LACONIC: Dense-Level Effectiveness for Scalable Sparse Retrieval via a Two-Phase Training Curriculum

Zhichao Xu
zhichao.xu@utah.edu
University of Utah
Salt Lake City, UT, USA

Shengyao Zhuang
s.zhuang@uq.edu.au
The University of Queensland
Brisbane, QLD, Australia

Crystina Zhang
x978zhan@uwaterloo.ca
University of Waterloo
Waterloo, ON, Canada

Xueguang Ma
x93ma@uwaterloo.ca
University of Waterloo
Waterloo, ON, Canada

Yijun Tian
meetyijun@gmail.com
University of Notre Dame
Notre Dame, IN, USA

Maitrey Mehta
maitrey@cs.utah.edu
University of Utah
Salt Lake City, UT, USA

Jimmy Lin
jimmylin@uwaterloo.ca
University of Waterloo
Waterloo, ON, Canada

Vivek Srikumar
svivek@cs.utah.edu
University of Utah
Salt Lake City, UT, USA

Abstract

While dense retrieval models have been the standard for state-of-the-art information retrieval, their deployment is often constrained by high memory requirements and reliance on GPU accelerators for vector similarity search at scale. Learned sparse retrieval offers a compelling alternative by enabling efficient search via inverted indices, yet it has historically received less attention than dense approaches. In this paper, we introduce `LACONIC`, a family of learned sparse retrievers based on the `Llama3` architecture (1B, 3B, and 8B). We propose a streamlined two-phase training curriculum consisting of (1) weakly supervised pre-finetuning to adapt causal LLMs for bidirectional contextualization and (2) high-signal finetuning using curated hard negatives. Our results demonstrate that `LACONIC` effectively bridges the performance gap with dense models: the 8B variant achieves a state-of-the-art 60.2 nDCG@10 on the MTEB Retrieval benchmark, ranking 15th on the leaderboard as of February 5th, 2026, while utilizing 74% less index memory than an equivalent dense model. By delivering high retrieval effectiveness on commodity CPU hardware with a fraction of the compute budget required by competing models, `LACONIC` provides a scalable and efficient solution for real-world search applications. We fully open source our code implementation and trained checkpoints to facilitate reproducibility.

CCS Concepts

• **Information systems** → **Retrieval models and ranking**.

Keywords

Learned sparse retrieval; document representation.

SIGIR '26, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3809869

ACM Reference Format:
Zhichao Xu, Shengyao Zhuang, Crystina Zhang, Xueguang Ma, Yijun Tian, Maitrey Mehta, Jimmy Lin, and Vivek Srikumar. 2026. LACONIC: Dense-Level Effectiveness for Scalable Sparse Retrieval via a Two-Phase Training Curriculum. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3805712.3809869

Code `laconic-sparse-retrieval`
Data `nomic-embed-pretrain-lite`
Models `LACONIC-1B` `LACONIC-3B` `LACONIC-8B`

1 Introduction

Information retrieval (IR) has undergone a paradigm shift from traditional term-matching methods like BM25 [37] to neural dense retrieval models [22, 25, 27, 44]. Dense retrievers excel at capturing semantic nuances by encoding queries and documents into continuous high-dimensional vectors. However, they often suffer from significant deployment overhead, requiring large memory footprints to store dense embeddings and specialized hardware (e.g., GPUs) for efficient vector similarity search.

Learned sparse retrieval, pioneered by earlier works such as `SNRM` [46], `DeepCT` [5], `SparTerm` [1], `SPARTA` [48] and popularized by the `SPLADE` framework [9, 10, 12, 23, 24, 28], offers a compelling middle ground. By projecting hidden states onto the vocabulary space and applying sparsity-inducing regularizations, these models produce high-dimensional but sparse representations. This allows for the use of efficient, CPU-friendly inverted index structures while maintaining the semantic richness of neural encoders. Despite their potential, a performance gap has historically persisted between sparse models and state-of-the-art dense retrievers, particularly as the latter have scaled to large language model (LLM) backbones [50].

In this paper, we introduce `LACONIC`, a series of learned sparse retrieval models based on the `Llama3` family [16] in 1B, 3B, and 8B parameter scales. We name our model `LACONIC` as a tribute to the historical tradition of *laconic* speech — the Spartan practice of using

the fewest words possible to deliver the maximum impact. This serves as a technical metaphor for our architecture: we leverage the vast knowledge of autoregressive decoders but constrain them to generate succinct, vocabulary-sparse representations that are "Spartan" in their resource requirements.

