PAIRDISTILL: Pairwise Relevance Distillation for Dense Retrieval

Anonymous ACL submission

Abstract

Effective information retrieval (IR) from vast datasets relies on advanced techniques to extract relevant information in response to queries. Recent advancements in dense passage retrieval (DPR) have showcased remarkable efficacy compared to traditional sparse retrieval methods. To further enhance retrieval performance, knowledge distillation techniques, often leveraging robust cross-encoder rerankers, have been extensively explored. However, existing approaches primarily distill knowledge from pointwise rerankers, which assign absolute relevance scores to documents, thus facing challenges related to inconsistent standards. This paper introduces Pairwise Relevance Distillation (PAIRDISTILL) to leverage pairwise reranking, offering fine-grained distinctions between similarly relevant documents to enrich the training of dense retrieval models. Our experiments demonstrate that PAIRDISTILL outperforms existing methods, achieving new state-of-the-art results across multiple benchmarks. This highlights the potential of PAIRDISTILL in advancing dense retrieval techniques effectively.[1]

1 Introduction

Information retrieval (IR) is the process of extracting relevant information from vast datasets, such as web pages or documents, based on user queries. Recently, deep learning methods, notably the dense passage retriever (DPR) (Karpukhin et al., 2020), have attracted attention for their superior performance compared to traditional sparse retrieval techniques like BM25. These methods, often termed dual-encoder models, encode both queries and documents into high-dimensional representations, facilitating efficient similarity computation and retrieval via nearest neighbor search (Douze et al., 2024).

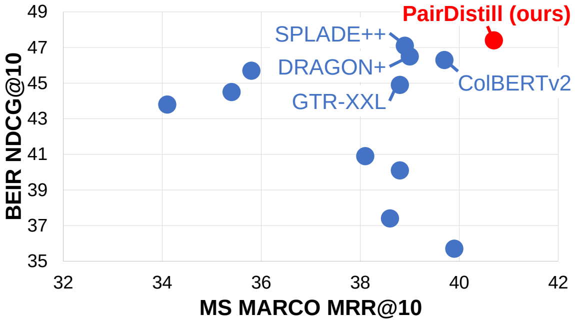

Figure 1: PAIRDISTILL, a model trained with our proposed pairwise relevance distillation, achieves the best performance in both in-domain evaluation (x-axis; MS MARCO dev set) and out-of-domain evaluation (y-axis; average performance over BEIR datasets).

Despite the effectiveness of dense retrievers, their modeling capacity is limited. To enhance retrieval performance, knowledge distillation is commonly employed (Izacard and Grave, 2020). Typically, knowledge from a robust cross-encoder reranker is distilled to train the dense retriever, achieving state-of-the-art results on retrieval benchmarks (Santhanam et al., 2022b). The efficacy of knowledge distillation largely relies on the performance of the reranker, which serves as the upper bound for the distilled retriever's performance.

However, existing studies primarily utilized *pointwise* rerankers for knowledge distillation, which an absolute relevance score for each document. Such scores are not trivial to compare due to inconsistent baselines. In contrast, *pairwise* reranking, an advanced technique comparing pairs of documents to assess their relative relevance to a query, has demonstrated superior reranking performance (Pradeep et al., 2021). By emphasizing relative comparison, pairwise rerankers can distinguish more finely between similarly relevant documents, yielding more precise relevance scores conducive to better distillation.

In this paper, we introduce Pairwise Relevance

[1] Our source code and trained models are released at `https://anonymous.4open.science/r/pair-distill-AE1F`

Distillation (PAIRDISTILL), a novel method leveraging the fine-grained training signals provided by pairwise rerankers. PAIRDISTILL enriches the training of dense retrieval models by distilling knowledge from pairwise comparisons, enabling the model to learn more nuanced distinctions between closely ranked passages. We conduct extensive experiments and demonstrate that PAIRDISTILL outperforms all baseline of similar size on multiple benchmark, as shown in Figure 1. In addition, we show that PAIRDISTILL is effective across difference architectures, i.e., ColBERT(Khattab and Zaharia, 2020) and DPR (Karpukhin et al., 2020), and in a domain adaptation setting. Furthermore, we demonstrate the potential of adopting LLM rerankers in PAIRDISTILL.

Our contributions are summarized as follows:

- We propose Pairwise Relevance Distillation (PAIRDISTILL), a novel method integrating the advantages of pairwise reranking into dense retrieval model training.
- Through extensive experiments, we demonstrate that PAIRDISTILL significantly outperforms existing dense retrieval models of similar size.
- We provide a comprehensive analysis, offering insights into the mechanisms driving the improvements achieved by PAIRDISTILL.

2 Related Work

Dense Passage Retrieval Dense retrieval has garnered attention for its efficacy in semantic space exploration. A notable technique in this domain is DPR (Karpukhin et al., 2020), employing both query and passage encoders for efficient retrieval. Various studies have delved into enhancing dense retrieval, including negative example mining techniques like RocketQA (Qu et al., 2021), and diverse data augmentation methods such as DRAGON (Lin et al., 2023a). ColBERT (Khattab and Zaharia, 2020; Santhanam et al., 2022b) introduced the late-interaction mechanism, offering an alternative architecture for dense retrieval.

