Effective Inference-Free Retrieval for Learned Sparse Representations

Franco Maria Nardini
ISTI-CNR
Pisa, Italy
francomaria.nardini@isti.cnr.it

Thong Nguyen
University of Amsterdam
Amsterdam, The Netherlands
t.nguyen2@uva.nl

Cosimo Rulli
ISTI-CNR
Pisa, Italy
cosimo.rulli@isti.cnr.it

Rossano Venturini
University of Pisa
Pisa, Italy
rossano.venturini@unipi.it

Andrew Yates
Johns Hopkins University, HLTCOE
Baltimore, MD, USA
andrew.yates@jhu.edu

Abstract

Learned Sparse Retrieval (LSR) is an effective IR approach that exploits pre-trained language models for encoding text into a learned bag of words. Several efforts in the literature have shown that sparsity is key to enabling a good trade-off between the efficiency and effectiveness of the query processor. To induce the right degree of sparsity, researchers typically use regularization techniques when training LSR models. Recently, new efficient—inverted index-based—retrieval engines have been proposed, leading to a natural question: has the role of regularization changed in training LSR models? In this paper, we conduct an extended evaluation of regularization approaches for LSR where we discuss their effectiveness, efficiency, and out-of-domain generalization capabilities. We first show that regularization can be relaxed to produce more effective LSR encoders. We also show that query encoding is now the bottleneck limiting the overall query processor performance. To remove this bottleneck, we advance the state-of-the-art of inference-free LSR by proposing *Learned Inference-free Retrieval* (Li-Lsr). At training time, Li-Lsr learns a score for each token, casting the query encoding step into a seamless table lookup. Our approach yields state-of-the-art effectiveness for both in-domain and out-of-domain evaluation, surpassing Splade-v3-Doc by 1 point of mRR@10 on MsMarco and 1.8 points of nDCG@10 on Beir.

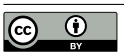
CCS Concepts

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

Keywords

Learned Sparse Retrieval, Inference-Free, Efficiency

ACM Reference Format:
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 (SIGIR '25), July 13–18, 2025, Padua, Italy.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3726302.3730185



SIGIR '25, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3730185

1 Introduction

Pretrained Language Models (PLM) have impacted IR in many ways [18]. Indeed, the possibility of producing vector representations, i.e., *embeddings*, of the input text by grasping semantic and contextual information is one of the most important [13, 15, 29]. Learning representations allows the problem of retrieving relevant documents for a query to be cast into the problem of finding the closest document embeddings w.r.t. the input query embedding according to a specific similarity metric, which is often dot product or cosine. Learned sparse representations (LSR) [7–9, 16, 17, 20, 30] map queries and documents into n-dimensional vectors where dimensions are grounded on the vocabulary of the collection. When a coordinate is nonzero in a sparse embedding, that indicates that the corresponding term is semantically relevant to the input. LSR models have shown to be competitive with dense retrieval models [6, 13, 15, 18, 24, 26, 29] and to effectively generalize to out-of-domain datasets [2, 17], while being interpretable by design. Moreover, sparse embeddings retain some of the benefits of classical lexical models such as BM25 [25], including the possibility to use retrieval machinery based on inverted indexes.

Despite the natural link that can be drawn between learned sparse representations and inverted indexes, their matching has only recently been fully realized. In fact, learned sparse representations show important differences w.r.t. normal query/document text, such as longer queries, and the non-Zipfian distribution of terms [21], which results in increased per-query latency. Several efforts have investigated how this performance gap can be closed when using inverted index-based methods. In this line, Bruch *et al.* recently introduced Seismic [3–5], a novel inverted index-based approximate nearest neighbors retrieval, which is shown to outperform state-of-the-art approaches for retrieval with learned sparse representations. Seismic is inherently built to work with Learned Sparse Representations and its properties, such as longer queries, longer documents, and non-Zipfian distribution.

In this paper, we explore the role of term expansion in Learned Sparse Retrieval, by proposing an extended evaluation of regularization approaches for LSR on both in- and out-of-domain benchmarks. We show that regularization can be relaxed in LSR, producing more effective sparse embeddings while still allowing for reduced retrieval times, thanks to modern retrieval engines. In this setting, query encoding can be a significant bottleneck (on CPU), limiting the performance of the query processor. We thus advance the state

of the art by introducing *Learned Inference-free Retrieval* (Li-Lsr), a new technique that learns a score for each token, casting the query encoding step to a fast table lookup operation.

