WAYFINDER AI

The Chunking Question

Nine Strategies, 36,450 Judgements, Two Winners

An empirical comparison of nine chunking strategies for RAG retrieval across five content types and three query-specificity levels.

Wayfinder Research

April 2026

Executive Summary

Chunking — the decision of how to split a source document into retrievable units — is a core configuration choice in any retrieval-augmented generation (RAG) pipeline. It sets the unit of retrieval and therefore the unit of context available to the reader model.

Despite its centrality, we could not find a published benchmark that compared a broad set of chunking strategies across multiple content types and query-specificity levels in one place. We wanted to produce one.

We ran nine chunking strategies against a corpus of 135 live web pages spanning five content types (blog, documentation, forum, knowledge_base, landing) and three query specificity levels (head, medium, specific). For each (content_type × specificity) cell we generated 27 queries, retrieved the top-10 chunks per strategy using MiniLM-L6-v2 embeddings, and had an LLM judge (Qwen 3.5 122B, temperature 0) grade each retrieved chunk on a 0/1/2 rubric. A 50-pair stratified spot-check recorded 50/50 agreement between the judge and a human reviewer under an informed-review protocol (see §2.6).

The headline result is that there are two winners, not one.

A **sliding-window strategy** (512-token chunks, 128-token step) wins every chunk-level metric: mean LLM-judge score 0.74 [95% CI 0.70, 0.78] vs 0.61 for the next-best strategy, a clean sweep of all 15 stratified cells, and the highest density of relevant chunks deep in the result list (P@10 = 0.53).

A **semantic-heading strategy** (chunks bounded by document headings) wins top-1 precision: P@1 = 0.86, MRR@10 = 0.91, nDCG@10 = 0.86. For any RAG pipeline that operates on only the single highest-ranked chunk, semantic_heading is competitive with sliding_window and slightly ahead on first-hit quality. However, the rank-distribution analysis in §4.3 shows that its advantage is confined to rank 1: from rank 2 onwards, sliding_window returns substantially more score-2 chunks.

A third strategy, **hierarchical chunking**, trails at chunk level (mean-judge 0.61) but ties sliding_window at 0.84 on mean-max-page-score once results are deduped to unique pages. It also produces the lowest unique-page count per top-10 (3.68 vs 4.23–7.52 for other strategies), meaning it returns multiple chunks from the same page more often than alternatives.

At the bottom of the table, **fixed_1024** — 1,024-token fixed chunks with no overlap — is the clear loser on every metric. Five other fixed-size and recursive-character variants cluster in a mid-tier band (mean-judge 0.42–0.48) with overlapping confidence intervals and no meaningful separation.

Four conclusions the data permits. Sliding-window dominates every chunk-level metric and every stratified cell in our data. Semantic-heading leads on top-1 precision and MRR but falls off sharply beyond rank 1. Hierarchical is competitive once results are deduped to pages. The five mid-tier variants are statistically indistinguishable on this benchmark.

1. The Chunking Question

Retrieval-Augmented Generation grounds LLM outputs in retrieved source documents. At the heart of any RAG pipeline sits a chunking strategy — the decision about how to split source text into retrievable units.

LangChain, LlamaIndex, and Haystack all ship multiple chunkers. We could not find a published benchmark that compared a broad set of them across multiple content types and query-specificity levels in one place — which is the gap this study aims to fill.

The study asks three questions:

- **Does a single strategy win across content types?** Blog, documentation, forum, knowledge_base, and landing pages differ in structural regularities (heading density, paragraph length, section coherence). We test whether these differences matter for chunker ranking.
- **Does the ranking hold as queries get harder?** We stratify queries by specificity — head ("what is DNS"), medium ("how do I set a CNAME record"), and specific ("what is the minimum TTL for a TXT record in Route 53") — and test whether chunker ranking is stable across the three.
- **Does the ranking depend on whether results are scored at chunk or page level?** A strategy that surfaces three chunks from one relevant page differs in chunk-level and page-level metrics.

We answer these questions across nine chunking strategies, five content types, three specificity levels, and 405 queries. The study does not attempt a universal ranking — the goal is a ranking that is stable under stratification and reproducible from the seeds and models specified in §2.

2. Methodology

2.1 Corpus

We built a corpus of 135 live web pages, evenly split across five content types:

- **blog** — long-form editorial and technical writing
- **documentation** — developer documentation (Cloudflare, Docker, Kubernetes, GitHub, Vercel and similar)
- **forum** — Stack Overflow, Server Fault, Reddit technical threads

- **knowledge_base** — open-source help centres and developer wikis (Arch Wiki, WordPress Support, NHS.uk, GOV.UK, MediaWiki Help, FreeBSD Handbook). This does not include commercial help centres such as Zendesk, Intercom, or Salesforce Knowledge; we discuss the scoping implications in §3.
- **landing** — product and marketing pages

Each page was fetched, stripped of boilerplate, and stored with its rendered HTML plus a plain-text body. All chunkers operate on the plain-text body against the same page set.