Another line of research is pre-training strategies for dense retrieval. Approaches like Contriever (Izacard et al., 2021), coCondenser (Gao and Callan, 2022), and COCO-DR (Yu et al., 2022) have proposed contrastive pre-training techniques tailored for retrieval tasks. Concurrently, CoT-MAE (Wu et al., 2023) and RetroMAE (Xiao et al., 2022) have focused on masked auto-encoding for pre-training.

As large language models (LLMs) advance, their integration into dense retrieval has become prevalent. GTR (Ni et al., 2022) utilized LLM encoders, showcasing performance gains with increased model size. Similarly, Promptagator (Dai et al., 2023) and InPars (Bonifacio et al., 2022) leveraged LLMs to synthesize query-document pairs for training dense retrievers.

Our contribution is orthogonal to these studies as we concentrate on refining training signals for knowledge distillation. This suggests that our approach holds potential for integration with other methods to achieve further improvements.

Knowledge Distillation for Dense Retrieval Enhancing the performance of dense retrievers often involves employing knowledge distillation techniques. Izacard and Grave (2020) pioneered the distillation of knowledge from the reader to the retriever, resulting in improved performance in open-domain question answering. Following this, RocketQAv2 (Chakrabarty et al., 2022) and Margin-MSE (Hofstätter et al., 2020) proposed knowledge distillation from cross-encoder rerankers to enhance dense retrievers, while CL-DRD (Zeng et al., 2022) introduced curriculum learning for cross-encoder distillation. Further advancements include PROD (Lin et al., 2023b), which proposed a progressive distillation framework, and ABEL (Jiang et al., 2023), introducing an alternating distillation framework with impressive zero-shot performance. Our method introduces pairwise relevance distillation, leveraging finer-grained training signals from pairwise rerankers.

Passage Reranking Passage reranking serves as a pivotal second-stage process following initial large-scale retrieval efforts. Various studies have introduced deep reranking models that assess the relevance of query-document pairs by encoding them and predicting relevance scores (Nogueira and Cho, 2019). For instance, MonoT5 (Nogueira et al., 2020) introduced a generation-based method for passage reranking by fine-tuning LLMs on MS-MARCO (Bajaj et al., 2016), distinguishing relevant from irrelevant documents. DuoT5 (Pradeep et al., 2021) proposed pairwise reranking, simultaneously comparing two documents to significantly enhance reranking performance. TART (Asai et al., 2022) fine-tunes LLMs via multi-task instruction tuning on diverse retriever datasets.

Another line of research is zero-shot passage reranking with LLMs, which eliminates the need for retrieval supervision. UPR (Sachan et al., 2022) pioneered unsupervised passage reranking, proposing to rerank passages by estimating the conditional likelihood of generating the query given the passage using LLMs. Additionally, (Sun et al., 2023) and (Ma et al., 2023) both proposed listwise passage reranking by leveraging prompts with ChatGPT.

Our method combines the superior performance of pairwise reranking with knowledge distillation, which improves retrieval performance significantly and results in state-of-the-art performance on multiple benchmarks.

3 Background

In this section, we detail two key tasks: dense retrieval and passage reranking. Following that, we explore knowledge distillation, a widely adopted technique aimed at bolstering the efficacy of dense retrievers. Note that we interchangeably use the terms "passage" and "document" in this paper.

3.1 Dense Retrieval

The goal of dense passage retrieval is to retrieve a subset of relevant passages, denoted as D^+, from a large collection of passages $\mathcal{D} = \{d_1, \cdots, d_n\}$. In order to efficiently retrieve from millions of passages, the most common architecture used for dense retrieval is the dual encoder architecture, where the queries and the passages are encoded by a query encoder and a passage encoder, respectively. We denote the query representation of a query q as $\mathbf{q}$ and the passage representation of a passage d as $\mathbf{d}$. This architecture enables offline encoding and indexing of all passages, thus reducing the computation required significantly during retrieval.

The relevance of a query q to a passage d_i is measured using a similarity function:

$$\mathrm{s}(q, d_i) = \mathrm{Sim}(\mathbf{q}, \mathbf{d}_i),$$

where a higher similarity score indicates a greater relevance of the passage to the query. Common choices of the similarity function are dot product, cosine similarity, or the Max-Sum operator introduced in ColBERT (Khattab and Zaharia, 2020).

Given a labeled dataset of relevant passage-query pairs (q, d^+), dense retriever are typically trained with a contrastive learning objective such as the InfoNCE loss (Oord et al., 2018):

$$\mathcal{L}_{CL} = -\log \frac{\exp(\mathrm{s}(q, d^+))}{\sum_{d \in \mathcal{D}'} \exp(\mathrm{s}(q, d))},$$

where $\mathcal{D}'$ denotes the union of the positive and negative examples. Optimizing this objective promotes the similarity of the positive pair $s(q, d^+)$ in contrast to the negative examples.

3.2 Passage Reranking

Due to the computational constraints, most dense retrievers utilize lightweight models such as *bert-base* (Devlin et al., 2019) as their backbone model. Consequently, a subsequent stage of passage reranking aims to refine the initially retrieved passages. Similar to dense retrieval, the task of passage reranking also aims to assign a relevance score $s_{\text{point}}(q, d_i)$ to each passage d_i given a query q. This reranking scheme is called *pointwise reranking*, where all passages are scored independently. Given the reduced number of candidate passages at this stage, it becomes feasible to deploy more computationally intensive models. This allows for the use of cross-encoder architectures and larger models, which are adept at capturing the fine-grained interactions between queries and passages, offering relevance scores that are more accurate. The candidate passages are then reranked based on their relevance scores $s_{\text{point}}(q, d_i)$.

