Towards Unsupervised Dense Information Retrieval with Contrastive Learning

Gautier Izacard[†,‡,◇] **Mathilde Caron**[†,◇,§] **Lucas Hosseini**[†] **Sebastian Riedel**[†,¶]
Piotr Bojanowski[†] **Armand Joulin**[†] **Edouard Grave**[†]
[†]Facebook AI Research [‡]ENS, PSL University [◇]Inria
[§]Université Grenoble Alpes [¶]University College London
gizacard@fb.com

Abstract

Information retrieval is an important component in natural language processing, for knowledge intensive tasks such as question answering and fact checking. Recently, information retrieval has seen the emergence of dense retrievers, based on neural networks, as an alternative to classical sparse methods based on term-frequency. These models have obtained state-of-the-art results on datasets and tasks where large training sets are available. However, they do not transfer well to new domains or applications with no training data, and are often outperformed by term-frequency methods such as BM25 which are not supervised. Thus, a natural question is whether it is possible to train dense retrievers without supervision. In this work, we explore the limits of contrastive learning as a way to train unsupervised dense retrievers, and show that it leads to strong retrieval performance. More precisely, we show on the BEIR benchmark that our model outperforms BM25 on 11 out of 15 datasets. Furthermore, when a few thousands examples are available, we show that fine-tuning our model on these leads to strong improvements compared to BM25. Finally, when used as pre-training before fine-tuning on the MS-MARCO dataset, our technique obtains state-of-the-art results on the BEIR benchmark.

1 Introduction

Document retrieval is the task of finding relevant documents in a large collection to answer specific queries. This is an important task by itself and a core component of many natural language processing (NLP) problems, such as open domain question answering (Chen et al., 2017a) or fact checking (Thorne et al., 2018). Traditionally, retrieval systems, or retrievers, leverage lexical similarities to match queries and documents, using, for instance, TF-IDF or BM25 weighting (Robertson & Zaragoza, 2009). These approaches, based on exact matches between tokens of the queries and documents, suffer from the lexical gap and do not generalize well (Berger et al., 2000). By contrast, approaches based on neural networks allow learning beyond lexical similarities, resulting in state-of-the-art performance on question answering benchmarks, such as MS-MARCO (Nguyen et al., 2016) or NaturalQuestions (Kwiatkowski et al., 2019).

As training neural networks requires large amount of data, their strong retrieval results were possible for domains and applications where large training datasets are available. In the case of retrieval, creating these datasets requires manually matching queries to the relevant documents in the collection. This is hardly possible when the collection contains millions or billions of element, resulting in many scenarios where only a few in-domain examples, if any, are available. A potential solution is to train a dense retriever on a large retrieval dataset such as MS-MARCO, and then apply it to new domains, a setting refered to as *zero-shot*. Unfortunately, in this setting dense retrievers are often outperformed by classical methods based on term-frequency, which do not require supervision (Thakur et al., 2021).

Thus, a natural alternative to transfer learning is unsupervised learning, which raises the following question: *is it possible to train dense retrievers without supervision, and match the performance of BM25?* Training dense retrievers without supervision can be achieved by using a pretext task that approximates retrieval. Given a document, one can generate a synthetic query and then train the network to retrieve the original document, among many others, given the query. The inverse Cloze task (ICT), proposed by Lee et al. (2019) to pretrain retrievers, uses a given sentence as query and predict the context surrounding it. While show-

ing promising results as pre-training (Chang et al., 2020; Sachan et al., 2021a), this approach still lags behind BM25 when used as a zero-shot retriever.

ICT is strongly related to contrastive learning (Wu et al., 2018), which has been widely applied in computer vision (Chen et al., 2020; He et al., 2020). In particular, computer vision models trained with the latest contrastive learning methods lead to features well suited to retrieval (Caron et al., 2021). In this work, we thus propose to revisit how well contrastive learning performs to train dense retrievers without supervision. We want to investigate how much the recent developments from computer vision, such as MoCo, can improve dense retrievers. Here, our goal is not to develop new techniques, but to determine how far we can go by pushing existing methods to their limits.

In this paper, we thus make the following contributions. First, we show that contrastive learning can lead to strong unsupervised retrievers: our model achieves recall@100 results competitive with BM25 on most of the BEIR benchmark. Second, in a few-shot setting, we show that our model benefits from few training examples, and obtains better results than transfering models from large datasets such as MS-MARCO. Third, when used as a pre-training method before fine-tuning on MS-MARCO, our technique leads to strong performance on the BEIR benchmark. Finally, we perform ablations to motivate our design choices, and show that cropping works better than the inverse Cloze task. Code and pre-trained models will be open-sourced.

2 Related work

In this section, we briefly review relevant work in information retrieval, and application of machine learning to this problem. This is not an exhaustive review, and we refer the reader to Manning et al. (2008), Mitra et al. (2018) and Lin et al. (2020) for a more complete introduction to the field.

Term-frequency based information retrieval. Historically, in information retrieval, documents and queries are represented as sparse vectors where each element of the vectors corresponds to a term of the vocabulary. Different weighing schemes have been proposed, to determine how important a particular term is to a document in a large dataset. One of the most used weighing scheme is known as TF-IDF, and is based on inverse document frequency, or term specificity (Jones, 1972). BM25, which is still widely used today, extends TF-IDF (Robertson et al., 1995). A well known limitation of these approaches is that they rely on exact match to retrieve documents. This lead to the introduction of latent semantic analysis (Deerwester et al., 1990), in which documents are represented as low dimensional dense vectors.