2.2 Chunking strategies

We compared nine strategies:

1. `fixed_256` — non-overlapping 256-token chunks
2. `fixed_512` — non-overlapping 512-token chunks
3. `fixed_1024` — non-overlapping 1,024-token chunks
4. `fixed_512_overlap_50` — 512-token chunks, 50-token overlap
5. `fixed_512_overlap_128` — 512-token chunks, 128-token overlap
6. `sliding_window_512_step_128` — 512-token window, 128-token step (~75% overlap)
7. `semantic_heading` — chunks bounded by document headings
8. `hierarchical` — paragraphs grouped under their headings and sub-headings (median chunk size ~20 tokens; see token-counting caveat below and Appendix D)
9. `recursive_character_512` — LangChain-style recursive character splitter targeting 512 tokens

Tokenisation for the six fixed-size and overlap-based strategies uses the `cl100k_base` tokeniser. The two structural strategies — `semantic_heading` and `hierarchical` — report chunk sizes as a word-count × 1.3 estimate rather than true `cl100k_base` tokens; the hierarchical ~20-token median should therefore be read as a rough-order-of-magnitude figure, not a precise tokeniser measurement. A full-tiktoken-everywhere re-count is straightforward and planned for a follow-up but would not change any chunker's rank order in this study (chunk-size is not a reported metric; only retrieval quality is).

The specific sliding-window parameters (512-token window, 128-token step) were fixed in the study's design document before any analysis was run. They were chosen as a plausible mid-range overlap configuration, not tuned against the benchmark. The same applies to the two explicit-overlap variants (50 and 128 tokens). We make no claim that these are the optimal settings for this pipeline; sweeping window and step jointly is a natural follow-up.

Two strategies from the broader literature — sentence-tiling and paragraph-level chunking — were scoped but not run in this study; we discuss this in Limitations (§3).

2.3 Query generation

For each of the 15 (content_type × specificity) cells, we generated 27 queries for a total of 405. Queries were produced by Qwen 3.5 122B with a task-specific prompt grounded in the page corpus, temperature 0, seed 20260417. A two-stage filter discarded queries that were trivially answerable from a page title or that did not have at least one plausible supporting page in the corpus. 780 queries passed both filters; we then stratified-sampled down to 405 (27 per cell) for cell balance.

Query specificity levels:

- **head** — broad, topical ("what is DNS")
- **medium** — mid-specificity ("how do I set a CNAME record")
- **specific** — narrow, technical, often named-entity bearing ("what is the minimum TTL for a TXT record in Route 53")

2.4 Retrieval

All strategies use the same retriever: MiniLM-L6-v2 sentence embeddings (384-dim), cosine similarity, top-10 chunks per query. Chunks are embedded once per strategy; queries are embedded once and reused across strategies. No reranking, no hybrid search, no filtering. This is an intentionally plain baseline: we wanted to isolate the effect of chunking, not compound it with retriever design choices.

2.5 LLM-judge relevance scoring

Each retrieved chunk was graded by an LLM judge:

- **0** — not relevant
- **1** — partially relevant (provides background or adjacent information)
- **2** — directly answers the query

The judge (Qwen 3.5 122B via Ollama, temperature 0, seed 20260417) received the query, the chunk, and a rubric, and returned a score plus a one-sentence reason. All 36,450 judgements (405 queries × 9 strategies × 10 ranks) were computed in a single pass with zero parse failures and zero judge errors.

2.6 Spot-check review

We ran a 50-pair stratified spot-check against a human reviewer: 10 pairs per content type, 20 score-0 pairs + 15 score-1 pairs + 15 score-2 pairs across the judge distribution. The review was **not blind** — the reviewer saw the judge's score and one-sentence reason before adjudicating, and recorded agree or disagree with the judge's label. This design tests whether the judge's per-pair reasoning holds up on inspection, not whether an independent rater would reach the same label from scratch. Agreement: 50/50, against a target of ≥ 45/50. The ≥

45/50 threshold was set in the study's design document before any judgements were collected; it was not derived post hoc from the observed agreement rate. A blind-calibration study is a natural follow-up (see §3).

2.7 Metrics

We report both chunk-level and page-level metrics:

- **Chunk-level:** mean LLM-judge score, P@1, P@5, P@10, strict P@5 (threshold = 2), MRR@10, nDCG@10
- **Page-level:** Unique pages in top-10, Page Success@5, Page Success@10, Page MRR, Page nDCG, mean-max page score

For page-level metrics we dedupe the top-10 chunks to unique pages, preserving rank order and taking the maximum chunk score per page as the page's grade. Strategies that produce many small chunks can place several chunks from the same page in the top-10; page-level aggregation reports retrieval quality at the granularity of the page rather than the chunk. Page Success@k counts a page as a success if its per-page max score is ≥ 1 (the page contains at least one partially-relevant chunk in the top-k); Page MRR and Page nDCG use the same threshold.