3.3 Knowledge Distillation for Dense Retrieval

Given the success of knowledge distillation of neural models (Hinton et al., 2015), a common approach to enhance the dense retrievers is distilling knowledge from the pointwise rerankers. Specifically, the relevance of a passage d_i to a query q predicted by a dense retrieval model can be defined as:

$$P(d_i \mid q) = \frac{\exp(s(q, d_i))}{\sum_{d \in \mathcal{D}'} \exp(s(q, d))}.$$

Similarly, the relevance predicted by a pointwise reranking model can be defined as:

$$P_{\text{point}}(d_i \mid q) = \frac{\exp(s_{\text{point}}(q, d_i)/\tau)}{\sum_{d \in \mathcal{D}'} \exp(s_{\text{point}}(q, d)/\tau)},$$

where τ is the temperature parameter for controlling the sharpness of the distribution. Finally, the loss function is the KL divergence between the two distributions:

$$\mathcal{L}_{KD} = \frac{1}{|\mathcal{B}|} \sum_{q \in \mathcal{B}} \mathrm{KL}(P_{\text{point}}(d \mid q) \parallel P(d \mid q)),$$

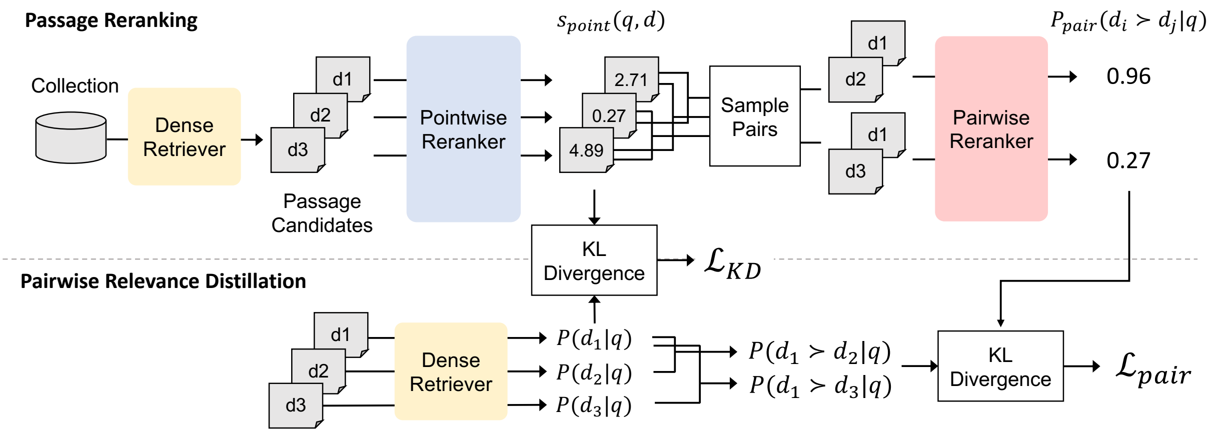

Figure 2: Illustration of our proposed method PAIRDISTILL. **Top**: The top-k retrieved passages go through pointwise reranking and pairwise reranking to obtain relevance scores. **Bottom**: Pairwise relevance distillation includes both pointwise distillation loss $\mathcal{L}_{KD}$ and pairwise distillation loss $\mathcal{L}_{pair}$.

where $|\mathcal{B}|$ denotes the size of the batch. By optimizing the KL divergence loss, the dense retriever learns to mimic the predictions of the pointwise reranker, thus improving its performance.

4 Our Method: PAIRDISTILL

In this section, we introduce our proposed method, pairwise relevance distillation (PAIRDISTILL). An illustration of the proposed framework is shown in Figure 2.

4.1 Pairwise Reranking

While the pointwise rerankers demonstrated superior performance over dense retrievers, reranking all passages independently poses a hard problem in calibrating the relevance score among passages, making the reranking performance of the pointwise rerankers suboptimal. We conduct preliminary analyses which can be found in Appendix A. To mitigate this problem, pairwise reranking techniques can be leveraged. Pairwise reranking produces better reranking results by comparing two passages simultaneously.

Formally, given a query q and two passages d_i and d_j, a pairwise reranker aims to estimate the probability that passage d_i is more relevant to the query than passage d_j:

$$s_{\text{pair}}(q, d_i, d_j) = P_{\text{pair}}(d_i \succ d_j \mid q). \tag{1}$$

This modeling choice effectively solve the calibration problem by only modeling the relative relevance of d_i and d_j. Note that in order to obtain the reranked list, an aggregation method is required which aggregates the relative relevance scores $s_{\text{pair}}(q)$. However, it is beyond the scope of this paper as our method does not require the final rankings. In this work, we adopt the following two pairwise reranking methods to estimate the pairwise relevance scores.

