Y-RET: A SEMANTIC INVERTED INDEX FOR QUERY-BOUNDED CONTEXTUAL RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural retrievers typically persist learned corpus state at the granularity of every searchable unit, from one vector per sentence in dense retrieval to many contextual vectors in late-interaction systems. We ask whether contextual state must be stored at the same granularity at which relevance is evaluated. We introduce Y-RET, a semantic inverted index that instead persists reusable lexical-semantic representatives and sparse sentence postings, then materializes full contextual sentence states only for a bounded candidate set. Across six retrieval benchmarks and four embedding backbones, Y-RET matches the corresponding dense retriever in macro effectiveness while reducing persistent retrieval state by 71.7–87.8%. On direct sentence-evidence retrieval, where BM25, dense, and Y-RET candidates are reranked by the same contextual verifier, Y-RET matches dense acquisition on SciFact. At 10.1 million sentence units on Natural Questions, the storage reduction grows to 88.4%, with a measurable effectiveness cost under the current scoring configuration. These results show that reusable semantic state can replace a large fraction of corpus-wide neural persistence while full contextual discrimination is deferred to query time.

1 INTRODUCTION

Neural retrieval has improved by attaching learned representations to increasingly fine-grained corpus objects. Dense bi-encoders store one vector per independently searchable sentence or passage (Reimers & Gurevych, 2019; Karpukhin et al., 2020; Xiong et al., 2021). Late-interaction systems preserve many contextual token vectors for every indexed unit (Khattab & Zaharia, 2020; Santhanam et al., 2022b), while learned sparse systems attach learned lexical weights to indexed texts and execute through inverted structures (Mallia et al., 2021; Formal et al., 2021; 2022). These choices differ in scoring and execution, but share a consequential design decision: learned corpus state is materialized at, or near, the granularity of the searchable object. As retrieval units become finer, this persistence choice can dominate index size even when the underlying corpus repeatedly expresses the same lexical-semantic content.

Many corpus occurrences share lexical-semantic identity while differing in their local context. The word *aspirin*, for example, may appear in thousands of sentences; a sentence-level dense retriever stores its contribution repeatedly inside sentence vectors, and a token-level late-interaction retriever stores occurrence-specific contextual state repeatedly. This motivates a different question: *must contextual neural state be persisted at the same granularity at which contextual relevance is ultimately evaluated?* If reusable semantic identity can be stored globally while occurrence-specific context is reconstructed only for plausible candidates, retrieval may occupy a different quality–storage regime.

Fine granularity is also what downstream use requires. A retrieval-augmented model consumes evidence tokens from a finite context (Guu et al., 2020; Lewis et al., 2020); retrieving a relevant document supplies its irrelevant sentences as well, whereas retrieving the evidence sentence does not. Sentence-level units are therefore the natural target for evidence retrieval, and they are also where per-unit persistence is most costly.

We introduce Y-RET, a semantic inverted index that makes this separation explicit (Figure 1). Offline, Y-RET independently encodes reusable lexical surface states, consolidates nearby states into lexical-semantic representatives, and associates each representative with a sparse posting list of sentence IDs. At query time, exact and semantic lexical evidence retrieve candidate sentences through these postings.

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.

Only the top $C = 200$ candidates are then encoded in full context and reranked with IDF-weighted late interaction. Y-RET therefore stores lexical-semantic identity and sentence incidence globally, but creates full occurrence-specific contextual state only after retrieval has narrowed the search space.

Across six heterogeneous retrieval benchmarks distributed through BEIR (Thakur et al., 2021) and four independently pretrained encoder families (BGE-small, GTE-small, MiniLM, and Contriever), the same Y-RET architecture matches the corresponding dense sentence retriever in six-task macro nDCG@10 while reducing persistent retrieval state by 71.7–87.8%. With BGE-small, Y-RET reaches 0.4182 macro nDCG@10 using 0.755 GB, versus 0.4111 and 3.016 GB for matched dense retrieval. Under the same sentence units, Y-RET also occupies a stronger quality–storage point than the evaluated SPLADE-v3-DistilBERT and ColBERTv2 baselines. A 10.1-million-sentence Natural Questions (Kwiatkowski et al., 2019) experiment increases the storage reduction to 88.4%, with the remaining effectiveness gap concentrated in acquisition and scoring.

These results support a representation-level separation between reusable semantic state that can be indexed globally and occurrence-specific context that can be instantiated only for plausible candidates.

Our contributions are:

- **A representation-persistence formulation.** We introduce a semantic inverted index whose dense neural states are attached to reusable lexical-semantic representatives shared across sentence occurrences.
- **Query-bounded contextual retrieval.** We combine semantic posting-list acquisition with contextual late interaction over a fixed candidate budget, avoiding corpus-wide persistent contextual token state.
- **Quality–storage evidence across paradigms, backbones, and scale.** We evaluate matched dense retrieval across four encoder families, sparse and late-interaction baselines, controlled mechanism ablations, a matched-verifier comparison on direct sentence evidence, and a 10.1M-sentence scale test.