In summary, the contributions of this paper are the following:

- We present an in-depth evaluation of regularization techniques for LSR, showing its pivotal role in in- and out-of-domain effectiveness. In particular, our setup shows how relaxing regularization improves retrieval quality in both scenarios.
- We evaluate the consequences of relaxing regularization from an efficiency perspective. The analysis shows that when leveraging modern retrieval engines, even longer (i.e., less sparse) representations can be retrieved efficiently.
- We propose *Learned Inference-free Retrieval* (Li-Lsr), a novel inference-free approach that allows the replacement of the query encoder with a fast lookup table. Li-Lsr learns the optimal score for each term at training time, enabling both effective and efficient query encoding. Li-Lsr outperforms state-of-the-art inference-free retrievers such as Splade-v3-Doc, with a margin of 1 point of mRR on MsMarco and 1.8 points of nDCG@10 on Beir.

2 Methodology

In this section, we define the methodology used to assess the impact of term expansion in modern LSR. Moreover, we introduce Li-Lsr, a novel approach to the inference-free paradigm that learns the importance score of each term at training time.

2.1 Exploring Term Expansion in LSR

Unlike naïve term-matching engines, which exploit terms that appear in the input text, learned sparse representations are trained to activate relevant terms regardless of whether they appear in the input tokens. In fact, LSR can encompass *term expansion* on both the query and the document, i.e., adding terms to the representation to improve its generalization power. Term expansion in LSR encoders is enabled and controlled by regularization terms applied to the loss function, which control the sparsity in the representations, thus enabling a practical—yet effective—use of the obtained embeddings.

In this section, we define a methodology that aims to assess the effectiveness of LSR as a function of the regularization applied during training. Equation 1 introduces a general loss function $\mathcal{L}$ used for learning sparse representations.

$$\mathcal{L} \;=\; \mathcal{L}_{\text{rank}} + \lambda_q \, \mathcal{L}^{Q}_{\text{reg}} \;+\; \lambda_d \, \mathcal{L}^{D}_{\text{reg}} \tag{1}$$

In state-of-the-art methods, $\mathcal{L}_{\text{rank}}$ is typically a distillation loss [8, 16, 17], encouraging the student model to align its scores with a strong teacher like a cross-encoder [28].

We investigate how λ_q and λ_d impact the effectiveness of an LSR model and how they interact with different loss functions. To do so, we opt for the following methodology. We first define a training configuration as a pair comprising a training loss and a regularizer. We then build three LSR models: Small, Medium, and Big, which are categorized based on the average number of non-zero values in the document representations for the collection, which we define as $|D|$. In Small models, we do not employ term expansion, i.e., $|D|$ is comparable to the number of tokens per document. For MsMarco, this number is approximately 60. In Medium models, we set term expansion to double the number of non-zero values in the documents compared to the Small configuration. This aligns with the expansion achieved by state-of-the-art models such as Splade (119) and Splade-v3 (168) [17]. For Big models, we set $|D|$ to be roughly 5× larger than the non-expanded scenario. This setting is particularly important. Expansion on both the query and document sides has traditionally been constrained to ensure efficient retrieval using classical inverted indexes. Our evaluation assesses how the retrieval effectiveness benefits from a larger expansion. Moreover, Big aligns with most common dense representations [19] regarding embedding memory footprint. A non-zero entry in a sparse embedding requires 16 bits for the token identifier, as long as the vocabulary size is smaller than 2^{16}, plus 16 bits for the token weight when using float16 precision. When storing dense vectors using a float16 per dimension, a sparse entry requires twice the memory of a dense one. Given that most common BERT-based dense encoders produce 768-dimensional vectors [12, 19], the memory footprint of embeddings generated by Big is approximately the same as those produced by a dense encoder, i.e., 1536 bytes per vector.

2.2 Learned Inference-Free Retrieval

The *inference-free* paradigm bypasses the costly query encoding phase by assigning a fixed value, such as 1, to each query term. While eliminating encoding costs brings efficiency gains, it significantly reduces retrieval effectiveness. For example, as noted by Geng *et al.* [11], Splade-v3 experiences a drop of 2.4 points in mRR@10 on MsMarco and 4.7 points in nDCG@10 on Beir when query encoding is removed [11].

