Unsupervised Dense Retrieval with Relevance-Aware Contrastive Pre-Training

Yibin Lei[1]*, **Liang Ding**[2]†, **Yu Cao**[3], **Chantong Zan**[4], **Andrew Yates**[1], **Dacheng Tao**[2]
[1]University of Amsterdam [2]JD Explore Academy
[3]Tencent IEG [4]China University of Petroleum (East China)
{y.lei, a.c.yates}@uva.nl, {liangding.liam, dacheng.tao}@gmail.com
rainyucao@tencent.com, zanct@s.upc.edu.cn

Abstract

Dense retrievers have achieved impressive performance, but their demand for abundant training data limits their application scenarios. Contrastive pre-training, which constructs pseudo-positive examples from unlabeled data, has shown great potential to solve this problem. However, the pseudo-positive examples crafted by data augmentations can be irrelevant. To this end, we propose relevance-aware contrastive learning. It takes the intermediate-trained model itself as an imperfect oracle to estimate the relevance of positive pairs and adaptively weighs the contrastive loss of different pairs according to the estimated relevance. Our method consistently improves the SOTA unsupervised Contriever model (Izacard et al., 2022) on the BEIR and open-domain QA retrieval benchmarks. Further exploration shows that our method can not only beat BM25 after further pre-training on the target corpus but also serves as a good few-shot learner. Our code is publicly available at https://github.com/Yibin-Lei/ReContriever.

1 Introduction

Dense retrievers, which estimate the relevance between queries and passages in the dense embedding space, have achieved impressive performance in various applications, including web search (Liu et al., 2021) and open-domain question answering (Karpukhin et al., 2020). One key factor for the success of dense retrievers is a large amount of human-annotated training data, e.g., MS-MARCO (Bajaj et al., 2016) with above 500,000 examples. However, a recent study (Thakur et al., 2021) shows that even trained with enormous labeled data, dense retrievers still suffer from a generalization issue, where they perform relatively poorly on novel domains in comparison to BM25.

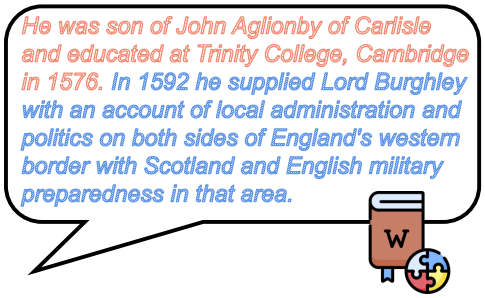

Figure 1: A text snippet from Wikipedia, where two nearby sentences are quite irrelevant. Random cropping may lead to a false positive query-passage pair.

Meanwhile, collecting human-annotated data for new domains is always hard and expensive. Thus improving dense retrievers with limited annotated data becomes essential, considering the significant domain variations of practical retrieval tasks.

Contrastive pre-training, which first generates pseudo-positive examples from a universal corpus and then utilizes them to contrastively pre-train retrievers, has shown impressive performance without any human annotations (Lee et al., 2019; Gao et al., 2021; Gao and Callan, 2022; Ram et al., 2022; Izacard et al., 2022). For instance, Contriever (Izacard et al., 2022) crafts relevant query-passage pairs by randomly cropping two random spans within the same document. However, owing to the high information density of texts, even nearby sentences in a document can be very irrelevant, as shown in Figure 1. These false positive samples may mislead the model to pull unrelated texts together in the embedding space and further harm the validity of representations.

Motivated by recent findings in computer vision that pre-training performance can be greatly boosted by reducing the effect of such false positives (Peng et al., 2022; Mishra et al., 2022), we propose Relevance-Aware Contrastive Retriever (ReContriever). At each training step, we utilize the trained models at the current step itself to estimate

* Work done when Yibin Lei was interning at JD Explore Academy.
† Corresponding author

Findings of the Association for Computational Linguistics: ACL 2023, pages 10932–10940
July 9-14, 2023 ©2023 Association for Computational Linguistics

the relevance of all the positives. Then the losses of different positive pairs are adaptively weighed using the estimated relevance, i.e., the pairs that receive higher relevance scores obtain higher weight. Moreover, simply applying lower weights to irrelevant pairs will result in insufficient usage of data, since many documents will contribute less to training. Therefore, we also introduce a one-document-multiple-pair strategy that generates multiple positive pairs from a single document, with a pair-weighting process conducted among samples originating from a single document. Such an operation makes sure that the model can learn positive knowledge from every document in the corpus.