2 Related Work

Lexical and learned sparse retrieval. Classical lexical retrieval separates a vocabulary from corpus-dependent posting lists; BM25 remains the standard probabilistic reference (Robertson & Zaragoza, 2009). Learned sparse systems preserve the inverted-index execution model while replacing hand-designed lexical weights with neural signals. DeepImpact predicts impact scores that can be served through a standard inverted index (Mallia et al., 2021); SPLADE learns sparse vocabulary expansion and weighting (Formal et al., 2021), with later training improvements strengthening effectiveness (Formal et al., 2022); LexMAE introduces a lexicon bottleneck during pretraining (Shen et al., 2023). Y-RET shares the reusable-vocabulary-plus-postings decomposition but stores a dense semantic state for each reusable lexical representative. Sentence identity is carried by integer postings, and contextual occurrence modeling is deferred to query time.

Dense retrieval. Dense bi-encoders map a query and each indexed unit to fixed-dimensional vectors, enabling nearest-neighbor retrieval after corpus-side encoding (Reimers & Gurevych, 2019; Karpukhin et al., 2020). ANCE improves dense retrieval through global hard-negative mining (Xiong et al., 2021), while Contriever shows that useful dense representations can be learned without task-specific relevance labels (Izacard et al., 2022). E5, BGE, and GTE further strengthen general-purpose text embeddings (Wang et al., 2022; Xiao et al., 2023; Li et al., 2023b). The dense interface offers simple online search, but its persistence unit is the complete searchable object: every sentence receives an independent vector. Y-RET instead attaches persistent dense state to reusable lexical representatives and reconstructs sentence specificity through postings plus bounded contextual verification.

Contextual lexical and late-interaction retrieval. ColBERT introduced contextualized token representations and a late-interaction MaxSim operator (Khattab & Zaharia, 2020). ColBERTv2 reduces the space footprint through residual compression (Santhanam et al., 2022b), while PLAID accelerates late-interaction search through centroid interaction and pruning (Santhanam et al., 2022a). COIL combines lexical inverted access with contextualized occurrence vectors (Gao et al., 2021), and

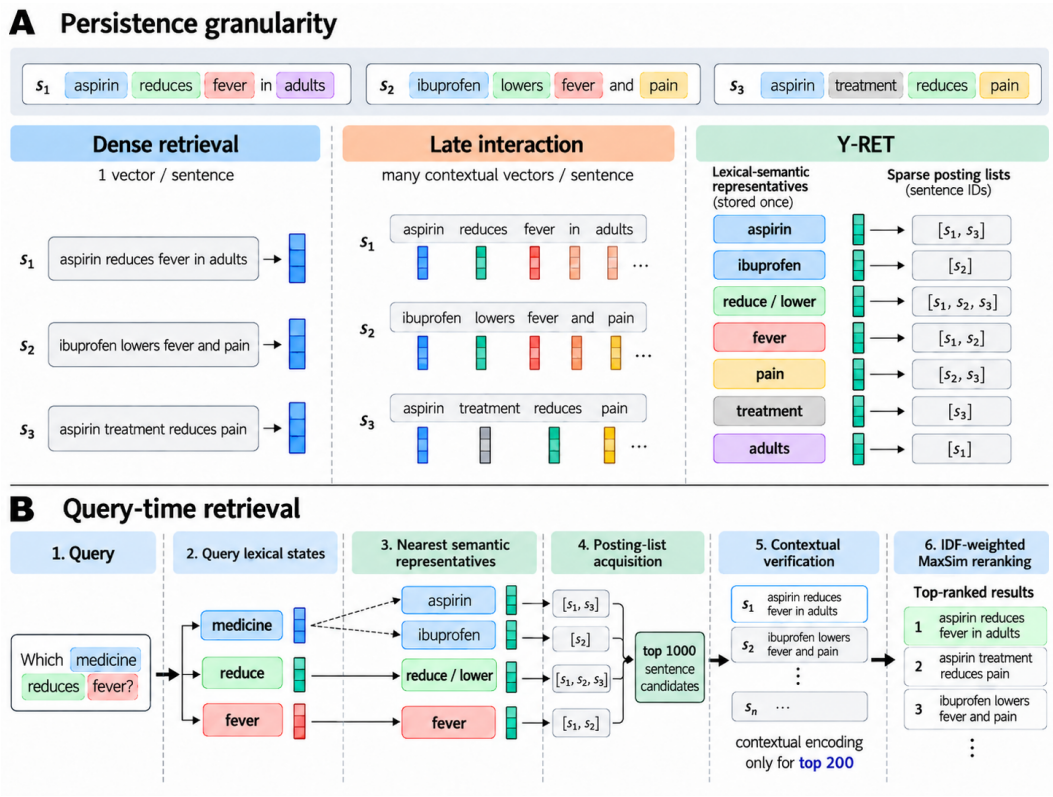


Figure 1: **Persistence granularity and query-time retrieval.** Dense retrieval attaches a learned vector to every sentence, while late interaction persists multiple contextual vectors per sentence. Y-RET instead stores reusable lexical-semantic representatives and sparse sentence postings. At query time, semantic posting-list acquisition produces a candidate set; only the top 200 candidates are contextually encoded and reranked with IDF-weighted MaxSim. Function words are omitted from the reusable semantic representatives in this illustration.

CITADEL learns dynamic lexical routing for multi-vector retrieval (Li et al., 2023a). These systems demonstrate that sparse access and contextual matching can be combined effectively while retaining, compressing, routing, or approximating corpus-side contextual state tied to indexed occurrences. Y-RET places full occurrence-specific contextual vectors outside the persistent retrieval index and applies late interaction after acquisition.