Neural network based information retrieval. Following the successful application of deep learning methods to natural language processing, neural networks techniques were introduced for information retrieval. Huang et al. (2013) proposed a deep bag-of-words model, in which representations of queries and documents are computed independently. A relevance score is then obtained by taking the dot product between representations, and the model is trained end-to-end on click data from a search engine. This method was later refined by replacing the bag-of-words model by convolutional neural networks (Shen et al., 2014) or recurrent neural network (Palangi et al., 2016). A limitation of bi-encoders is that queries and documents are represented by a single vector, preventing the model to capture fine-grained interactions between terms. Nogueira & Cho (2019) thus introduced a cross-encoder model, based on a pre-trained BERT model (Devlin et al., 2019), which jointly encodes queries and documents. The application of a strong pre-trained model, as well as the cross-encoder architecture, lead to important improvement on the MS-MARCO benchmark (Bajaj et al., 2016).

The methods described in the previous paragraph were applied to re-rank documents, which were retrieved with a traditional IR system such as BM25. Gillick et al. (2018) first studied whether continuous retrievers, based on bi-encoder neural models, could be viable alternative to re-ranking. In the context of question answering, Karpukhin et al. (2020) introduced a dense passage retriever (DPR) based on the bi-encoder architecture. This model is initialized with a BERT network, and trained discriminatively using pairs of queries and relevant documents, with hard negatives from BM25. Xiong et al. (2020) further extended this work by mining hard negatives with the model itself during optimization, and trained on the MS-MARCO dataset. Once a collection of documents, such as Wikipedia articles, is encoded, retrieval is performed with a fast k-nearest neighbors library such as FAISS (Johnson et al., 2019). To alleviate the limitations of bi-encoders, Humeau et al. (2019) in-

troduces the poly-encoder architecture, where documents are encoded by multiple vectors. Similarly, Khattab et al. (2020) proposes the ColBERT model, which keeps a vector representation for each term of the queries and documents. To make the retrieval tractable, the term-level function is approximated to first retrieve an initial set of candidates, which are then re-ranked with the true score. In the context of question answering, knowledge distillation has been used to train retrievers, either using the attention scores of the reader of the downstream task as synthetic labels (Izacard & Grave, 2021), or the relevance score from a cross encoder (Yang & Seo, 2020). Luan et al. (2020) compares, theoretically and empirically, the performance of sparse and dense retrievers, including bi-, cross- and poly-encoders. Dense retrievers, such as DPR, can lead to indices that weigh close to 100 GB when encoding document collections such as Wikipedia. Izacard et al. (2020) shows how to compress such indices, with limited impact on performance, making them more practical to use.

Self-supervised learning for NLP. Following the success of word2vec (Mikolov et al., 2013), many self-supervised techniques have been proposed to learn representation of text. Here, we briefly review the ones that are most related to our approach: sentence level models and contrastive techniques. Jernite et al. (2017) introduced different objective function to learn sentence representations, including next sentence prediction and binary order prediction. These objectives were later used in pre-trained models based on transformers, such as BERT (Devlin et al., 2019) and AlBERT (Lan et al., 2019). In the context of retrieval, Lee et al. (2019) introduced the inverse cloze task (ICT), whose purpose is to predict the context surrounding a span of text. Guu et al. (2020) integrated a bi-encoder retriever model in a BERT pre-training scheme. The retrieved documents are used as additional context in the BERT task, and the whole system is trained end-to-end in an unsupervised way. Similarly, Lewis et al. (2020) proposed to jointly learn a retriever and a generative seq2seq model, using self-supervised training. Chang et al. (2020) compares different pre-training tasks for retrieval, including the inverse cloze task. Constrastive learning was introduced in computer vision by Wu et al. (2018), followed by several modifications to improve the training (He et al., 2020; Chen et al., 2020; Caron et al., 2020). In the context of natural language processing, Fang et al. (2020) proposed to apply MoCo where positive pairs of sentences are obtained using back-translation. Different works augmented the masked language modeling objective with a contrastive loss (Giorgi et al., 2020; Wu et al., 2020; Meng et al., 2021). Finally, SBERT (Reimers & Gurevych, 2019) uses a siamese network similar to contrastive learning to learn a BERT-like model that is adapted to matching sentence embeddings. Their formulation is similar to our work but requires aligned pairs of sentences to form positive pairs while we propose to use data augmentation to leverage large unaligned textual corpora. Concurrent to this work, Gao & Callan (2021) have also shown the potential of contrastive learning for information retrieval; building on the same observation that both tasks share a similar structure.

3 Method

In this section, we describe how to train a dense retriever with no supervision. We review the model architecture and then describe contrastive learning - a key component of its training.

The objective of a retriever is to find relevant documents in a large collection for a given query. Thus, the retriever takes as input the set of documents and the query and outputs a relevance score for each document. A standard approach is to encode each query–document pair with a neural network (Nogueira & Cho, 2019). This procedure requires re-encoding every document for any new query and hence does not scale to large collections of documents. Instead, we propose to use a bi-encoder architecture, where documents and queries are encoded independently (Huang et al., 2013; Karpukhin et al., 2020). One can compute the relevance score by taking the dot product (or cosine similarity) between the document's representation and the representation of the query. More precisely, given a pair of query q and document d, we encode each of them independently using the same model, f_θ, parametrized by θ. The relevance score $s(q, d)$ is then the dot product of the resulting representations:

$$s(q, d) = \langle f_\theta(q), f_\theta(d) \rangle.$$

In practice, we use a transformer network for f_θ to embed both queries and documents. The representation $f_\theta(q)$ (resp. $f_\theta(d)$) for a query (resp.

document) is obtained by averaging the hidden representations of the last layer. Following previous work on dense retrieval with neural networks, we use the BERT base uncased architecture and refer the reader to Devlin et al. (2019) for more details.