We adopt a streamlined two-phase training curriculum — pre-finetuning on weakly-supervised data followed by high-quality hard-negative finetuning — that allows LACONIC to close the performance gap with its dense counterparts. Although such multi-stage curricula have been extensively studied and validated in dense retrieval, we show that this paradigm is equally critical for learned sparse retrievers, where it plays a central role in adapting large causal language models to bidirectional information and relevance modeling. In contrast to SPLADE-v2's curriculum [9], which couples hard-negative mining with cross-encoder distillation atop a BERT backbone, our recipe targets the distinct challenges of adapting large causal LLMs for bidirectional sparse encoding without auxiliary teacher models, complementing recent decoder-based LSR efforts [8, 11, 35, 41, 47]. Our LACONIC-8B model achieves an impressive 60.2 nDCG@10 on the MTEB Retrieval benchmark, ranking 15th on the leaderboard as of February 5th, 2026. Notably, LACONIC achieves these results using a fraction of the compute budget of its competitors while maintaining a significantly smaller index memory footprint. We detail our architecture, training methodology, an extensive evaluation of retrieval efficiency and effectiveness, and fully open source our implementation and trained models.

2 Proposed Approach

The performance of LACONIC stems from the integration of powerful LLM backbones with a streamlined two-phase training curriculum based on contrastive training.

2.1 Model Architecture

LACONIC is a bi-encoder retrieval model [19, 36], which builds upon the SPLADE framework [12, 23] while incorporating architectural best practices for scaling sparse retrievers [8, 35, 41].

Denote query Q and document D, and a language model's vocabulary as $\mathcal{V}$. $D = \{t_1, t_2, \ldots, t_{|D|}\}$ where t_i is the i-th token. The document's corresponding contextualized representation can be written as $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots \boldsymbol{h}_{|D|}\}$. For each $\boldsymbol{h}_i$, we project the hidden representation to a vocabulary-sized vector $\boldsymbol{H}_i \in \mathbb{R}^{|\mathcal{V}|}$ with the language modeling head. The j-th dimension of $\boldsymbol{H}_i$ represents the importance of token j (in vocabulary $\mathcal{V}$) to token i in the input sequence, which in practice is the logit_j from the LM head output. Given $\boldsymbol{H}_D = \{\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots \boldsymbol{H}_{|D|}\}$ of tensor shape $(|\mathcal{V}|, |D|)$, we apply a max-pooling along the sequence length dimension, i.e., across all tokens, followed by ReLU activation and log rescaling to get the vocabulary-sized representation for the input document d:

$$\boldsymbol{D} = \log\Big(1 + \text{ReLU}\big(\text{MaxPooling}(\boldsymbol{H}_D)\big)\Big) \in \mathbb{R}^{|\mathcal{V}|} \tag{1}$$

A similar operation can also be applied to query Q to get query representation $\boldsymbol{Q} \in \mathbb{R}^{|\mathcal{V}|}$.

We adopt Llama3 family models as the backbone, specifically Llama3ForCausalLM. Note that the above Equation (1) applies a pooling along the sequence length dimension, which is disadvantageous for causal language models with unidirectional attention. Prior works explored different mitigations, including using "echo" input [8, 38] and enabling bidirectional attention via lightweight adaptation training [2, 41, 47]; in contrast, we adopt a streamlined approach of directly enabling the bidirectional attention of the causal language models by removing the causal attention mask, and letting the models "self-adapt" in the subsequent contrastive training.

To summarize, LACONIC differs from SPLADE by using a bidirectional variant of the stronger Llama3 backbone language model, which is implementation-wise straightforward and achieves impressive empirical performance with a correct training curriculum.

2.2 Training Objective

We adopt a standard InfoNCE loss [33] in training LACONIC. Denote a training pair (Q, D^+), where D^+ is relevant to query Q, and $\{D_N\}$ is a list of documents not relevant to Q, score function $s(Q, D) = \langle \boldsymbol{Q}, \boldsymbol{D} \rangle$, the ranking loss is formulated as:

$$\begin{aligned}\mathcal{L}_{rank}(Q, D^+, \{D_N\}) &= -\log p(D = D^+ | Q) \\ &= -\log \frac{e^{s(Q,D^+)}}{e^{s(Q,D^+)} + \sum_{D_i^- \in \{D_N\}} e^{s(Q,D_i^-)}}\end{aligned} \tag{2}$$

Practically we use in-batch negatives and/or hard negatives in different phases of training, following prior practices [32, 45]. To enforce the sparsity of the encoded sparse representations, we adopt FLOPs regularization [34], the same as SPLADE. Denote FLOPs regularization loss for Q and D as $\mathcal{L}_{reg}^Q$ and $\mathcal{L}_{reg}^D$, respectively, and λ_Q, λ_D as the corresponding coefficients, the final loss is:

$$\mathcal{L} = \mathcal{L}_{rank}(Q, D^+, \{D_N\}) + \lambda_Q \mathcal{L}_{reg}^Q + \lambda_D \mathcal{L}_{reg}^D$$

where λ_Q and λ_D are tuned as hyperparameters.

