WAYFINDER AI

The Embedder Question

Seven Models, 28,350 Judgements, Two Frontiers

An empirical comparison of open-weights embedding models for RAG retrieval

Shaun Myandee, Wayfinder AI

2026-04-24

Abstract

The embedding model is a core configuration choice in any retrieval-augmented generation pipeline. It determines how queries and document chunks are mapped into a shared vector space, and therefore which chunks are surfaced for a given query. Published benchmarks such as MTEB evaluate embedders on curated academic corpora; we could not find an equivalent evaluation on the kind of messy live-web content that production RAG systems actually ingest.

This paper replicates the Wayfinder Study 01 benchmark methodology — 405 stratified queries across 135 live web pages, judged by an LLM on a 0 / 1 / 2 rubric — but holds the chunker constant at `sliding_window_512_step_128` (the Study 01 deep-retrieval winner) and varies the embedder across seven open-weights models: `all-MiniLM-L6-v2`, `all-MiniLM-L12-v2`, `BGE-small-en-v1.5`, `BGE-large-en-v1.5`, `nomic-embed-text-v1.5`, `mxbai-embed-large-v1`, and `pplx-embed-context-v1-4b`. Total: 28,350 LLM judgements.

The top model (`pplx-embed-context-v1-4b`) scores 0.86 on the mean-judge rubric; the bottom model (`all-MiniLM-L12-v2`) scores 0.54. Five of the six non-baseline models beat the Study 01 baseline (`all-MiniLM-L6-v2`) at $p < 0.001$; one (`all-MiniLM-L12-v2`) falls below it at the same significance level. Two Pareto frontiers emerge when cost is considered — one on storage, one on throughput — and they give different answers: three models on the storage frontier, five on the throughput frontier. Two statistical near-ties surface as well: `mxbai-embed-large-v1` and `BGE-large-en-v1.5` are indistinguishable on mean at overlapping 95% CIs, and `BGE-small-en-v1.5` and `nomic-embed-text-v1.5` produce identical aggregate scores from observably different retrieval behaviours.

1. Introduction

The embedding model is the most replaceable component of a retrieval-augmented generation (RAG) pipeline. Swapping the embedder is usually a one-line change; swapping the chunker or the reader model is a structural decision. This asymmetry makes the embedder the most tempting component to "tune" in practice, and the most important one to understand empirically.

Published evaluations of embedding models are abundant. The MTEB benchmark (Muennighoff et al., 2022) provides a standardised multi-task evaluation across classification, clustering, retrieval, and pairwise similarity; the MTEB leaderboard is the de facto reference for embedder selection. Retrieval-specific benchmarks such as BEIR (Thakur et al., 2021) evaluate on academic information-retrieval corpora with well-specified relevance judgements.

Our concern is that these benchmarks evaluate embedders on corpora and queries that look different from production RAG inputs. Real RAG pipelines ingest blogs, documentation, forum threads, knowledge-base articles, and marketing pages — content with inconsistent structure, noise, boilerplate, and drift. Real queries are generated by users or agents and vary in

specificity. An embedder that performs well on MS MARCO passages may or may not perform well on a live-crawled landing page with a cookie banner, a navigation menu, and five levels of nested headings.

Study 01 (Myandee, 2026) measured nine chunking strategies on a corpus of 135 live web pages and 405 stratified queries, using an LLM as a 0 / 1 / 2 relevance judge. This paper reuses that corpus, that query set, and that judge — and varies only the embedder. The chunker is held constant at the Study 01 deep-retrieval winner.

The research question: given a corpus and query distribution representative of production RAG workloads, which embedding models perform best on retrieval quality, and what do the cost-quality trade-offs look like?

2. Methodology

2.1 Study design

The benchmark replicates Study 01's retrieval-and-judge pipeline with one variable changed. Specifically:

- **Corpus**: 135 live web pages across five content types (blog, documentation, forum, knowledge_base, landing). Identical to Study 01.
- **Queries**: 405 queries stratified by (content_type × specificity), with three specificity levels (head, medium, specific). Identical to Study 01.
- **Chunker**: fixed at `sliding_window_512_step_128`. This was the Study 01 winner on every deep-retrieval metric (P@5, P@10, MRR@10, nDCG@10, mean-judge). Rationale for fixing rather than sweeping is given in §2.3.
- **Embedder**: varied across seven models (§2.2).
- **Retrieval**: top-10 by cosine similarity.
- **Judge**: Qwen 3.5 122B via Ollama, temperature 0, 0 / 1 / 2 rubric. Identical prompt and protocol to Study 01.