3.1 Unsupervised training on unaligned documents

In this section, we describe our unsupervised training pipeline. We briefly review the loss function traditionally used in contrastive learning. We then discuss obtaining positive pairs from a single text document, a critical ingredient for this training paradigm.

3.1.1 Contrastive learning

Contrastive learning is an approach that relies on the fact that every document is, in some way, unique. This signal is the only information available in the absence of manual supervision. The resulting algorithm learns by discriminating between documents, using a contrastive loss (Wu et al., 2018). This loss compares either positive (from the same document) or negative (from different documents) pairs of document representations. More formally, given a positive pair of representations (q, k_+) and a set of negative pairs $(q, k_i)_{i=0..K}$, the contrastive InfoNCE loss is defined as:

$$\mathcal{L}(q, k_+) = \frac{\exp(s(q, k_+)/\tau)}{\sum_{i=0}^{K} \exp(s(q, k_i)/\tau)},$$

where τ is a temperature parameter. This loss encourages the relevance score of similar examples to be high and that of dissimilar examples to be low. Another interpretation of this loss function is the following: given the query representation q, the goal is to recover, or retrieve, the representation k_+ corresponding to the positive document, among all the negatives k_i. In the following, we refer to the left-hand side representations in the score s as queries and the right-hand side representations as keys.

3.1.2 Building positive pairs from a single document

A crucial element of contrastive learning is how to build positive pairs from a single input. In computer vision, this step relies on applying two independent data augmentations to the same image, resulting in two "views" that form a positive pair (Wu et al., 2018; Chen et al., 2020). While we consider similar independent text transformation, we also explore dependent transformations designed to reduce the correlation between views.

Inverse Cloze Task is a data augmentation that generates two mutually exclusive views of a document, introduced in the context of retrieval by Lee et al. (2019). The idea is to take a span of tokens to form one view, and its complementary to form the other view. More precisely, given a sequence of text $(w_1, ..., w_n)$, ICT samples a span a, b, where $1 \leq a < b \leq n$, uses the tokens of the span as the query and the rest of the tokens as the document (or key). In the original implementation by Lee et al. (2019) the span corresponds to a sentence, and is kept in the document 10% of the time to encourage lexical matching.

Independent cropping is a common independent data augmentation used for images where views are generated independently by cropping the input. In the context of text, cropping is equivalent to sampling a span of tokens. This strategy thus samples independently two spans from a document to form a positive pair. As opposed to the inverse Cloze task, in *cropping* both views of the example correspond to contiguous subsequence of the original data. A second difference between cropping and ICT is the fact that the task is symmetric: both the queries and documents follow the same distribution. Independent cropping can also lead to overlap between the two views of the data, hence encouraging the network to learn exact matches between the query and document, in a way that is similar to lexical matching methods like BM25. In practice, we can either fix the length of the span for the query and the key, or sample them.

Additional data augmentation. Finally, we also consider additional data augmentations such as random word deletion, random word replacement or random word masking. We use these perturbation in addition to random cropping.

3.1.3 Building large set of negative pairs

An important aspect of contrastive learning is to maintain a large set of negative pairs. Most standard frameworks differ from how the negatives are handled, and we briefly describe two of them, in-batch negative sampling and MoCo, that we use in this work.

Negative pairs within a batch. A first solution is to generate the negative pairs by using the other examples from the same batch: each example in

a batch is transformed twice to generate positive pairs, and we generate negative pairs by using the views from the other examples in the batch. We will refer to this technique as "in-batch negatives". In that case, the gradient is back-propagated through the representations of both the queries and the keys. A downside of this approach is that it requires extremely large batch sizes to work well (Chen et al., 2020). This method has been widely used to train information retrieval models with supervised data (Chen et al., 2017b; Karpukhin et al., 2020) and was also considered when using ICT to pre-train retrievers by Lee et al. (2019).

Negative pairs across batches. An alternative approach is to store representations from previous batches in a queue and use them as negative examples in the loss (Wu et al., 2018). This allows for smaller batch size but slightly changes the loss by making it asymmetric between "queries" (one of the view generated from the elements of the current batch), and "keys" (the other generated views, as well as the elements stored in the queue). Gradient is only backpropagated through the "queries", and the representation of the "keys" are considered as fixed. In practice, the features stored in the queue from previous batches comes form previous iterations of the network. This leads to a drop of performance when the network rapidly changes during training. Instead, He et al. (2020) proposed to generate representations of keys from a second network that is updated more slowly. This approach, called MoCo, considers two networks: one for the keys, parametrized by θ_k, and one of the query, parametrized by θ_q. The parameters of the query network are updated with backpropagation and stochastic gradient descent, similarly to when using in-batch negatives, while the parameters of the key network, or Momentum encoder, is updated from the parameters of the query network by using a exponential moving average:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q,$$

where m is the momentum parameter that takes its value in $[0, 1]$.

4 Experiments

In this section, we empirically evaluate dense retrievers trained with contrastive learning. First, we compare our best models to the state-of-the-art on competitive retrieval benchmarks in unsupervised, zero-shot and few-shot settings. Then, we provide an ablation study to motivate the technical choices leading to our best retriever, called *Contriever* (contrastive retriever) and which uses MoCo with random cropping. We give more technical details about our models in Appendix A.