Classification-based The classification method involves training a binary classifier that predicts whether a given passage d_i is more relevant to a query q than another passage d_j. The classifier takes as input a triplet (q, d_i, d_j) and encodes them together in one sequence, allowing modeling the interaction among the query and two passages. The output of the classifier will be normalized via a sigmoid function, which can then be interpreted as the probability $P_{\text{pair}}(d_i \succ d_j \mid q)$. The training objective for this classifier is typically a binary cross-entropy loss, where the model is trained to minimize the difference between the predicted probability and the ground truth relevance ordering of the passages. This method requires a training dataset consists of triplets and their annotated relative relevance:

$$y = \begin{cases} 1 & \text{if } d_i \succ d_j \\ 0 & \text{otherwise} \end{cases}$$

Instruction-based In cases where training data is not available, we adopt instruction-based reranking with LLMs for zero-shot reranking. We instruct the LLM to select the passage that is more relevant to the query and assign the probability of selecting the index of d_i as the score.

$$P_{\text{pair}}(d_i \succ d_j \mid q) = P_{\text{LLM}}(i \mid q, d_i, d_j),$$

where $P_{\text{LLM}}(i \mid q, d_i, d_j)$ is the probability predicted by the LLM of d_i being more relevant to the query q than d_j. The detailed instructions for this method can be found in Appendix C.1.

4.2 Pairwise Relevance Distillation

Given the pairwise relevance scores from the pairwise reranker, we can leverage knowledge distillation to further enhance the performance of the dense retriever. The goal is to make the dense retriever imitate the output distribution of the pairwise reranker, which is defined above in Equation 1. To specify, we define the pairwise relevance distribution predicted by the dense retriever as:

$$P(d_i \succ d_j \mid q) = \frac{\exp(s(q, d_i))}{\exp(s(q, d_i)) + \exp(s(q, d_j))},$$

which applies the softmax function to the individual relevance scores $s(q, d_i)$ and $s(q, d_j)$. Consequently, the training objective for pairwise relevance distillation is defined as the KL divergence between the pairwise relevance distributions from the dense retreiver and the pairwise reranker:

$$\mathcal{L}_{pair} = \frac{1}{|\mathcal{B}|} \sum_{q \in \mathcal{B}} \Bigg(\sum_{d_i, d_j \sim \mathcal{D}_{pair}} \text{KL}\Big(P_{\text{pair}}(d_i \succ d_j \mid q) \parallel P(d_i \succ d_j \mid q) \Big) \Bigg),$$

where $\mathcal{D}_{pair} = \{(d_i, d_j) \mid d_i, d_j \in \text{ret}_k(q), i \neq j, |i - j| < \delta\}$ denotes the set of all possible pairs among $\text{ret}_k(q)$, which denotes the tok-k documents retrieved given the query q. We introduce a simple heuristic, $|i - j| < \delta$, to constrain the possible pairs, where δ is a hyperparameter. The intuition is that documents which are ranked further apart are less likely to provide meaningful training signal, as they are already easily distinguishable by the retriever.

In practice, the process begins by using a retriever to retrieve the top-k documents. These documents are then reranked by a pointwise reranker to refine the ranking and establish the top-k reranked documents. Finally, we apply pairwise reranking to the pointwise reranked documents, which allows us to derive pairwise relevance scores for the distillation process. The full loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{CL} + \lambda_{KD} \cdot \mathcal{L}_{KD} + \lambda_{pair} \cdot \mathcal{L}_{pair},$$

where λ_{KD} and λ_{pair} are hyperparameters representing the weight for the distillation losses. Our proposed method can also be applied to scenarios where no labeled training data is available. In such cases, the contrastive loss $\mathcal{L}_{CL}$ is discarded:

$$\mathcal{L}_{ZS} = \mathcal{L}_{KD} + \lambda_{pair} \cdot \mathcal{L}_{pair}.$$

4.3 Iterative Training

To enhance the performance of the retriever and mitigate the risk of overfitting to a static set of top-k passages, we adopt an iterative training strategy. In each iteration, the retriever trained in the previous iteration is used to build an index and retrieve the top-k documents. Subsequently, the top-k documents are reranked with pointwise reranking and pairwise reranking, and the trained retriever is fine-tuned with pairwise relevance distillation. The fine-tuned retriever then becomes the retriever for the next iteration. This iterative training allows for refreshing the retrieved documents in each iteration, avoiding training on the fixed set of documents. Furthermore, the performance of the retriever can be improved iteratively.

5 Experiments

Our proposed method, pairwise relevance distillation, can be applied to both supervised datasets and zero-shot domain adaptation tasks. In this section, we conduct extensive experiments on passage retrieval tasks to validate and analyze the effectiveness of the proposed method.

5.1 Datasets

Following previous work, we use MS MARCO (Bajaj et al., 2016) as the supervised dataset to perform knowledge distillation. We evaluate our model on the official dev set of MS MARCO. Additionally, we perform zero-shot evaluation on TREC 19 and 20 (Craswell et al., 2020, 2021), BEIR (Thakur et al., 2021), and LoTTE (Santhanam et al., 2022b). Detailed description of the datasets can be found in Appendix B.1.

We report evaluation metrics based on the common practice of each dataset: MRR@10 and Recall@1000 for MS MARCO, NDCG@10 for TREC and BEIR, and Success@5 for LoTTE.

