LaPraDoR: Unsupervised Pretrained Dense Retriever for Zero-Shot Text Retrieval

Canwen Xu[1]*, **Daya Guo**[2]*, **Nan Duan**[3], **Julian McAuley**[1]
[1]University of California, San Diego, [2]Sun Yat-sen University, [3]Microsoft Research Asia
[1]{cxu,jmcauley}@ucsd.edu, [2]guody5@mail2.sysu.edu.cn,
[3]nanduan@microsoft.com

Abstract

In this paper, we propose LaPraDoR, a pretrained dual-tower dense retriever that does not require any supervised data for training. Specifically, we first present Iterative Contrastive Learning (ICoL) that iteratively trains the query and document encoders with a cache mechanism. ICoL not only enlarges the number of negative instances but also keeps representations of cached examples in the same hidden space. We then propose Lexicon-Enhanced Dense Retrieval (LEDR) as a simple yet effective way to enhance dense retrieval with lexical matching. We evaluate LaPraDoR on the recently proposed BEIR benchmark, including 18 datasets of 9 zero-shot text retrieval tasks. Experimental results show that LaPraDoR achieves state-of-the-art performance compared with supervised dense retrieval models, and further analysis reveals the effectiveness of our training strategy and objectives. Compared to re-ranking, our lexicon-enhanced approach can be run in milliseconds (22.5× faster) while achieving superior performance.[1]

1 Introduction

Dense retrieval uses dense vectors to represent documents and retrieve documents by similarity scores between query vectors and document vectors. Different from cross-encoders (Reimers and Gurevych, 2019; Gao et al., 2020; MacAvaney et al., 2020) or late-interaction models (Khattab and Zaharia, 2020; Gao et al., 2021a), which predict a match score for each query-document pair thus are computationally costly, dense retrieval can be run in milliseconds, with the help of an approximate nearest neighbor (ANN) retrieval library, e.g., FAISS (Johnson et al., 2021).

As a drawback, dense retrieval models often require large supervised datasets like MS-MARCO (Nguyen et al., 2016) (533k training examples) or NQ (Kwiatkowski et al., 2019) (133k training examples) for training. Unfortunately, Thakur et al. (2021) empirically show that models trained on one dataset suffer from an out-of-domain (OOD) problem when transferring to another. This hinders the applications of dense retrieval systems. On the other hand, creating a large supervised training dataset for dense retrieval is time-consuming and expensive. For many low-resource languages, there is even no existing supervised dataset for retrieval and it can be extremely difficult to construct one.

The recently proposed BEIR benchmark (Thakur et al., 2021) highlights the generalization ability of text retrieval systems. The benchmark features a setting where models are trained on a large supervised dataset MS-MARCO (Nguyen et al., 2016) and then tested on 18 heterogeneous datasets of 9 tasks. In this paper, we propose **La**rge-scale **Pr**etr**a**ined **D**ense Zer**o**-shot **R**etriever (LaPraDoR), a fully unsupervised pretrained retriever for zero-shot text retrieval. While existing dense retrievers need large supervised data and struggle to compete with a lexical matching approach like BM25 (Robertson and Zaragoza, 2009) for zero-shot retrieval, we take a different approach by complementing lexical matching with semantic matching. Without any supervised data, LaPraDoR outperforms all dense retrievers on BEIR. LaPraDoR achieves state-of-the-art performance on BEIR with a further fine-tuning, outperforming re-ranking, despite being 22.5× and 42× faster on GPU and CPU, respectively.

Training LaPraDoR faces two challenges: **(1) Training Efficiency.** For large-scale pretraining, training efficiency can be important. In contrastive learning, more negative instances often lead to better performance (Giorgi et al., 2021; Wu et al., 2020; Gao et al., 2021b). However, traditional in-batch negative sampling is bottlenecked by limited

*Equal contribution.
[1]Code and pretrained weights can be found at https://github.com/JetRunner/LaPraDoR.

GPU memory. To alleviate this problem, we propose Iterative Contrastive Learning (ICoL), which iteratively trains the query and document encoders with a cache mechanism. Compared to existing solutions MoCo (He et al., 2020) and xMoCo (Yang et al., 2021), ICoL does not introduce extra encoders and can solve the mismatching between representation spaces, thus demonstrating superior performance. **(2) Versatility.** There are different types of downstream tasks from various domains in both BEIR and real-world applications. We use a large-scale multi-domain corpus, C4 (Raffel et al., 2020), to train our LaPraDoR model. To make LaPraDoR versatile, besides conventional query-document retrieval, we also incorporate document-query, query-query, and document-document retrieval into the pretraining objective. We further share the weights between the query and document encoders and obtain an all-around encoder that fits all retrieval tasks.

To summarize, our contribution is three-fold: (1) We train LaPraDoR, an all-around unsupervised pretrained dense retriever that achieves state-of-the-art performance on the BEIR benchmark. (2) We propose Iterative Contrastive Learning (ICoL) for training a retrieval model effectively. (3) We propose Lexicon-Enhanced Dense Retrieval as an efficient way for combining BM25 with a dense retriever, compared to the widely-used re-ranking paradigm.

2 Related Work

Dense Retrieval DPR (Karpukhin et al., 2020) initializes a bi-encoder model with BERT (Devlin et al., 2019) and achieves better results than earlier dense retrieval methods. RocketQA (Qu et al., 2021) exploits a trained retriever to mine hard negatives and then re-train a retriever with the mined negatives. ANCE (Xiong et al., 2021) dynamically mines hard negatives throughout training but requires periodic encoding of the entire corpus. TAS-B (Hofstätter et al., 2021) is a bi-encoder trained with balanced topic-aware sampling and knowledge distillation from a cross-encoder and a ColBERT model (Khattab and Zaharia, 2020), in addition to in-batch negatives. xMoCo (Yang et al., 2021) adapt MoCo (He et al., 2020), a contrastive learning algorithm that is originally proposed for image representation, to text retrieval by doubling its fast and slow encoders. Although these dense retrieval systems demonstrate effectiveness on some datasets, the BEIR benchmark (Thakur et al., 2021) highlights a main drawback of these dense retrieval systems - failure to generalize to out-of-domain data. This motivates pretraining as a solution for better domain generalization (Gururangan et al., 2020). Dense retrieval has also been applied in many other tasks (Guo et al., 2019, 2020).

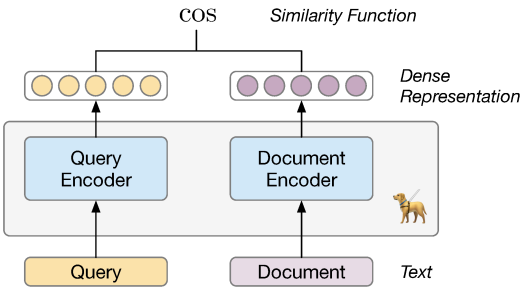


Figure 1: Dual-tower architecture for text retrieval.