4.1 BEIR benchmark

The BEIR benchmark, introduced by Thakur et al. (2021), contains 18 retrieval datasets with a focus on diversity. Each dataset is made of a set of queries, the corresponding relevant documents and a large collection of documents to retrieve from. These datasets correspond to nine different retrieval tasks, such as fact checking, question answering or citation prediction, and cover multiple domains, such as Wikipedia, news articles or scientific publications. There is also diversity in terms of documents and queries length, with queries ranging from 3 to 190 words and documents from 11 to 630 words. Most datasets from the BEIR benchmark do not contain a training set, and the focus of the benchmark is *zero-shot retrieval*. However, most machine learning based retrievers are still trained on supervised data, such as the large scale retrieval dataset MS-MARCO (Bajaj et al., 2016). Following standard practice, we report two metrics on this benchmark, the nDGC@10 and the recall@100. These two metrics are complementary and both important: nDCG focuses on the ranking of the top 10 retrieved documents, and is good at evaluating rankings returned to humans, for example in a search engine. On the other hand, recall@100 is relevant to evaluate retrievers that are used in marchine learning systems, such as question answering. Indeed, such models can process hundreds of documents, and ignore the ranking of these documents (Izacard & Grave, 2020).

4.2 Baselines

First, we compare our retriever to BM25, which does not require supervision. However, the term relevance is computed over each dataset independently, thus leading to a different ranking function for each dataset of BEIR. As dense unsupervised baselines, we consider the retriever from REALM (Guu et al., 2020), as well as RoBERTa large fine-tuned with SimCSE (Gao et al., 2021). We also compare to ML-based retrievers trained on MS-MARCO, classified in three categories: sparse, dense and late-interaction. For sparse methods, we compare to *Splade v2* (Formal et al., 2021), which computes sparse representations of docu-

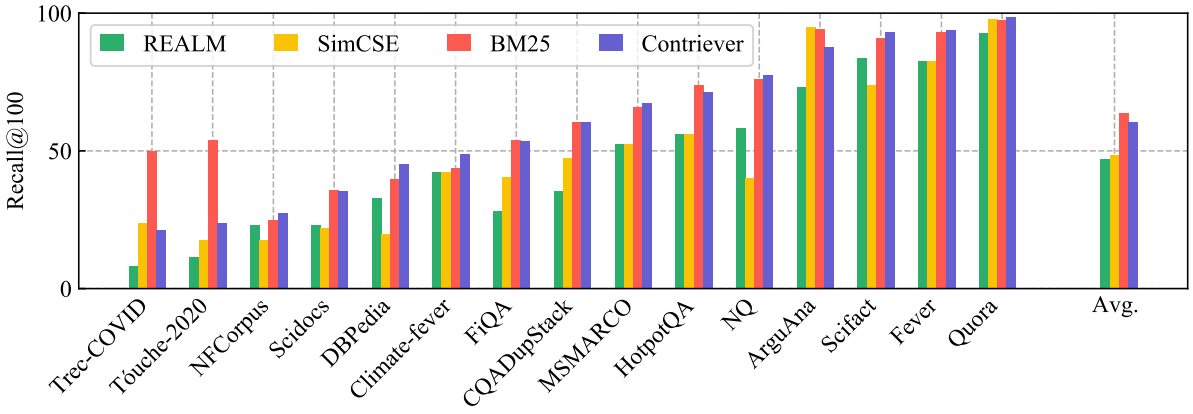

Figure 1: **Unsupervised retrieval.** We compare our pre-training without using *any* annotated data to REALM (Guu et al., 2020), SimCSE (Gao et al., 2021) and BM25. For SimCSE we report results of the model using RoBERTa large. REALM uses annotated entity recognition data for training. We highlight that our unsupervised pre-training is on par with BM25 but on 2 datasets.

	NaturalQuestions			TriviaQA		
	R@5	R@20	R@100	R@5	R@20	R@100
Inverse Cloze Task (Sachan et al., 2021b)	32.3	50.9	66.8	40.2	57.5	73.6
Masked salient spans (Sachan et al., 2021b)	41.7	59.8	74.9	53.3	68.2	79.4
BM25 (Ma et al., 2021)	-	62.9	78.3	-	**76.4**	**83.2**
Contriever	**47.2**	**67.2**	**81.3**	**59.5**	74.2	**83.2**
Supervised topline: DPR (Karpukhin et al., 2020)	-	78.4	85.4	-	79.4	85.0

Table 1: **Unsupervised recall@k** on the test sets of NaturalQuestions and TriviaQA. For Inverse Cloze Task and Masked Salient Spans we report the results of Sachan et al. (2021b). The Masked Salient Spans model uses annotated named entity recognition data.

ments with BERT pre-trained model. For dense methods, we use *DPR* (Karpukhin et al., 2020) and *ANCE* (Xiong et al., 2020), which are bi-encoders trained on supervised data such as NaturalQuestions or MS-MARCO. We also compare to *TAS-B* (Hofstätter et al., 2021), which performs distillation from a cross-encoder to a bi-encoder, and GenQ, which creates synthetic query-document pairs with a generative model.[1] For late-interaction, we use ColBERT (Khattab et al., 2020), which computes pairwise scores between contextualized representations of queries and documents, as well as a cross-encoder used to re-rank documents retrieved with BM25. Interestingly, we note that BM25 is a competitive baseline, outperforming all dense retrievers on average on BEIR.

4.3 Results

First, we compare the performance of fully unsupervised models, i.e., without fine-tuning on MS-MARCO or other annotated data. In Figure 1 we report the recall@100 performance of unsupervised models on the BEIR benchmark. Interestingly, we observe that in this setting, contriever is competitive compared to BM25 on all datasets, but TREC-COVID and Tóuche-2020. In particular, it obtains better performance than BM25 on 11 out of 15 datasets from the benchmark. Contriever also outperforms previously proposed unsupervised dense retrievers, which obtains lower performance than BM25 in general. In Table 1, we report the retrieval performance on two question answering datasets: NaturalQuestions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017). Here, our model is also competitive with a strong BM25 baseline (Ma et al., 2021), for example leading to 3 points improvement for the recall@100 on NaturalQuestions. It also outperforms previously proposed dense retrievers which were trained with ICT or salient span masking. Overall, these results show the potential of contrastive learning to train fully unsupervised dense retrievers.