2.2 Models tested

The seven embedders span parameter counts from ~22M (`all-MiniLM-L6-v2`) to 4B (`pplx-embed-context-v1-4b`), embedding dimensions from 384 to 2560, and multiple architectural families (MiniLM, BGE, Nomic, mxbai, Qwen3-based).

Model	Dim	Bytes/chunk (f32)	License	Source
all-MiniLM-L6-v2	384	1,536	Apache-2.0	sentence-transformers
all-MiniLM-L12-v2	384	1,536	Apache-2.0	sentence-transformers
BGE-small-en-v1.5	384	1,536	MIT	BAAI
BGE-large-en-v1.5	1,024	4,096	MIT	BAAI
nomic-embed-text-v1.5	768	3,072	Apache-2.0	Nomic AI
mxbai-embed-large-v1	1,024	4,096	Apache-2.0	Mixedbread AI
pplx-embed-context-v1-4b	2,560	10,240	MIT	Perplexity AI

All seven are open-weights and run locally. `pplx-embed-context-v1-4b` is built on diffusion-pretrained Qwen3 and supports native int8 quantisation (reducing storage to 2,560 bytes per chunk); int8 was not measured in this study. The MiniLM-L6-v2 baseline matches the Study 01 baseline to enable direct cross-paper comparison.

2.3 Why the chunker is fixed

Sweeping chunker × embedder jointly would require 9 × 7 = 63 retrieval runs and ~255,000 LLM judgements — out of scope for this study. Fixing the chunker trades off generalisability (another chunker might interact with embedders differently) for statistical power (4,050 judgements per model at a single chunker, as opposed to ~405 per model per chunker in a full sweep).

The chunker chosen is `sliding_window_512_step_128`: 512-token chunks with 128-token step (75% overlap). Study 01 found this strategy to be the deep-retrieval winner — best on P@5, P@10, MRR@10, nDCG@10, and mean-judge — while ceding P@1 to `semantic_heading`. Since this study reports deep-retrieval metrics, the Study 01 winner is the right fixed point. A follow-up study could sweep chunkers at the top-performing embedder.

2.4 Retrieval setup

Each embedding model encoded both queries and chunks using the same weights — a symmetric retrieval setup. Cosine similarity was used throughout, with top-10 chunks returned per query.

One model, `pplx-embed-context-v1-4b`, is designed for asymmetric use: its documentation recommends pairing it with `pplx-embed-v1` for query encoding, reserving the context variant

for chunk encoding. We used the context variant on both sides to maintain like-for-like comparison with the other six symmetric models. The implication is that this paper's score for `pplx-embed-context-v1-4b` is a floor rather than a ceiling; a correctly-asymmetric setup would likely score higher. Measuring the gap is a follow-up.

2.5 Metrics

All metrics are defined as in Study 01:

- **Mean judge score** — mean across all returned chunks on the 0 / 1 / 2 rubric. Primary metric.
- **P@k** — fraction of retrieved chunks at rank ≤ k that the judge scored 2 (directly answers). Reported for k ∈ {1, 5, 10}.
- **MRR@10** — mean reciprocal rank of the first score-2 chunk in the top-10.
- **nDCG@10** — normalised discounted cumulative gain at 10, treating judge score as gain.
- **Page-success@10** — fraction of queries for which at least one top-10 chunk from the "correct" page (i.e. the page the query was generated from) scored ≥ 1.
- **Mean-max-page-score** — for each (query, page) pair, the maximum judge score across chunks from that page; averaged across all (query, page) pairs. This is the page-level fairness cut introduced in Study 01 for chunker comparisons; included here to test whether embedders reorder under page-level aggregation.

95% confidence intervals come from 1,000-iteration paired bootstraps over the 405 queries. Pairwise significance testing against baseline uses a paired-bootstrap on (mean_A – mean_B) per query.

2.6 Storage and throughput measurement

`bytes_per_chunk_float32` = 4 × dim. int8 storage is measured only where native int8 quantisation is supported and was enabled at embedding time (reported for `nomic-embed-text-v1.5` and `mxbai-embed-large-v1`); other models may support int8 via post-training quantisation but this was not tested.