We propose to advance the state of the art in inference-free retrieval by introducing *Learned Inference-free Retrieval* (Li-Lsr). Li-Lsr learns a static relevance score for each token during training by projecting the output of word embeddings into a scalar value using a simple linear layer. This transforms query encoding into a seamless lookup in a term-score dictionary. Unlike existing methods that rely on fixed scores or lexical statistics, Li-Lsr enables the model to learn the optimal value for each token at training time.

Let us consider a tokenizer that maps a query input sequence q into a set of tokens $x = \{x_0, \ldots, x_l\}$, where l is the number of tokens and $x_i \in \mathrm{I}_V$, being I_V the set of valid indices in V. Our goal is to learn a mapping $S = \{x_i : s_i\}_{i=0}^{|V|}$ to reduce query encoding to a simple lookup in S.

Transformer-based encoders rely on an embedding module, E, to project x into a vector representation $E(x) \in \mathbb{R}^{l \times d}$, where d is the vector dimensionality. Typically, E consists of multiple components, including but not limited to word embeddings, positional embeddings, and a normalization layer. In our approach, we consider only the word embedding module, E_W, because it provides a *context-independent* representation for a given token x_i, namely $E_W(x) = [E_W(x_1), \ldots, E_W(x_l)]$. In contrast, positional embeddings depend on token position, $E_P(x_i) = E_P(x_i; i)$, and layer normalization requires the computation over the entire input sequence.

We introduce a linear layer w with a bias b to project the output of the word embedding module into a per-token score, and then we apply a positivity constraint, similarly to the Splade log-saturation function, to obtain s_i:

$$s_i = \log(1 + \mathrm{ReLU}[w^T E_W(x_i) + b]). \tag{2}$$

Loss	Reg.	Size	λ_q	λ_d	|Q|	|D|	MsMarco	Beir
KL	L_1	Small	1e-3	1e-4	8	73	39.08	48.83
		Medium	5e-5	5e-5	14	128	39.44	49.55
		Big	1e-5	1e-5	16	304	**39.81**	49.04
	FLOPs	Small	1e-2	1e-2	17	78	38.49	48.52
		Medium	2e-3	2e-3	14	145	38.91	50.02
		Big	2e-4	2e-4	14	355	39.51	49.24
MSE	L_1	Small	5e-2	5e-2	13	38	37.83	48.07
		Medium	5e-3	5e-3	30	133	38.68	50.71
		Big	1e-3	1e-3	45	326	38.86	50.65
	FLOPs	Small	2.50	2.50	34	69	37.64	47.30
		Medium	5e-1	5e-1	43	138	38.38	50.35
		Big	5e-2	5e-2	38	294	38.84	**51.02**

Table 1: Performance of an LSR Model with different levels of expansion. Best results in terms of mRR@10 and nDCG@10 are reported in bold.

Whether the same token appears more than once in the input sequence, the final score of the token is multiplied by the number of times it appears.

3 Experimental Evaluation

Datasets and Evaluation. We train our models, i.e., Small, Medium, Big, on the MsMarco [23] dataset over four different combinations of training loss, i.e., Kullback-Leibler (KL) or Mean Squared Error (MSE), and regularization strategies (L_1 or FLOPs), distilling from the *msmarco-hard-negatives*.[1] We use a single negative for each query. We employ a batch size of 128, a learning rate of 5e-5, and we train for 150,000 steps. We start our training from the Co-Condenser [10] checkpoint available on HuggingFace.[2] We evaluate our LSR models both on in-domain and out-of-domain benchmarks. For the former, we use the 6,980 queries in the *dev/small* set of MsMarco, evaluating the ranking performance using mRR@10. For the latter, we use the 13 datasets in the Beir [27] collections, and we adopt nDCG@10 as a metric, following [11, 17].