Bootstrap confidence intervals use 1,000 resamples with seed 20260419. Pairwise significance tests use paired bootstrap on mean-judge differences.

---

3. Limitations

We name the limitations of this study before the findings so readers can calibrate what follows. Some are things we scoped but did not do; others are scope boundaries worth flagging.

Sample size: 27 queries per cell. The research plan reserved the option to extend to 40 queries per cell (600 total) as a sensitivity check. We did not run the extension. Anyone wanting to re-run at 40 queries per cell can do so from the specified seed; the methodology is deterministic.

Chunker coverage: nine strategies, not eleven. Sentence-tiling (TextTiling / sentence-boundary chunking) and paragraph-level chunking were originally scoped but not run. Including them would broaden the comparison surface; we do not claim to generalise beyond the nine strategies tested.

End-to-end RAG evaluation. This study measures retrieval quality only — whether a chunking strategy surfaces relevant chunks and pages for a given query. It does not measure whether a downstream LLM reader, given those chunks, produces a faithful or useful answer. Answer-level metrics (faithfulness, hallucination rate, answer quality) require a separate evaluation setup.

Alternative embedding models. All retrieval uses MiniLM-L6-v2 for both chunks and queries. We make no claims about whether chunker ranking is stable under different embedding models; re-running across additional embedding tiers is a natural follow-up.

LLM-judge review design. The 50-pair spot-check was informed, not blind: the reviewer saw the judge's score and reason before recording agree/disagree. This validates that the judge's per-pair reasoning held up on inspection, but does not establish independent inter-rater agreement. A blind-rating exercise — where the reviewer scores chunks from scratch and agreement is computed post hoc — would produce a stricter calibration figure. Similarly, the 50 pairs did not test for systematic biases such as whether the judge scores longer chunks or heading-initial chunks differently at equal relevance; an adversarial audit pairing matched chunks would be required to characterise those.

Knowledge_base scope. As noted in §2.1, our knowledge_base bucket contains developer wikis and open-source help centres, not commercial help centres. Readers should not generalise findings on knowledge_base content to Zendesk/Intercom/Salesforce Knowledge-style surfaces without further testing.

Judge model. The LLM-judge is Qwen 3.5 122B via Ollama. Agreement on the stratified 50-pair spot-check is 50/50. We make no claim about how a different judge model would score the same (query, chunk) pairs; re-running with an alternative judge is a natural follow-up. Seeds and prompts are specified for reproducibility.

---

4. Key Findings

4.1 Two winners, different jobs

The headline table, ranked by mean LLM-judge score:

Strategy	Mean judge	95% CI	P@1	P@5	P@10	Strict P@5	MRR@10	nDCG@10
sliding_window_512_step_128	**0.74**	[0.70, 0.78]	0.83	**0.66**	**0.53**	**0.31**	0.89	0.86
hierarchical	0.61	[0.58, 0.65]	0.72	0.59	0.47	0.22	0.82	0.80
fixed_512_overlap_128	0.48	[0.45, 0.51]	0.82	0.50	0.37	0.17	0.87	0.84
fixed_256	0.48	[0.45, 0.51]	0.82	0.51	0.38	0.16	0.87	0.85
semantic_heading	0.47	[0.44, 0.50]	**0.86**	0.50	0.37	0.17	**0.91**	**0.86**
fixed_512_overlap_50	0.43	[0.40, 0.47]	0.78	0.46	0.34	0.16	0.84	0.82
fixed_512	0.43	[0.40, 0.46]	0.77	0.44	0.34	0.15	0.83	0.81
recursive_character_512	0.42	[0.40, 0.45]	0.79	0.45	0.33	0.15	0.85	0.82
fixed_1024	0.38	[0.35, 0.41]	0.72	0.41	0.30	0.13	0.79	0.78

Strict P@5 uses relevance threshold judge_score = 2 (directly answers). Standard P@k and MRR@10 use threshold ≥ 1. All values rounded to two decimal places; full-precision numbers are available in the released judgements file (Appendix B).