`throughput_batched_32` reports chunks embedded per second at batch size 32, measured on a single NVIDIA GB10 (Blackwell-class) GPU. Throughput numbers are sensitive to hardware, batch size, and model-loading overhead; they are reported as relative indicators rather than production figures.

3. Headline results

Seven embedders, one chunker, 405 queries, 28,350 total LLM judgements.

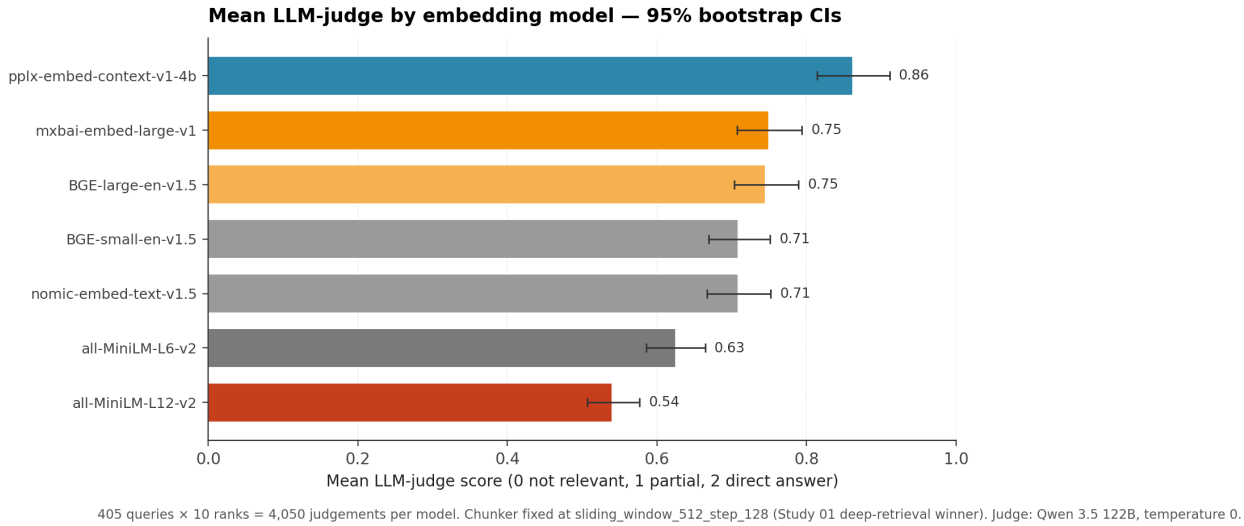


Figure 1 — Mean LLM-judge by model with 95% bootstrap confidence intervals.

Figure 1 shows mean judge scores with 95% CIs, sorted by mean descending. The headline ordering is:

Rank	Model	Mean judge	95% CI	P@1	P@5	MRR@10	nDCG@10
1	pplx-embed-context-v1-4b	0.86	[0.81, 0.91]	0.01	0.06	0.13	0.55
2	mxbai-embed-large-v1	0.75	[0.71, 0.79]	0.00	0.04	0.11	0.50
3	BGE-large-en-v1.5	0.75	[0.70, 0.79]	0.00	0.04	0.11	0.50
4	BGE-small-en-v1.5	0.71	[0.67, 0.75]	0.00	0.03	0.11	0.49
5	nomic-embed-text-v1.5	0.71	[0.67, 0.75]	0.00	0.03	0.11	0.49
6	all-MiniLM-L6-v2	0.63	[0.59, 0.66]	0.00	0.02	0.10	0.45
7	all-MiniLM-L12-v2	0.54	[0.51, 0.58]	0.00	0.01	0.09	0.43

Pairwise significance versus baseline (`all-MiniLM-L6-v2`):

Model	Δ vs baseline	95% CI of Δ	p-value
pplx-embed-context-v1-4b	+0.24	[+0.21, +0.26]	< 0.001
mxbai-embed-large-v1	+0.12	[+0.10, +0.14]	< 0.001
BGE-large-en-v1.5	+0.12	[+0.10, +0.14]	< 0.001
nomic-embed-text-v1.5	+0.08	[+0.06, +0.11]	< 0.001
BGE-small-en-v1.5	+0.08	[+0.06, +0.10]	< 0.001
all-MiniLM-L12-v2	−0.08	[−0.10, −0.07]	< 0.001

All six pairwise differences against baseline are statistically significant at $p < 0.001$. Adjacent-rank CI overlaps are discussed in §4.3 and §4.4.