Retrieval granularity and evidence access. DensePhrases makes phrases independently retrievable (Lee et al., 2021), illustrating that the retrievable unit itself is a first-class design choice. Our setting similarly ranks deterministic sentence units because the target is directly usable evidence. SciFact provides a claim-to-evidence setting (Wadden et al., 2020), while Natural Questions supplies naturally occurring information-seeking queries at much larger scale (Kwiatkowski et al., 2019). This evidence-access perspective is directly relevant to retrieval-augmented systems such as REALM and RAG (Guu et al., 2020; Lewis et al., 2020): the retriever determines which evidence becomes available downstream. Y-RET focuses on how this retrieval state is persisted and accessed.

Research gap. Prior work has extensively studied how to compress, sparsify, route, or approximate neural retrieval representations. We study a different boundary: whether occurrence-specific contextual state must be persisted at corpus scale at all. Y-RET stores reusable semantic identity globally, sparse postings preserve sentence incidence, and contextual occurrence state is instantiated only for retrieved candidates.

3 METHOD

3.1 RETRIEVAL SETTING

Let the corpus contain deterministic sentence retrieval units $\mathcal{S} = \{s_1, \ldots, s_N\}$. Each sentence s has a parent identifier $\pi(s)$. The retrieval operation is sentence-level. For benchmarks whose relevance judgments are defined over parent documents, evaluation projects the ranked sentence list to parents by first occurrence: the first retrieved occurrence of a parent determines that parent's rank. These projected metrics evaluate parent retrieval under a shared sentence-to-parent protocol and are distinct from native sentence-level relevance judgments.

3.2 REUSABLE LEXICAL-SEMANTIC STATES

Let w denote a retained lexical surface type, extracted from the corpus by surface-word segmentation with a fixed function-word list removed. The backbone f_θ independently encodes each reusable type,

$$e_w = f_\theta(w), \qquad \|e_w\|_2 = 1. \tag{1}$$

Lexical states are consolidated by a direct-to-representative rule. A state w may join representative r only when

$$e_w^\top e_r \geq \tau_m, \tag{2}$$

with no transitive chaining. Each retained representative owns a sentence posting list

$$P(r) = \{s \in \mathcal{S} : \text{a lexical state mapped to } r \text{ occurs in } s\}. \tag{3}$$

The persistent retrieval state consists of representative vectors, posting offsets and sentence IDs, lexical aliases, sentence-length metadata, and parent/provenance mappings, with full contextual token vectors created only for retrieved candidates.

3.3 QUERY LEXICAL STATES

For each unique query lexical type i, we obtain an isolated vector e_i^{iso} and a full-query conditioned vector e_i^{ctx} using the same backbone. The acquisition vector is

$$q_i = \operatorname{norm}\left(\tfrac{1}{2} e_i^{\text{iso}} + \tfrac{1}{2} e_i^{\text{ctx}}\right). \tag{4}$$

The contextual query pass also provides occurrence-level query states for the final scorer. For BGE-small we use its retrieval instruction; the other backbones use the same structural pipeline without a BGE-specific prefix.

3.4 SEMANTIC POSTING-LIST ACQUISITION

Let a_i be query IDF. Exact lexical evidence is obtained by mapping query type i to its representative r_i: every sentence in $P(r_i)$ receives contribution a_i. For semantic expansion, the top query lexical states by IDF retrieve their K nearest representatives $\mathcal{N}_K(q_i)$. Similarity is transformed by

$$g(u) = \begin{cases} 0, & u < \tau_s, \\ \dfrac{u - \tau_s}{1 - \tau_s}, & u \geq \tau_s. \end{cases} \tag{5}$$

For sentence s, the strongest semantic-neighbor evidence from query lexical unit i is

$$m_i(s) = \max_{\substack{r \in \mathcal{N}_K(q_i),\ r \neq r_i \\ s \in P(r)}} a_i \, g(q_i^\top e_r). \tag{6}$$

with $m_i(s) = 0$ when no such representative exists, including for query units outside the semantic-expansion budget. The pre-normalization evidence score is

$$E(q, s) = \sum_i \left[a_i \, \mathbb{I}\{s \in P(r_i)\} + m_i(s)\right]. \tag{7}$$

With lexical sentence length ℓ_s, average length $\bar{\ell}$, and $b = 0.25$, acquisition uses

$$S_{\text{acq}}(q, s) = \frac{E(q, s)}{(1 - b) + b\,\ell_s/\bar{\ell}}. \tag{8}$$

Representatives whose posting frequency exceeds 5% of sentence units are excluded from semantic traversal. We retain the top 1000 acquisition candidates. The fixed acquisition configuration is $K = 16$, top 48 query lexical states by IDF for semantic expansion, $\tau_s = 0.55$, and lexical merge threshold $\tau_m = 0.85$.

3.5 Query-bounded contextual verification

Only the top $C = 200$ acquisition candidates are contextually encoded. For candidate sentence s, the backbone processes the complete raw sentence. Subword states overlapping each surface word are averaged and unit-normalized to obtain contextual surface-word vectors c_j. Let contextual query occurrence states be q_i^{ctx}. Final candidate relevance is

$$S_{\text{ctx}}(q, s) = \sum_i a_i \max_j \left(q_i^{\text{ctx}}\right)^\top c_j. \tag{9}$$

The top-200 candidates are reranked by S_{ctx}, with acquisition score and sentence ID used only for deterministic tie breaking.