Next, we report nDCG@10 on the BEIR

[1]GenQ thus leads to one different model for each dataset.

	BM25	BM25+CE	DPR	ANCE	TAS-B	Gen-Q	ColBERT	Splade v2	Ours	Ours+CE
MS MARCO	22.8	41.3	17.7	38.8	40.8	40.8	40.1	43.3	40.7	**47.0**
Trec-COVID	65.6	**75.7**	33.2	65.4	48.1	61.9	67.7	71.0	59.6	70.1
NFCorpus	32.5	**35.0**	18.9	23.7	31.9	31.9	30.5	33.4	32.8	34.4
NQ	32.9	53.3	47.4	44.6	46.3	35.8	52.4	52.1	49.8	**57.7**
HotpotQA	60.3	70.7	39.1	45.6	58.4	53.4	59.3	68.4	63.8	**71.5**
FiQA	23.6	34.7	11.2	29.5	30.0	30.8	31.7	33.6	32.9	**36.7**
ArguAna	31.5	31.1	17.5	41.5	42.9	**49.3**	23.3	47.9	44.6	41.3
Touche-2020	**36.7**	27.1	13.1	24.0	16.2	18.2	20.2	36.4	23.0	29.8
CQADupStack	29.9	37.0.	15.3	29.6	31.4	34.7	35.0	-	34.5	**37.7**
Quora	78.9	82.5	24.8	85.2	83.5	83.0	85.4	83.8	**86.5**	82.4
DBPedia	31.3	40.9	26.3	28.1	38.4	32.8	39.2	43.5	41.3	**47.1**
Scidocs	15.8	16.6	7.7	12.2	14.9	14.3	14.5	15.8	16.5	**17.1**
FEVER	75.3	**81.9**	56.2	66.9	70.0	66.9	77.1	78.6	75.8	**81.9**
Climate-FEVER	21.3	25.3	14.8	19.8	22.8	17.5	18.4	23.5	23.7	**25.8**
Scifact	66.5	68.8	31.8	50.7	64.3	64.4	67.1	**69.3**	67.7	69.2
Avg. w/o CQA	42.5	48.9	25.7	41.1	43.5	42.9	44.8	50.0	47.1	50.9
Avg.	41.7	48.1	25.0	40.4	42.7	42.4	44.1	-	46.2	50.0
Best on	1	3	0	0	0	1	0	1	1	9

Table 2: **BEIR Benchmark.** We report nDCG@10 on the test sets from the BEIR benchmark for bi-encoder methods without re-ranker. We also report the average and number of datasets where a method is the best ("Best on") over the entire BEIR benchmark (excluding three datasets because of their licence). Bold is the best overall. MS-MARCO is excluded from the average.

benchmark for different retrievers trained on MS-MARCO in Table 2 (recall@100 can be found in Table 8 of appendix). We individually report results on each dataset as well as the average over 14 datasets of the BEIR Benchmark (excluding 3 for license reasons). We observe that when used as pre-training, contrastive learning leads to strong performance: contriever obtains the best results among dense methods for the nDCG@10, and is state-of-the-art for the recall@100 (improving the average recall@100 from 65.0 to 67.1). This strong recall@100 performance can be further exploited by using a cross-encoder[2] to re-rank the retrieved documents: this leads to the state-of-the-art on 8 datasets of the BEIR benchmark, as well as on average. It should be noted that our fine-tuning procedure on MS-MARCO is simpler than for other retrievers, as we use a simple strategy for negative mining and do not use distillation. Our model would probably also benefits from improvements proposed by these retreivers, but this is beyond the scope of this paper.

Finally, we illustrate the benefit of our retriever compared to BM25 in a *few-shot* setting, where we have access to a small number of in-domain retrieval examples. This setting is common in practice, and lexical based methods, like BM25, cannot leverage the small training sets to adapt its weights. In Table 3, we report nDCG@10 on three datasets from BEIR associated with the smallest training sets, ranging from 729 to 5,500 queries. We observe that on these small datasets, our pre-training leads to better results than BERT pre-training, even when BERT is fine-tuned on MS-MARCO as an intermediate step. Our pre-trained model also outperforms BM25, showing the advantage of dense retriever over lexical methods in the few-shot setting. More details are given in Appendix A.

[2]We use the existing `ms-marco-MiniLM-L-6-v2` cross-encoder model to perform the re-ranking.

	Add'l. data	SciFact	NFCorpus	FiQA
# queries		729	2,590	5,500
BM25	-	66.5	32.5	23.6
BERT	-	75.2	29.9	26.1
Contriever	-	84.0	33.6	36.4
BERT	MS-MARCO	80.9	33.2	30.9
Contriever	MS-MARCO	**84.8**	**35.8**	**38.1**

Table 3: **Few-shot retrieval.** Test nDCG@10 after training on a small in-domain training set. We compare BERT and our model, with and without an intermediate fine-tuning step on MS-MARCO. Note that our unsupervised pre-training alone outperforms BERT with intermediate MS-MARCO fine-tuning.

	NFCorpus	NQ	FiQA	ArguAna	Quora	DBPedia	SciDocs	FEVER	AVG
MoCo	26.2	13.1	13.7	33.0	69.5	20.0	11.9	57.6	30.1
in-batch negatives	24.2	21.6	13.0	33.7	74.9	17.9	13.6	56.1	31.9

Table 4: **MoCo vs. in-batch negatives.** In this table, we report nDCG@10 on the BEIR benchmark for in-batch negatives and MoCo, without fine-tuning on the MS-MARCO dataset.