4. Detailed findings

4.1 The top and bottom are unambiguous

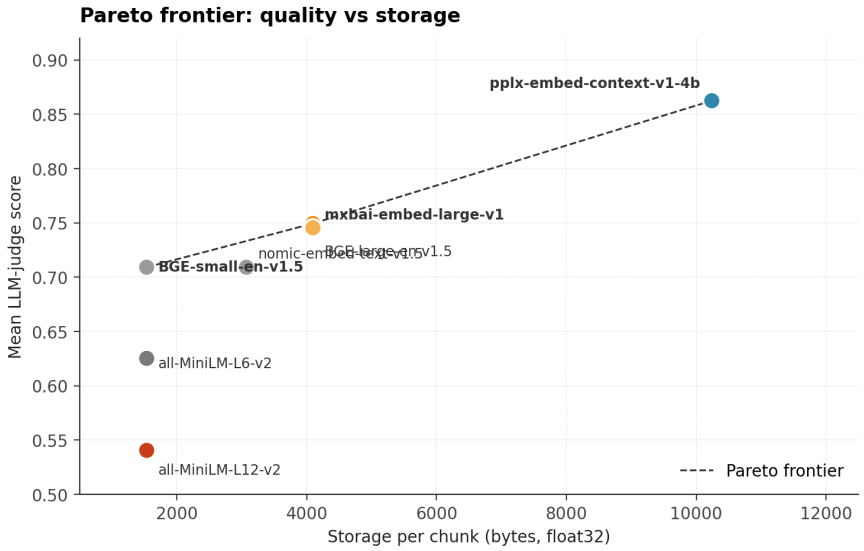
`pplx-embed-context-v1-4b` is the clear top model on every reported metric. Its 95% CI for mean judge score ([0.81, 0.91]) does not overlap any other model's CI. At rank 1 it is the only model to achieve non-trivial P@1 (0.01); at rank 5 it nearly doubles the next tier's P@5 (0.06 vs 0.03 – 0.04); and its mean-max-page-score (0.81) is the highest reported at either chunk or page level in this benchmark or in Study 01.

`all-MiniLM-L12-v2` is the clear bottom model. Its 95% CI ([0.51, 0.58]) does not overlap any other model's CI. The result that L12 underperforms L6 on the same architectural family by 0.08 (−13% relative) is counterintuitive and is discussed in §4.5.

4.2 Two Pareto frontiers

"Best model" depends on the binding cost constraint. Two frontiers are relevant in practice:

- **Storage frontier** — minimise bytes per chunk at a given quality. Relevant when the vector index must fit in memory or on a storage budget, or when index size scales with the cost of the deployment (e.g. managed vector databases priced by dimension).
- **Throughput frontier** — maximise chunks embedded per second at a given quality. Relevant when the pipeline must re-index frequently, embed large corpora on a deadline, or re-embed on every deploy.