Pretraining for Retrieval Lee et al. (2019) first propose to pretrain a bi-encoder retriever with an Inverse Cloze Task (ICT), which constructs a training pair by randomly selecting a sentence from a passage as the query and leaving the rest as the document. Chang et al. (2020) propose two pretraining tasks for Wikipedia and attempt to combine them with ICT and masked language modeling (MLM). Guu et al. (2020) pretrain a retriever and a reader together for end-to-end question answering (QA). Very recently, DPR-PAQ (Oğuz et al., 2021) highlight the importance of domain matching by using both synthetic and crawled QA data to pretrain and then fine-tune the model on downstream datasets for dialogue retrieval. Condenser (Gao and Callan, 2021a) is a new Transformer variant for MLM pretraining. It exploits an information bottleneck to facilitate learning for information aggregation. On top of that, coCondenser (Gao and Callan, 2021b) adds an unsupervised corpus-level contrastive loss to warm up the passage embedding space. Different from these works, LaPraDoR is the first pretrained retriever that does not require fine-tuning on a downstream dataset and can perform zero-shot retrieval.

3 Methodology

3.1 Dual-Tower Architecture

Two Encoders The dual-tower architecture, as illustrated in Figure 1, is widely used in dense retrieval systems (Lee et al., 2019; Karpukhin et al., 2020; Xiong et al., 2021). The dual-tower archi-

tecture has a query encoder E_Q and a document encoder E_D, which in our work are both BERT-like bidirectional text encoders (Devlin et al., 2019). Compared with cross-attention models (Reimers and Gurevych, 2019; Gao et al., 2020; MacAvaney et al., 2020), the dual-tower architecture enables pre-indexing and fast approximate nearest neighbor search (to be detailed shortly), thus is popular in production.

Dense Representation Given an input document (query) $\boldsymbol{x} = \{\texttt{[CLS]}, w_1, \ldots, w_l, \texttt{[SEP]}\}$, we use a document (query) encoder E_D (E_Q) to encode the input sequence into hidden states $\boldsymbol{h} = \{v_{\texttt{[CLS]}}, v_1, \ldots, v_l, v_{\texttt{[SEP]}}\}$, where w_i is the i-th token; `[CLS]` and `[SEP]` are special tokens that mark the start and end of a sentence, respectively. To obtain a dense representation, we use mean pooling over hidden states $\boldsymbol{h}$ as the representation $\boldsymbol{h_x}$ of the input $\boldsymbol{x}$. Some prior works (Lee et al., 2019; Chang et al., 2020; Karpukhin et al., 2020) use $v_{\texttt{[CLS]}}$ as the representation for the input $\boldsymbol{x}$, but Huang et al. (2021) empirically find that applying mean pooling to hidden states $\boldsymbol{h}$ outperforms taking $v_{\texttt{[CLS]}}$ as the representation.

Similarity Function After obtaining the representation for both the query q and the document d, we use the cosine function as a similarity function to measure the similarity between them:

$$\text{sim}(q, d) = \frac{E_Q(q) \cdot E_D(d)}{\|E_Q(q)\| \|E_D(d)\|} \quad (1)$$

Approximate Nearest Neighbor In practice, for the dual-tower architecture, the documents are encoded offline and their dense representations can be pre-indexed by a fast vector similarity search library (e.g., FAISS, Johnson et al., 2021). The library can utilize GPU acceleration to perform approximate nearest neighbor (ANN) search in sublinear time with almost no loss in recall. Thus, compared to a cross-encoder (i.e., an encoder that accepts the concatenation of the query and every candidate document), a pre-indexed ANN-based retrieval system is at least 10 times faster (to be detailed in Section 4.2).

3.2 Constructing Positive Instances

In this section, we first introduce how we build the positive instances with two self-supervised tasks, namely Inverse Cloze Task (ICT) and Dropout as Positive Instance (DaPI).

Inverse Cloze Task (ICT) First introduced in Lee et al. (2019), ICT is an effective way to pretrain a text retrieval model (Chang et al., 2020). Given a passage p consisting of sentences $p = \{s_1, \ldots, s_n\}$, we randomly select a sentence s_k as query q and treat its context as document $d = \{s_1, \ldots, s_{k-1}, s_{k+1}, \ldots, s_n\}$. ICT is designed to mimic a text retrieval task where a short query is used to retrieve a longer document which is semantically relevant. Also, unlike some pretraining tasks, e.g., Wiki Link Prediction or Body First Selection (Chang et al., 2020), ICT is fast and does not rely on a specific corpus format (e.g., Wikipedia) thus can be scaled to a large multi-source corpus (e.g., C4, Raffel et al., 2020).

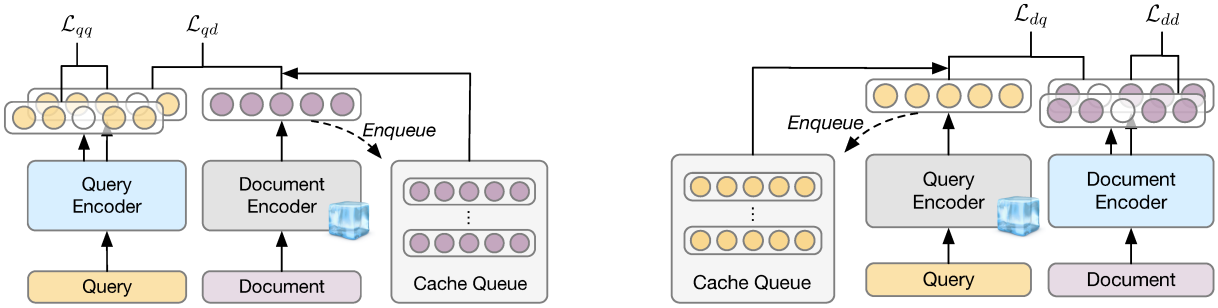
Dropout as Positive Instance (DaPI) DaPI is originally proposed in SimCSE (Gao et al., 2021c) as a simple strategy for perturbing intermediate representations and thus can serve as data augmentation.[2] A similar idea is also presented in Liu et al. (2021). We apply a dropout rate of 0.1 to the fully-connected layers and attention probabilities in the Transformer encoders, as in BERT (Devlin et al., 2019). The same input is fed to the encoder twice to obtain two representations, of which one is used as the positive instance of the other. Gao et al. (2021c) conduct experiments and conclude that the dropout strategy outperforms all commonly-used discrete perturbation techniques including cropping, word deletion, masked language modeling and synonym replacement. Note that different from SimCSE, we only calculate gradients for one of the two passes. In our experiments, we find that the addition of DaPI only increases the memory use by 2%, since it mostly reuses the computational graph for the ICT objective.

3.3 Iterative Contrastive Learning

Previous studies (Giorgi et al., 2021; Wu et al., 2020; Gao et al., 2021b) show that the number of negative instances is critical to the performance of the model. Since the batch size on a single GPU is limited, we propose Iterative Contrastive Learning (ICoL) to mitigate the insufficient memory on a single GPU and allow more negative instances for better performance. We illustrate LaPraDoR training in Figure 2.

Iterative Training We iteratively train the query encoder and document encoder. To be specific, we

[2] To avoid confusion with the SimCSE model, we address the dropout strategy as DaPI here.



(a) Query encoder training.

(b) Document encoder training.