3.6 Persistence complexity

Let N_s be the number of sentence units, N_t the number of indexed token occurrences, R the number of retained lexical representatives, d the embedding dimension, d_c the contextual token dimension, and P the number of posting entries. A sentence-level dense index stores $\Theta(N_s d)$ learned vector state, while contextual multi-vector retrieval stores $\Theta(N_t d_c)$ before compression. Y-RET stores

$$B_{\text{YRET}} = \Theta(Rd + P) \tag{10}$$

plus small offsets and metadata. This shifts dense neural vector state from sentence or contextual-occurrence granularity to reusable lexical representatives, while sparse postings preserve sentence incidence.

4 Experimental Setup

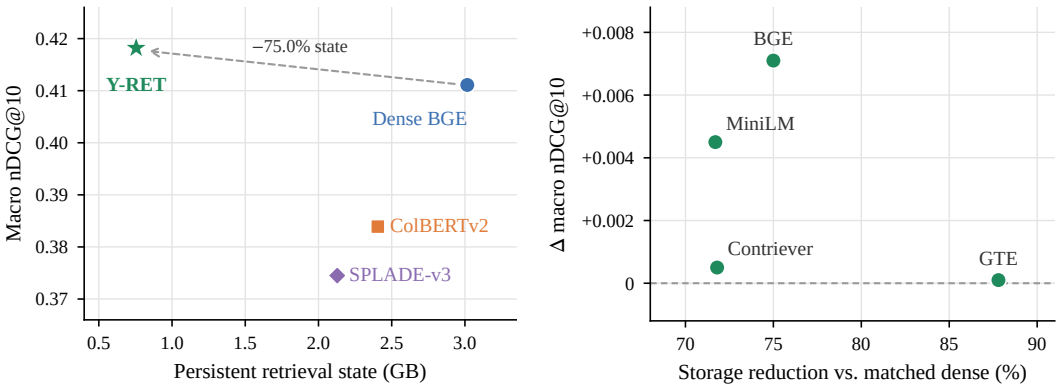
Benchmarks and metrics. The core suite contains ArguAna, SciFact, NFCorpus, SCIDOCS, FiQA, and TREC-COVID, using their BEIR distributions (Thakur et al., 2021). All systems rank the same deterministic sentence units. When qrels are parent-document-level, the same first-occurrence/MaxP parent projection is applied before evaluation. The primary metric is nDCG@10. Natural Questions is reported separately as the large-scale extension.

Backbones and baselines. We evaluate matched dense retrieval and Y-RET using BGE-small-v1.5 (Xiao et al., 2023), GTE-small (Li et al., 2023b), all-MiniLM-L6-v2 (Wang et al., 2020), and Contriever-MS MARCO (Izacard et al., 2022). The BGE cross-paradigm comparison additionally includes BM25 (Robertson & Zaragoza, 2009), BM25+dense reciprocal-rank fusion (Cormack et al., 2009), SPLADE-v3-DistilBERT (Formal et al., 2021; 2022), and ColBERTv2 (Santhanam et al., 2022b). ColBERTv2 uses 128-dimensional token representations and 2-bit residual compression in the evaluated index.

Persistent-state accounting. The primary efficiency quantity is physical persistent retrieval state. For Y-RET this includes representative vectors, postings and offsets, alias mapping, sentence-length metadata, and parent/provenance mappings. Model parameters and raw corpus text are excluded from the neural retrieval index for all methods. For matched dense retrieval we count the stored float32 sentence embedding matrix. For SPLADE and ColBERTv2 we count their physical stored retrieval structures. Because Y-RET fetches candidate text for query-time contextualization, the text store is reported separately in the appendix and text-fetch time is included in end-to-end Y-RET latency. On Natural Questions, the random-access text store is 1.400 GB and is reported separately from the 1.804 GB persistent Y-RET retrieval index.

Table 1: Core six-task effectiveness (nDCG@10). All methods rank the same sentence units and use the same parent projection when required.

	Task nDCG@10						
Method	Arg.	Sci.	NFC	SDoc	FiQA	T-C	Macro
BM25	.3818	.5977	.2885	.1373	.1601	.4013	.3278
Dense BGE	.4033	.6826	**.3589**	.1727	.2291	**.6198**	.4111
BM25 + BGE RRF	.4370	.6820	.3470	.1745	.2175	.5957	.4089
SPLADE-v3	.3457	.6390	.3360	.1349	.2316	.5599	.3745
ColBERTv2	.3525	.6612	.3385	.1400	**.2559**	.5553	.3839
Y-RET (BGE)	**.4502**	**.7014**	.3555	**.1781**	.2397	.5845	**.4182**



(a) Cross-paradigm BGE frontier (b) Matched-backbone frontier

Figure 2: **Quality–storage frontier.** (a) Macro nDCG@10 versus six-task aggregate persistent retrieval state for BGE-based systems. (b) Across four backbones, Y-RET reduces storage by 71.7–87.8% while changing macro nDCG@10 by at most +0.0071 relative to matched dense retrieval.

Execution environment. All experiments were run on an NVIDIA A100. Y-RET uses BF16 execution where supported, SDPA attention, candidate length bucketing, GPU surface-word pooling, and GPU MaxSim. Batch-1 latency is measured after initialization with no cross-query contextual-vector or tokenization cache in the headline timing.