5.2 Implementation Details

We adopt the pretrained ColBERTv2 (Santhanam et al., 2022b) as the initial retriever with the PLAID engine (Santhanam et al., 2022a) using their official implementation[2]. Following ColBERTv2, we

[2] https://github.com/stanford-futuredata/ColBERT

Representation	**Sparse**	**Dense**													**Mul-vec**	
Model	SPLADE++	GTR-XXL	CL-DRD	RocketQAv2	CoT-MAE	RetroMAE	coCondenser	Contriever	DRAGON+	ABEL-FT	COCO-DR	GPL	PTR	ColBERTv2	PairDistill (Ours)
Pre-training	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗
Distillation	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✗	✓	✗	✓	✓
Target Corpus	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗
MS MARCO (Supervised)															
Dev (RR@10)	38.9	38.8	38.1	38.8	39.9†	35.4	38.6	34.1	39.0	-	35.8	-	-	39.7	**40.7**
Dev (R@1K)	98.2	**99.0**	97.9	98.1	98.5	97.5	98.4	97.9	98.6	-	97.9	-	-	98.4	98.5
DL2019	74.3	-	72.5	-	70.0	68.8	71.5	67.8	74.4	-	74.1	-	-	74.6	**75.2**
DL2020	71.8	-	68.3	-	67.8	71.4	68.1	66.1	72.3	-	69.7	-	-	**75.2**	75.1
BEIR (Zero-shot)															
TREC-COVID	71.1	50.1	58.4	67.5	56.1	77.2	71.2	59.6	75.9	76.5	**78.9**	70.0	72.7	73.2	74.2
NFCorpus	34.5	34.2	31.5	29.3	32.1	30.8	32.5	32.8	33.9	35.1	**35.5**	34.5	33.4	33.9	34.5
FiQA-2018	35.1	**46.7**	30.8	30.2	28.3	31.6	27.6	32.9	35.6	34.3	31.7	34.4	40.4	35.6	37.1
ArguAna	52.1	54.0	41.3	45.1	27.8	43.3	29.9	44.6	46.9	**56.9**	49.3	55.7	53.8	45.8	46.8
Tóuche-2020	24.4	25.6	20.3	24.7	21.9	23.7	19.1	23.0	26.3	19.5	23.8	25.5	**26.6**	26.5	26.4
Quora	81.4	**89.2**	82.6	74.9	75.6	84.7	85.6	86.5	87.5	84.5	86.7	83.6	-	85.1	85.3
SCIDOCS	15.9	16.1	14.6	13.1	13.2	15.0	13.7	16.5	15.9	**17.4**	16.0	16.9	16.3	15.5	16.2
SciFact	69.9	66.2	62.1	56.8	60.1	65.3	61.5	67.7	67.9	**72.6**	70.9	67.4	62.3	69.1	71.5
NQ	54.4	56.8	50.0	50.5	48.3	51.8	48.7	49.5	53.7	50.2	50.5	48.3	-	56.3	**58.3**
HotpotQA	68.6	59.9	58.9	53.3	53.6	63.5	56.3	63.8	66.2	65.7	61.6	58.2	60.4	67.4	**69.3**
DBPedia	44.2	40.8	38.1	35.6	35.7	39.0	36.3	41.3	41.7	41.4	39.1	38.4	36.4	44.6	**46.0**
FEVER	79.6	74.0	73.4	67.6	50.6	77.4	49.5	75.8	78.1	74.1	75.1	75.9	76.2	79.0	**80.4**
Climate-FEVER	22.8	**26.7**	20.4	18.0	14.0	23.2	14.4	23.7	22.7	21.8	21.1	23.5	21.4	18.2	19.4
CQADupStack	34.1	**39.9**	32.5	-	29.7	34.7	32.0	34.5	35.4	36.9	37.0	35.7	-	36.7	38.0
Robust04	45.8	**50.6**	37.7	-	30.8	44.7	35.4	47.6	47.9	50.0	44.3	43.7	-	46.8	48.7
Signal-1M	29.6	27.3	28.2	-	21.1	26.5	28.1	19.9	30.1	28.0	27.1	27.6	-	30.7	**31.2**
TREC-NEWS	39.4	34.6	38.0	-	26.1	42.8	33.7	42.8	44.4	**45.4**	40.3	42.1	-	42.0	41.9
BioASQ	50.4	32.4	37.4	-	26.2	42.1	25.7	38.3	43.3	45.4	42.9	44.2	-	52.2	**54.8**
Avg. PTR-11	47.1	44.9	40.9	40.1	35.7	44.5	37.4	43.8	46.5	46.9	45.7	45.5	45.5	46.3	**47.4**
Avg. BEIR-13	50.3	49.3	44.8	43.6	39.8	48.2	42.0	47.5	50.2	50.0	49.2	48.6	-	50.0	**51.2**
Avg. All-18	47.4	45.8	42.0	-	36.2	45.4	38.9	44.5	47.4	47.5	46.2	45.9	-	47.7	**48.9**
LoTTE (Zero-shot)															
Search (pooled)	70.9	-	65.8	69.8	63.4	66.8	62.5	66.1	73.5	-	67.5	-	-	71.4	**73.9**
Forum (pooled)	62.3	-	55.0	57.7	51.9	58.5	52.1	58.9	62.1	-	56.8	-	-	63.2	**65.5**

Table 1: Retrieval performance on benchmarks (%). We report NDCG@10 for MS MARCO and BEIR unless otherwise noted. Recall@5 is reported for LoTTE following previous work. The best result for each dataset is **bolded** and the second best result is underlined. †The model was trained on a non-standard MS MARCO corpus which includes the title of the passages.

employ MiniLM[3] as the pointwise cross-encoder reranker (Thakur et al., 2021), which achieves comparable performance as MonoT5-3B[4] in our preliminary experiment. We adopt duoT5-3B[5] (Pradeep et al., 2021) as our pairwise reranker, which is trained on MS MARCO. We will discuss the feasibility of using instruction-based reranking with LLMs in Section 6.2.