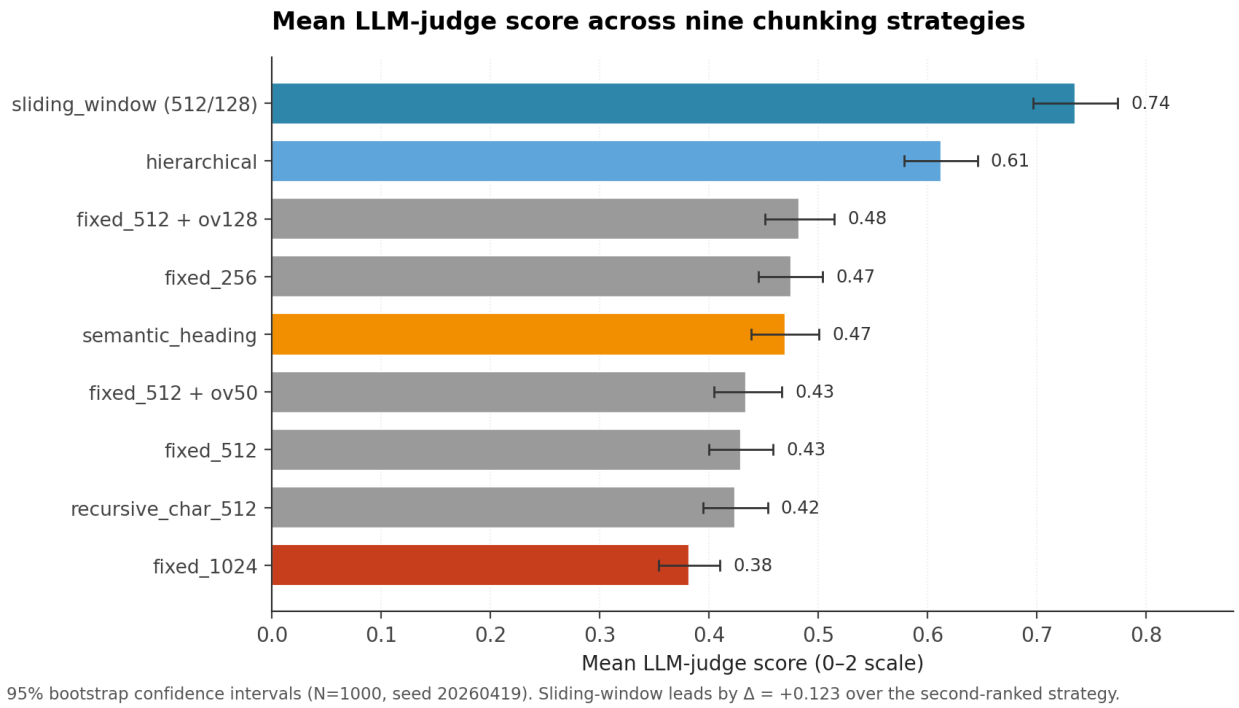


Figure 1: Headline metrics across nine chunking strategies. Mean LLM-judge score with 95% bootstrap confidence intervals. Sliding-window leads by Δ = +0.12 over the second-ranked strategy.

Two strategies lead on different metrics.

`sliding_window_512_step_128` leads on every chunk-level metric except P@1, MRR@10, and nDCG@10. It beats the second-ranked strategy (hierarchical) by Δ = +0.12 on mean judge score (paired bootstrap 95% CI [+0.09, +0.16]; no overlap with zero). All pairwise comparisons vs sliding_window on mean-judge are significant at 95% (see Appendix A).

`semantic_heading` leads on top-ranked metrics: P@1 = 0.86, MRR@10 = 0.91, nDCG@10 = 0.86. The differences vs sliding_window at the top of the ranking are small (P@1 Δ ≈ 0.03, nDCG@10 Δ < 0.01). We did not test whether these differences are statistically significant at this sample size.

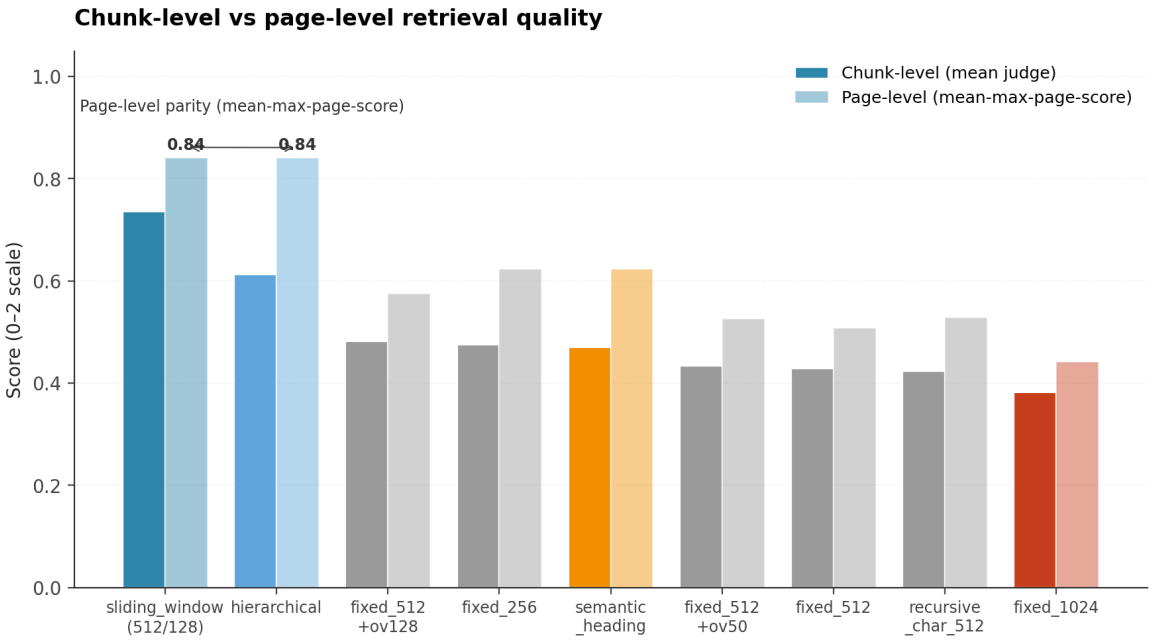
The two strategies lead different metric families: sliding_window leads mean-judge, P@5, P@10, and strict P@5; semantic_heading leads P@1, MRR@10, and nDCG@10. §4.3 shows that semantic_heading's top-1 advantage does not extend beyond rank 1.