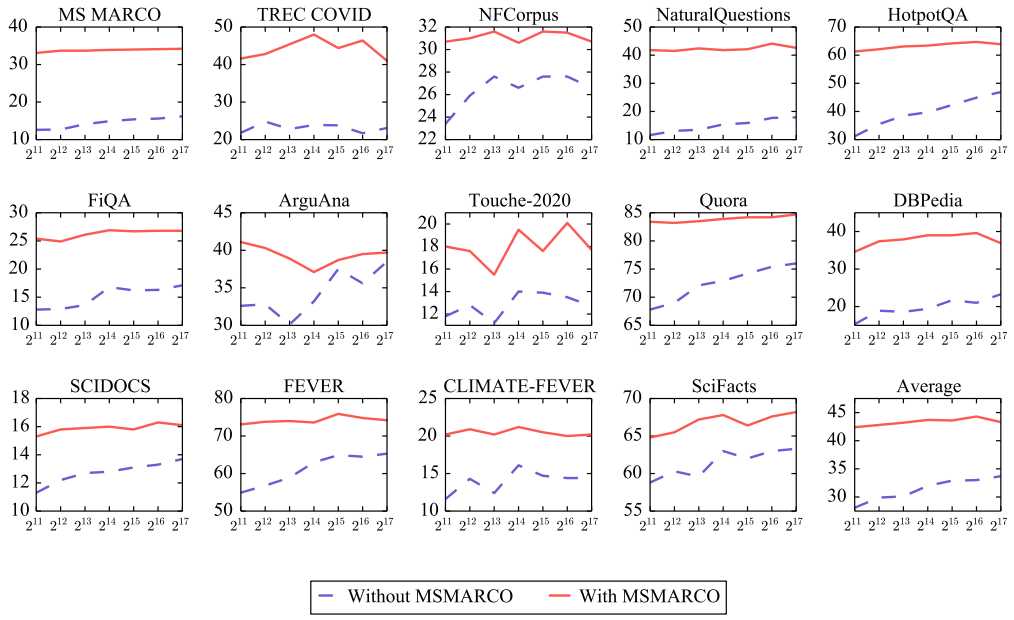

Figure 2: **Impact of the number of negatives.** We report nDCG@10 as a function of the queue size, with and without fine-tuning on MS-MARCO. We report numbers using the MoCo framework where the keys for the negatives are computed with the momentum encoder and stored in a queue.

5 Ablation studies

In this section, we investigate the influence of different design choices on our method. In these ablations, all the models are pre-trained on Wikipedia for 200k gradient steps, with a batch size of 2,048 (on 32 GPUs). Each fine-tuning on MS-MARCO takes 20k gradient steps with a batch size of 512 (on 8 GPUs), using AdamW and no hard negatives.

MoCo vs. in-batch negatives. First, we compare the two contrastive pre-training methods: MoCo and in-batch negatives. As in in-batch negatives, the number of negative examples is equal to the batch size, we train models with a batch size of 4,096 and restrict the queue in MoCo to the same number of elements. This experiment measures the effect of using of momentum encoder for the keys instead of the same network as for the queries. Using a momentum also prevents from backpropagating the gradient through the keys. We report results, without fine-tuning on MS-MARCO in Table 4. We observe that the difference of performance between the two methods is small, especially after fine-tuning on MS-MARCO. We thus propose to use MoCo as our contrastive learning framework, since it scales to a larger number of negative examples without the need to increase the batch size.

Number of negative examples. Next, we study the influence of the number of negatives used in the contrastive loss, by varying the queue size of the MoCo algorithm. We consider values ranging from 2,048 to 131,072, and report results in Figure 2. We see that on average over the BEIR benchmark, increasing the number of negatives leads to better retrieval performance, especially in the unsupervised setting. However, we note that this effect is not equally strong for all datasets.

Data augmentations. Third, we compare different ways to generate pairs of positive examples from a single document or chunk of text. In particular, we compare random cropping, which leads to pairs with overlap, and the inverse cloze task,

	NFCorpus	NQ	ArguAna	Quora	DBPedia	SciDocs	FEVER	Overall
ICT	23.2	19.4	31.6	27.6	21.3	10.6	55.6	25.9
Crop	27.6	17.7	35.6	75.4	21.0	13.3	64.5	32.2
Crop + delete	26.8	20.8	35.8	77.3	21.5	14.0	67.9	33.8
Crop + replace	27.7	18.7	36.2	75.6	22.0	13.0	66.8	32.9

Table 5: **Impact of data augmentions.** nDCG@10 without fine-tuning on MS-MARCO.

	NFCorpus	NQ	FiQA	ArguAna	Quora	DBPedia	SciDocs	FEVER	Overall
Wiki	27.6	17.7	16.3	35.6	75.4	21.0	13.3	64.5	33.0
CCNet	29.5	25.8	26.2	35.2	80.6	20.5	14.9	60.9	34.9
Uniform	31.0	19.4	25.1	37.8	80.4	21.5	14.7	59.8	33.9
50/50%	31.5	18.6	23.3	36.2	79.1	22.1	13.7	64.1	34.7

Table 6: **Training data.** We report nDCG@10 without fine-tuning on MS-MARCO.

	NFCorpus	NQ	FiQA	ArguAna	Quora	DBPedia	SciDocs	FEVER	Overall
BERT	28.2	44.6	25.9	35.0	84.0	34.4	13.0	69.8	42.0
Contriever	33.2	50.2	28.8	46.0	85.4	38.8	16.0	77.7	46.5

Table 7: **Fine-tuning.** We report nDCG@10 after fine-tuning BERT and our model on MS-MARCO.

which was previously considered to pre-train retrievers. Interestingly, as shown in Table 5, the random cropping strategy outperforms the inverse cloze task in our setting. We believe that random cropping, leading to the identical distributions of keys and queries, leads to more stable training with MoCo compared to ICT. This might explains part of the difference of performance between the two methods. We also investigate whether additional data perturbations, such as random word deletion or replacement, are beneficial for retrieval.

