.863	0.815	0.892
NDCG@10	0.914	0.887	0.918	0.912	0.895	0.849	0.916
Finance and Supply Chain (150)	0.939	0.917	0.928	0.906	0.895	0.815	0.853
NDCG@10	0.952	0.937	0.943	0.925	0.922	0.844	0.884
IT Operations Management (195)	0.912	0.805	0.870	0.893	0.716	0.813	0.823
NDCG@10	0.934	0.851	0.903	0.917	0.779	0.842	0.862
IT Service Management (311)	0.883	0.837	0.858	0.838	0.780	0.798	0.796
NDCG@10	0.907	0.873	0.889	0.875	0.830	0.837	0.842
OTHER/LONGTAIL (1298)	0.895	0.896	0.872	0.867	0.851	0.835	0.815
NDCG@10	0.918	0.918	0.898	0.896	0.881	0.865	0.855
Other (2516)	0.821	0.806	0.816	0.801	0.781	0.758	0.759
NDCG@10	0.853	0.843	0.854	0.840	0.825	0.799	0.806
Strategic Portfolio Management (507)	0.945	0.917	0.921	0.932	0.860	0.867	0.900
NDCG@10	0.955	0.936	0.937	0.945	0.892	0.894	0.920
Unknown (2812)	0.852	0.829	0.810	0.800	0.766	0.781	0.752
NDCG@10	0.880	0.862	0.847	0.839	0.811	0.818	0.799

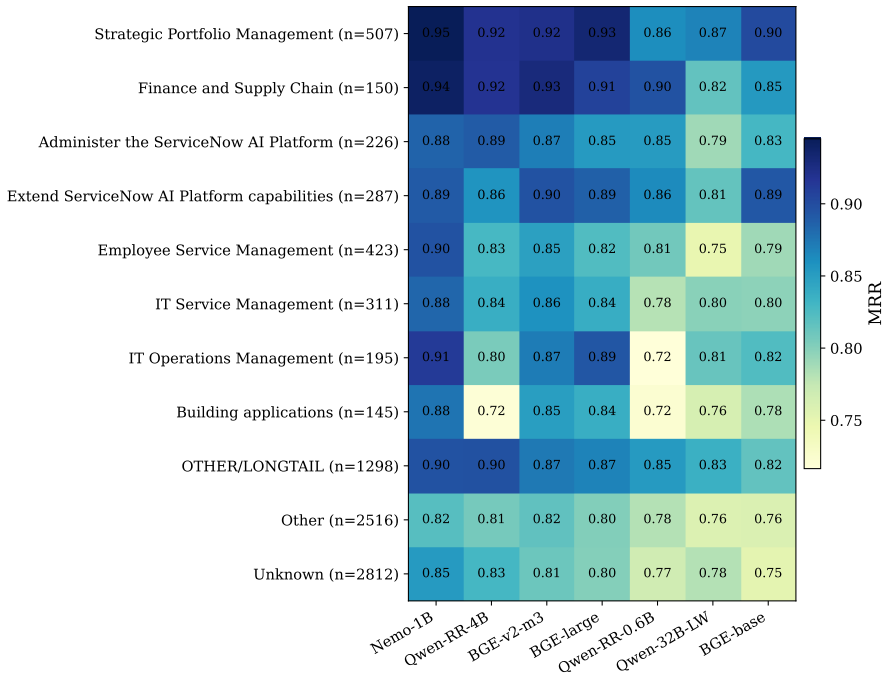

Figure 10: A5: per-product-area MRR heatmap, same data as Table 19. By per-area average, the spread runs from *Strategic Portfolio Management* (∼0.91, strongest) down to *Building applications*/*Other* (∼0.79, weakest), an ∼11-point range; the single lowest *cell* in the table (IT Operations Management, Qwen3-Reranker-0.6B, 0.716, main-text finding) is lower still, since it reflects one model-area pair rather than an area average.

H.14 A6: CONTENT TYPE

Slices by the gold document's raw ServiceNow content type. Categories with $n<30$ are shown for completeness but are descriptive only and excluded from comparative findings.

Table 20: A6: content type - MRR (top) / NDCG@10 (bottom) per bucket, per model. n = queries in bucket.