Figure 2: Training of LaPraDoR with Iterative Contrastive Learning (ICoL). We iteratively train the query encoder and document encoder while freezing the other (marked with an ice cube icon). For $\mathcal{L}_{qd}$ and $\mathcal{L}_{dq}$, we obtain additional negative instances from the cache queue. For each batch of data, we enqueue the representation encoded by the frozen encoder into the cache queue as future negative instances. The cache queue is cleared when switching the encoder to train from one to the other.

first arbitrarily select an encoder to start training. Here we assume to start with the query encoder E_Q. The training loss consists of two terms. First, we calculate the loss for query-query retrieval with DaPI to optimize the negative log likelihood of the positive instance:

$$\begin{aligned}&\mathcal{L}_{qq}(q_i, \{q_i^+, q_{i,1}^-, \dots, q_{i,n}^-\}) \\ &= -\log \frac{e^{\text{sim}(q_i, q_i^+)}}{e^{\text{sim}(q_i, q_i^+)} + \sum_{j=1}^{n} e^{\text{sim}(q_i, q_{i,j}^-)}}\end{aligned} \tag{2}$$

where q_i and q_i^+ are the same query that are encoded by E_Q with different dropout masks; $\{q_{i,1}^-, ..., q_{i,n}^-\}$ is a set of randomly sampled negative instances; $\text{sim}(\cdot, \cdot)$ is the cosine similarity function defined in Equation 1.

The second term is to retrieve the corresponding document d_i^+ with the query q_i, where q_i and d_i^+ are a pair constructed with ICT. Similarly, we optimize the negative log likelihood of the positive instance by:

$$\begin{aligned}&\mathcal{L}_{qd}(q_i, \{d_i^+, d_{i,1}^-, \dots, d_{i,n}^-, d_{\mathcal{Q},1}^-, \dots, d_{\mathcal{Q},|\mathcal{Q}|}^-\}) \\ &= -\log \frac{e^{\text{sim}(q_i, d_i^+)}}{e^{\text{sim}(q_i, d_i^+)} + \sum_{j=1}^{n} e^{\text{sim}(q_i, d_{i,j}^-)} + \sum_{k=1}^{|\mathcal{Q}|} e^{\text{sim}(q_i, d_{\mathcal{Q},k}^-)}}\end{aligned} \tag{3}$$

where $\{d_{i,1}^-, ..., d_{i,n}^-\}$ is a set of freshly sampled documents that are encoded at the current step i; $\{d_{\mathcal{Q},1}^-, ..., d_{\mathcal{Q},|\mathcal{Q}|}^-\}$ is a set of representations that are currently stored in the cache queue $\mathcal{Q}$. Then, we optimize the sum of the two losses with a weight coefficient λ:

$$\mathcal{L}_q = \mathcal{L}_{qd} + \lambda \mathcal{L}_{qq} \tag{4}$$

Note that the query q_i only needs to be encoded once and can be used for calculation of both $\mathcal{L}_{qd}$ and $\mathcal{L}_{qq}$.

After a predefined number of steps, the E_Q becomes frozen as the training for E_D starts. Similarly, for d_i, a document encoded by E_D, we have the training objective:

$$\begin{aligned}&\mathcal{L}_{dd}(d_i, \{d_i^+, d_{i,1}^-, \dots, d_{i,n}^-\}) \\ &= -\log \frac{e^{\text{sim}(d_i, d_i^+)}}{e^{\text{sim}(d_i, d_i^+)} + \sum_{j=1}^{n} e^{\text{sim}(d_i, d_{i,j}^-)}}\end{aligned} \tag{5}$$

$$\begin{aligned}&\mathcal{L}_{dq}(d_i, \{q_i^+, q_{i,1}^-, \dots, q_{i,n}^-, q_{\mathcal{Q},1}^-, \dots, q_{\mathcal{Q},|\mathcal{Q}|}^-\}) \\ &= -\log \frac{e^{\text{sim}(d_i, q_i^+)}}{e^{\text{sim}(d_i, q_i^+)} + \sum_{j=1}^{n} e^{\text{sim}(d_i, q_{i,j}^-)} + \sum_{k=1}^{|\mathcal{Q}|} e^{\text{sim}(d_i, q_{\mathcal{Q},k}^-)}}\end{aligned} \tag{6}$$

$$\mathcal{L}_d = \mathcal{L}_{dq} + \lambda \mathcal{L}_{dd} \tag{7}$$

where d_i^+ and q_i^+ are positive instances constructed by DaPI and ICT, respectively; $\{d_{i,1}^-, \dots, d_{i,n}^-\}$ is a set of randomly sampled document negatives; $\{q_{i,1}^-, \dots, q_{i,n}^-\}$ is a set of freshly sampled queries encoded at step i; $\{q_{\mathcal{Q},1}^-, \dots, q_{\mathcal{Q},|\mathcal{Q}|}^-\}$ are the cached query representations. To speed up training, we apply the in-batch negatives technique (Yih et al., 2011; Henderson et al., 2017; Gillick et al., 2019) that can reuse computation and train b queries/documents in a mini-batch simultaneously.

Cache Mechanism To enlarge the size of negative instances, we maintain a cache queue $\mathcal{Q}$ that

stores previously encoded representations that can serve as negative instances for the current step, extending an earlier study (Wu et al., 2018). Our cache queue is implemented as first-in-first-out (FIFO) with a maximum capacity m, which is a hyperparameter set based on the GPU memory size. When training with multiple GPUs, $\mathcal{Q}$ can be shared across GPUs. Since the representations in the queue are encoded with a frozen encoder and thus do not require gradients, m can be set large to supplement the numbers of negative instances. When $\mathcal{Q}$ is full, the earliest cached representations will be dequeued. When we switch the training from one encoder to the other, the queue will be cleared to ensure that all representations in $\mathcal{Q}$ lie in the same hidden space and are encoded with the currently frozen encoder.

ICoL vs. MoCo Previously, similar to our method, MoCo (He et al., 2020) exploits a queue for storing encoded representations. Specifically, MoCo consists of a slow encoder and a fast encoder to encode queries and documents, respectively. The slow encoder is updated as a slow moving average of the fast encoder to reduce inconsistency of encoded document representations between training steps. A queue is maintained to allow the encoded document representations to be reused in later steps as negative instances.

However, we argue there are a two limitations that make MoCo not ideal for training a text retrieval model: (1) As pointed out by Yang et al. (2021), unlike the image matching task in the original paper of MoCo, in text retrieval, the queries and documents are distinct from each other thus not interchangeable. Yang et al. (2021) propose xMoCo, which incorporates two sets of slow and fast encoders, as a simple fix for this flaw. (2) The cached representations are in different hidden spaces. Although the fast encoders in both MoCo and xMoCo are updated with momentum, the already-encoded representations in the queue will never be updated. This creates a semantic mismatch between newly encoded and cached old representations and creates noise during training. In ICoL, all representations used for contrastive learning are aligned in the same hidden space. Besides, ICoL is more flexible than xMoCo since it does not introduce additional fast encoders and even the weights of its query encoder and document encoder can be shared. We conduct experiments to compare ICoL with MoCo and xMoCo in Section 4.2.1.