4.2 Page-level results differ from chunk-level results

At chunk level, hierarchical trails sliding_window by 0.12 mean-judge points. At page level, hierarchical ties sliding_window at 0.84 mean-max-page-score.

Page-level aggregation method: for each (query, strategy), take the top-10 chunks, dedupe to unique pages preserving rank order, and assign each page the maximum score of any chunk from that page in the top-10.

Strategy	Unique pages in top-10	Page Success@5	Page Success@10	Page MRR	Page nDCG	Mean-max page score
sliding_window_512_step_128	4.23	0.96	0.98	0.86	0.94	**0.84**
hierarchical	**3.68**	0.93	0.97	0.77	0.91	**0.84**
semantic_heading	5.78	0.96	0.98	**0.89**	0.93	0.62
fixed_256	5.44	0.95	0.97	0.86	0.91	0.62
fixed_512_overlap_128	6.05	0.95	0.97	0.86	0.90	0.58
recursive_character_512	6.46	0.92	0.95	0.83	0.88	0.53
fixed_512_overlap_50	6.48	0.93	0.96	0.83	0.88	0.53
fixed_512	6.57	0.91	0.95	0.82	0.87	0.51
fixed_1024	7.52	0.91	0.94	0.78	0.83	0.44



Page-level score is mean across queries of max chunk score per unique page in top-10. Hierarchical's chunk-to-page delta is +0.229 — the largest of any strategy.

Figure 2: Chunk-level vs page-level mean-max-page-score. Hierarchical's chunk-to-page delta is +0.23 — the largest of any strategy — and its page-level score ties sliding-window at 0.84.

Three observations from the table:

Hierarchical's chunk-to-page delta is +0.23 — the largest of any strategy. It ties sliding_window at 0.84 mean-max-page-score. Hierarchical has the lowest unique-page count per top-10 (3.68), meaning on average it returns multiple chunks from the same page.

Page Success@10 is above 0.9 for every strategy. Variation across strategies at page level concentrates in Page MRR and mean-max-page-score, not in whether a relevant page is retrieved at all.

Unique-pages-per-top-10 and mean-max-page-score are inversely ranked in our data: the strategy with the highest unique-page count (fixed_1024, 7.52) has the lowest mean-max-page-score (0.44); the strategy with the lowest unique-page count (hierarchical, 3.68) is tied for the highest mean-max-page-score (0.84).

A side observation on chunk-token budget. Page-level parity between sliding_window and hierarchical at 0.84 mean-max-page-score does not imply parity in the total number of chunk tokens a downstream reader model would receive. Hierarchical's median chunk size is ~20 tokens (Appendix D, with the tokeniser caveat from §2.2), so its top-10 retrieved chunks contain on the order of 200 tokens of chunk text in aggregate. Sliding_window's chunks are 512 tokens, so its top-10 aggregate is roughly an order of magnitude larger in raw chunk tokens. We note this asymmetry for completeness and make **no judgements** in this paper about its practical consequences — prompt-budget fit, context-window utilisation, latency, answer quality, or any other downstream effect. This study measures retrieval quality only; characterising the reader-side impact of chunk-token volume is a separate evaluation (see §3, "End-to-end RAG evaluation").