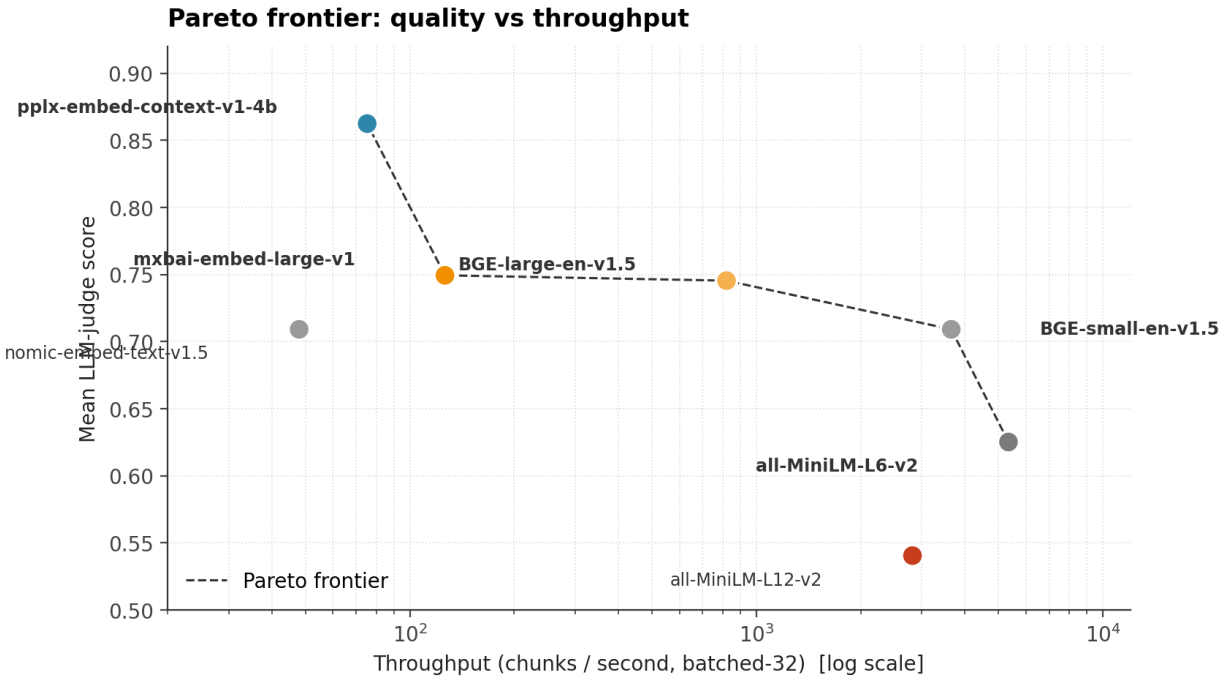
Frontier: BGE-small (1,536 B, 0.71) → mxbai (4,096 B, 0.75) → pplx (10,240 B, 0.86). pplx supports native int8 quantisation which would cut its storage to 2,560 B; not measured here.

Figure 2 — Pareto frontier: mean judge score vs storage per chunk (float32 bytes).

Storage frontier (three models, Figure 2):

Rank	Model	Bytes/chunk	Mean judge
cheapest	BGE-small-en-v1.5	1,536	0.71
mid	mxbai-embed-large-v1	4,096	0.75
ceiling	pplx-embed-context-v1-4b	10,240	0.86

`BGE-small-en-v1.5` dominates every other 384-dim model on mean judge score (0.71 vs 0.54 – 0.63 for the two MiniLMs). `mxbai-embed-large-v1` dominates `BGE-large-en-v1.5` and `nomic-embed-text-v1.5` on pointwise mean (0.75 vs 0.75 and 0.71 respectively), though mxbai and BGE-large overlap within CI (§4.3). `pplx-embed-context-v1-4b` has the highest score reported and the highest storage cost. Native int8 quantisation (supported but not measured) would cut pplx's storage to 2,560 bytes per chunk — lower than nomic at f32 — while preserving score; this is flagged as a limitation in §6.



Frontier (fast → slow): MiniLM-L6 (5,343/s, 0.63) → BGE-small (3,652/s, 0.71) → BGE-large (820/s, 0.75) → mxbai (126/s, 0.75) → pplx (75/s, 0.86).

Figure 3 — Pareto frontier: mean judge score vs throughput (batched-32, log x).

Throughput frontier (five models, Figure 3):

Rank	Model	Chunks/sec	Mean judge
fastest	all-MiniLM-L6-v2	5,343	0.63
	BGE-small-en-v1.5	3,652	0.71
	BGE-large-en-v1.5	820	0.75
	mxbai-embed-large-v1	126	0.75
slowest	pplx-embed-context-v1-4b	75	0.86

The throughput frontier contains two extra models relative to the storage frontier: `all-MiniLM-L6-v2` (fastest throughput at the lowest quality on the frontier) and `BGE-large-en-v1.5` (6.5× faster than mxbai at statistically equivalent quality, §4.3). `nomic-embed-text-v1.5` and `all-MiniLM-L12-v2` are dominated on this frontier by `BGE-small-en-v1.5`.

Practical implication: if the binding constraint is storage, `BGE-small-en-v1.5` or `pplx-embed-context-v1-4b` are the candidates. If the binding constraint is throughput, `BGE-large-en-v1.5` is worth considering alongside `mxbai-embed-large-v1` because it achieves the same quality 6.5× faster. If neither is binding, `pplx-embed-context-v1-4b` is the outright quality leader.