Development disclosure. SciFact, FiQA, and TREC-COVID contributed to architecture validation. The reported matched-backbone study uses one frozen Y-RET architecture.

5 Results

5.1 RQ1: Can reusable semantic state preserve quality while reducing persistent state?

Y-RET preserves strong retrieval effectiveness while occupying a substantially smaller persistent-state footprint. With BGE-small, Y-RET obtains 0.4182 macro nDCG@10, compared with 0.4111 for matched dense retrieval (Table 1). Task-level effects vary: Y-RET improves strongly on ArguAna and SciFact, is close on NFCorpus and SCIDOCS, improves modestly on FiQA, and trails dense BGE on TREC-COVID. Across the suite, aggregate effectiveness is preserved under a substantially smaller persistence architecture.

The storage difference is large. Across the six BGE datasets, the Y-RET indexes occupy 0.755 GB versus 3.016 GB for matched dense BGE, a 75.0% reduction. The evaluated ColBERTv2 index occupies 2.405 GB and SPLADE-v3-DistilBERT 2.128 GB; Y-RET is therefore 68.6% smaller than ColBERTv2 and 64.5% smaller than SPLADE under the same sentence units. Figure 2 places these systems on the resulting quality–storage frontier.

Table 2: Matched-backbone robustness. Storage is the aggregate persistent state over the same six sentence-retrieval tasks.

	Macro nDCG@10			Persistent state	
Backbone	Y-RET	Dense	Δ	Y-RET / Dense	Reduction
BGE-small	.4182	.4111	+.0071	.755 / 3.016 GB	75.0%
GTE-small	.4142	.4141	+.0001	.369 / 3.016 GB	87.8%
MiniLM-L6-v2	.3992	.3947	+.0045	.854 / 3.016 GB	71.7%
Contriever-MS	.3745	.3741	+.0005	1.703 / 6.033 GB	71.8%

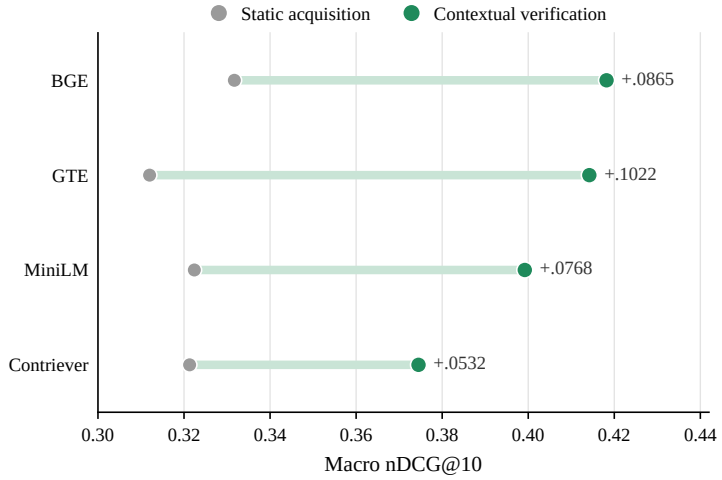


Figure 3: **Effect of contextual verification.** Across all four backbones, contextual IDF-weighted MaxSim over the fixed top-200 candidate set substantially improves macro effectiveness relative to static acquisition.

5.2 RQ2: Does the result depend on one embedding geometry?

The result survives substantially different embedding geometries (Table 2). Across all four backbones, Y-RET matches the corresponding dense retriever in six-task macro nDCG@10, while persistent state is 71.7–87.8% lower. The near-zero GTE and Contriever deltas show that the dense operating point is preserved while the storage coordinate shifts by factors of 3.5–8.2×.

Task-level differences remain heterogeneous. For BGE, Y-RET wins four of six datasets; for GTE, four; MiniLM, three; Contriever, two. The direction of individual deltas also changes by backbone. Stable macro performance across these varying task-level patterns supports the robustness of reusable semantic state across encoder families. Full per-dataset matrices are in Appendix A.

5.3 RQ3: Which components produce the quality–storage trade-off?

The semantic index provides candidate acquisition, while contextual verification supplies the final fine-grained relevance discrimination. Across the four backbones, contextual verification consistently raises effectiveness: BGE improves from 0.3317 to 0.4182 macro nDCG@10, GTE from 0.3120 to 0.4142, MiniLM from 0.3224 to 0.3992, and Contriever from 0.3213 to 0.3745 (Figure 3).

Two controlled BGE ablations isolate the other mechanisms (Table 3). First, disabling semantic-neighbor expansion while retaining exact lexical routing and the identical top-200 contextual verifier reduces macro nDCG@10 from 0.4182 to 0.4083. The drop appears on five of six datasets and is largest on NFCorpus and TREC-COVID, showing that semantic posting traversal contributes retrieval evidence beyond lexical identity. Second, disabling lexical-state consolidation changes macro nDCG@10 only from 0.4182 to 0.4175, but increases the measured persistent lexical-index

Table 3: Mechanism ablations with BGE-small. The exact-only and unmerged variants retain the same top-200 contextual verifier. Storage for the merged/unmerged comparison uses byte-exact ablation accounting.