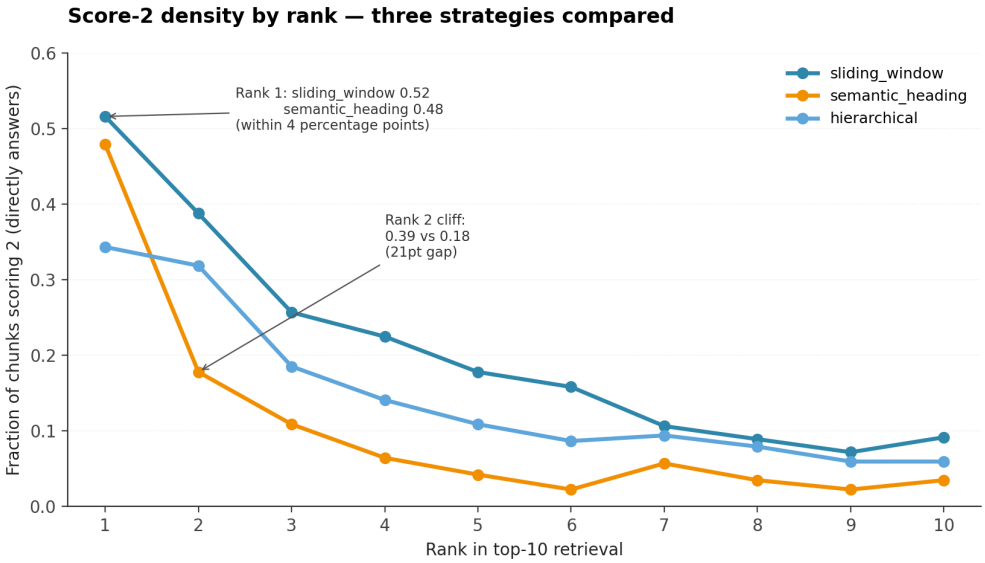
4.3 Top-1 is not the whole story

The headline table shows semantic_heading leading on P@1, MRR@10, and nDCG@10 while sliding_window leads on P@5, P@10, strict P@5, and mean-judge. Those two facts sit in tension only if you stop at the table. Looking at score distribution rank-by-rank explains what's going on.

Rank	sliding_window (frac score-2)	semantic_heading (frac score-2)
1	0.52	0.48
2	0.39	0.18
3	0.26	0.11
4	0.22	0.06
5	0.18	0.04
10	0.09	0.04

At rank 1 the two strategies are essentially tied on the fraction of retrieved chunks that directly answer the query (semantic_heading 0.48, sliding_window 0.52). From rank 2 onwards they diverge sharply: sliding_window retains roughly twice the score-2 density of semantic_heading at every rank through 10.

The mechanism is redundancy. Sliding_window's 75% overlap means rank 2 is usually the window adjacent to the rank-1 hit, so it carries most of the same relevant content. Semantic_heading's chunks are heading-bounded and therefore largely independent: if the directly-answering section is at rank 1, the rank-2 chunk is a different section of the same page (or a different page entirely) and typically scores 0 or 1.



Sliding_window and semantic_heading tie at rank 1. Semantic_heading falls off sharply from rank 2; sliding_window's 75% overlap spreads relevant content across adjacent ranks.

Figure 4: Fraction of retrieved chunks scoring 2 ("directly answers") by rank, top-10. Sliding_window and semantic_heading tie at rank 1; semantic_heading falls off sharply from rank 2 while sliding_window's score-2 density decays gradually through rank 10.

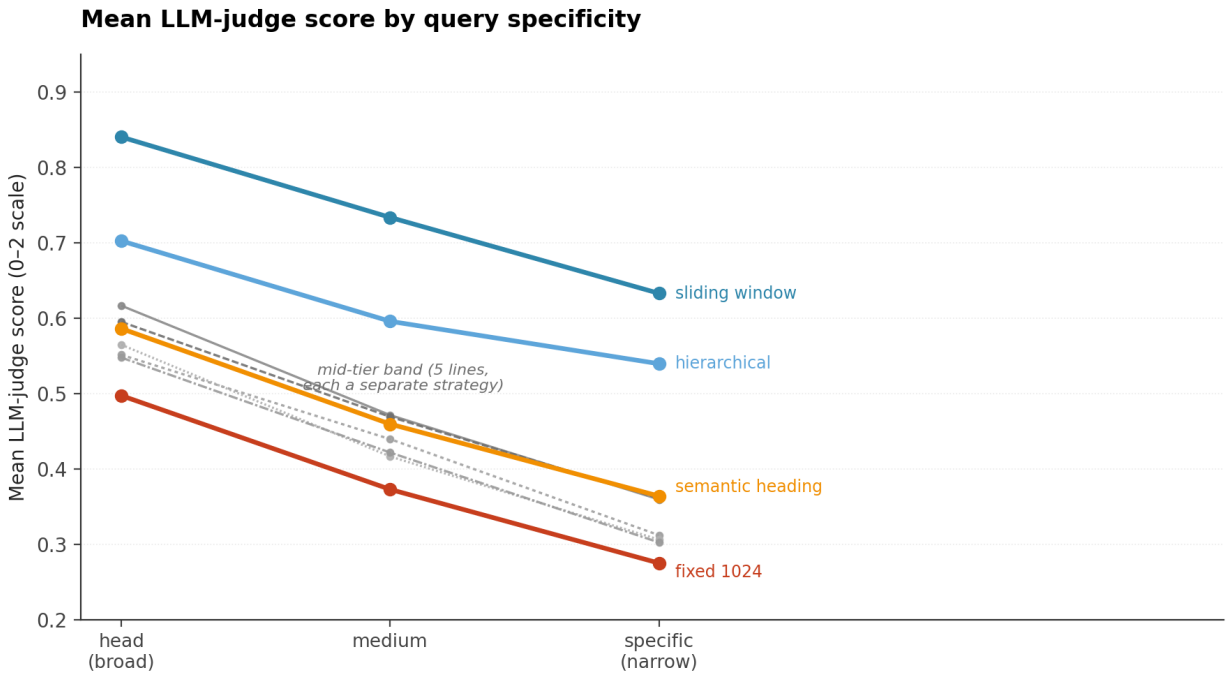
This has a direct practical implication. For a RAG pipeline that passes only the top-1 retrieved chunk to the reader model, semantic_heading and sliding_window are interchangeable on this benchmark. For any pipeline that passes top-k chunks with k ≥ 2 — which is standard — sliding_window delivers materially more directly-answering content. The strict-P@k numbers in §4.1 reflect this: sliding_window's P@5 (0.66) is 0.16 points above semantic_heading's (0.50) because the four chunks after rank 1 carry real retrieval weight for sliding_window and very little for semantic_heading.