2.3 Pre-finetuning

The goal of this training phase is to adapt the backbone language model for bidirectional attention (Section 2.1) while training it to encode sparse representations to model query and document relevance using large-scale, weakly-supervised data.

Dataset. We follow prior recipes [17, 32, 45] to use weakly-supervised contrastive pairs, i.e., (Q, D) pairs curated from noisy data sources. Given the limited compute budget, we use a subset of Nomic Embedding Unsupervised Data released under Apache 2.0 license.[1] As our focus is on asymmetric retrieval tasks, we use 11 splits: wikipedia, gooaq, agnews, ccnews, npr, eli5, cnn, squad, quora, simplewiki, stackexchange_duplicate_questions. We selected these splits via lightweight manual inspection, informed by our compute budget. Our final mixture consists of about 9M pairs, which is a small fraction of the original dataset's 470M pairs or Arctic-Embed-v2's 308M pairs [45]. We refer to this mixture as Nomic-embed-pretrain-lite. We hypothesize that scaling the pre-finetuning data could further improve performance.

[1] https://huggingface.co/datasets/nomic-ai/nomic-embed-unsupervised-data

Training. In this training phase, we use in-batch negatives. Prior works have reported the efficacy of scaling up batch size in contrastive training [14]. Notably, `Nomic-text-v1` [32] used 16,384 global batch size while `Arctic-embed-v2` [45] reported 32,768 global batch size using 32 × H100 GPUs. In our experiments, we use 2,048 global batch size consistently per our compute budget.

We carefully tune the training schedule and hyperparameters. We train for 3, 2, 1 epochs for 1B, 3B, and 8B variants of `LACONIC`, respectively. We use LoRA training [18] and set rank=32 for 1B and 3B models, and rank=16 for the 8B model. We use cosine learning rate scheduling, and adopt a separate exponential warmup for the `FLOPs` regularization loss, same as [9, 12]. We set $\lambda_Q = \lambda_D = 1 \times 10^{-3}$ for {1B, 3B, 8B} models. We truncate the queries to 64 tokens and documents to 192 tokens. After this training phase, we merge the LoRA adapter back to the base model and use the merged checkpoint in the subsequent finetuning phase.

2.4 Finetuning

After the pre-finetuning phase, the model has already learned the sparsity pattern required for sparse retrieval and acquired the basic "capability" of identifying relevance. We then move on to the next phase of finetuning with dedicated hard negatives.

Dataset. We adopt a recently released `RLHN` dataset [39], which consists of 690K $(Q, D^+, \{D_N\})$ triplets.[2] `RLHN` is a lite version of the larger BGE training mixture [26] with further relabeled hard negatives, and has been reported to improve retrieval and ranking performance while reducing training time [39, 43]. More specifically, the BGE mixture's negatives are mined as top-ranked but non-relevant candidates by an ensemble of strong dense and sparse retrievers; `RLHN` then re-judges these candidates with an LLM judge to remove false negatives, yielding cleaner per-query hard negatives than typical mining-only pipelines.

Training. We use hard negatives and in-batch negatives in this training phase. Specifically, each query is paired with 1 relevant document and 15 hard negatives, together with all other documents in this batch. We use a consistent 32 global batch size, which implies 512 negatives per query. Similar to the pre-finetuning phase, we finetune for 2, 2, 1 epochs for 1B, 3B, and 8B variants of `LACONIC`. Again, we use LoRA training, set rank=32 for 1B and 3B models and rank=16 for the 8B model, together with cosine learning rate scheduling, a separate exponential warmup for `FLOPs` regularization loss and use a consistent $\lambda_Q = \lambda_D = 1 \times 10^{-3}$. We truncate both queries and documents to 192 tokens.

3 Experiments

We describe the experimental setup (Section 3.1) and discuss results and analysis (Section 3.2).

3.1 Experimental Setup

Implementation Details. We initialize `LACONIC` with open-weight `Llama3.2-1B`,[3] `Llama3.2-3B`,[4] `Llama3.1-8B`[5] models, all licensed for academic use. Note that we use the Base models (without post-training). We implement `LACONIC` in PyTorch on top of the `Tevatron` framework [13]. To improve training scalability, we use gradient checkpointing, gradient accumulation, mixed-precision training (BF16), Flash Attention 2 [6], and PyTorch FSDP [49]. Our compute infrastructure is based on a cluster of A100 SXM4 40GB GPUs with NVSwitch inter-GPU connectivity.