To summarize, our contributions in this paper are three-fold: 1) We propose relevance-aware contrastive learning for dense retrieval pre-training, which aims to reduce the false positive problem. 2) Experiments show our method brings consistent improvements to the SOTA unsupervised Contriver model on 10/15 tasks on the BEIR benchmark and three representative open-domain QA retrieval datasets. 3) Further explorations show that our method works well given no or limited labeled data. Specifically, on 4 representative domain-specialized datasets it outperforms BM25 when only unsupervised pre-training on the target corpora, and with only a few annotated samples its accuracy can be on par with DPR (Karpukhin et al., 2020) which is trained on thousands of annotated examples.

2 Method

2.1 Preliminary

In this section, we briefly describe the bi-encoder structure used in dense retrieval and the SOTA Contriever model, on which we build our model.

Bi-Encoder Structure Dense retrievers are always a bi-encoder composed of two separate encoders to transform the query and document into a single vector each. The relevance score is obtained by computing the similarity (e.g., inner-product) between the encoded vectors of queries and documents. The typical way to train a dense retriever is using a contrastive loss that aims to pull relevant passages closer to the query and irrelevant passages farther in the embedding space. For each query, the training data involves one positive passage labeled by annotators and a pool of negative passages, which are usually random passages in the corpus.

Contriever It crafts pseudo-positive pairs by randomly cropping two spans of the same document. As negative texts have shown to be a key to the success of retrieval training (Xiong et al., 2021), Contriever also applies the MoCo mechanism (He et al., 2020) to utilize negatives in the previous batches to increase the number of negatives. These two factors make Contriever obtain significant decent performance without any human annotations.

2.2 Relevance-Aware Contrastive Learning

We start by 1) producing a larger number of positives (***one-document-multi-pair***) and 2) forcing the model to pay more attention to the ones with higher relevance (***relevance-aware contrastive loss***).

One-Document-Multi-Pair Given a text snippet T, previous pre-training methods always craft only one positive (query-passage) pair (q, d^+). To exploit T more effectively, our one-document-multi-pair strategy generates n positive pairs, denoting as $\{(q, d_1^+), (q, d_2^+), \ldots, (q, d_n^+)\}$, from T by repeating the procedure several times. We keep the query q unchanged to ensure the relevance comparison is fair among pairs within the same snippet, which is used in our following step. Building upon Contriever, we craft n pairs by random cropping $n + 1$ spans and setting 1 span as the fixed query for the left n spans. And it is easy to extend this strategy to other contrastive pre-training methods.

Relevance-Aware Contrastive Loss The ordinary contrastive loss for training dense retrievers is the InfoNCE loss. Given a positive pair (q, d^+) and a negative pool $\{d_i^-\}_{i=1..D}$, InfoNCE (q, d^+) is computed by:

$$-\log \frac{\exp\left(s\left(q, d^+\right)/\tau\right)}{\exp\left(s\left(q, d^+\right)/\tau\right) + \sum_{i=1}^{D} \exp\left(s\left(q, d_i^-\right)/\tau\right)}, \quad (1)$$

where $s(\cdot)$ and τ denote the similarity function and temperature parameter. Then the overall loss of a batch is usually the average across all the $m \times n$ positive pairs from m snippets: $L = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \text{InfoNCE}(q_i, d_{ij}^+)$.