Bucket (n)	Nemo-1B	qw-rr-4B	bge-v2m3	bge-lg	qw-rr-0.6B	qw32B-lw	bge-bs
Automation Overview (3)	1.000	0.833	0.556	0.442	0.500	0.733	0.437
NDCG@10	1.000	0.877	0.667	0.567	0.620	0.796	0.563
Business Process (10)	0.950	0.950	0.950	1.000	0.950	0.917	0.933
NDCG@10	0.963	0.963	0.963	1.000	0.963	0.936	0.950
Development Notes (3)	0.833	0.436	1.000	1.000	0.778	0.281	1.000
NDCG@10	0.877	0.477	1.000	1.000	0.833	0.450	1.000
General Information (7858)	0.859	0.837	0.836	0.824	0.792	0.783	0.782
NDCG@10	0.886	0.870	0.869	0.860	0.835	0.821	0.825
Guideline (14)	0.696	0.676	0.650	0.750	0.611	0.964	0.660
NDCG@10	0.751	0.716	0.712	0.769	0.645	0.974	0.702
Landing Page (13)	0.823	0.723	0.808	0.769	0.683	0.710	0.720
NDCG@10	0.868	0.788	0.855	0.827	0.734	0.756	0.764
Policy (6)	0.722	0.415	0.462	0.437	0.480	0.313	0.186
NDCG@10	0.794	0.556	0.541	0.476	0.510	0.437	0.238
Solution (7)	0.886	0.893	0.878	0.893	0.893	0.563	0.878
NDCG@10	0.912	0.919	0.905	0.919	0.919	0.627	0.905
Technical Service Overview (1)	1.000	1.000	1.000	1.000	1.000	1.000	1.000
NDCG@10	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Troubleshooting Alerts (14)	0.869	0.917	0.946	0.943	0.907	0.911	0.907
NDCG@10	0.902	0.938	0.959	0.956	0.930	0.933	0.930
Unknown (169)	0.905	0.919	0.901	0.895	0.898	0.912	0.884
NDCG@10	0.923	0.933	0.918	0.912	0.916	0.933	0.903
Work Instruction (46)	0.970	0.969	0.910	0.960	0.958	0.939	0.909
NDCG@10	0.977	0.970	0.932	0.969	0.962	0.948	0.925
Workaround (15)	0.894	0.911	0.939	0.947	0.933	0.750	0.775
NDCG@10	0.919	0.933	0.933	0.959	0.951	0.794	0.829
html (711)	0.890	0.843	0.853	0.852	0.814	0.809	0.795
NDCG@10	0.916	0.873	0.883	0.881	0.848	0.839	0.836

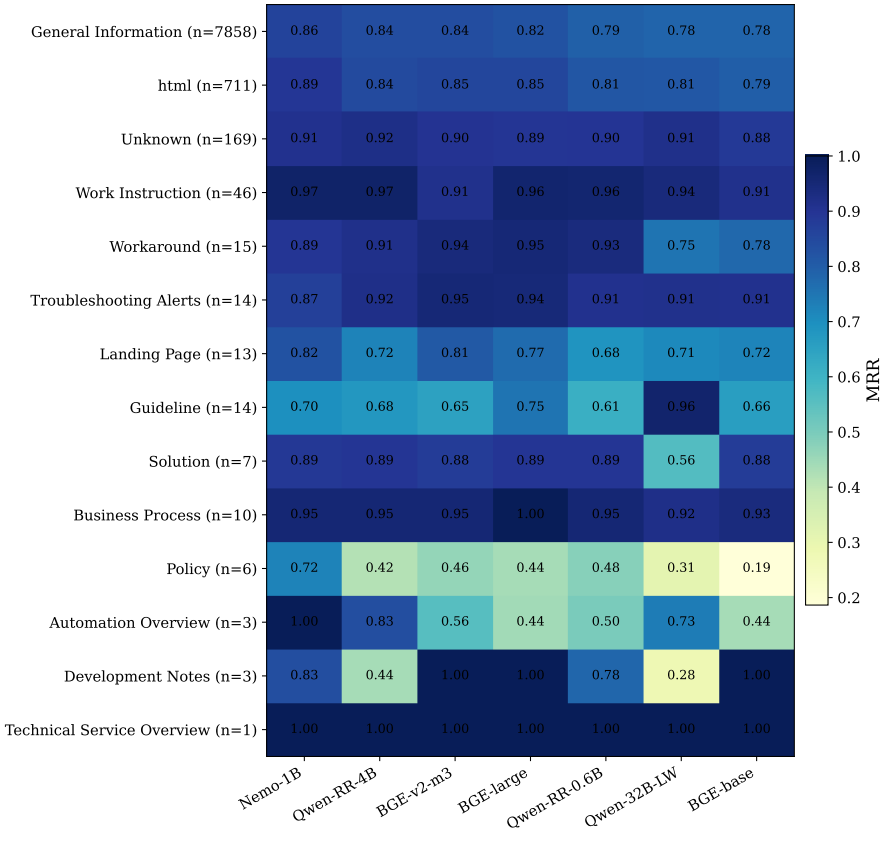

Figure 11: A6: per-content-type MRR heatmap, same data as Table 20. The dominant type (*General Information*, top row, 88.6% of queries) is stable; the tiny-n minority types (bottom rows, $n \leq 14$, several $n \leq 3$) swing widely – visual confirmation these are too small for confident per-type claims.