Baselines. For dense retrieval models, we include open-weight models: `Nomic-Embed-v1` [32][6], `Arctic-Embed-v2` [45].[7] We also implement a dense baseline following `RepLlama`'s training recipe with the `Llama3.1-8B` backbone and `RLHN` dataset, which we term `RepLlama3`. Note that `RepLlama3` did not undergo the same pre-finetuning training phase as `LACONIC`. For sparse retrieval models, we include the performant `SPLADE-v3` [24] and `CSPLADE` [41].

Indexing and Evaluation. We use Seismic – an efficient inverted index structure [3, 4] that enables accurate and fast approximate nearest neighbor search without GPU accelerators. We also compare retrieval latency against dense baselines using a Faiss FlatIP dense index [21].

We use nDCG@10 to evaluate `LACONIC`'s retrieval performance on the 15 retrieval tasks of the MTEB benchmark [29], which we refer to as MTEB-R.

3.2 Results and Analysis

Retrieval Performance. We report the retrieval performance in Table 1. The lite `LACONIC-1B` significantly outperforms the state-of-the-art sparse retrieval models `SPLADE` and `CSPLADE`, averaging 57.6 nDCG@10 on 14 BEIR datasets versus `SPLADE`'s 51.3 and `CSPLADE`'s 54.6. `LACONIC-1B` also outperforms the competitive lightweight `Nomic-text-v1` and `Arctic-embed-v2`, despite being trained on only a fraction of pre-finetuning and finetuning datasets. Our largest model, `LACONIC-8B` achieves an impressive average 60.2 nDCG@10 on 15 MTEB Retrieval datasets, which is ranked 15th on the leaderboard as of February 5th, 2026, being the only learned sparse retrieval model at this position. We note that on the MS MARCO development set specifically, `LACONIC` trails `SPLADE-v3` and `CSPLADE` by approximately 1.5–3.0 nDCG@10. We attribute this to `SPLADE-v3` and `CSPLADE` being trained extensively on MS MARCO supervision with cross-encoder distillation, whereas `LACONIC`'s training mixture targets broad-domain generalization without MS MARCO-specific tuning. The substantial gains on the cross-domain MTEB Retrieval datasets suggest this trade-off favors generalization, while domain-specific finetuning could further improve `LACONIC`'s performance on individual benchmarks.

Index Search Efficiency. We study the index search efficiency of `LACONIC` compared to dense and sparse baselines, with results shown in Figure 1. Unless otherwise stated, we report query-time search latency only, excluding embedding computation and index construction costs. For dense retrieval, we use Faiss GPU index (`GpuIndexFlatIP`) with 8× A100 SXM4 40GB GPUs or CPU-only `IndexFlatIP`, while `LACONIC` uses the Seismic inverted index on CPU without accelerator support. All benchmarks are conducted

[2] https://huggingface.co/datasets/rlhn/rlhn-680K
[3] https://huggingface.co/meta-llama/Llama-3.2-1B
[4] https://huggingface.co/meta-llama/Llama-3.2-3B
[5] https://huggingface.co/meta-llama/Llama-3.1-8B
[6] https://huggingface.co/nomic-ai/nomic-embed-text-v1
[7] https://huggingface.co/Snowflake/snowflake-arctic-embed-l-v2.0

Table 1: Model Performance (nDCG@10). We use baseline results reported by their respective papers. Average BEIR14* excludes the result on CQADupstack datasets.

Category	Model	Size	MS	Arg	Clm	CQA	DBP	FEV	FiQ	Hot	NFC	NQ	Quo	SCI	ScF	TRC	Tou	Avg BEIR14	MTEB
Sparse	SPLADE-v3	110M	45.6	50.9	23.3	32.3	45.0	79.6	37.4	69.2	35.7	58.6	81.4	15.8	71.0	74.8	29.3	51.3	50.0
	CSPLADE	8B	46.5	48.9	29.4	–	44.5	86.5	40.5	69.8	37.2	60.9	87.1	17.6	73.9	83.2	38.9	54.6	–
Dense	Nomic-v1	137M	43.1	49.3	40.5	38.3	45.0	85.0	38.4	73.6	35.0	59.4	87.7	18.3	70.5	79.9	28.2	53.9	52.8
	Arctic-v2	529M	44.0	58.0	38.3	47.2	43.9	91.6	44.0	72.4	35.9	64.6	88.7	20.3	71.8	80.3	29.8	56.0	55.4
	RepLlama3	8B	45.3	60.2	42.3	44.2	45.8	91.8	57.3	85.0	41.7	70.1	85.9	29.6	78.6	85.8	33.7	60.9	59.8
Ours	LACONIC-1B	1B	43.5	62.1	37.4	34.3	46.8	88.3	43.4	79.2	39.2	66.3	85.2	24.2	75.6	83.9	31.2	57.6	56.0
	LACONIC-3B	3B	44.0	72.0	36.4	41.5	49.0	88.5	50.7	81.7	41.0	69.8	86.7	27.3	78.2	83.4	30.7	60.0	58.7
	LACONIC-8B	8B	44.1	73.0	38.8	42.3	50.2	89.8	55.0	83.9	41.9	72.8	86.1	29.3	79.7	85.3	31.3	**61.5**	**60.2**