For each query, we retrieve top-100 passages from the MS MARCO collection and perform pointwise reranking. We sample 50 pairs of passages from all possible pairs and obtain pairwise relevance scores through pairwise reranking. We use all 800K queries for knowledge distillation, while the 500K labeled queries are used for contrastive learning. δ is set to 10 in our experiments.

[3] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

[4] https://huggingface.co/castorini/monot5-3b-msmarco

[5] https://huggingface.co/castorini/duot5-3b-msmarco

All experiments are conducted with 4 V100 GPUs with 32GB memory each. Detailed hyperparameters can be found in Appendix C.2.

	NQ	TriviaQA	SQuAD
BM25	44.6	67.6	50.6
SPLADEv2	65.6	74.7	60.4
ColBERTv2	68.9	76.7	65.0
PAIRDISTILL	**71.8**	**77.4**	**66.9**

Table 2: Recall@5 performance on open-domain question answering datasets (%).

5.3 Main Results

We compare the performance of our proposed PAIRDISTILL to various baseline models, including state-of-the-art models, e.g., SPLADE++, ColBERTv2, DRAGON+, and ABEL-FT. The evaluation results on MS MARCO, BEIR, and LoTTE are shown in Table 1. Note that we follow Lin et al. and compare with models trained on MS MARCO without title for a fair comparison.

5.3.1 In-domain Evaluation

Following previous work (Santhanam et al., 2022b; Lin et al., 2023a; Jiang et al., 2023), we consider MS MARCO dev set, TREC DL19 and DL20 as in-domain evaluation sets. As shown in Table 1, our proposed method PAIRDISTILL achieves 40.7 in terms of MRR@10, which is the best performance on MS MARCO Dev set. Our model significantly outperforms ColBERTv2 (40.7 v.s. 39.7), which is the initialization of our model. This result demonstrates that the proposed pairwise relevance distillation effectively improves the performance of dense retrievers. PAIRDISTILL also achieves the best performance on TREC DL19 and the second best performance on TREC DL20. Note that coCondenser and CoT-MAE are fine-tuned on the MS MARCO passage corpus that has been augmented with title, which makes their performance not directly comparable to our method.

5.3.2 Out-of-domain Evaluation

Next, we evaluate the trained model on out-of-domain evaluation dataset to validate its generalizability. On the BEIR evaluation datasets (Thakur et al., 2021), PAIRDISTILL achieves the best overall performance in three different subsets, demonstrating that our model also excels at out-of-domain generalization. Considering individual datasets, PAIRDISTILL achieves the best performance among all compared models in 6 out of 18 tasks. Notably, our method outperforms domain-specific models, e.g., ABEL-FT (Jiang et al., 2023) and Promptagator (Dai et al., 2023), which leverage the target domain corpus for specialized domain adaptation. Additionally, our method consistently outperforms ColBERTv2 in 16 out of 18 datasets, showing that pairwise relevance distillation offers consistent out-of-domain improvement.

On the LoTTE evaluation sets (Santhanam et al., 2022b), PAIRDISTILL achieves state-of-the-art performance in both search and forum subsets, significantly outperforms all compared models. Notably, DRAGON+ (Lin et al., 2023a) performs comparably to our model in the search subset, which shows that diverse data augmentation might further improve our model in this scenario.

We also evaluate our model on open-domain question answering datasets, i.e., NaturalQuestions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and SQuAD (Rajpurkar et al., 2016). We follow ColBERTv2 (Santhanam et al., 2022b) which reports the performance on the dev set of each dataset in terms of Recall@5. The results are reported in Table 2. PAIRDISTILL consistently outperforms all baseline models on all datasets, demonstrating that our method is suitable for retrieving passages for open-domain question answering as well.

6 Discussions

6.1 Ablation Study

We conduct ablation studies on MS MARCO dev set to assess the effectiveness of each component in PAIRDISTILL. Table 3 shows the results of the ablation studies.

In the first experiment, we remove each distillation loss during training. Removing both $\mathcal{L}_{pair}$ and $\mathcal{L}_{KD}$ results in degraded performance. Notably, training with only $\mathcal{L}_{pair}$ slightly hurts performance. Our hypothesis is that since our pairwise distillation objective effectively demotes the score of the lower-ranked passage, we might demote the passage too much during training if we do not refresh the top-k passages. We also remove the heuristic for pair sampling, where we sample from all possible pairs. Removing the heuristic shows slight degradation, demonstrating the heuristic contributes to the improvement.