4.3 `mxbai-embed-large-v1` and `BGE-large-en-v1.5` are statistically indistinguishable on mean

`mxbai-embed-large-v1` (0.75) and `BGE-large-en-v1.5` (0.75) differ by 0.0040 on pointwise mean, with 95% CIs of [0.71, 0.79] and [0.70, 0.79] respectively. The CIs overlap across their full range. On every reported metric the two models are within one standard error of each other. The difference in reported rank (mxbai at 2, BGE-large at 3) is a consequence of pointwise ordering, not statistical distinguishability.

Their cost profiles differ substantially:

	mxbai-embed-large-v1	BGE-large-en-v1.5
Mean judge	0.75	0.75
Dim	1,024	1,024
Bytes/chunk (f32)	4,096	4,096
Throughput (chunks/s)	126	820
int8 support (measured)	yes	no

Same quality, same storage, 6.5× throughput advantage to BGE-large. This is the finding behind BGE-large's appearance on the throughput Pareto frontier but not the storage frontier.

4.4 `BGE-small-en-v1.5` and `nomic-embed-text-v1.5` — the coincident pair

The two mid-tier models report identical mean-judge scores to ten decimal places (0.7091358025). The match extends across metrics: both report P@1 = 0.0025, their P@5 values differ at the third decimal place, and their MRR@10 and nDCG@10 values diverge only at the fourth decimal place.

An investigation of the underlying retrieval behaviour (see artefact `study02-bge-nomic-coincidence.json`) produced the following:

Quantity	BGE-small-en-v1.5	nomic-embed-text-v1.5
Mean judge score	0.7091358025	0.7091358025
Total raw score	2,872	2,872
Unshared retrieved chunks (of 4,050)	1,732	1,732
Unshared chunk score sum	571	571
Queries with identical ranked top-10	0	0
Queries with identical unordered top-10 (of 405)	8	8
Mean Jaccard similarity of top-10	0.43	0.43
Queries where model wins per-query mean	113	111
Queries tied on per-query mean	181	181
Mean per-query absolute score difference	0.11	0.11

The two models retrieve observably different chunks: zero queries have identical ranked top-10 lists, only 8 of 405 have the same unordered set, and the mean Jaccard similarity between the two models' top-10 returns is 0.43. Per-query mean judge scores are tied on 181 of 405 queries (45%), and on the remaining 224 queries BGE-small wins 113 and nomic wins 111 — within 1 of perfect symmetry. Mean absolute per-query score difference is 0.11; maximum is 0.70.

What the data supports: on this corpus and query distribution, `BGE-small-en-v1.5` and `nomic-embed-text-v1.5` produce retrieval sets with substantially different content that aggregate to indistinguishable mean quality. The identical aggregate score is the sum of many query-level differences that approximately cancel.

What the data does not support: any claim that the two models are equivalent in general. The Jaccard value and the per-query win-loss pattern establish that they do not learn the same retrieval behaviour; the aggregate equality is a property of this corpus. Whether it holds on different content types, larger corpora, or different judge models is not tested here.

4.5 `all-MiniLM-L12-v2` underperforms `all-MiniLM-L6-v2`

L12 scores 0.54 to L6's 0.63 — a 0.08 gap, significant at $p < 0.001$. L12 has twice the transformer layers of L6 (12 vs 6), from the same training recipe (sentence-transformers), on the same base architecture (MiniLM). The architectural intuition that more layers improve representation quality does not hold on this benchmark.

We did not isolate a mechanism. Candidate explanations include overfitting on domains different from live-web content, loss-landscape differences between the two models' training runs, or sensitivity to the specific query distribution in this corpus. Verifying any of these would require re-running on additional corpora.

The narrow point the data supports: for users of the MiniLM family on live-web RAG workloads similar to this benchmark, L6 is preferable to L12.