Variant	Macro nDCG@10	Δ vs full	Persistent state
Static acquisition only	.3317	-.0865	.755 GB
Exact-only acquisition + context	.4083	-.0099	.753 GB
Unmerged lexical states + context	.4175	-.0008	1.084 GB
Full Y-RET	**.4182**	–	.755 GB

Table 4: Natural Questions scale experiment. Persistent-state accounting uses the same sentence retrieval units.

Quantity	Y-RET	Dense BGE
Sentence units	10,135,155	10,135,155
nDCG@10	.3702	.4231
Persistent retrieval state	1.804 GB	15.568 GB
Storage reduction	88.4%	–

state from approximately 0.753 GB to 1.084 GB. Consolidation therefore removes about 30.5% of that index state with essentially unchanged aggregate effectiveness.

Together, the ablations map directly onto the architecture: semantic routing contributes effectiveness, lexical consolidation removes redundant persistent state, and bounded contextualization supplies fine-grained relevance over the retrieved candidate set.

5.4 RQ4: What happens at larger scale?

Natural Questions stresses the same representation at 2,681,468 parent documents, 10,135,155 sentence units, and 3,452 queries. Y-RET requires 1.804 GB of persistent retrieval state, compared with 15.568 GB for the matched float32 dense BGE sentence matrix: an 88.4% reduction, or 8.63× less persistent state. The measured storage reduction therefore increases at this larger scale.

The quality result exposes the current boundary. Y-RET obtains 0.3702 nDCG@10 versus 0.4231 for matched dense BGE. Static acquisition alone reaches 0.2168, and contextual verification raises it by +0.1534. Relevant-parent exposure is 0.707 at the top 200 and 0.839 at the top 1000, localizing much of the remaining gap to acquisition and candidate ordering. These query-time scoring components can be optimized while preserving the stored lexical representatives and postings.

Latency. The six-task mean batch-1 p50/p95 latency is 184/315 ms for BGE, 177/228 ms for GTE, 146/192 ms for MiniLM, and 167/219 ms for Contriever. On Natural Questions, BGE Y-RET reaches 380/817 ms. These end-to-end measurements include candidate handling and contextual verification and use no cross-query contextual caching. They characterize the evaluated A100 implementation and provide a baseline for further scoring and systems optimization.

5.5 RQ5: Does the result hold on direct sentence evidence under a shared verifier?

Table 1 compares Y-RET, whose final ranking includes contextual verification, with first-stage baselines that do not. To isolate the persistent index, we apply the identical top-200 verifier of Section 3.5 to BM25 and dense BGE candidates and evaluate direct sentence evidence rather than projected parents. On SciFact, each retrieval unit is the article title followed by one sentence (45,952 units), and the 188 development claims with annotated rationales define the gold sentences (Appendix D). FullRationale@k requires every sentence of at least one gold rationale set to appear in the top k.

With the verifier held fixed, Y-RET and dense acquisition produce closely matched evidence rankings (Table 5). Both reach 0.8032 FullRationale@5 and 0.8777 FullRationale@10, and Recall@5 is

Table 5: Direct sentence-evidence retrieval on SciFact (188 claims). "+ verifier" applies the identical top-200 contextual verifier of Section 3.5 to each system's candidates. Y-RET acquisition is the static stage alone.

		Recall			FullRationale	
System	MRR@10	@5	@10	@20	@5	@10
BM25	.5703	.5538	.6824	.7645	.6915	.7819
BM25 + verifier	.6376	.6944	.8195	.8635	.7872	.8670
Dense BGE	.6311	.6755	.8095	.8797	.7819	.8564
Dense BGE + verifier	.6397	.7026	.8305	**.8809**	**.8032**	**.8777**
Y-RET acquisition	.5788	.5919	.7367	.8009	.7074	.8032
Y-RET	**.6424**	**.7030**	**.8314**	.8706	**.8032**	**.8777**

0.7030 versus 0.7026. Paired bootstrap intervals over claims give Y-RET a small MRR@10 advantage (+0.0027, 95% CI [+0.0005, +0.0055]) and dense a small Recall@20 advantage (-0.0103, [-0.0210, -0.0022]); the other differences have intervals containing zero. The Recall@20 difference follows candidate exposure: dense places 98.1% of gold sentences within its top 200 candidates, and Y-RET 93.3%. With the same verifier, Y-RET exceeds BM25 on every reported metric, so its acquisition contributes beyond lexical matching.

6 Limitations

The clearest measured limitation is large-scale scoring. On Natural Questions, the persistent representation remains 8.63× smaller than matched dense BGE, while the current acquisition and ranking configuration trails matched-dense effectiveness. Exposure diagnostics localize a meaningful part of the gap before contextual verification. Both scoring and query-time execution remain optimization opportunities, and improvements can retain the same persistent representation.

7 Conclusion

Y-RET reframes neural retrieval around a persistence boundary. It stores persistent learned state as reusable lexical-semantic representatives with sparse sentence incidence, then materializes full contextual state only for a bounded candidate set. Across four encoder families, this preserves matched-dense aggregate retrieval quality while reducing persistent state by 71.7–87.8%; controlled ablations show that semantic routing, lexical-state reuse, and query-bounded contextual verification each contribute to the resulting operating point. Under an identical contextual verifier, Y-RET matches dense acquisition on direct sentence evidence. At 10.1 million sentence units, the storage reduction reaches 88.4%, while the current scorer exposes a measurable large-scale quality gap. These results identify persistence granularity as a first-class retrieval design choice: reusable semantic state can carry corpus-wide identity, while full context can be deferred to bounded query-time computation.