The finding does not reverse §4.1 — semantic_heading still leads on the top-1 metric family — but it qualifies how that leadership should be read. "Wins P@1" is a narrower claim than it sounds if the rest of the top-10 collapses.

4.4 The specificity cliff

Query difficulty cuts across all strategies — but not equally.

Strategy	head	medium	specific
sliding_window_512_step_128	0.84	0.73	0.63
hierarchical	0.70	0.60	0.54
fixed_512_overlap_128	0.62	0.47	0.36
fixed_256	0.60	0.47	0.36
semantic_heading	0.59	0.46	0.36
fixed_512_overlap_50	0.55	0.44	0.31
fixed_512	0.56	0.42	0.31
recursive_character_512	0.55	0.42	0.30
fixed_1024	0.50	0.37	0.28



Every strategy degrades monotonically head → medium → specific. Sliding-window degrades least in absolute terms (0.840 → 0.633, drop 0.207).

Figure 3: Mean LLM-judge score by query specificity. Monotonic degradation head → medium → specific for every strategy. Sliding-window degrades least in absolute terms.

Every strategy degrades monotonically from head to medium to specific queries. Sliding-window degrades least in absolute terms (0.84 → 0.63, drop of 0.21). The gap between sliding_window and the mid-tier is larger at the specific level than at head or medium.

4.5 Cross-content-type robustness

We stratified results across the five content types to test whether chunker ranking inverts by content type. It does not.

Strategy	blog	documentation	forum	knowledge_base	landing
sliding_window_512_step_128	0.75	0.71	0.66	**0.82**	0.74
hierarchical	0.64	0.55	0.57	0.72	0.58
fixed_512_overlap_128	0.47	0.44	0.41	0.50	0.59
fixed_256	0.48	0.44	0.40	0.50	0.56
semantic_heading	0.45	0.43	0.39	0.50	0.57
fixed_512_overlap_50	0.47	0.37	0.36	0.44	0.53
fixed_512	0.45	0.37	0.37	0.42	0.54
recursive_character_512	0.41	0.38	0.37	0.42	0.53
fixed_1024	0.38	0.34	0.29	0.38	0.52

`sliding_window_512_step_128` leads every (content_type × specificity) cell — all 15 — on mean judge score. Hierarchical is second in every content type. The rankings below the top two shuffle but the top two are stable across content types.

Knowledge_base has the highest mean scores across every strategy; forum has the lowest; landing, blog, and documentation fall in between. The top-two ranking is unchanged across content types.

4.6 Mid-tier equivalence class

Five strategies cluster in a band of mean-judge 0.42–0.48 with overlapping 95% confidence intervals: `fixed_256`, `fixed_512`, `fixed_512_overlap_50`, `fixed_512_overlap_128`, `recursive_character_512`. Pairwise differences within this group are not significant at 95% (paired bootstrap).

Within the band, two directional patterns hold in our data: overlap increases score (`fixed_512_overlap_128` > `fixed_512_overlap_50` > `fixed_512`), and smaller chunk size increases score (`fixed_256` > `fixed_512`). Tuning overlap from 50 to 128 tokens moves mean-judge from 0.43 to 0.48. These patterns do not move any mid-tier strategy into the CI range of sliding_window or within the top-1-metric range of semantic_heading.

4.7 fixed_1024 is last on every metric

`fixed_1024` ranks last on every metric reported in this study: mean-judge, P@1, P@5, P@10, strict P@5, MRR@10, nDCG@10, Page MRR, Page nDCG, mean-max page score. Its 95% CI on mean-judge does not overlap with the next-worst strategy (`recursive_character_512`).

This result is specific to the retriever (MiniLM-L6-v2) and judge (Qwen 3.5 122B) used in this study. We do not claim to generalise beyond those models.

---

5. Implications

The findings in §4 support a small number of claims specific to the benchmark, retriever, and judge described in §2. We state them narrowly.

`sliding_window_512_step_128` leads every chunk-level metric except P@1, MRR@10, and nDCG@10, leads every (content_type × specificity) cell on mean-judge, and has the smallest absolute head → specific drop. The paired-bootstrap difference vs the next-ranked strategy (hierarchical, Δ = +0.12 mean-judge) does not overlap zero at 95%.