3.4 Lexicon-Enhanced Dense Retrieval

Although dense retrieval achieves state-of-the-art performance, its performance significantly degenerates on out-of-domain data (Thakur et al., 2021). On the other hand, BM25 (Robertson and Zaragoza, 2009) demonstrates good performance without training. Early attempts at combining lexical match with dense retrieval often formulate it to a re-ranking task (Nguyen et al., 2016). First, BM25 is used to recall the top-k documents from the corpus. Then, a cross-encoder is applied to further re-rank candidate documents. Recently, COIL (Gao et al., 2021a) highlights the importance of lexical match and incorporates exact lexical matching into dense retrieval. Different from these works, we propose a fast and effective way, namely Lexicon-Enhanced Dense Retrieval (LEDR) to enhance dense retrieval with BM25. The similarity score of BM25 is defined as:

$$\begin{aligned}\mathrm{BM25}(q,d) &= \sum_{t\in q\cap d} \mathrm{IDF}(t) h_q(q,t) h_d(d,t) \\ h_q(q,t) &= \frac{\mathrm{TF}_{t,q}\,(1+k_2)}{\mathrm{TF}_{t,q}+k_2} \\ h_d(d,t) &= \frac{\mathrm{TF}_{t,d}\,(1+k_1)}{\mathrm{TF}_{t,d}+k_1\left(1-b+b\frac{|d|}{\mathrm{avgdl}}\right)}\end{aligned} \tag{8}$$

where $\mathrm{TF}_{t,d}$ and $\mathrm{TF}_{t,q}$ refer to term frequency of term t in document d and query q, respectively; $\mathrm{IDF}(t)$ is the inverse document frequency; b, k_1 and k_2 are hyperparameters. For inference, we simply multiply the BM25 score with the similarity score for dense retrieval:

$$score(q,d) = \mathrm{sim}(q,d) \times \mathrm{BM25}(q,d) \tag{9}$$

In this way, we consider both lexical and semantic matching. This combination makes LaPraDoR more robust on unseen data in zero-shot learning.

4 Experiments

4.1 Experimental Setting

Benchmark We use BEIR (Thakur et al., 2021), a recently released benchmark for zero-shot evaluation of information retrieval models. BEIR includes 18 heterogeneous datasets, focusing on evaluating a retrieval system that works across different domains (bio-medical, scientific, news, social media, etc.). The benchmark uses Normalized Discounted Cumulative Gain (nDCG) (Järvelin and Kekäläinen, 2002) as the evaluation metric, which is a measure

Model		Dense Retrieval				Lexical	Late Interaction	Re-ranking	Lexicon-Enhanced Dense	
		DPR	ANCE	GenQ	TAS-B	BM25†	ColBERT	BM25 + CE	LaPraDoR†	LaPraDoR FT
Encoding Speed	Qry/s (GPU/CPU)	4000/170	4000/170	4000/170	7000/350	-	4000/170	7000/350	7000/350	7000/350
	Doc/s (GPU/CPU)	540/30	540/30	540/30	1100/70	-	540/30	1100/70	1100/70	1100/70
Index size		3 GB	3 GB	3 GB	3 GB	0.4 GB	20 GB	0.4 GB	3.4 GB	3.4 GB
Retrieval Latency	GPU	19 ms	20 ms	14 ms	14 ms	-	350 ms	450 ms	20 ms	20 ms
	CPU	230 ms	275 ms	125 ms	125 ms	20 ms	-	6100 ms	145 ms	145 ms
MS-MARCO	nDCG@10	0.177	0.388	0.408	0.408	0.228	0.401	0.413	0.262	0.366
Zero-shot (nDCG@10)	TREC-COVID	0.332	0.654	0.619	0.481	0.656	0.677	0.757	0.728	**0.779**
	BIOASQ	0.127	0.306	0.398	0.383	0.465	0.474	**0.523**	0.500	0.511
	NFCorpus	0.189	0.237	0.319	0.319	0.325	0.305	**0.350**	0.346	0.347
	NQ	0.474	0.446	0.358	0.463	0.329	0.524	**0.533**	0.359	0.479
	HotpotQA	0.391	0.456	0.534	0.584	0.603	0.593	**0.707**	0.625	0.666
	FiQA	0.112	0.295	0.308	0.300	0.236	0.317	**0.347**	0.317	0.343
	Signal-1M	0.155	0.249	0.281	0.289	0.330	0.274	0.338	0.343	**0.344**
	TREC-NEWS	0.161	0.382	0.396	0.377	0.398	0.393	0.431	0.470	**0.480**
	Robust04	0.252	0.392	0.362	0.427	0.408	0.391	0.475	**0.490**	0.484
	ArguAna	0.175	0.415	0.493	0.429	0.315	0.232	0.311	0.507	**0.508**
	Touche-2020	0.131	0.240	0.182	0.162	0.367	0.202	0.271	0.322	**0.333**
	CQADupStack	0.153	0.296	0.347	0.314	0.299	0.350	**0.370**	0.222	0.290
	Quora	0.248	0.852	0.830	0.835	0.789	0.854	0.825	0.863	**0.875**
	DBPedia	0.263	0.281	0.328	0.384	0.313	0.392	**0.409**	0.361	0.391
	SCIDOCS	0.077	0.122	0.143	0.149	0.158	0.145	0.166	**0.185**	0.184
	FEVER	0.562	0.669	0.669	0.700	0.753	0.771	**0.819**	0.671	0.763
	Climate-FEVER	0.148	0.198	0.175	0.228	0.213	0.184	0.253	0.228	**0.261**
	SciFact	0.318	0.507	0.644	0.643	0.665	0.671	0.688	**0.697**	0.687
	Avg.	0.237	0.389	0.410	0.415	0.423	0.431	0.476	0.457	**0.485**

Table 1: Experimental results on the BEIR benchmark (Thakur et al., 2021). The estimated average retrieval latency and index sizes are for a single query in DBPedia. The encoding speed is reported on a 8-core Intel Xeon Platinum 8168 CPU @ 2.70GHz and a single Nvidia V100 GPU, respectively. "LaPraDoR FT" is a LaPraDoR model fine-tuned on MS-MARCO with the official BEIR training script. †Unsupervised method.

of ranking quality and often used to measure effectiveness of search algorithms or retrieval models. Details of the BEIR benchmark and the evaluation metric are included in Appendix A.

Model Settings In our preliminary experiments on Wikipedia (see Table 2), we find that sharing weights between the query encoder E_Q and document encoder E_D has no negative effect on downstream performance. For weight sharing between E_Q and E_D, we simply copy the weights of E_Q to E_D when switching to training of E_D, vice versa. This design eliminates nearly half of the parameters. An additional benefit is that weight sharing makes the encoder versatile to handle not only query-document retrieval, but also query-query and document-document retrieval.

In our preliminary experiments on Wikipedia, we observed a diminishing return when increasing the model size from 6 layers to 12 layers, or 24 layers. Thus, we initialize our encoder with the 6-layer DistilBERT (Sanh et al., 2019), which has ∼67M parameters. For BM25, we use the implementation and default settings of Elastic Search[3]. BM25 scores after the top 1,000 retrieved text are set to 0 to save computation.