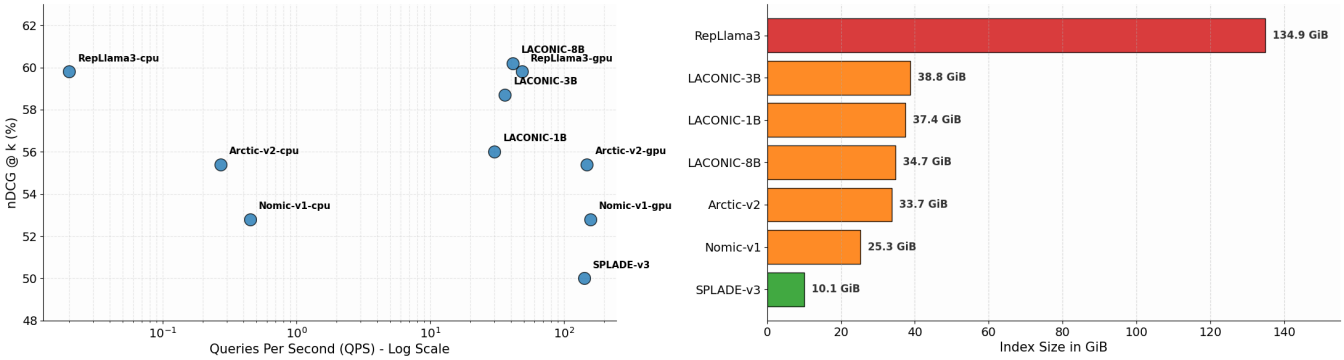

Figure 1: Efficiency comparison. Left plot shows the index search latency on MS MARCO dataset, measured by queries per second, versus retrieval performance on MTEB-R benchmark. Right plot shows memory requirement to load retrieval index. Notice that LACONIC improves the performance-latency frontier compared to baselines without requiring accelerators for efficient index search. We reproduce SPLADE-v3's result using Seismic.

on a compute node with an Intel Xeon Platinum 8275L CPU and 1152 GB RAM.

As shown in the left panel, LACONIC achieves a superior effectiveness–latency trade-off compared to dense baselines, enabling fast approximate nearest neighbor search using inverted indices alone. Notably, LACONIC does not require GPU accelerators at index search time, significantly reducing deployment complexity. We note that this latency excludes query encoding, which for an 8B decoder is non-trivial; encoding speed is reported separately in Figure 2, and we defer a fully integrated end-to-end latency benchmark to future work.

The right panel highlights the memory efficiency of learned sparse indexing. Compared to the dense RepLlama3 model, which requires 134.9 GiB to index the corpus, LACONIC-8B requires only 34.7 GiB — corresponding to a 3.9× reduction in index size. This substantially smaller memory footprint enables retrieval on commodity hardware and underscores the practical advantages of learned sparse representations. For a closer comparison within the LSR family, SPLADE-v3 and CSPLADE index the same corpus at approximately 10 GiB and >30 GiB, respectively. The roughly 3× overhead of LACONIC over BERT-scale SPLADE-v3 is broadly aligned with other LLM-backed LSR systems and reflects the higher token-activation density of LLM encoders rather than an artifact of our training objective; tighter FLOPs regularization scheduling could narrow this gap. We also note that dense indices admit complementary compression techniques such as product quantization (e.g., OPQ) and Matryoshka representation learning (MRL); these are orthogonal to LSR, and a head-to-head comparison at matched memory budgets remains valuable future work.

Encoding Speed. We report encoding speed (queries per second, QPS) in Figure 2 at BF16 precision. All results are based on a single A100 SXM4 GPU. Compared to baselines, LACONIC demonstrates competitive QPS due to native support of Flash Attention.

Ablation Studies. In Table 2, we compare the sparse retrieval model with the dense model that undergoes the same training curriculum. This ablation is carried out at 1B model scale. We include additional baselines Contriever [20] and e5-base [40] (both pre-finetuned and finetuned variants) to showcase the effectiveness of pre-finetuning.

We note the importance of the pre-finetuning phase and using a stronger backbone model. The dense 1B retriever already achieves 48.8 nDCG@10 on MTEB Retrieval datasets, outperforming the fine-tuned e5-base baseline. We observe that LACONIC underperforms its dense counterpart after pre-finetuning, which we hypothesize is

Table 2: Ablation study of dense versus learned sparse retrievers at 1B scale.