The relevance-aware contrastive loss aims to force the model to focus more on true positive pairs by 1) utilizing trained model θ at present itself as an imperfect oracle to compute the relevance score $s_\theta(q, d^+)$ between all pairs; and 2) adaptively assigning weights to different pairs according to the

DATASET	BM25	BERT	SimCSE	RetroMAE	coCondenser	Contriever	Contriever (reproduced)	ReContriever
MS MARCO	**22.8**	0.6	8.8	4.5	7.7	20.6	21.1	21.8†
Trec-COVID	**65.6**	16.6	38.6	20.4	17.3	27.4	42.0	40.5
NFCorpus	**32.5**	2.5	14.0	15.3	14.4	31.7	30.0	31.9†
NQ	**32.9**	2.7	12.6	3.4	3.9	25.4	29.5	31.0†
HotpotQA	**60.3**	4.9	23.3	25.0	24.4	48.1	44.1	50.1†
FiQA-2018	23.6	1.4	14.8	9.3	5.2	24.5	**26.2**	**26.2**
ArguAna	31.5	23.1	**45.6**	37.6	34.5	37.9	43.4	39.8
Touche-2020	**36.7**	3.4	11.6	1.9	3.0	16.7	16.7	16.6
CQADupStack	**29.9**	2.5	20.2	17.0	9.8	28.4	28.4	28.7†
Quora	78.9	3.9	81.5	69.0	66.7	83.5	83.6	**84.3**†
DBPedia	**31.3**	3.9	13.7	4.6	15.1	29.2	27.6	29.3†
SCIDOCS	**15.8**	2.7	7.4	7.4	1.9	14.9	15.0	15.6†
FEVER	**75.3**	4.9	20.1	7.1	25.3	68.2	66.9	68.9†
Climate-fever	**21.3**	4.1	17.6	4.4	9.8	15.5	15.6	15.6
SciFact	**66.5**	9.8	38.5	53.1	48.1	64.9	65	66.4
Avg	**41.7**	8.7	24.6	18.7	8.9	35.8	37.0	37.8
Avg Rank	**1.9**	7.9	4.9	6.1	6.3	3.4	2.7	2.2

Table 1: **NDCG@10 of BEIR Benchmark**. All models are **unsupervised trained without any human-annotated data**. **Bold** indicates the best result. The average and rank across the entire benchmark are included. Four datasets are excluded because of their licenses. "†" means ReContriever performs significantly better than our reproduced Contriever, as determined by a t-test with p-value 0.05 as threshold.

estimated relevance. Then the relevance-aware contrastive loss $L_{\text{relevance}}$ can be expressed as:[1]

$$\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{s_\theta\left(q_i, d_{ij}^+\right)}{\sum_{k=1}^{n} s_\theta\left(q_i, d_{ik}^+\right)}\text{InfoNCE}(q_i, d_{ij}^+). \tag{2}$$

In this way, for each text snippet, positive pairs with more confidence to be relevant will thus be more focused on by the model, or vice versa.

3 Experiments

In this section, we evaluate our model in several settings after describing our experimental setup. We consider unsupervised retrieval performance and two practical use cases: further pre-training on the target domain and few-short retrieval. We then conduct an ablation study to separate the impact of our method's two components.

3.1 Setup

• **Datasets** We evaluate retrieval models on the BEIR (Thakur et al., 2021) benchmark and three representative open-domain QA retrieval benchmarks: Natural Questions (NQ; (Kwiatkowski et al., 2019)), TriviaQA (Joshi et al., 2017) and WebQuestions (Berant et al., 2013).

[1]Equation (2) will be invalid when $s_\theta\left(q_i, d_{ij}^+\right)$ is negative. In the preliminary study, we found the value is always positive and thus ignore this special case for simplicity.

• **Baselines** We compare our model with two types of unsupervised models, namely models based on contrastive pre-training and on auto-encoding pre-training. The former models include SimCSE (Gao et al., 2021), coCondenser (Gao and Callan, 2022), Spider (Ram et al., 2022) and Contriever (Izacard et al., 2022). The latter category includes the recently proposed RetroMAE (Xiao et al., 2022). BM25 (Robertson and Zaragoza, 2009) and uncased BERT-base model (Devlin et al., 2019) are also involved for reference. We use the official checkpoints for evaluation.

• **Implementation Details** We apply our method to the SOTA Contriever model and use its default settings. The pre-training data is a combination of Wikipedia and CCNet (Wenzek et al., 2020), same as Contriever. We generate 4 positive pairs for each document. Refer to Appendix A for more details. We conduct a t-test with p-value 0.05 as threshold to compare the performance of ReContriever and our reproduced Contriever.