References

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, 2009.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2288–2292, 2021. doi: 10.1145/3404835.3463098.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. From distillation to hard negative sampling: Making sparse neural ir models more effective. *arXiv preprint arXiv:2205.04733*, 2022.

Luyu Gao, Zhuyun Dai, and Jamie Callan. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3030–3042, 2021. doi: 10.18653/v1/2021.naacl-main.241.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 3929–3938, 2020.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781, 2020. doi: 10.18653/v1/2020.emnlp-main.550.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 39–48, 2020.

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 Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. Learning dense representations of phrases at scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 6634–6647, 2021. doi: 10.18653/v1/2021.acl-long.518.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Minghan Li, Sheng-Chieh Lin, Barlas Oguz, Asish Ghoshal, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. Citadel: Conditional token interaction via dynamic lexical routing for efficient and effective multi-vector retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 11891–11907, 2023a. doi: 10.18653/v1/2023.acl-long.663.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023b.

Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. Learning passage impacts for inverted indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1723–1727, 2021. doi: 10.1145/3404835.3463030.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3982–3992, 2019. doi: 10.18653/v1/D19-1410.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. Plaid: An efficient engine for late interaction retrieval. *arXiv preprint arXiv:2205.09707*, 2022a.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3715–3734, 2022b. doi: 10.18653/v1/2022.naacl-main.272.

Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Xiaolong Huang, Binxing Jiao, Linjun Yang, and Daxin Jiang. Lexmae: Lexicon-bottlenecked pretraining for large-scale retrieval. *arXiv preprint arXiv:2208.14754*, 2023.

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

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*, 2023.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021.

A FULL MATCHED-BACKBONE RESULTS

Table 6: Per-dataset difference $\Delta = \mathrm{nDCG}_{\text{Y-RET}}@10 - \mathrm{nDCG}_{\text{dense}}@10$.

Dataset	BGE	GTE	MiniLM	Contriever
ArguAna	+.0469	+.0223	-.0029	+.0213
SciFact	+.0188	+.0136	+.0131	-.0182
NFCorpus	-.0034	+.0085	-.0039	-.0069
SCIDOCS	+.0054	-.0054	-.0222	-.0084
FiQA	+.0106	+.0008	+.0022	-.0128
TREC-COVID	-.0353	-.0390	+.0403	+.0278

B BGE EFFECTIVENESS AND STORAGE BY DATASET

Table 7: BGE-small Y-RET versus matched dense retrieval. Storage figures are physical persistent retrieval state.

Dataset	Y-RET nDCG@10	Dense nDCG@10	Y-RET state	Dense state
ArguAna	.4502	.4033	47.4 MB	99.7 MB
SciFact	.7014	.6826	52.5 MB	74.9 MB
NFCorpus	.3555	.3589	38.9 MB	59.0 MB
SCIDOCS	.1781	.1727	126.8 MB	309.6 MB
FiQA	.2397	.2291	127.1 MB	591.5 MB
TREC-COVID	.5845	.6198	362.3 MB	1881.7 MB
Aggregate / macro	.4182	.4111	0.755 GB	3.016 GB

C ABLATION DETAILS

Table 8: Per-dataset nDCG@10 for the BGE mechanism ablations.

Variant	Arg.	Sci.	NFC	SDoc	FiQA	T-C	Macro
Exact-only + context	.4505	.6918	.3344	.1713	.2339	.5680	.4083
Unmerged + context	.4530	.6938	.3550	.1763	.2386	.5882	.4175
Full Y-RET	.4502	.7014	.3555	.1781	.2397	.5845	.4182

D DIRECT SENTENCE-EVIDENCE PROTOCOL

SciFact. We use the public development split, whose rationale labels are released. Of its claims, 188 carry SUPPORT or CONTRADICT rationales, giving 338 gold rationale sets and 366 gold sentence assignments. Each of the 45,952 retrieval units is the article title followed by one sentence, identically for every system; gold rationale sentences are aligned to these units. The lexical index contains 46,687 lexical types, merged into 31,747 representatives, with 1,037,682 representative–sentence incidences (22.58 per sentence). Recall@k is the fraction of a claim's gold sentences in the top k; FullRationale@k is 1 if every sentence of at least one gold rationale set is in the top k.

E ALGORITHMIC SUMMARY

Offline index construction. (1) Segment the corpus into deterministic sentence units and record each parent ID. (2) Extract lexical surface types. (3) Encode each unique type independently with the chosen backbone and L2-normalize. (4) Consolidate lexical states by direct comparison to retained representatives at threshold $\tau_m = 0.85$, without transitive chaining. (5) Build representative-to-sentence posting lists, alias mappings, sentence lengths, and parent mappings.

Query-time retrieval. (1) Extract query lexical states and form the 50/50 isolated/contextual acquisition vectors. (2) Add exact posting evidence. (3) For the 48 highest-IDF query lexical states, traverse up to $K = 16$ nearest representatives satisfying the semantic gate and frequency guard. (4) Rank static candidates and retain the top 1000. (5) Fetch raw text for the top 200, contextually encode these sentences, pool subwords to surface words, and compute IDF-weighted MAXSIM. (6) Rerank the candidate set and project sentences to parent IDs for document-level qrels.