Category	Method	MS	Arg	Clm	CQA	DBP	FEV	FiQ	Hot	NFC	NQ	Quo	SCI	ScF	TRC	Tou	Avg
Unsupervised	Contriever	20.6	37.9	15.5	28.4	29.2	68.2	24.5	48.1	31.7	25.4	83.5	14.9	64.9	27.4	19.3	36.0
	E5-base	26.0	42.2	15.4	35.4	35.4	63.4	40.0	52.4	35.8	39.0	85.7	21.1	73.7	61.0	16.9	42.9
Supervised	Contriever	40.7	44.6	23.7	34.5	41.3	75.8	32.9	63.8	32.8	49.8	86.5	16.5	67.7	59.6	23.0	46.2
	E5-base	43.1	51.4	15.4	38.9	41.0	58.2	36.4	62.2	36.6	60.0	87.9	19.0	73.1	79.6	28.3	48.7
Dense	Pre-FT	32.4	54.8	22.9	42.0	37.6	74.7	42.9	63.1	36.6	45.5	88.3	21.2	73.7	71.5	24.5	48.8
	FT	43.6	57.6	38.0	42.4	46.4	89.7	46.6	78.9	37.9	65.3	86.9	25.4	76.6	84.7	32.1	**56.8**
Sparse	Pre-FT	26.6	52.5	22.1	33.4	33.1	59.9	34.0	54.9	35.7	33.6	84.5	18.4	69.5	67.0	13.3	42.6
	FT	43.5	62.1	37.4	34.3	46.8	88.3	43.4	79.2	39.2	66.3	85.2	24.2	75.6	83.9	31.2	56.0

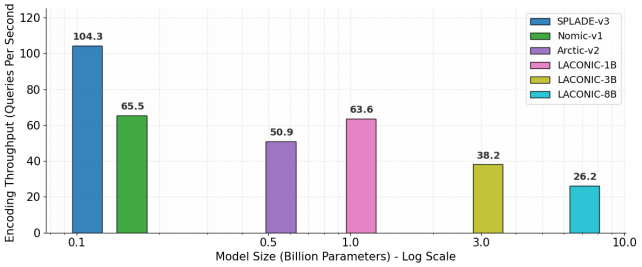

Figure 2: Encoding speed comparison at batch_size=1.

because the dense model by default uses `#hidden_dimension` features while the sparse retriever relies on a much smaller activated feature dimension with non-negative learned token importance. On the other hand, after the finetuning phase, LACONIC achieves performance comparable to its dense counterpart (56.0 versus 56.8). This result suggests the synergy of the two training phases in our training curriculum: the pre-finetuning phase adapts the pretrained causal language model for bidirectional information and sparsity pattern, laying the foundation for the subsequent finetuning; and the high-signal finetuning phase enhances the retriever's ability to identify fine-grained, more nuanced query-document relevance patterns, which is critical for retrieval performance.

4 Conclusion and Future Work

In this paper, we presented LACONIC, a family of learned sparse retrieval models that demonstrate the effectiveness of scaling learned sparse retrieval to LLM backbones. By combining a bidirectional adaptation of `Llama3` with a targeted two-phase training curriculum, we have shown that sparse retrieval can achieve performance parity with dense models while maintaining superior efficiency in index search. `LACONIC-8B` currently stands as the most performant sparse retriever on the MTEB Retrieval leaderboard as of February 5th, 2026, offering a scalable solution for high-precision retrieval on commodity hardware. We view these results as evidence that efficiency and scale need not be opposing forces in neural retrieval, and that LLM-based architectures can produce sparse, index-friendly representations without sacrificing effectiveness, pointing toward a unified design space that bridges neural and classical IR. More broadly, we envision a new generation of retrieval systems in which large language models serve not only as generators or re-rankers, but as efficient first-stage retrievers—enabling scalable, high-quality access to information across domains, modalities, and deployment environments.

Future work will explore several promising directions. In particular, we aim to extend LACONIC to multilingual and multimodal settings [31], further optimize the training data mixture to improve performance, and investigate inference-free learned sparse retrieval [7, 15, 30]. In parallel, improving end-to-end system efficiency—particularly through faster query encoding and integration with inference optimization techniques—remains important for real-world deployment, alongside continued benchmarking against emerging retrieval models and strengthening evaluation practices as the field evolves.

References

[1] Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. SparTerm: Learning term-based sparse representation for fast text retrieval. arXiv:2010.00768 [cs.IR] https://arxiv.org/abs/2010.00768

[2] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders. In *First Conference on Language Modeling*. https://openreview.net/forum?id=IW1PR7vEBf

[3] Sebastian Bruch, Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. 2024. Efficient inverted indexes for approximate retrieval over learned sparse representations. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 152–162.