3.2 Main Results

3.2.1 BEIR

The NDCG@10 of ReContriever and other fully unsupervised models across 15 public datasets of BEIR are shown in Table 1. ReContriever achieves consistent improvements over Contriever on 10/15 datasets, with a significant improvement observed in 9 of those datasets. Notably, it also only sees

Model	NQ			TriviaQA			WQ		
	Top-5	Top-20	Top-100	Top-5	Top-20	Top-100	Top-5	Top-20	Top-100
Supervised Model									
DPR	-	78.4	85.4	-	79.4	85.0	-	73.2	81.4
Unsupervised Models									
BM25	43.8	62.9	78.3	**66.3**	**76.4**	83.2	41.8	62.4	75.5
RetroMAE	23.0	40.1	58.8	47.0	61.4	74.2	25.8	43.8	62.3
SimCSE	5.4	11.5	23.0	3.7	7.6	17.0	3.3	8.7	19.4
coCondenser	28.9	46.8	63.5	7.5	13.8	24.3	30.2	50.7	68.7
Spider	49.6	68.3	81.2	63.6	75.8	83.5	46.8	65.9	79.7
Contriever	47.3	67.8	80.6	59.5	73.9	82.9	43.5	65.7	80.1
Contriever (reproduced)	48.9	68.3	81.4	61.2	74.6	83.4	47.0	67.0	80.5
ReContriever	**50.3**†	**69.4**†	**82.6**†	63.4†	75.9†	**84.1**†	**48.3**	**68.0**	**81.1**

Table 2: **Recall of open-domain retrieval benchmarks**. **Bold**: the best results across unsupervised models. "†" means ReContriever performs significantly better than our reproduced Contriever, as determined by a t-test with p-value 0.05 as threshold.

very slight decreases on datasets without promotion (e.g., FiQA, Touche and Climate-Fever with at most -0.1 decrease). Moreover, our method obtains an average rank of 2.2, proving our method to be the best unsupervised dense retriever. BM25 is still a strong baseline under the fully unsupervised scenario, but ReContriever greatly narrows the gap between dense retrievers and it.

3.2.2 Open-Domain QA Retrieval

Table 2 shows the Recall performance of ReContriever on open-domain QA retrieval benchmarks, where supervised DPR (Karpukhin et al., 2020) is involved for reference. Obviously, ReContriever outperforms BM25 by a large margin except for Recall@5 and Recall@10 on TriviaQA with relatively smaller differences, verifying the effect of our method. Moreover, among all unsupervised methods, ReContriever obtains the best performance in nearly all cases, especially substantial improvement over Contriever. Our ReContriever promisingly narrows the gaps between supervised and unsupervised models, making it more valuable.

Model	SciF	SCID	Arg	CQA	Avg.
BM25	66.5	15.8	31.5	29.9	35.9
Contriever	64.9	14.9	43.4	28.4	37.9
+ corpus pretrain	66.3	**17.1**	52.4	30.6	41.6$^{\Uparrow+3.7}$
ReContriever	66.4	15.6	39.8	28.4	37.6
+ corpus pretrain	**67.1**	16.6	**54.6**†	**30.7**	**42.3**$^{\Uparrow+4.7}$

Table 3: **NDCG@10 after further pre-training on the target domain corpus**. "⇑" denotes the gains of further pre-training. "†" means ReContriever performs significantly better than our reproduced Contriever.

3.3 Practical Use Cases

In this section, we explore the applicability of ReContriever in more practical scenarios[2], where only texts in the target corpus (pre-training on the target domain) or very limited annotated training data (few-shot retrieval) are available.

Model	NQ		
	Top-5	Top-20	Top-100
Reference			
DPR	-	78.4	85.4
BM25	43.8	62.9	78.3
8 examples			
Spider	49.7	68.3	81.4
Contriever	51.7	70.6	83.1
ReContriever	52.9	71.6	84.2†
32 examples			
Spider	50.2	69.4	81.7
Contriever	52.6	70.9	83.1
ReContriever	53.5	71.9†	84.7†
128 examples			
Spider	57.0	74.3	85.3
Contriever	55.1	72.4	83.7
ReContriever	55.9	74.1†	85.1†

Table 4: **Few-shot Retrieval on NQ**. Results are report with Recall. "†" means ReContriever performs significantly better than our reproduced Contriever.