Training Details For pretraining, we optimize the model with the AdamW optimizer with a learning rate of 2e-4. The model is trained with 16 Nvidia V100 32GB GPUs with FP16 mixed precision training. The batch size for each GPU is set to 256. The maximum lengths set for queries and documents are 64 and 350, respectively. Training switches between E_Q and E_D every 100 steps. The cache queue has a maximum capacity m of 100k. The loss weight hyperparameter λ is fixed to 1. For our main results, we train LaPraDoR on C4 (Raffel et al., 2020) for 1M steps, which takes about 400 hours. For the ablation study, since training on C4 is very costly, we train LaPraDoR on Wikipedia[4] for 100k steps. When calculating the loss, we apply a re-scaling trick of multiplying the cosine similarity score by 20 for better optimization (Thakur et al., 2021). Our implementation of LaPraDoR is based on Hugging Face Transformers (Wolf et al., 2020) and Datasets (Lhoest et al., 2021).

We test LaPraDoR under two settings: **(1) No supervised data at all.** We directly use the pretrained model for zero-shot retrieval on BEIR. **(2) Fine-**

[3] https://github.com/elastic/elasticsearch

[4] https://huggingface.co/datasets/wikipedia

Model		In-Batch (shared)	MoCo	xMoCo	ICoL	ICoL (shared)
#Encoder		1	2	4	2	1
MS-MARCO	nDCG@10	0.255	0.222	0.255	0.255	**0.262**
Zero-shot (nDCG@10)	TREC-COVID	0.705	0.537	**0.724**	0.706	0.710
	BIOASQ	0.451	0.260	0.423	**0.468**	0.459
	NFCorpus	0.315	0.271	0.312	**0.317**	0.314
	NQ	0.332	0.279	**0.355**	**0.355**	0.351
	HotpotQA	0.599	0.552	0.584	0.598	**0.610**
	FiQA	0.213	0.156	0.242	**0.256**	0.251
	Signal-1M	0.329	0.307	0.323	0.327	**0.335**
	TREC-NEWS	0.441	0.405	0.441	0.444	**0.445**
	Robust04	0.419	0.439	0.439	0.465	**0.470**
	ArguAna	0.477	0.465	0.491	0.496	**0.503**
	Touche-2020	0.302	0.261	0.330	**0.331**	0.328
	CQADupStack	0.109	0.052	0.118	0.132	**0.140**
	Quora	0.832	0.834	0.822	0.828	**0.839**
	DBPedia	0.349	0.318	0.359	**0.374**	0.364
	SCIDOCS	0.173	0.154	0.170	0.173	**0.178**
	FEVER	0.537	0.540	0.651	**0.686**	0.653
	Climate-FEVER	0.206	0.183	**0.244**	0.242	0.242
	SciFact	0.660	0.659	0.667	0.683	**0.689**
	Avg.	0.414	0.371	0.428	**0.438**	**0.438**

Table 2: Comparison of different methods for contrastive learning. The models are trained on Wikipedia.

Model	LaPraDoR		LaPraDoR FT			
	Full	w/o LEDR	Full	w/o LEDR	w/o PT	w/o LEDR & PT
TREC-COVID	**0.728**	0.227	**0.779**	0.492	0.735	0.482
BIOASQ	**0.500**	0.205	**0.511**	0.308	0.489	0.281
NFCorpus	**0.346**	0.311	**0.347**	0.335	0.323	0.267
NQ	**0.359**	0.181	**0.479**	0.473	0.454	0.443
HotpotQA	**0.625**	0.303	**0.666**	0.495	0.642	0.484
FiQA	**0.317**	0.203	**0.343**	0.314	0.308	0.245
Signal-1M	**0.343**	0.186	0.344	0.231	**0.354**	0.247
TREC-NEWS	**0.470**	0.345	**0.480**	0.374	0.449	0.350
Robust04	**0.490**	0.319	**0.484**	0.368	0.459	0.332
ArguAna	**0.507**	0.459	**0.508**	0.469	0.495	0.412
Touche-2020	**0.322**	0.094	0.333	0.182	**0.346**	0.156
CQADupStack	**0.222**	0.220	0.290	0.288	**0.306**	0.250
Quora	**0.863**	0.787	**0.875**	0.847	0.867	0.840
DBPedia	**0.361**	0.250	**0.391**	0.338	0.384	0.303
SCIDOCS	**0.185**	0.133	**0.184**	0.155	0.173	0.127
FEVER	**0.671**	0.368	**0.763**	0.646	0.750	0.664
Climate-FEVER	**0.228**	0.138	**0.261**	0.209	0.247	0.206
SciFact	**0.697**	0.555	**0.687**	0.599	0.678	0.529
Avg.	**0.457**	0.294	**0.485**	0.396	0.470	0.368

Table 3: Effect of pretraining (PT) and Lexicon-Enhanced Dense Retrieval (LEDR). Pretraining is on C4. The results of "w/o PT" directly use DistilBERT (Sanh et al., 2019) for fine-tuning, which is also used to initialize our model.

tuning on MS-MARCO (Nguyen et al., 2016) and zero-shot transfer to the other datasets. This is the original setting for BEIR. We use BEIR's official script[5] to fine-tune LaPraDoR. The batch size is set to 75 per GPU and the learning rate is 2e-5.

Baselines For dense retrieval, we compare our model to the dual-tower models: DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2021), TAS-B (Hofstätter et al., 2021) and GenQ (Thakur et al., 2021). For lexical matching, we use the BM25 results reported in Thakur et al. (2021). We also consider a late interaction baseline **ColBERT** (Khattab and Zaharia, 2020). The model computes multiple contextualized embeddings for each token of queries and documents, and then maximizes a similarity function to retrieve relevant documents. For re-ranking, we use the **BM25+CE** baseline implemented in Thakur et al. (2021) that uses BM25 to retrieve top-100 documents and a cross-encoder model to further re-rank. As shown in Table 1, the latency for both lexical and dense retrieval is low whereas re-ranking introduces significantly higher latency, with late-interaction in-between. Details of the baselines can be found in Appendix B.

4.2 Experimental Results

We list the results of LaPraDoR on the BEIR benchmark in Table 1. Our model achieves state-of-the-art performance on BEIR to date (November 15, 2021). Without any supervised data, LaPraDoR outperforms the previous state-of-the-art for zero-shot dense retrieval, TAS-B (Hofstätter et al., 2021), on 13 tasks (out of 18) of BEIR with an average advantage of 0.042, though TAS-B applies additional query clustering and knowledge distillation. When further fine-tuned on MS-MARCO, LaPraDoR can outperform all baselines, including late interaction and re-ranking, whose latency on GPU is 17.5× and 22.5× higher than our method. Compared to dense retrieval, we only add 0.4 GB of BM25 indices and almost no additional latency.