[4] Sebastian Bruch, Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. 2025. Efficient Sketching and Nearest Neighbor Search Algorithms for Sparse Vector Sets. arXiv:2509.24815 [cs.DS] https://arxiv.org/abs/2509.24815

[5] Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 985–988.

[6] Tri Dao. 2023. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. arXiv:2307.08691 [cs.LG] https://arxiv.org/abs/2307.08691

[7] DatologyAI, Luke Merrick, Alex Fang, Aldo Carranza, Alvin Deng, Amro Abbas, Brett Larsen, Cody Blakeney, Darren Teh, David Schwab, Fan Pan, Haakon Mongstad, Haoli Yin, Jack Urbanek, Jason Lee, Jason Telanoff, Josh Wills, Kaleigh Mentzer, Paul Burstein, Parth Doshi, Paul Burnstein, Pratyush Maini, Ricardo Monti, Rishabh Adiga, Scott Loftin, Siddharth Joshi, Spandan Das, Tony Jiang, Vineeth Dorna, Zhengping Wang, Bogdan Gaza, Ari Morcos, and Matthew Leavitt. 2025. Luxical: High-Speed Lexical-Dense Text Embeddings. arXiv:2512.09015 [cs.CL] https://arxiv.org/abs/2512.09015

[8] Meet Doshi, Vishwajeet Kumar, Rudra Murthy, Vignesh P, and Jaydeep Sen. 2024. Mistral-SPLADE: LLMs for better Learned Sparse Retrieval. arXiv:2408.11119 [cs.IR] https://arxiv.org/abs/2408.11119

[9] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. arXiv:2109.10086 [cs.IR] https://arxiv.org/abs/2109.10086

[10] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From Distillation to Hard Negative Sampling: Making Sparse Neural IR Models More Effective. In *Proceedings of the 45th International ACM SIGIR*

Conference on Research and Development in Information Retrieval (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 2353–2359. doi:10.1145/3477495.3531857

[11] Thibault Formal, Maxime Louis, Hervé Dejean, and Stéphane Clinchant. 2026. Learning Retrieval Models with Sparse Autoencoders. arXiv:2603.13277 [cs.LG] https://arxiv.org/abs/2603.13277

[12] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. 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*. 2288–2292.

[13] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Tevatron: An efficient and flexible toolkit for dense retrieval. arXiv:2203.05765 [cs.IR] https://arxiv.org/abs/2203.05765

[14] Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021. Scaling Deep Contrastive Learning Batch Size under Memory Limited Setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, Anna Rogers, Iacer Calixto, Ivan Vulić, Naomi Saphra, Nora Kassner, Oana-Maria Camburu, Trapit Bansal, and Vered Shwartz (Eds.). Association for Computational Linguistics, Online, 316–321. doi:10.18653/v1/2021.repl4nlp-1.31

[15] Zhichao Geng, Yiwen Wang, Dongyu Ru, and Yang Yang. 2025. Towards Competitive Search Relevance For Inference-Free Learned Sparse Retrievers. arXiv:2411.04403 [cs.IR] https://arxiv.org/abs/2411.04403

[16] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The Llama 3 herd of models. arXiv:2407.21783 [cs.CL] https://arxiv.org/abs/2407.21783

[17] Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2024. Jina Embeddings 2: 8192-Token General-Purpose Text Embeddings for Long Documents. arXiv:2310.19923 [cs.CL] https://arxiv.org/abs/2310.19923

[18] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

[19] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SkxgnnNFvH

[20] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research* (2022). https://openreview.net/forum?id=jKN1pXi7b0

[21] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. arXiv:1702.08734 [cs.CV] https://arxiv.org/abs/1702.08734

[22] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 6769–6781. doi:10.18653/v1/2020.emnlp-main.550

[23] Carlos Lassance and Stéphane Clinchant. 2022. An Efficiency Study for SPLADE Models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 2220–2226. doi:10.1145/3477495.3531833

[24] Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. SPLADE-v3: New baselines for SPLADE. arXiv:2403.06789 [cs.IR] https://arxiv.org/abs/2403.06789

[25] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 6086–6096. doi:10.18653/v1/P19-1612

[26] Chaofan Li, Minghao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Defu Lian, Yingxia Shao, and Zheng Liu. 2025. Making Text Embedders Few-Shot Learners. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=wfLuiDjQ0u

[27] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2022. *Pretrained transformers for text ranking: BERT and beyond*. Springer Nature.