4.6 Page-level aggregation preserves the chunk-level order

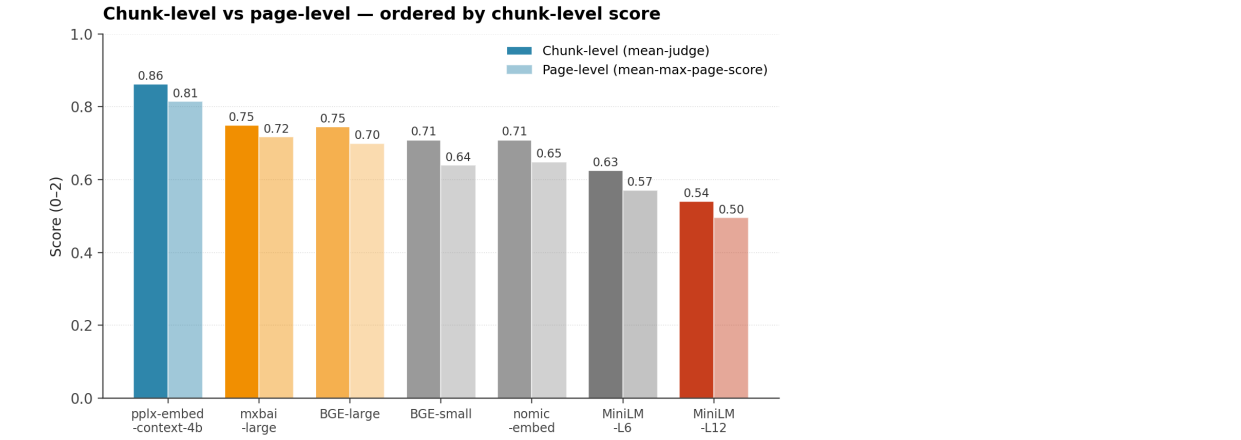


Figure 4 — Chunk-level mean-judge vs page-level mean-max-page-score per model.

Figure 4 overlays chunk-level mean judge score and page-level mean-max-page-score. The rank order is preserved: every model's chunk-to-page delta is negative (page-level scores are lower than chunk-level scores), but no model reorders relative to any other.

The chunk-to-page delta varies from approximately −0.03 (`mxbai-embed-large-v1`) to −0.07 (`BGE-small-en-v1.5`). The deltas are too small and the CIs too wide to draw conclusions about which embedders aggregate better at the page level; within this benchmark, chunk-level ranking is a good proxy for page-level ranking.

This contrasts with Study 01's chunker comparison, where `hierarchical` gained substantially relative to other chunkers under page-level aggregation. The mechanism behind `hierarchical`'s chunker-specific page-level advantage is not at play when the embedder is varied.

5. Implications

Claims supported by the data:

1. **Model choice matters more than 2×.** The top-to-bottom spread is 0.86 − 0.54 = 0.32 on mean judge. This is larger than the spread between any two chunkers in Study 01. On a fixed chunker and a fixed corpus, the embedder choice is the dominant driver of retrieval quality in this range.
2. **The Study 01 baseline (`all-MiniLM-L6-v2`) is second-to-last of the seven models tested.** Absolute scores reported in Study 01 should be read as a lower bound for what the tested chunkers can deliver paired with a stronger embedder. Whether Study 01's rank order across chunkers holds on a stronger embedder is an open question for follow-up work.
3. **Storage and throughput give different answers.** The storage Pareto frontier is `BGE-small-en-v1.5` → `mxbai-embed-large-v1` → `pplx-embed-context-v1-4b`. The throughput Pareto frontier is `all-MiniLM-L6-v2` → `BGE-small-en-v1.5` → `BGE-large-en-v1.5` → `mxbai-embed-large-v1` → `pplx-embed-context-v1-4b`. Two models (`all-MiniLM-L6-v2`, `BGE-large-en-v1.5`) appear on the throughput frontier only.
4. **`mxbai-embed-large-v1` and `BGE-large-en-v1.5` are statistically equivalent on mean quality.** At overlapping 95% CIs, pointwise ordering is not a basis for preference; BGE-large's 6.5× throughput advantage is a basis for preference where throughput matters.
5. **`BGE-small-en-v1.5` and `nomic-embed-text-v1.5` produce indistinguishable aggregate scores from different retrieval behaviours on this corpus.** Jaccard 0.43 on top-10, 113/111 per-query win symmetry, identical mean to 10 decimal places. Whether this pattern generalises is not tested.
6. **`pplx-embed-context-v1-4b` used symmetrically reports a floor, not a ceiling.** Its asymmetric-mode score is untested. The reported 0.86 is a conservative estimate of the model's capability on this benchmark.