Pre-Training on the Target Domain Four domain-specialized datasets (SciFact (Wadden et al., 2020) (SciF; citation-prediction), SCIDOCS (Cohan et al., 2020) (SCID; fact checking), ArguAna (Wachsmuth et al., 2018) (Arg; argument

[2]We report results of our reproduced Contriever as they are slightly better than the original ones (Izacard et al., 2022).

Model	MS MARCO	NFCoprus	NQ	Hotpot	FiQA	Touche	Quora	SCIDOCS	Avg.
Contriever	19.1	25.1	26.7	43.2	23.2	18.6	82.3	14.6	31.6
+ relevance-aware loss	0.2	2.5	0.0	0.2	0.5	0.6	57.6	14.5	9.5
+ one-document-multiple-pair	19.9	29.5	27.5	44.2	21.9	15.9	82.8	14.5	32.0
ReContriever	20.8	28.1	29.6	49.9	23.4	18.2	83.3	14.7	33.5

Table 5: **Ablation Study.** Results are reported with NDCG@10.

retrieval) and CQADupStack (Hoogeveen et al., 2015) (CQA; StackExchange retrieval) with only corpus available are picked as a testbed, shown in Table 3. Before further pre-training on the corresponding corpus, ReContriever underperforms BM25 on 3 of 4 datasets. Surprisingly, after pre-training, ReContriever is able to consistently beat BM25. Moreover, our model obtains an average +4.7 improvement after further pre-training, which is substantially better than Contriever (+3.7).

Few-Shot Retrieval. Results on training with limited annotated data of NQ are shown in Table 4. Following the same setting, our model trained on 128 samples can perform on par on Recall@100 with DPR which has seen thousands of annotated samples. In addition, when training data is scarce (below 100 examples), ReContriever still shows stronger few-shot performance compared to Spider and Contriever.

3.4 Ablation Study

We conduct an ablation study to investigate the contributions of our proposed loss and pairing strategies within ReContriever, using 100,000 training steps. Solely adding relevance-aware loss means estimating the relevance of N pairs from N documents and then normalizing the relevance among the N pairs within a batch, which slightly differs from equation (2) that normalizes over 4 pairs from the same document. As shown in Table 5, solely adding relevance-aware contrastive loss to Contriever will lead to a noticeable degeneration, owing to the missing information from the documents with low adjusted weights and the unstable relevance comparison without a fixed query. Applying the one-document-multi-pair strategy can obtain a slight improvement which can be attributed to the effective usage of the unlabeled data. Combining both strategies (i.e., ReContriever) can lead to an obvious improvement, which demonstrates the necessity of both components in our method.

4 Conclusion

In this work, we propose ReContriever to further explore the potential of contrastive pre-training to reduce the demand of human-annotated data for dense retrievers. Benefiting from multiple positives from the same document as well as relevance-aware contrastive loss, our model achieves remarkable performance under zero-shot cases. Additional results on low data resources further verify its value under various practical scenarios.

Limitations

Although ReContriever narrows the gap between BM25 and unsupervised dense retrievers, it still lags behind BM25 when acting as a general-purpose retriever. This issue may make ReContriever not directly usable when facing a new domain, thus limiting its practicality. Also, as ReContriever is initialized from the language model $\text{BERT}_{\text{base}}$, there may exist social biases (Zhao et al., 2017) in ReContriever and thus have the risk of offending people from under-represented groups.

Ethics Statement

We strictly adhere to the ACL Ethics Policy. This paper focuses on reducing the false positives problem of unsupervised dense retrieval. The datasets used in this paper are publicly available and have been widely adopted by researchers. We ensure that the findings and conclusions of this paper are reported accurately and objectively.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint*.

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

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER:

Document-level representation learning using citation-informed transformers. In *ACL*.