[28] Joel Mackenzie, Shengyao Zhuang, and Guido Zuccon. 2023. Exploring the Representation Power of SPLADE Models. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval* (Taipei, Taiwan) *(ICTIR '23)*. Association for Computing Machinery, New York, NY, USA, 143–147. doi:10.1145/3578337.3605129

[29] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Andreas Vlachos and Isabelle Augenstein (Eds.). Association for Computational Linguistics, Dubrovnik, Croatia, 2014–2037. doi:10.18653/v1/2023.eacl-main.148

[30] Franco Maria Nardini, Thong Nguyen, Cosimo Rulli, Rossano Venturini, and Andrew Yates. 2025. Effective inference-free retrieval for learned sparse representations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2936–2940.

[31] Thong Nguyen, Yibin Lei, Jia-Huei Ju, Eugene Yang, and Andrew Yates. 2025. Milco: Learned Sparse Retrieval Across Languages via a Multilingual Connector. arXiv:2510.00671 [cs.IR] https://arxiv.org/abs/2510.00671

[32] Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2025. Nomic Embed: Training a Reproducible Long Context Text Embedder. arXiv:2402.01613 [cs.CL] https://arxiv.org/abs/2402.01613

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. arXiv:1807.03748 [cs.LG] https://arxiv.org/abs/1807.03748

[34] Biswajit Paria, Chih-Kuan Yeh, Ian EH Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. 2020. Minimizing FLOPs to Learn Efficient Sparse Representations. In *International Conference on Learning Representations*.

[35] Jingfen Qiao, Thong Nguyen, Evangelos Kanoulas, and Andrew Yates. 2025. Leveraging decoder architectures for learned sparse retrieval. In *International Workshop on Knowledge-Enhanced Information Retrieval*. Springer, 19–35.

[36] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 3982–3992. doi:10.18653/v1/D19-1410

[37] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *NIST Special Publication Sp* 109 (1995), 109.

[38] Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2024. Repetition improves language model embeddings. arXiv:2402.15449 [cs.CL] https://arxiv.org/abs/2402.15449

[39] Nandan Thakur, Crystina Zhang, Xueguang Ma, and Jimmy Lin. 2025. Hard Negatives, Hard Lessons: Revisiting Training Data Quality for Robust Information Retrieval with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 9064–9083. doi:10.18653/v1/2025.findings-emnlp.481

[40] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. arXiv:2212.03533 [cs.CL] https://arxiv.org/abs/2212.03533

[41] Zhichao Xu, Aosong Feng, Yijun Tian, Haibo Ding, and Lin Lee Cheong. 2025. CSPLADE: Learned Sparse Retrieval with Causal Language Models. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh (Eds.). The Asian Federation of Natural Language Processing and The Association for Computational Linguistics, Mumbai, India, 99–114. doi:10.18653/v1/2025.ijcnlp-long.7

[42] Zhichao Xu, Ashim Gupta, Tao Li, Oliver Bentham, and Vivek Srikumar. 2024. Beyond Perplexity: Multi-dimensional Safety Evaluation of LLM Compression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 15359–15396. doi:10.18653/v1/2024.findings-emnlp.901

[43] Zhichao Xu, Zhiqi Huang, Shengyao Zhuang, and Vivek Srikumar. 2025. Distillation versus Contrastive Learning: How to Train Your Rerankers. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh (Eds.). The Asian Federation of Natural Language Processing and The Association for Computational Linguistics, Mumbai, India, 564–578. doi:10.18653/v1/2025.findings-ijcnlp.33

[44] Zhichao Xu, Fengran Mo, Zhiqi Huang, Crystina Zhang, Puxuan Yu, Bei Wang Phillips, Jimmy Lin, and Vivek Srikumar. 2026. A Survey of Model Architectures in Information Retrieval. *Transactions on Machine Learning Research* (2026). https://openreview.net/forum?id=xAIbTbHRrX Survey Certification.

[45] Puxuan Yu, Luke Merrick, Gaurav Nuti, and Daniel Campos. 2024. Arctic-Embed 2.0: Multilingual Retrieval Without Compromise. arXiv:2412.04506 [cs.CL] https://arxiv.org/abs/2412.04506

[46] Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 497–506.

[47] Hansi Zeng, Julian Killingback, and Hamed Zamani. 2025. Scaling sparse and dense retrieval in decoder-only LLMs. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2679–2684.

[48] Tiancheng Zhao, Xiaopeng Lu, and Kyusong Lee. 2021. SPARTA: Efficient Open-Domain Question Answering via Sparse Transformer Matching Retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 565–575. doi:10.18653/v1/2021.naacl-main.47

[49] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. 2023. PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel. arXiv:2304.11277 [cs.DC] https://arxiv.org/abs/2304.11277

[50] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. arXiv:2308.07107 [cs.IR] https://arxiv.org/abs/2308.07107