Training data. Finally, we study the impact of the pre-training data on the performance of our retriever, by training on Wikipedia, CCNet or a mix of both sources of data. We report results in Table 6, and observe that there is no clear winner between the two data sources. Unsurprisingly, training on the more diverse CCNet data leads to strong improvements on datasets from different domains than Wikipedia, such as FiQA or Quora. On the other hand, on a dataset like FEVER, training on Wikipedia leads to better results. To get the best of both worlds, we consider two strategies to mix the two data sources. In the "50/50%" strategy, examples are sampled uniformly across domain, meaning that half the batches are from Wikipedia and the other half from CCNet. In the "uniform" strategy, examples are sampled uniformly over the union of the dataset. Since CCNet is significantly larger than Wikipedia, this means that most of the batches are from CCNet.

Impact of fine-tuning on MS-MARCO. To isolate the impact of pre-training from the impact of fine-tuning on MS-MARCO, we apply the same fine-tuning to the BERT base uncased model. We report results in Table 7, and observe that when applied to BERT, our fine-tuning leads to results that are lower than the state-of-the-art. Hence, we believe that most of the improvement compared to the state-of-the-art retrievers can be attributed to our contrastive pre-training strategy.

Discussion

In this work, we propose to explore the limits of contrastive pre-training to learn dense text retrievers. We use the MoCo technique, which allows to train models with a large number of negative examples. We make several interesting observations: first, we show that neural networks trained without supervision using contrastive learning exhibits good retrieval performance, which are competitive with BM25 (albeit not state-of-the-art). These results can be further improved by fine-tuning on the supervised MS MARCO dataset, leading to strong results, in particular for recall@100. Based on that observation, we use a cross-encoder to rerank documents retrieved with our model, leading to new state-of-the-art on the competitive BEIR

benchmark. We also performed extensive ablation experiments, and observed that independent random cropping seems to be a strong alternative to the inverse Cloze task for training retrievers.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 192–199, 2000.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932*, 2020.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In *Proc. ACL*, 2017a.

Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. On sampling strategies for neural network-based collaborative filtering. *arXiv preprint arXiv:1706.07881*, 2017b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL*, 2019.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*, 2020.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*, 2021.

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

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

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. End-to-end retrieval in continuous space. *arXiv preprint arXiv:1811.08008*, 2018.

John M Giorgi, Osvald Nitski, Gary D Bader, and Bo Wang. Declutr: Deep contrastive learning for unsupervised textual representations. *arXiv preprint arXiv:2006.03659*, 2020.

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

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. *arXiv preprint arXiv:2104.06967*, 2021.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pp. 2333–2338, 2013.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*, 2019.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*, 2020.

Gautier Izacard and Edouard Grave. Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=NTEz-6wysdb.

Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Sebastian Riedel, and Edouard Grave. A memory efficient baseline for open domain question answering. *arXiv preprint arXiv:2012.15156*, 2020.

Yacine Jernite, Samuel R Bowman, and David Sontag. Discourse-based objectives for fast unsupervised sentence representation learning. *arXiv preprint arXiv:1705.00557*, 2017.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proc. ACL*, 2017.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

Omar Khattab, Christopher Potts, and Matei Zaharia. Relevance-guided supervision for openqa with colbert, 2020.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. ASAM: adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. *arXiv preprint arXiv:2102.11600*, 2021.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proc. ACL*, 2019.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. Pre-training via paraphrasing. *arXiv preprint arXiv:2006.15020*, 2020.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: Bert and beyond. *arXiv preprint arXiv:2010.06467*, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. Sparse, dense, and attentional representations for text retrieval. *arXiv preprint arXiv:2005.00181*, 2020.

Xueguang Ma, Kai Sun, Ronak Pradeep, and Jimmy Lin. A replication study of dense passage retriever, 2021.

Christopher D Manning, Hinrich Schütze, and Prabhakar Raghavan. *Introduction to information retrieval*. Cambridge university press, 2008.

Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. Coco-lm: Correcting and contrasting text sequences for language model pretraining. *arXiv preprint arXiv:2102.08473*, 2021.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

Bhaskar Mitra, Nick Craswell, et al. An introduction to neural information retrieval. *Foundations and Trends® in Information Retrieval*, 13(1):1–126, 2018.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*, 2016.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):694–707, 2016.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

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

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

Devendra Singh Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L Hamilton, and Bryan Catanzaro. End-to-end training of neural retrievers for open-domain question answering. *arXiv preprint arXiv:2101.00408*, 2021a.

Devendra Singh Sachan, Siva Reddy, William Hamilton, Chris Dyer, and Dani Yogatama. End-to-end training of multi-document reader and retriever for open-domain question answering, 2021b.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd international conference on world wide web*, pp. 373–374, 2014.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 2021.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

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

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*, 2020.