H.15 A7: QUERY-INTENT TYPE

Slices by the 7-way LLM-tagged query-intent label (§3.4). We exclude the *other/factual_lookup/configuration_reference* triangle from fine-grained comparative claims because it is the low-agreement region of the $\kappa = 0.785$ test–retest check (Appendix B); its values remain descriptive.

Table 21: A7: query-intent type - MRR (top) / NDCG@10 (bottom) per bucket, per model. n = queries in bucket.

Bucket (n)	Nemo-1B	qw-rr-4B	bge-v2m3	bge-lg	qw-rr-0.6B	qw32B-lw	bge-bs
capability_inquiry (245)	0.903	0.861	0.867	0.855	0.800	0.823	0.807
NDCG@10	0.927	0.892	0.891	0.887	0.839	0.851	0.846
comparison (16)	0.969	0.896	0.896	0.906	0.785	0.797	0.896
NDCG@10	0.977	0.923	0.923	0.931	0.837	0.847	0.923
configuration_reference (1007)	0.840	0.814	0.840	0.817	0.792	0.775	0.780
NDCG@10	0.868	0.850	0.874	0.852	0.833	0.815	0.825
factual_lookup (315)	0.895	0.860	0.849	0.848	0.766	0.813	0.780
NDCG@10	0.918	0.886	0.881	0.880	0.815	0.839	0.828
how_to_procedural (5175)	0.857	0.830	0.834	0.815	0.791	0.784	0.770
NDCG@10	0.885	0.864	0.867	0.852	0.833	0.822	0.816
other (353)	0.897	0.862	0.882	0.856	0.818	0.731	0.819
NDCG@10	0.920	0.893	0.906	0.884	0.855	0.781	0.854
troubleshooting_diagnostic (1759)	0.876	0.868	0.837	0.861	0.816	0.811	0.823
NDCG@10	0.900	0.896	0.870	0.890	0.855	0.845	0.857

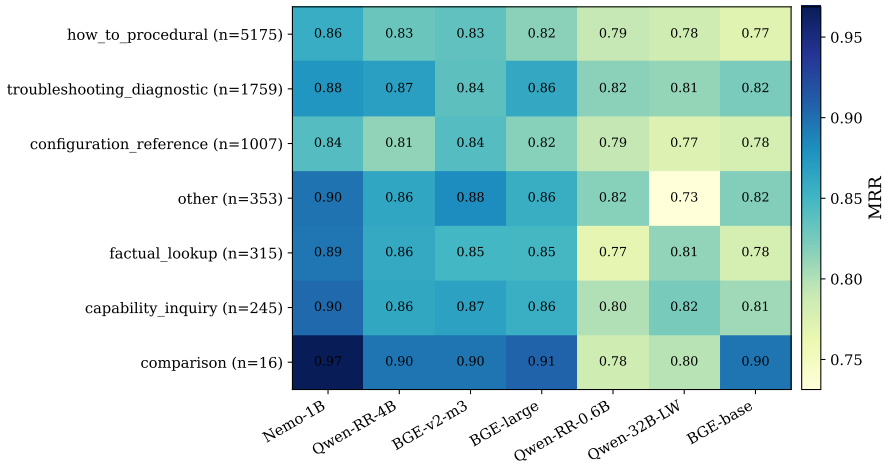

Figure 12: A7: per-intent MRR heatmap, same data as Table 21. The two dominant intents (top two rows, 78.1% of queries combined) sit close together; the smallest bucket (*comparison*, n=16, bottom row) is the noisiest.

H.16 A8: DUPLICATE QUERY TEXT

Splits rows by whether their exact query string is shared with another row (multi-positive augmentation, §3.1). Quantifies the scoring-risk limitation directly - see §6 and the correction noted below.

Table 22: A8: duplicate query text - MRR (top) / NDCG@10 (bottom) per bucket, per model. n = queries in bucket.