Indexing and Retrieval. We index our collection using Seismic [3]. Seismic allows both for exhaustive and approximate search. We employ exhaustive search in effectiveness-oriented evaluations. We resort to approximate search when comparing the efficiency-effectiveness trade-off achieved by different LSR models. We perform a grid search over the length of the truncated posting lists of Seismic, varying it in $\{2000, 4000, 6000, 8000\}$. Following Bruch *et al.* [5], we set the summary energy, i.e., α, to 0.4 and the number of centroids per list to one-tenth of the posting list length. The efficiency evaluation was conducted on a machine equipped with two Intel Xeon Silver 4314 CPU, clocked at 2.40GHz, 256GiB of RAM. All the experiments involving retrieval and encoding were conducted using a single execution thread.

3.1 Impact of Term Expansion in LSR

We present the results of our comparison in Table 1. For each row, we report the type of loss function used for training (KL or MSE), the regularization method (L_1 or FLOPs), and the corresponding values of query and document regularization (λ_q and λ_d). Additionally, we provide the resulting number of non-zero entries per query ($|Q|$) and per document ($|D|$), along with their retrieval performance on MsMarco (mRR@10) and Beir (nDCG@10).

Table 1 shows that both query and document density positively correlate with effectiveness, particularly for documents. In fact, increasing document length proves to be an effective strategy for improving the retrieval quality of the encoder. The only exception is nDCG@10 on Beir for models trained with *KL*, where performance peaks with Medium models rather than Big ones. We also observe that $|Q|$ is influenced not only by λ_q but also by λ_d. This indicates that even when query regularization is disabled ($\lambda_q = 0$), query expansion remains constrained. Additionally, we find that models perform best when λ_q and λ_d are equal, with the exception of Small in the L_1 + *KL* scenario.

Regarding in-domain and out-of-domain effectiveness, the *KL* divergence achieves the best results on MsMarco, reaching a maximum mRR@10 of 39.81 with L_1 regularization at the maximum allowed expansion. The Small model, trained using the same configuration, attains an mRR@10 of 39.08, while offering the advantage of requiring 2× and 4× smaller embedding collections compared to Medium and Big, respectively. On the other hand, MSE is more effective for out-of-domain retrieval, particularly when combined with FLOPs regularization. The best-performing model is obtained using FLOPs regularization and MSE as the training loss, achieving an nDCG@10 of 51.02 on Beir. We attribute this to the larger query expansion induced by MSE loss compared to *KL* divergence.

3.2 Efficiency-Effectiveness Trade-off

In this section, we investigate the relationship between expansion and retrieval efficiency. Figure 1 reports the mRR@10 on MsMarco as a function of the average query time (AQT, in μsec.) for each of the models in Table 1, grouped into Small, Medium, and Big.

Despite the large number of non-zero terms in queries and documents, models from the Big family still enable highly efficient retrieval. In particular, the model trained with *KL* divergence as the loss function and L_1 regularization achieves an mRR@10 of 39.80—over 99.9% of the peak effectiveness—in approximately 800 μsec. using single-threaded execution. Notably, this level of mRR@10 is comparable to that of multi-vector encoders such as ColBERTv2 [26]. As an example, we tested the optimized EMVB [22] retrieval engine in the $k = 10$, $m = 32$ setting; while yielding a mRR@10 of 39.7, retrieving with EMVB takes about 60 *m*sec., almost two orders of magnitude slower than Seismic.

As mentioned above, query expansion plays a key role in out-of-domain effectiveness. While a decrease in query sparsity hinders the usage of standard inverted indexes [3], Seismic's efficiency is resilient to query expansion. For example, the FLOPs + MSE model can quickly reach 99.5% of its maximum performance in 1.3 *m*sec.

Comparison with encoding time We complement the efficiency-effectiveness trade-off analysis by measuring the time required to encode a query with a BERT-based model [14]. We measure the inference time using the `rust-bert` Rust package [1]. To align the evaluation with the methodology used for retrieval, we measure it on the CPU in single-thread execution with a batch size of 1. We

[1] https://huggingface.co/datasets/sentence-transformers/msmarco-hard-negatives
[2] https://huggingface.co/Luyu/co-condenser-marco

Model	Neg.	Teach.	PT	MM	DL19	DL20	Beir
Splade-Doc-Distill	1	1	✗	36.5	69.8	-	45.0
Splade-v3-Doc	M	M	✗	37.8	71.5	70.3	47.0
Geng *et al.* [11]	M	M	✓	37.8	72.1	-	50.4
Li-Lsr							
Medium	1	1	✗	37.6	69.6	69.6	47.6
Big	1	1	✗	38.8	72.1	70.5	48.8
Li-Lsr + IDF							
Medium	1	1	✗	37.8	71.3	68.3	48.2
Big	1	1	✗	38.0	71.6	69.6	48.9

Table 2: Comparison between inference-free LSR models.

employ MsMarco *dev/small* as a benchmark. The average encoding time is 24 *m*sec., one order of magnitude slower than almost-exact retrieval with Seismic. This evidence poses the interesting challenge of reducing the query encoding time for sparse retrievers.