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

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *ACL*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *EMNLP*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*.

Doris Hoogeveen, Karin M. Verspoor, and Timothy Baldwin. 2015. Cqadupstack: A benchmark data set for community question-answering research. In *ADCS*.

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

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*.

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

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

Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model for web-scale retrieval in baidu search. In *KDD*.

Shlok Kumar Mishra, Anshul Shah, Ankan Bansal, Janit K Anjaria, Abhyuday Narayan Jagannatha, Abhishek Sharma, David Jacobs, and Dilip Krishnan. 2022. Object-aware cropping for self-supervised learning. *TMLR*.

Xiangyu Peng, Kai Wang, Zheng Zhu, Mang Wang, and Yang You. 2022. Crafting better contrastive views for siamese representation learning. In *CVPR*.

Ori Ram, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. 2022. Learning to retrieve passages without supervision. In *NAACL*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*

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 *NeurIPS Datasets and Benchmarks*.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *ACL*.

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*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, et al. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *LREC*.

Shitao Xiao, Zheng Liu, Shao Yingxia, and Cao Zhao. 2022. Retromae: Pre-training retrieval-oriented language models via masked auto-encoder. In *EMNLP*.

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*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *EMNLP*.

A Implementation Details

Basic Infrastructure Basic backbones of our implementation involve Pytorch[3] and Huggingface-Transformers[4]. We build our code upon the released code of Contriever[5]. Models are evaluated using evaluation scripts provided by the BEIR[6] (for BEIR evaluation) and Spider[7] (for open-domain QA retrieval evaluation) GitHub repositories. The pre-training experiments are conducted on 16 NVIDIA A100 GPUs and the few-shot experiments are conducted on a single NVIDIA A100 GPU. We report the results on a single run with a fixed random seed 0 (same as the setting of Contriever).

[3] https://pytorch.org/

[4] https://github.com/huggingface/transformers

[5] https://github.com/facebookresearch/contriever

[6] https://github.com/beir-cellar/beir

[7] https://github.com/oriram/spider

Details of ReContriever Following the default settings of Contriever, we pre-train ReContriever for 500,000 steps with a batch size of 2048, initializing from the uncased $\text{BERT}_{\text{base}}$ model with 110 million parameters. The pre-training data is a combination of Wikipedia and CCNet (Wenzek et al., 2020). The learning rate is set to $5 \cdot 10^{-5}$ with a warm-up for the first 20,000 steps and a linear decay for the remaining steps. Average pooling over the whole sequence is used for obtaining the final representation of the query or document. For each document, we generate 4 positive pairs.

For experiments on target domain pre-training, we initialize the model from our pre-trained Contriever/ReContriever checkpoints. To avoid overfitting, the models are further pre-trained with 5000 warm-up steps to a learning rate of $1.25 \cdot 10^{-7}$ on all 4 picked datasets with a batch size of 1024 on 8 NVIDIA A100 GPUs.

For few-shot retrieval experiments, we adopt the training procedure from (Karpukhin et al., 2020): exploiting BM25 negatives and not including negatives mined by the model itself (Xiong et al., 2021) for few-shot fine-tuning. The hyper-parameters are directly borrowed from (Karpukhin et al., 2020) except for the batch size and number of training epochs. We fine-tune all the models with 80 epochs. For 8 examples, the batch size is set to 8. The batch size is 32 when there are 32 or 128 examples.

B Dataset Statistics

Details about the number of examples in the there open-domain QA retrieval datasets are shown in Table 6.

Dataset	Train	Dev	Test
NQ	58880	8757	3610
TriviaQA	60413	8837	11313
WQ	2474	361	2032

Table 6: **Statistics of Open-Domain QA Retrieval Datasets**

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☑ A2. Did you discuss any potential risks of your work?
Section Limitations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3

☑ B1. Did you cite the creators of artifacts you used?
Section 3

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 3

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Our work is solely for research without any intention for commercial usage.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The datasets used in our work are publicly available and widely used.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix C

C ☑ Did you run computational experiments?

Section 3, Appendix A

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix B

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix B

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3, Appendix B

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix B

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.