Bucket (n)	Nemo-1B	qw-rr-4B	bge-v2m3	bge-lg	qw-rr-0.6B	qw32B-lw	bge-bs
unique (3974)	0.874	0.841	0.831	0.814	0.795	0.771	0.767
NDCG@10	0.901	0.875	0.866	0.852	0.839	0.811	0.813
dup (4896)	0.854	0.838	0.845	0.840	0.797	0.803	0.800
NDCG@10	0.881	0.869	0.875	0.872	0.837	0.837	0.840

H.17 A9: EXACT-DUPLICATE GOLD-ARTICLE TEXT

Splits the 44 queries whose gold article has an exact-text duplicate elsewhere in the corpus from the other 8,826. This flag concerns duplicate corpus content, not reuse of one gold article across multiple generated queries.

Table 23: A9: exact-duplicate gold-article text - MRR (top) / NDCG@10 (bottom) per bucket, per model. n = queries in bucket.

Bucket (n)	Nemo-1B	qw-rr-4B	bge-v2m3	bge-lg	qw-rr-0.6B	qw32B-lw	bge-bs
unique text (8826)	0.863	0.839	0.839	0.828	0.796	0.788	0.785
NDCG@10	0.890	0.872	0.871	0.863	0.838	0.825	0.828
duplicate text (44)	0.864	0.842	0.910	0.915	0.803	0.853	0.843
NDCG@10	0.898	0.882	0.931	0.936	0.852	0.884	0.881

I QUALITATIVE EXAMPLES

Three real records, chosen to make the diagnostic-ablation findings of §4.6 (Appendix H.8) concrete rather than only statistical. Each shows the query, the gold document, and where representative models placed the gold document within its 21-document pool (Eq. 1); rank 1 is correct, higher numbers mean the model buried the right answer under distractors.

EXAMPLE 1 - LEXICAL OVERLAP $\rho(q)=0$: SEMANTIC MATCH VS. SURFACE MATCH (FINDING A1)

Query (q7958): *"I need to list folder contents"*
Gold document: *"Use the GetFiles component"* - an RPA Desktop Design Studio component that returns the names of files in a directory. Zero content words are shared between the query and this title/text ($\rho(q)=0$); solving it requires knowing that "list folder contents" *is* what a "GetFiles component" does, not matching any word.
Distractors in the same pool include: *Microsoft SharePoint Online Spoke*, *Microsoft OneDrive Spoke*, *Use the MoveDirectory component* - all topically adjacent (files, folders, cloud storage) but not the answer.

nemotron-rerank-1b-v2 (cross-encoder): rank **1/21**. Correctly identifies the component from its function, not its wording.

bge-reranker-v2-m3 (cross-encoder): rank **9/21**. Ranks *Microsoft SharePoint Online Spoke* first instead - pulled toward the topically-adjacent but wrong cloud-storage distractors. Direct illustration of A1's finding that bge-v2-m3 rides lexical/topical overlap far more than nemotron does (§4.6).

EXAMPLE 2 - SAME WEIGHTS, OPPOSITE FORMULATION (TASK-FORMULATION FINDING, §4.2)

Query (q1338): *"How do I get a total count of incidents using GlideAggregate?"*
Gold document: *"Scoped GlideAggregate – orderByDesc(String name)"*, a database-aggregation API reference page. Lexical overlap is high ($\rho(q)=0.83$) and the pool is hard ($h(q)=0.36$, max 0.9999 - an almost-indistinguishable distractor is present).

Qwen3-32B, pointwise (same weights as below): rank **1/21**.

Qwen3-32B, listwise (identical checkpoint, identical pool): rank **21/21** - dead last. Nothing about the model changed; only the task formulation (score-each-document independently vs. rank-the-whole-list-at-once) did. This is a single concrete instance of the same-weights ablation's headline result (§4.2): pointwise formulation helped this checkpoint, and here the gap between formulations is total, not marginal.

EXAMPLE 3 - PERFECT LEXICAL OVERLAP, STILL BURIED: DOCUMENT-LENGTH DILUTION (FINDING A4)

Query (q4547): *"Why does my GlideRecord query return all incidents?"*
Gold document: *"Scoped GlideRecord – addUserEncodedQuery(String query)"*, one method's reference entry inside a 10,352-token page documenting the entire GlideRecord API. Every query token appears in the document ($\rho(q)$=1.0, the maximum possible) - lexically, this should be the easiest case in the benchmark.

nemotron-rerank-1b-v2 (cross-encoder, statistically tied for best overall): rank **21/21** - dead last, despite perfect lexical overlap. The one relevant method is a small fraction of a very long page; chunking recovers *some* signal per chunk, but the specific answer is diluted among dozens of other API entries the chunk-and-max-pool scoring must search through. Concrete instance of A4, the single largest effect of any diagnostic slice (§4.6): matching content is not sufficient when the correct answer is a small part of a long document.