3.3 Learned Inference-Free Retrieval

We now evaluate the effectiveness of our *Learned Inference-Free Retrieval* (Li-Lsr) technique, which aims at introducing a learned inference-free query encoding. We compare our approach with state-of-the-art inference-free retrievers in Table 2, including Splade-Doc-Distill [8], Splade-v3-Doc [17], and a method by Geng *et al.* [11] that uses inverse document frequency as query term weights. For each model, we report the in-domain performance measured as mRR@10 on MsMarco ("MM"), and as nDCG@10 on TrecDL-2019 ("DL19") and TrecDL-2020 ("DL20"). The out-of-domain effectiveness is measured in terms of nDCG@10 on Beir. We also describe its training methodology by specifying the number of negatives used per query ("Neg.") and whether it employs single or multiple teachers ("Teach."). To this regard, we highlight that Geng *et al.* [11] use a vast pre-training on different data sources before fine-tuning on MsMarco [11]. Our approach is trained with the settings detailed in Section 3, namely with a single teacher and a single negative per query. We use L_1 as the regularization function and KL as loss. In our experiments, we observed that the KL loss yields consistently better results than MSE for Li-Lsr. Capitalizing on the lessons learned in Section 3.1, we train models with different document expansions. We report the results of Medium (λ_d = 1e−4) and Big (λ_d = 1e−5), the former to assess the effectiveness of Li-Lsr compared to other inference-free techniques, the latter to explore

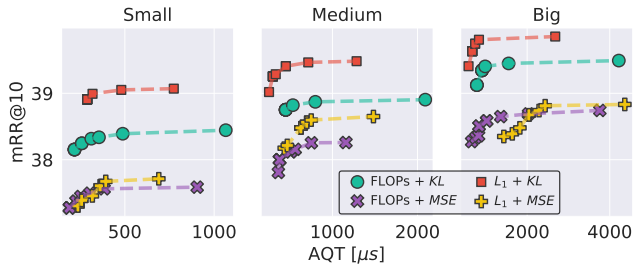


Figure 1: Efficiency-effectiveness trade-off (AQT vs. mRR@10) of different LSR models with different expansion.

the potential of inference-free retrieval when paired with boosted document expansion.

Table 2 shows the effectiveness of Li-Lsr. Our Li-Lsr- Medium model rivals the performance of Splade-v3-Doc for in-domain and surpasses it for out-of-domain, despite Splade-v3-Doc employing multiple teachers and multiple negatives per query. With the same training budget, Li-Lsr-Medium outperforms Splade-Doc-Distill by 1.1 points of mRR@10 on MsMarco and 2.6 of nDCG@10 on Beir. Li-Lsr- Big surpasses by 1.0 points both Splade-v3 and the approach by Geng *et al.* [11], despite the vast disparity between the training budget. This showcases the effectiveness of our approach to learning the scores to assign to query terms rather than relying on fixed, predetermined values. We also combine our approach with the one from Geng *et al.* [11] by computing the query term scores by multiplying our learned word embedding projection with the IDF value of the corresponding token. These configurations are reported as "Li-Lsr + IDF". Our experiments show that combining IDF with our learned approach allows smaller models (Li-Lsr + IDF - Medium) to improve the out-of-domain-effectiveness. In contrast, the large model, i.e., Li-Lsr + IDF - Big, poorly benefits from it.