4.2.1 Effect of Iterative Contrastive Learning

We set a baseline that only uses in-batch negatives and compare our proposed Iterative Contrastive Learning (ICoL) to MoCo (He et al., 2020) and xMoCo (Yang et al., 2021) for training LaPraDoR on Wikipedia in Table 2. The aforementioned two flaws of MoCo hinder its performance and lead to a performance drop instead of an improvement. In contrast, our ICoL approach outperforms the in-batch baseline on all datasets. It also beats the competitive MoCo variant for text retrieval, xMoCo, on 15 out of 18 tasks. ICoL only uses two encoders (which can be further shared) which can alleviate the GPU memory problem and thus can fit more in-batch negatives. Meanwhile, MoCo uses two encoders and xMoCo uses four (two sets of MoCo's encoders). Moreover, we observe no performance drop on average if we share the encoder between query and document (as we do when training LaPraDoR on C4). Thus, we can eliminate half of the parameters by simply sharing the encoder.

[5] https://github.com/UKPLab/beir/blob/main/examples/retrieval/training/train_msmarco_v3.py

Model	LaPraDoR	w/o DaPI	w/o ICT
TREC-COVID	0.710	**0.714**	0.612
BIOASQ	**0.459**	0.457	0.270
NFCorpus	0.314	**0.316**	0.257
NQ	0.351	**0.353**	0.221
HotpotQA	**0.610**	0.608	0.431
FiQA	**0.251**	0.247	0.145
Signal-1M	**0.335**	0.330	0.306
TREC-NEWS	0.445	**0.448**	0.336
Robust04	**0.470**	0.458	0.307
ArguAna	**0.503**	0.497	0.389
Touche-2020	**0.328**	0.310	0.248
CQADupStack	**0.140**	0.137	0.064
Quora	**0.839**	0.839	0.774
DBPedia	**0.364**	0.363	0.242
SCIDOCS	**0.178**	0.173	0.113
FEVER	**0.653**	0.639	0.376
Climate-FEVER	**0.242**	0.231	0.118
SciFact	0.689	**0.690**	0.533
Avg.	**0.438**	0.434	0.319

Table 4: Effect of ICT and DaPI in the loss function. The "w/o ICT" variant is equal to the original SimCSE approach (Gao et al., 2021c). The pretraining is on Wikipedia.

4.2.2 Effect of Pretraining and Lexicon-Enhanced Dense Retrieval

We conduct an ablation study for both pretraining and Lexicon-Enhanced Dense Retrieval to verify the effectiveness of these designs. As shown in Table 3, Lexicon-Enhanced Dense Retrieval (LEDR) improves performance of dense retrieval on most tasks for both fully unsupervised and fine-tuned LaPraDoR. Furthermore, as illustrated in Table 4, we test the effectiveness of the two components in our loss function. We can see that both ICT and DaPI significantly contribute to the performance of our model ($p < 0.01$) while ICT has a large impact on the final performance.

4.3 Case Study

We conduct a case study to intuitively demonstrate the effectiveness of LaPraDoR. As shown in Figure 3, for Q1, the lexical method (i.e., BM25) can successfully find the corresponding document in its top-2 retrieved results. However, due to lower lexical overlap, the score of the ground truth is lower than that of the first document. Although the phrase *"prepare for his departure"* in the first document indicates that *Aeneas has not left Carchage yet* and provides strong evidence that this document is incorrect, BM25 fails to correctly rank the ground truth due to its lack of ability in semantic matching. By incorporating both lexical and semantic matching, LaPraDoR can successfully retrieve the ground truth.

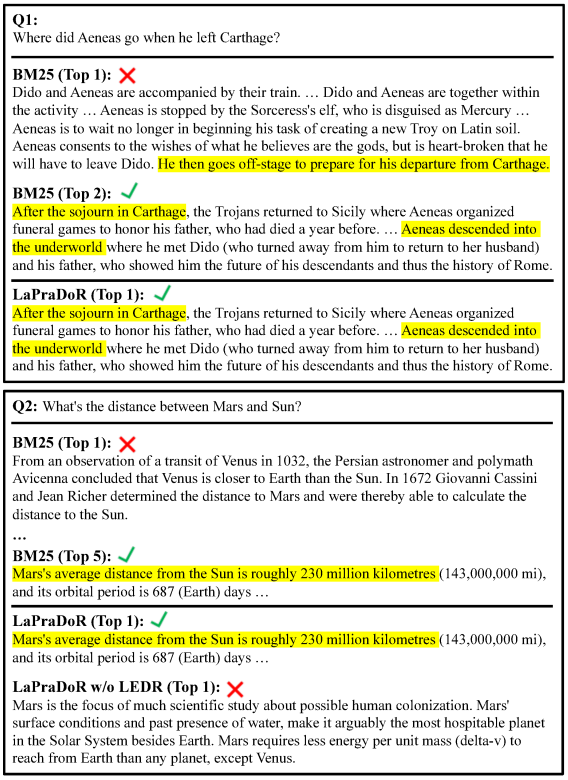


Figure 3: Examples from the NQ dataset (Kwiatkowski et al., 2019). The key clues are highlighted.

For Q2, with the powerful semantic matching, LaPraDoR successfully retrieves the ground truth whereas BM25 fails to distinguish among the documents that contain both the keywords *Mars* and *Sun*. On the other hand, after removing lexical matching, LaPraDoR without LEDR suffers from noise: the key entity *Sun* does not appear in its top-1 retrieved document. LEDR helps filter out such noise and allows the dense retriever to focus on fine-grained semantic matching. Please find more cases from other datasets on Appendix C.

5 Conclusion and Future Work

In this paper, we introduce LaPraDoR, an unsupervised pretrained dense retriever that achieves state-of-the-art performance on the zero-shot text retrieval benchmark BEIR. We propose Iterative Contrastive Learning (ICoL) for efficiently training LaPraDoR and Lexicon-Enhanced Dense Retrieval (LEDR) to combine lexical matching with LaPraDoR. Our experiments verify the effectiveness of both ICoL and LEDR, shedding light on a new paradigm for unsupervised text retrieval. For future work, we plan to extend unsupervised LaPraDoR to multilingual and multi-modal retrieval.

Broader Impact

Ethical Concerns LaPraDoR is trained with web-crawled data, which may contain inappropriate content. However, due to the nature of text retrieval, our retriever has lower ethical risk compared to a generative auto-regressive language model (Bender et al., 2021). Meanwhile, our unsupervised retrieval model enables high-performance text retrieval for low-resource languages where there is no supervised query-document dataset. This contributes to equality and diversity of language technology.

Carbon Footprint To conduct all experiments in this paper, we estimate to have consumed 3,840 kWh of electricity and emitted 1,420.8 kg (3,132.3 lbs) of CO_2. All emitted carbon dioxide has already been offset by the cloud service provider.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments. We would like to thank the authors of BEIR (Thakur et al., 2021), Nandan Thakur and Nils Reimers, for their support. Canwen wants to thank Minghua Liu's Labrador, Jojo, for the inspiration to name this paper. This project is partly supported by NSF Award #1750063.

References

J Allan. 2004. Overview of the trec 2004 robust retrieval track. In *TREC*, volume 13.

Petr Baudis and Jan Sedivý. 2015. Modeling of the question answering task in the yodaqa system. In *CLEF*, volume 9283 of *Lecture Notes in Computer Science*, pages 222–228. Springer.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *FAccT*, pages 610–623. ACM.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, pages 1533–1544. ACL.

Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. 2020. Overview of Touché 2020: Argument Retrieval. In *Working Notes Papers of the CLEF 2020 Evaluation Labs*, volume 2696 of *CEUR Workshop Proceedings*.

Vera Boteva, Demian Gholipour Ghalandari, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *ECIR*, volume 9626 of *Lecture Notes in Computer Science*, pages 716–722. Springer.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *ICLR*. OpenReview.net.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. SPECTER: document-level representation learning using citation-informed transformers. In *ACL*, pages 2270–2282. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims.

Luyu Gao and Jamie Callan. 2021a. Condenser: a pre-training architecture for dense retrieval. *arXiv preprint arXiv:2104.08253*.

Luyu Gao and Jamie Callan. 2021b. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2020. Modularized transfomer-based ranking framework. In *EMNLP*, pages 4180–4190. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021a. COIL: revisit exact lexical match in information retrieval with contextualized inverted list. In *NAACL-HLT*, pages 3030–3042. Association for Computational Linguistics.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021b. Scaling deep contrastive learning batch size under memory limited setup. In *The 6th Workshop on Representation Learning for NLP (RepL4NLP)*, pages 316–321.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021c. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego García-Olano. 2019. Learning dense representations for entity retrieval. In *CoNLL*, pages 528–537. Association for Computational Linguistics.

John M. Giorgi, Osvald Nitski, Bo Wang, and Gary D. Bader. 2021. Declutr: Deep contrastive learning for unsupervised textual representations. In *ACL-IJCNLP*, pages 879–895. Association for Computational Linguistics.

Daya Guo, Duyu Tang, Nan Duan, Jian Yin, Daxin Jiang, and Ming Zhou. 2020. Evidence-aware inferential text generation with vector quantised variational autoencoder. In *ACL*, pages 6118–6129. Association for Computational Linguistics.

Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2019. Coupling retrieval and meta-learning for context-dependent semantic parsing. In *ACL*, pages 855–866. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *ACL*, pages 8342–8360. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. Dbpedia-entity v2: A test collection for entity search. In *SIGIR*, pages 1265–1268. ACM.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735. Computer Vision Foundation / IEEE.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

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 *SIGIR*, pages 113–122. ACM.

Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2021. Improving efficient neural ranking models with cross-architecture knowledge distillation.

Doris Hoogeveen, Karin M Verspoor, and Timothy Baldwin. 2015. Cqadupstack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian document computing symposium*, pages 1–8.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. Whiteningbert: An easy unsupervised sentence embedding approach. *arXiv preprint arXiv:2104.01767*.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Trans. Big Data*, 7(3):535–547.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, pages 1601–1611. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*, pages 6769–6781. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *SIGIR*, pages 39–48. ACM.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *ACL*, pages 6086–6096. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Sasko, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *EMNLP (Demos)*, pages 175–184. Association for Computational Linguistics.

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021. Fast, effective and self-supervised:

Transforming masked languagemodels into universal lexical and sentence encoders. *arXiv preprint arXiv:2104.08027*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Efficient document re-ranking for transformers by precomputing term representations. In *SIGIR*, pages 49–58. ACM.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. WWW'18 open challenge: Financial opinion mining and question answering. In *WWW (Companion Volume)*, pages 1941–1942. ACM.

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. In *CoCo@NIPS*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

Barlas Oğuz, Kushal Lakhotia, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen-tau Yih, Sonal Gupta, et al. 2021. Domain-matched pretraining tasks for dense retrieval. *arXiv preprint arXiv:2107.13602*.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *NAACL-HLT*, pages 5835–5847. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*, pages 3980–3990. Association for Computational Linguistics.

Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William R Hersh. 2020. Trec-covid: rationale and structure of an information retrieval shared task for covid-19. *Journal of the American Medical Informatics Association*, 27(9):1431–1436.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Ian Soboroff, Shudong Huang, and Donna Harman. 2019. Trec 2019 news track overview. In *TREC*.

Axel Suarez, Dyaa Albakour, David P. A. Corney, Miguel Martinez-Alvarez, and José Esquivel. 2018. A data collection for evaluating the retrieval of related tweets to news articles. In *ECIR*, volume 10772 of *Lecture Notes in Computer Science*, pages 780–786. Springer.

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 *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*, pages 809–819. Association for Computational Linguistics.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):1–28.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *ACL*, pages 241–251. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *EMNLP*, pages 7534–7550. Association for Computational Linguistics.

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. In *NeurIPS*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP (Demos)*, pages 38–45. Association for Computational Linguistics.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742. Computer Vision Foundation / IEEE Computer Society.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

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

Nan Yang, Furu Wei, Binxing Jiao, Daxing Jiang, and Linjun Yang. 2021. xmoco: Cross momentum contrastive learning for open-domain question answering. In *ACL-IJCNLP*, pages 6120–6129. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pages 2369–2380. Association for Computational Linguistics.

Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *CoNLL*, pages 247–256. ACL.

A The BEIR Benchmark

Datasets We list the statistics of the BEIR benchmark in Table 5. The 18 English zero-shot evaluation datasets come from 9 heterogeneous retrieval tasks, including bio-medical information retrieval, question answering, tweet retrieval, news retrieval, argument retrieval, duplicate question retrieval, citation prediction, and fact checking.

Metric To measure effectiveness of search algorithms or retrieval models, the benchmark uses Normalized Discounted Cumulative Gain (nDCG) (Järvelin and Kekäläinen, 2002) as the evaluation metric. We will give the definition of the metric in the following.

Given top k retrieved documents $\{d_1, d_2, .., d_k\}$ with their relevance $\{r_1, r_2, .., r_k\}$ for a query, the traditional formula of discounted cumulative gain (DCG) accumulated at a particular rank position k is defined in Equation 10, where r_i is 1 if d_i is the ground truth otherwise 0.

$$DCG@K = \sum_{i=1}^{K} \frac{r_i}{log_2(i+1)} \tag{10}$$

Since the length of ground truth list depends on the query, using DCG to compare the performance of retrieval models from one query to the next cannot be consistently achieved. Therefore, the discounted cumulative gain is normalized (nDCG) as:

$$nDCG@K = \frac{DCG@K}{IDCG@K} \tag{11}$$

where IDCG@K is the DCG@K score for the list of relevant documents (ordered by their relevance) in the corpus up to position k. Since IDCG@K producs the maximum possible DCG through position k, the value of nDCG@K is in the range 0 to 1.

B Baselines

We use the baselines from the current BEIR leaderboard (Thakur et al., 2021). These baselines can be divided into four groups: dense retrieval, lexical retrieval, late interaction and re-ranking.

Dense Retrieval For dense retrieval, the baselines are the same dual-tower model as ours. We consider **DPR** (Karpukhin et al., 2020), **ANCE** (Xiong et al., 2021), **TAS-B** (Hofstätter et al., 2021) and **GenQ** (Thakur et al., 2021) in this paper.

- **DPR** uses a single BM25 retrieval example and in-batch examples as hard negative examples to train the model. Following Thakur et al. (2021), we use Multi-DPR as the baseline. The model is a BERT-base model and is trained on four QA datasets, including NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), WebQuestions (Berant et al., 2013) and CuratedTREC (Baudis and Sedivý, 2015).