Next, as ColBERTv2 is an already well-trained

	MS MARCO Dev
Distillation Loss	
PairDistill	**40.7**
- $\mathcal{L}_{pair}$	39.7
- $\mathcal{L}_{KD}$	39.4
- pair sampling heuristic	40.3
Initialization	
ColBERTv2	**40.7**
bert-base-uncased	40.3
Different Architecture	
DPR	34.8
+ $\mathcal{L}_{KD}$	36.1
+ $\mathcal{L}_{KD} + \mathcal{L}_{pair}$	**36.8**
Iterative Training	
Iteration 1	40.2
Iteration 2	**40.7**
Iteration 3	**40.7**

Table 3: Results of ablation studies. We report performance on MS MARCO dev set by removing components of our proposed method.

model, we train our model with different initializations to verify if our method is effective for other pretrained models. As the results demonstrate, initializing our model with bert-base-uncased achieves 40.3 on MS MARCO dev set. This result shows that our method is effective regardless of the initialization.

Our proposed method is agnostic to the architecture used for dense retrieval as long as it produces a relevance score for each query-passage pair. Therefore, we conduct experiments with a different dense retrieval architecture, i.e., DPR (Karpukhin et al., 2020), to verify if the improvement is consistent across different architectures. Experimental results shows consistent improvement over vanilla DPR, where using both pointwise and pairwise distillation losses achieves the best performance. This result demonstrates that our proposed method can improve performance across different dense retrieval architectures.

Finally, we evaluate our trained models from each iteration to verify the effectiveness of the iterative training framework. The result shows that we can achieve state-of-the-art performance with only 1 iteration, while the second iteration further improves the result. The improvement converges after 2 iteration.

	FiQA	BioASQ	C-FEVER
ColBERTv2	35.6	52.2	18.2
PairDistill	37.1	54.8	19.4
Domain Adaptation			
$\mathcal{L}_{KD}$ only	38.2	57.0	21.4
$\mathcal{L}_{pair}$	**39.5**	**59.4**	**22.6**

Table 4: Performance of zero-shot domain adaptation on FiQA, BioASQ, and Climate-FEVER.

6.2 Zero-shot Domain Adaptation

As discussed in Section 4.1, it is possible to leverage LLMs to perform zero-shot instruction-based reranking. In this section, we conduct a study where we utilize LLMs for zero-shot domain adaptation. Specifically, we replace the supervised rerankers with LLMs (flan-t5-xl) for instruction based pointwise and pairwise reranking.

To evaluate the effectiveness of zero-shot domain adaptation with LLMs, we select 3 datasets from BEIR, FiQA, BioASQ, and Climate-FEVER, where training queries are available. Note that our method only utilize the queries, not the labeled pairs. We fine-tune ColBERTv2 with $\mathcal{L}_{ZS}$ on each dataset and evaluate the models on the corresponding test set.

Table 4 shows the results of zero-shot domain adaptation. Training with $\mathcal{L}_{pair}$ consistently improves performance in the target domain compared to using $\mathcal{L}_{KD}$ only and the baseline models trained on MS MARCO only. The results demonstrate that performing domain adaptation on queries from the target domain with LLMs are effective.

7 Conclusion

In this paper, we introduce Pairwise Relevance Distillation (PairDistill), a novel distillation method for dense retrieval that leverages the finer-grained training signal provided by the pairwise rerankers. Through extensive experiments, we demonstrate that PairDistill achieves state-of-the-art performance in both in-domain and out-of-domain evaluation. Further analyses show that the proposed method offers consistent improvements across domains and architectures. We hope this study could provide insights into distillation methods for dense retrieval and prompt more advance distillation techniques.

8 Limitations

While the proposed method leverages pairwise relevance for enhancing the training of dense retrievers, it is important to acknowledge certain limitations. One notable concern is the potential requirement for a larger number of training pairs compared to methods utilizing pointwise relevance. This reliance on a larger volume of training pairs may pose challenges in terms of computational resources required for training.

Therefore, future work in this domain should focus on addressing this limitation by exploring strategies to mitigate the need for an extensive number of training pairs while maintaining or even improving the effectiveness of knowledge distillation. This could involve investigating techniques to optimize the selection of training pairs to reduce the computational cost. Addressing the challenge of reducing the required training pairs for knowledge distillation would contribute to the scalability and applicability of the proposed method in real-world retrieval scenarios.

References

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. 2022. Task-aware retrieval with instructions. *arXiv preprint arXiv:2211.09260*.

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

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2387–2392.

Tuhin Chakrabarty, Yejin Choi, and Vered Shwartz. 2022. It's not rocket science: Interpreting figurative language in narratives. *Transactions of the Association for Computational Linguistics*, 10:589–606.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track. corr abs/2102.07662 (2021). *arXiv preprint arXiv:2102.07662*.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv preprint arXiv:2401.08281*.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853, Dublin, Ireland. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning.

Gautier Izacard and Edouard Grave. 2020. Distilling knowledge from reader to retriever for question answering.

Fan Jiang, Qiongkai Xu, Tom Drummond, and Trevor Cohn. 2023. Boot and switch: Alternating distillation for zero-shot dense retrieval. *arXiv preprint arXiv:2311.15564*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

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

2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

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. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023a. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*.

Zhenghao Lin, Yeyun Gong, Xiao Liu, Hang Zhang, Chen Lin, Anlei Dong, Jian Jiao, Jingwen Lu, Daxin Jiang, Rangan Majumder, et al. 2023b. Prod: Progressive distillation for dense retrieval. In *Proceedings of the ACM Web Conference 2023*, pages 3299–3308.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy J. Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *ArXiv*, abs/2101.05667.

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 *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022a. Plaid: an efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1747–1756.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022b. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agent. *arXiv preprint arXiv:2304.09542*.