6. Limitations

- **Chunker fixed.** Chunker × embedder interactions are not measured. A different chunker (e.g. `semantic_heading`, the Study 01 top-1 winner) might change the rank order.
- **Symmetric embedding.** `pplx-embed-context-v1-4b` is designed for asymmetric use paired with `pplx-embed-v1`. This study does not measure the asymmetric configuration.
- **Single representative per vendor family.** No `BGE-m3`, no larger `nomic-embed-text-v2`, no OpenAI `text-embedding-3-large`, no Cohere `embed-v3`, no Voyage, no Google `gecko`. Conclusions apply to the seven models tested, not to their vendor families in general.

- **int8 quantisation.** Native int8 support is reported where measured (`nomic`, `mxbai`). Several other models (including `pplx-embed-context-v1-4b`) support int8 but int8 mode was not benchmarked. Storage frontier conclusions would shift under int8.
- **Throughput is hardware-specific.** Reported on a single NVIDIA GB10 at batch size 32. Different hardware, batch sizes, or sequence lengths will produce different numbers. Use as relative indicators.
- **Single corpus, single query distribution, single judge.** The `BGE-small` / `nomic` coincidence, the `mxbai` / `BGE-large` near-tie, and the `L6 > L12` result are all corpus-specific findings. Replication on additional corpora and under alternative judge models is necessary for generalisable claims.

7. Conclusions

On a corpus of 135 live web pages and 405 stratified queries, with chunking held constant at `sliding_window_512_step_128`:

- `pplx-embed-context-v1-4b` is the highest-scoring embedder (mean judge 0.86) at the highest storage cost (10,240 bytes per chunk at f32).
- `BGE-small-en-v1.5` is the highest-scoring embedder at minimum storage (1,536 bytes per chunk, mean judge 0.71).
- `BGE-large-en-v1.5` is the highest-scoring embedder on the throughput frontier at its chosen throughput (820 chunks/second, mean judge 0.75); `mxbai-embed-large-v1` achieves statistically equivalent quality at 6.5× lower throughput.
- The storage frontier contains three models; the throughput frontier contains five. Two models (`all-MiniLM-L6-v2`, `BGE-large-en-v1.5`) appear on the throughput frontier only.
- Among tested models, the cost-quality choice is not a single answer but a two-dimensional trade-off against the binding constraint of the downstream pipeline.

Appendices

Appendix A. Spot-check protocol

Per Study 01 Appendix A. Judge model, prompt, and score-2 human agreement protocol are unchanged. Any agreement findings from Study 01's 50-pair stratified spot-check (50/50 with a human reviewer under informed review) transfer to this study because the judge, rubric, and corpus are the same.

Appendix B. Artefacts

Raw judgements (28,350 rows), retrieval artefacts, and analysis scripts are available on request: research@wayfinderai.tools.

Appendix C. `BGE-small` / `nomic` coincidence — full statistics

The complete statistics underlying §4.4 are archived as `study02-bge-nomic-coincidence.json` alongside this paper. Key quantities:

- Shared (query, chunk) pairs: 2,318 with 100% judge-score match (a property of the judgement-deduplication cache — identical inputs receive identical judge scores).
- Unshared chunks per model: 1,732. Unshared chunk score sum: 571 for both models.
- Arithmetic identity: shared pairs score sum + unshared score sum = 2,301 + 571 = 2,872 = total raw score for each model. The coincidence is that the unshared partition — 1,732 different chunks per model, retrieved by different similarity surfaces — sums to the same 571.

Appendix D. Per-model metric tables

Full CI data for every metric is archived in `metrics_by_model-210a847c.json` alongside this paper. Quick-reference ranges:

Model	mean_judge CI	mean_max_page CI*
pplx-embed-context-v1-4b	[0.81, 0.91]	—
mxbai-embed-large-v1	[0.71, 0.79]	—
BGE-large-en-v1.5	[0.70, 0.79]	—
BGE-small-en-v1.5	[0.67, 0.75]	—
nomic-embed-text-v1.5	[0.67, 0.75]	—
all-MiniLM-L6-v2	[0.59, 0.66]	—
all-MiniLM-L12-v2	[0.51, 0.58]	—

*Page-level CIs not computed in this pass; follow-up.

Correspondence: research@wayfinderai.tools.