4 Conclusions and Future Work

We presented a study about the role of term expansion in sparse encoders, both in terms of effectiveness and efficiency. We found that a lighter regularization—allowing more non-zero terms in queries and documents—enables for improved retrieval quality. Thanks to new state-of-the-art retrieval engines explicitly tailored for LSR, even with light regularization, retrieval can be done in a few milliseconds per query. These considerations facilitate more effective sparse encoders that rival multi-vector representations in effectiveness while being almost two orders of magnitude faster. Moreover, we introduce Li-Lsr, a new method for inference-free retrieval. Li-Lsr enables the encoding phase to be converted into a fast dictionary lookup of a learned score for each token. Our approach outperforms existing inference-free methods, even when relying on much simpler training recipes. For example, it outperforms Splade-v3-Doc by 1 point of mRR on MsMarco and 1.8 points of nDCG@10 on Beir. As future work, given the importance of document expansion highlighted by this analysis, we intend to investigate compression techniques for learned sparse representation.

Acknowledgments. We acknowledge the support of "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI" funded by the European Union (EU) under the NextGeneration EU programme, and by the Horizon Europe RIA "Extreme Food Risk Analytics" (EFRA) funded by the European Commission under the NextGeneration EU programme grant agreement n. 101093026. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the EU or European Commission-EU. Neither the EU nor the granting authority can be held responsible for them. SoBigData.it receives funding from European Union - NextGenerationEU - National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR) - Project: "SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics" - Prot. IR0000013 - Avviso n. 3264 del S28/12/2021. This research was partially supported by the Dutch Research Council (NWO) under project number VI.Vidi.223.166.

References

[1] Guillaume Becquin. 2020. End-to-end NLP Pipelines in Rust. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*. Association for Computational Linguistics, 20–25. https://www.aclweb.org/anthology/2020.nlposs-1.4

[2] Sebastian Bruch, Siyu Gai, and Amir Ingber. 2023. An Analysis of Fusion Functions for Hybrid Retrieval. *ACM Transactions on Information Systems* 42, 1, Article 20 (August 2023), 35 pages.

[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. 2024. Pairing Clustered Inverted Indexes with κ-NN Graphs for Fast Approximate Retrieval over Learned Sparse Representations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 3642–3646.

[5] Sebastian Bruch, Franco Maria Nardini, Cosimo Rulli, Rossano Venturini, and Leonardo Venuta. 2025. Investigating the Scalability of Approximate Sparse Retrieval Algorithms to Massive Datasets. In *European Conference on Information Retrieval*. Springer, 437–445.

[6] Majid Daliri, Juliana Freire, Christopher Musco, Aécio Santos, and Haoxiang Zhang. 2023. Sampling Methods for Inner Product Sketching. arXiv:2309.16157 [cs.DB]

[7] 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]

[8] 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). 2353–2359.

[9] 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* (Virtual Event, Canada). 2288–2292.

[10] Luyu Gao and Jamie Callan. 2022. Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2843–2853.

[11] Zhichao Geng, Dongyu Ru, and Yang Yang. 2024. Towards Competitive Search Relevance For Inference-Free Learned Sparse Retrievers. *arXiv preprint arXiv:2411.04403* (2024).

[12] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 113–122.

[13] 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*. 6769–6781.

[14] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, Vol. 1. Minneapolis, Minnesota.

[15] Omar Khattab and Matei Zaharia. 2020. 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*. 39–48.

[16] 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). 2220–2226.

[17] Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. SPLADE-v3: New baselines for SPLADE. *arXiv preprint arXiv:2403.06789* (2024).

[18] Jimmy Lin, Rodrigo Frassetto Nogueira, and Andrew Yates. 2021. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Morgan & Claypool Publishers.

[19] Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to Train Your Dragon: Diverse Augmentation Towards Generalizable Dense Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 6385–6400.

[20] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Expansion via Prediction of Importance with Contextualization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China). 1573–1576.

[21] Joel Mackenzie, Andrew Trotman, and Jimmy Lin. 2021. Wacky Weights in Learned Sparse Representations and the Revenge of Score-at-a-Time Query Evaluation. arXiv:2110.11540 [cs.IR]

[22] Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. 2024. Efficient Multi-vector Dense Retrieval with Bit Vectors. In *European Conference on Information Retrieval*. Springer, 3–17.

[23] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. (November 2016).

[24] 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*. Association for Computational Linguistics.

[25] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3.. In *TREC (NIST Special Publication, Vol. 500-225)*, Donna K. Harman (Ed.). National Institute of Standards and Technology (NIST), 109–126.

[26] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. 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*. 3715–3734.

[27] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[28] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems* 33 (2020), 5776–5788.

[29] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*.

[30] 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*. 565–575.