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 (Round 2)*.

Xing Wu, Guangyuan Ma, Meng Lin, Zijia Lin, Zhongyuan Wang, and Songlin Hu. 2023. Contextual masked auto-encoder for dense passage retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4738–4746.

Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. RetroMAE: Pre-training retrieval-oriented language models via masked auto-encoder. In *Proceedings of the 2022 Conference on Empirical Methods in*

Natural Language Processing, pages 538–548, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Peilin Yang, Hui Fang, and Jimmy J. Lin. 2018. Anserini: Reproducible ranking baselines using lucene. *ACM J. Data Inf. Qual.*, 10:16:1–16:20.

Yue Yu, Chenyan Xiong, Si Sun, Chao Zhang, and Arnold Overwijk. 2022. COCO-DR: Combating distribution shift in zero-shot dense retrieval with contrastive and distributionally robust learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1462–1479, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hansi Zeng, Hamed Zamani, and Vishwa Vinay. 2022. Curriculum learning for dense retrieval distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1979–1983.

MSMARCO	MRR@10
ColBERTv2	39.7
MiniLM (pointwise)	40.5
MonoT5 (pointwise)	40.6
duoT5 (pairwise)	41.5

Table 5: Reranking performance of different rerankers (%).

λ_{pair}	MS MARCO Dev
1.0	40.3
3.0	40.7

Table 6: Results of varying the value of λ_{pair}.

A Additional Analyses

A.1 Reranking Performance

In order to better motivate the proposed method, we compare the reranking performance of the pairwise reranker to pointwise rerankers. Results are shown in Table 5. The results demonstrate that pairwise reranking offers greater reranking performance, which makes better distillation targets.

A.2 Difference between pairwise and pointwise reranking

In addition to the reranking performance, we conduct another experiment to analyze the difference between pairwise and pointwise rerankers. In this experiment, we compare the pairwise rank disagreement rate between the rerankers. We found that the pointwise reranker (MiniLM) disagrees with the more accurate pairwise reranker (duoT5) in 31% of the pairs sampled via our heuristic. This result shows that pairwise rerankers provide very different distillation targets for the retrievers. Combined with the fact that pairwise reranker achieves higher reranking performance, we believe that these experiments demonstrate the necessity of the proposed pairwise relevance distillation.

A.3 Effect of hyperparameters

We conduct an experiment where we vary the value of the hyperparameter λ_{pair}. The results are shown in Table 6. As shown in the results, varying the value of λ_{pair} has a slight effect on the final performance. Setting the value to 3.0 achieves the best performance.

B Evaluation Details

B.1 Dataset Details

- **MS MARCO** (Bajaj et al., 2016): Following previous work (Santhanam et al., 2022b; Lin et al., 2023a; Jiang et al., 2023), we use MS MARCO as the supervised dataset, which consists of 502K training queries with 8.8 million passages in the collection. Additionally, there are 306K unlabeled queries that can be used for distillation. The main evaluation is conducted on the official dev set of MS MARCO, which is a standard evaluation set.
- **TREC** (Craswell et al., 2020, 2021): We also perform evaluation on the TREC DL19 and DL20 evaluation sets, which are consider as in-domain datasets as they use the same collection as MS MARCO.
- **BEIR** (Thakur et al., 2021): BEIR is a benchmark consisting of 18 retrieval datasets, aiming to assess the out-of-domain retrieval performance of retrievers. We conduct zero-shot evaluation on all 18 datasets.
- **LoTTE** (Santhanam et al., 2022b): LoTTE consists of questions and answers posted on StackExchange with five topics including writing, recreation, science, technology, and lifestyle. A pooled set is also provided where passages and queries from all five topics are aggregated.

B.2 Baseline Models

We mostly follow the evaluation procedure from the prior work. In Table 1, most results are refered directly from DRAGON (Lin et al., 2023a) and ABEL-FT (Jiang et al., 2023). We reran all results of ColBERTv2 to offer a fair comparison to our method. All evaluation results are computed with the trec_eval tool from Anserini (Yang et al., 2018).

For the open-domain question answering datasets, all baseline results are referred directly from ColBERTv2 (Santhanam et al., 2022b).

B.3 Inference

During inference, we utilize the PLAID engine (Santhanam et al., 2022a) for efficient indexing and retrieval. Following prior work (Santhanam et al., 2022b), we set the maximum length of documents to 300 for BEIR and LoTTE. The maximum length of queries is set to 300 for Arguana and 64 for Climate-Fever. We set the compression to 2 bits in the PLAID engine.

C Implementation Details

C.1 Instruction-based Reranking

For pointwise reranking, we use the following instruction:

```
Is the document relevant to the query
(Yes or No)?
Query: {query}
Document: {document}
```

For pairwise reranking, we use the following instruction:

```
Which document is more relevant to the query?
Answer only 'A' or 'B'.
Query: {query}
Document: {document}
```

C.2 Hyperparameters

The hyperparameters used for pairwise relevance distillation training are listed in Table 7

hyperparameters	
batch size	32
# passages per question	64
max passage length	180
max query length	32
max training steps	100000
learning rate	1e-5
optimizer	AdamW
temperature τ	1.0
λ_{KD}	1.0
λ_{pair}	3.0

Table 7: Hyperparameters used in the knowledge distillation stage.