F STORAGE ACCOUNTING NOTES

The main paper reports persistent retrieval structures separately from model parameters and shared raw corpus text. For Y-RET, the counted structures are representative vectors, posting offsets, posting sentence IDs, lexical aliases, sentence-length state, and parent/provenance metadata. For dense retrieval, the counted structure is the float32 sentence embedding matrix. ColBERTv2 and SPLADE use the physical retrieval-index files produced by their evaluated implementations. Raw candidate text is required by Y-RET at query time and is therefore reported separately; on Natural Questions it occupies 1.400 GB in the random-access store and candidate-text fetch is included in end-to-end latency.

G LATENCY DETAILS

All experiments were run on an NVIDIA A100 40 GB. Y-RET headline timing is batch-1 after model/index initialization, with no cross-query contextual-vector or tokenization cache. BGE p50/p95 ranges from 127/158 ms on NFCorpus to 213/720 ms on TREC-COVID, with ArguAna at 311/453 ms. The six-task mean p50/p95 is 184/315 ms for BGE, 177/228 ms for GTE, 146/192 ms for MiniLM, and 167/219 ms for Contriever. Natural Questions BGE timing is 380.2/817.4 ms.

H IMPLEMENTATION AND REPRODUCIBILITY DETAILS

Sentence and lexical preprocessing. Sentence units are generated with PySBD for English, followed by a deterministic conservative fallback that re-splits only abnormally long units (at least 120 whitespace-delimited tokens) when multiple clear sentence-ending boundaries remain. Lexical surface types are lower-cased and extracted with a fixed surface tokenizer that recognizes alphabetic words, internal hyphen/apostrophe compounds, alphanumeric tokens, and formatted numbers; commas inside numeric tokens are removed during normalization. The fixed omitted function-word set is {`a`, `an`, `the`, `this`, `that`, `these`, `those`, `and`, `or`, `if`, `whether`, `because`, `although`, `though`, `while`, `whereas`, `which`, `who`, `whom`, `whose`, `of`, `in`, `for`, `with`, `by`, `on`, `at`, `to`, `from`, `into`, `through`, `over`, `under`, `between`, `among`, `during`, `before`, `after`, `within`, `upon`, `via`, `be`, `am`, `is`, `are`, `was`, `were`, `been`, `being`}.

IDF and deterministic consolidation. All corpus frequencies are measured over sentence units. For a lexical surface type w,

$$\mathrm{idf}(w) = \log \frac{N+1}{\mathrm{df}(w)+1} + 1. \tag{11}$$

An out-of-vocabulary query type therefore receives the maximum smoothed IDF but has no exact posting list. The global lexical vocabulary is sorted lexicographically before encoding. Consolidation visits lexical states in descending sentence document frequency, breaking frequency ties by this stable vocabulary order. The first unassigned state becomes the representative; another unassigned state joins it only when its cosine similarity to that representative itself is at least 0.85. This is the direct-star rule of Section 3.2 and does not use transitive chaining. Query lexical types are de-duplicated in first-occurrence order; the top-48 semantic anchors are selected by descending IDF, with this query order as the deterministic tie break.

Encoder instantiation and query formatting. The exact encoder identifiers are `BAAI/bge-small-en-v1.5`, `thenlper/gte-small`, `sentence-transformers/all-MiniLM-L6-v2`, and `facebook/contriever-msmarco`. BGE queries use the literal prefix `Represent this sentence for searching relevant passages: `; the

other three backbones use no query prefix. An isolated lexical type is encoded by averaging its non-special final-layer subword states and L2-normalizing the result; isolated lexical inputs are truncated at 64 subword tokens. Contextual query/candidate inputs are truncated at 256 subword tokens. For each retained surface word, final-layer states of all overlapping subwords are averaged and L2-normalized before the IDF-weighted MAXSIM score. The paper-run implementation uses batches of 256 isolated lexical types and 32 contextual sentences; these batching choices affect throughput but not the scoring definition.

Baseline settings. Matched dense retrieval uses the same four checkpoint identifiers above, L2-normalized float32 sentence embeddings, and exact inner-product search. Dense encoding batches are 256 for BGE and GTE, 512 for MiniLM, and 128 for Contriever. BM25 uses `bm25s==0.3.11` with Lucene scoring ($k_1 = 1.5$, $b = 0.75$), English stemming via PyStemmer, and the package's English stopword list. BM25+dense reciprocal-rank fusion uses $k = 60$. The evaluated specialist checkpoints are `naver/splade-v3` and `colbert-ir/colbertv2.0`; the ColBERT implementation uses `colbert-ai==0.2.22`. All first-stage systems retain at most 1000 sentence candidates before any parent projection, and the matched-verifier SciFact comparison reranks only each system's top 200 candidates with the identical verifier described in Section 3.5.

Randomness and confidence intervals. The broad paper-run seed is 20260920. The direct SciFact sentence-evidence experiment uses seed 20260926. Its paired confidence intervals use 10,000 bootstrap resamples over the 188 claims, sampling claims with replacement and reporting the 2.5th and 97.5th percentiles of the paired mean difference. All reported GPU experiments and latency measurements use a 40GB NVIDIA A100.

I USE OF LARGE LANGUAGE MODELS

We used large language models (LLMs) in three roles: to aid and polish the writing of this paper; for retrieval and discovery, such as finding related work; and to assist with debugging experimental code. The authors reviewed all LLM-assisted outputs and take full responsibility for the content of this paper.