- **ANCE** constructs hard negative examples from an ANN index of the corpus. The hard negative training instances are updated in parallel during fine-tuning of the model. The model is a RoBERTa (Liu et al., 2019) model trained on MS-MARCO for 600k steps.

- **TAS-B** is trained with Balanced Topic Aware Sampling using dual supervision from a cross-encoder and a ColBERT model (Khattab and Zaharia, 2020). The model is trained with a combination of a pairwise Margin-MSE (Hofstätter et al., 2021) loss and an in-batch negative loss function.

Split (→)					Train	Dev	Test			Avg. Word Lengths	
Task (↓)	Domain (↓)	Dataset (↓)	Title	Relevancy	#Pairs	#Query	#Query	#Corpus	Avg. D / Q	Query	Document
Passage-Retrieval	Misc.	MS MARCO (2016)	✗	Binary	532,761	—-	6,980	8,841,823	1.1	5.96	55.98
Bio-Medical	Bio-Medical	TREC-COVID (2020)	✓	3-level	—-	—-	50	171,332	493.5	10.60	160.77
Information	Bio-Medical	NFCorpus (2016)	✓	3-level	110,575	324	323	3,633	38.2	3.30	232.26
Retrieval (IR)	Bio-Medical	BioASQ (2015)	✓	Binary	32,916	—-	500	14,914,602	4.7	8.05	202.61
Question	Wikipedia	NQ (2019)	✓	Binary	132,803	—-	3,452	2,681,468	1.2	9.16	78.88
Answering	Wikipedia	HotpotQA (2018)	✓	Binary	170,000	5,447	7,405	5,233,329	2.0	17.61	46.30
(QA)	Finance	FiQA-2018 (2018)	✗	Binary	14,166	500	648	57,638	2.6	10.77	132.32
Tweet-Retrieval	Twitter	Signal-1M (RT) (2018)	✗	3-level	—-	—-	97	2,866,316	19.6	9.30	13.93
News	News	TREC-NEWS (2019)	✓	5-level	—-	—-	57	594,977	19.6	11.14	634.79
Retrieval	News	Robust04 (2004)	✗	3-level	—-	—-	249	528,155	69.9	15.27	466.40
Argument	Misc.	ArguAna (2018)	✓	Binary	—-	—-	1,406	8,674	1.0	192.98	166.80
Retrieval	Misc.	Touché-2020 (2020)	✓	3-level	—-	—-	49	382,545	19.0	6.55	292.37
Duplicate-Question	StackEx.	CQADupStack (2015)	✓	Binary	—-	—-	13,145	457,199	1.4	8.59	129.09
Retrieval	Quora	Quora	✗	Binary	—-	5,000	10,000	522,931	1.6	9.53	11.44
Entity-Retrieval	Wikipedia	DBPedia (2017)	✓	3-level	—-	67	400	4,635,922	38.2	5.39	49.68
Citation-Prediction	Scientific	SCIDOCS (2020)	✓	Binary	—-	—-	1,000	25,657	4.9	9.38	176.19
Fact Checking	Wikipedia	FEVER (2018)	✓	Binary	140,085	6,666	6,666	5,416,568	1.2	8.13	84.76
	Wikipedia	Climate-FEVER (2020)	✓	Binary	—-	—-	1,535	5,416,593	3.0	20.13	84.76
	Scientific	SciFact (2020)	✓	Binary	920	—-	300	5,183	1.1	12.37	213.63

Table 5: Statistics of datasets in the BEIR benchmark. The table is taken from Thakur et al. (2021). Few datasets contain documents without titles. Relevancy indicates the query-document relation: binary (relevant, non-relevant) or graded into sub-levels. Avg. D/Q indicates the average relevant documents per query.

- **GenQ** fine-tunes a T5-base (Raffel et al., 2020) model on MS MARCO for 2 epochs and generate 5 queries for each document as additional training data to continue to fine-tune the TAS-B model.

Lexical Retrieval Lexical retrieval is a score function for token matching calculated between two high-dimensional sparse vectors with token weights. BM25 (Robertson and Zaragoza, 2009) is the most commonly used lexical retrieval function. We use the BM25 results reported in Thakur et al. (2021) for comparison.

Late Interaction We also consider a late interaction baseline, namely **ColBERT** (Khattab and Zaharia, 2020). The model computes multiple contextualized embeddings for each token of queries and documents, and then uses a maximum similarity function to retrieve relevant documents. This type of matching requires significantly more disk space for indexes and has a higher latency.

Re-ranking Re-ranking based approaches use the output of a first-stage retrieval system (e.g., BM25), and then re-rank the retrieved documents using a cross-encoder (Nogueira and Cho, 2020). In this paper, we use the **BM25+CE** baseline implemented in Thakur et al. (2021) that uses BM25 to retrieve top-100 documents and a 6-layer MiniLM (Wang et al., 2020) model to further re-rank the recalled documents.

C More Examples

In addition to examples in Section 4.3, we provide more examples here, from a commonsense question answering dataset HotpotQA (Yang et al., 2018).

Q1: In what month is the annual documentary film festival, that is presented by the fort nightly published British journal of literary essays, held?

BM25 (Top 1): ✗
The DOXA Documentary Film Festival is a documentary film festival based in Vancouver, British Columbia, Canada. It is held annually held for 10 days in May and is presented by The Documentary Media Society, a non-profit organization.

BM25 (Top 2): ✓
The London Review of Books (LRB) is a British journal of literary essays. It is published fortnightly.

LaPraDoR (Top 1): ✓
The London International Documentary Festival (or LIDF) is an annual documentary film festival that takes place in the months of March and April every year. [1] The event is presented in association with the London Review of Books. [2]

LaPraDoR (Top 2): ✓
The London Review of Books (LRB) is a British journal of literary essays. [3] It is published fortnightly. [4]

Q2: Ethel Winter worked with which avant-garde theater director?

BM25 (Top 1): ✗
Avant-garde refers to a style in experimental work in art, music, culture, or politics.

BM25 (Top 2): ✗
Christoph Marthaler (born October 17, 1951, Erlenbach, Switzerland) is a Swiss director and musician, working in the style of avant-garde theater, such as Expressionism and Dada, a theater of the absurd elements.

LaPraDoR (Top 1): ✓
Ethel Winter (June 18, 1924 – March 10, 2012) [5] was an American ballet dancer and dance instructor. Winter was an early ballet dancer with the Martha Graham Dance Company from the 1940s to the 1960s, working with other notable early members of the company, including Martha Graham, Yuriko, Yuriko, Ethel Butler, Jean Erdman, and Patricia Birch. [6] She later taught dance and ballet at the Juilliard School.

LaPraDoR (Top 2): ✓
Jean Erdman (born February 20, 1916) [7] is an American dancer and choreographer of modern dance as well as an avant-garde theater director. [8]

Figure 4: Examples from the HotpotQA dataset (Yang et al., 2018). The key facts are highlighted. The reasoning path for Q1 is [3]→[4]→[2]→[1] and for Q2 is [5]→[6]→[7]→[8].