`semantic_heading` leads P@1 (0.86), MRR@10 (0.91), and nDCG@10 (0.86). The rank-distribution analysis (§4.3) shows this advantage is confined to rank 1; from rank 2 onwards semantic_heading's score-2 density collapses to roughly half that of sliding_window. We did not test whether the top-1 differences between semantic_heading and sliding_window are statistically significant at this sample size.

`hierarchical` ties `sliding_window_512_step_128` at 0.84 mean-max-page-score and has the lowest unique-page count per top-10 (3.68). Its chunk-to-page delta (+0.23) is the largest of any strategy in the study.

Five strategies (`fixed_256` , `fixed_512` , `fixed_512_overlap_50` , `fixed_512_overlap_128` , `recursive_character_512`) cluster in a mid-tier band with overlapping 95% confidence intervals on mean-judge. Pairwise differences within this group are not significant at 95%.

`fixed_1024` ranks last on every metric reported, with a 95% CI on mean-judge that does not overlap with the next-worst strategy.

These claims apply to the corpus, embedding model, and judge used in this study. We discuss practical takeaways for RAG engineering and content structuring in a companion blog post, which is outside the scope of this paper.

Technical Appendix

A. Full pairwise significance tests

Paired bootstrap, 1,000 resamples, seed 20260419. Δ is defined as `mean(sliding_window) − mean(other strategy)` on mean LLM-judge score; positive values indicate sliding_window scoring higher. Strategies ordered by the comparison's mean-judge descending.

Comparison	Δ (sliding_window – other)	95% CI	Significant at 95%
sliding_window vs hierarchical	+0.12	[+0.09, +0.16]	Yes
sliding_window vs fixed_512_overlap_128	+0.25	[+0.22, +0.29]	Yes
sliding_window vs fixed_256	+0.26	[+0.23, +0.30]	Yes
sliding_window vs semantic_heading	+0.27	[+0.23, +0.30]	Yes
sliding_window vs fixed_512_overlap_50	+0.30	[+0.27, +0.34]	Yes
sliding_window vs fixed_512	+0.31	[+0.27, +0.34]	Yes
sliding_window vs recursive_character_512	+0.31	[+0.28, +0.35]	Yes
sliding_window vs fixed_1024	+0.35	[+0.32, +0.39]	Yes

All pairwise comparisons against sliding_window are significant at 95%; no confidence interval overlaps with zero.

B. Reproducibility

Seeds. Query generation: 20260417. LLM-judge: temperature 0, seed 20260417. Bootstrap: 20260419.

Models.

- Embeddings: `sentence-transformers/all-MiniLM-L6-v2` (384-dim)

- Query generator: Qwen 3.5 122B via Ollama, temperature 0
- LLM-judge: Qwen 3.5 122B via Ollama, temperature 0

Corpus. 135 pages across 5 content types.

Artefacts. Rebuild scripts, analysis code (headline metrics, stratified breakdowns, page-level aggregation), and the full judgements file (36,450 records of `(query, strategy, rank, chunk_id, page_id, score, reason)` tuples) are available on request: research@wayfinderai.tools.

C. Spot-check composition and protocol

50 stratified pairs: 10 per content type, 20 score-0 pairs + 15 score-1 pairs + 15 score-2 pairs across the judge distribution. For each pair the reviewer saw the (query, chunk, judge_score, judge_reason) tuple in full and recorded agree or disagree with the judge's label. The reviewer was not blind to the judge's output; the review tests whether the judge's reasoning holds up on inspection, not independent inter-rater agreement from scratch. Agreement: 50/50. Pre-registered target: ≥ 45/50.

D. Strategy-level hyperparameters

Strategy	Target chunk size	Step / overlap	Size reported in	Notes
fixed_256	256 tokens	none	`cl100k_base` tokens	Hard boundary, token-aligned
fixed_512	512 tokens	none	`cl100k_base` tokens	
fixed_1024	1024 tokens	none	`cl100k_base` tokens	
fixed_512_overlap_50	512 tokens	50-token overlap	`cl100k_base` tokens	~10% overlap
fixed_512_overlap_128	512 tokens	128-token overlap	`cl100k_base` tokens	25% overlap
sliding_window_512_step_128	512 tokens	128-token step	`cl100k_base` tokens	~75% overlap; parameters pre-registered in the study design, not tuned
semantic_heading	variable	heading-bounded	word count × 1.3 estimate	Chunks align with H1/H2/H3
hierarchical	variable (median ~20 tokens, estimated)	heading hierarchy	word count × 1.3 estimate	Paragraph-under-heading structure; the ~20-token median is an estimate, not a `cl100k_base` measurement (see §2.2)
recursive_character_512	512 tokens (target)	language-aware splits	`cl100k_base` tokens	LangChain default recursive splitter

E. Distribution of LLM-judge scores

Across all 36,450 judgements: 0 = 62%, 1 = 27%, 2 = 11%. The distribution is stable across strategies (within ±3% on each bucket).

Correspondence: research@wayfinderai.tools.