	BM25	ANCE	TAS-B	ColBERT	Splade v2	Ours
MS MARCO	65.8	85.2	88.4	86.5	-	**89.1**
Trec-COVID	**49.8**	45.7	38.7	46.4	12.3	40.7
NFCorpus	25.0	23.2	28.0	25.4	27.7	**30.0**
NQ	76.0	83.6	90.3	91.2	**93.0**	92.5
HotpotQA	74.0	57.8	72.8	74.8	**82.0**	77.7
FiQA	53.9	58.1	59.3	60.3	62.1	**65.6**
ArguAna	94.2	93.7	94.2	91.4	97.2	**97.7**
Touche-2020	**53.8**	45.8	43.1	43.9	35.4	29.4
CQADupStack	60.6	57.9	62.2	62.4	-	**66.3**
Quora	97.3	98.7	98.6	98.9	98.7	**99.3**
DBPedia	39.8	31.9	49.9	46.1	**57.5**	54.1
Scidocs	35.6	26.9	33.5	34.4	36.4	**37.8**
Fever	93.1	90.0	93.7	93.4	**95.1**	94.9
Climate-fever	43.6	44.5	53.4	44.4	52.4	**57.4**
Scifact	90.8	81.6	89.1	87.8	92.0	**94.7**
Avg. w/o CQA	63.6	60.1	65.0	64.5	64.8	67.1
Avg.	63.4	60.0	64.8	64.3	-	67.0
Best on	2	0	0	0	4	8

Table 8: **BEIR Benchmark.** We report the recall@100 on the test sets from the BEIR benchmark for bi-encoder methods. We report the capped recall@100 on Trec-COVID following the original BEIR setup. Note that using a cross-encoder to re-rank the top-100 documents do not change the recall@100, hence, we do not include these methods in this table. We also report the average and number of datasets where a method is the best ("Best on") over the entire BEIR benchmark (excluding three datasets because of their licence). Bold is the best overall. MS-MARCO is excluded from the average.

Sohee Yang and Minjoon Seo. Is retriever merely an approximator of reader? *arXiv preprint arXiv:2010.10999*, 2020.

A Technical details

Contrastive pre-training. For the model with fine-tuning on MS-MARCO, we use the MoCo algorithm (He et al., 2020) with a queue of size 131,072, a momentum value of 0.9995 and a temperature of 0.05. We use the random cropping data augmentation, with documents of 256 tokens and span sizes sampled between 5% and 50% of the document length. Documents are simply random piece of text sampled from a mix between Wikipedia and CC-net data (Wenzek et al., 2020), where half the batches are sampled from each source. We also apply word deletion with a probability of 10%. We optimize the model with the AdamW (Loshchilov & Hutter, 2019) optimizer, with learning rate of $5 \cdot 10^{-5}$, batch size of 2,048 and 500,000 steps. We initialize the network with the publicly available BERT base uncased model.

Fine-tuning on MS-MARCO. For the fine-tuning on MS-MARCO we do not use the MoCo algorithm and simply use in-batch negatives. We use the ASAM optimizer (Kwon et al., 2021), with a learning rate of 10^{-5} and a batch size of 1024 with a temperature of 0.05, also used during pre-training. We train an initial model with random negative examples for 20000 steps, mine hard negatives with this first model, and re-train a second model with those. Each query is associated with a gold document and a negative document, which is a random document in the first phase and a hard negative 10% of the time in the second phase. For each query, all documents from the current batch aside of the gold document are used as negatives.

Few-shot training. For the few-shot evaluation presented in Table 3, we train for 500 epochs on each dataset with a batch size of 256 with in-batch random negatives. We evaluate performance performance on the development set every 100 gradient updates and perform early stopping based on this metric. For SciFact, we hold out randomly 10% of the training data and use them as development set, leading to a train set containing 729 samples.

Model (→)	BM25	BERT	SimCSE	REALM	Contriever
Dataset (↓)			Recall@100		
MS MARCO	65.8	3.5	52.6	52.6	**67.2**
Trec-COVID	**49.8**	10.6	23.9	8.1	17.2
NFCorpus	25.0	6.7	17.5	23.0	**29.4**
NQ	76.0	14.3	40.0	58.1	**77.1**
HotpotQA	**74.0**	15.8	56.1	56.1	70.4
FiQA-2018	53.9	6.9	40.3	28.0	**56.2**
ArguAna	94.2	59.1	**95.1**	73.1	90.1
Tóuche-2020	**53.8**	3.0	17.5	11.5	22.5
CQADupStack	60.6	11.0	47.3	35.5	**61.4**
Quora	97.3	74.6	97.9	92.7	**98.7**
DBPedia	39.8	7.1	19.7	33.0	**45.3**
SCIDOCS	35.6	11.3	22.0	23.1	**36.0**
Fever	93.1	13.6	82.6	82.6	**93.6**
Climate-fever	43.6	12.8	42.3	42.3	**44.1**
SciFact	90.8	35.2	73.8	83.8	**92.6**
Avg.	63.4	20.1	48.3	46.5	59.6
Best on	3	0	1	0	11
			NDCG@10		
MS MARCO	**22.8**	0.6	15.2	15.2	20.6
Trec-COVID	**65.6**	16.6	36.4	20.1	27.4
NFCorpus	**32.5**	2.5	12.3	24.1	31.7
NQ	**32.9**	2.7	11.1	15.2	25.4
HotpotQA	**60.3**	4.9	40.5	40.5	48.1
FiQA-2018	23.6	1.4	13.8	9.7	**24.5**
ArguAna	31.5	23.1	**45.3**	22.8	37.9
Tóuche-2020	**36.7**	3.4	10.4	7.3	19.3
CQADupStack	**29.9**	2.5	19.0	13.5	28.4
Quora	78.9	3.9	47.8	71.6	**83.5**
DBPedia	**31.3**	3.9	12.9	22.7	29.2
SCIDOCS	**15.8**	2.7	7.0	9.0	14.9
FEVER	**75.3**	4.9	42.9	42.9	68.2
Climate-fever	**21.3**	4.1	14.3	14.3	15.5
SciFact	**66.5**	9.8	36.9	47.1	64.9
Avg.	43.0	9.3	27.4	25.8	37.1
Best on	12	0	1	0	2

Table 9: **Unsupervised retrieval.** Performance of unsupervised methods on the BEIR datasets. We report the capped recall@100 on Trec-COVID following the original BEIR setup. For SimCSE we report results of the model using RoBERTa large. REALM uses annotated entity recognition data for training.