2/9/2026 3:04:36 PM

Compare Results

Old File:		New File:
2024.emnlp-main.336.pdf	versus	**2024_emnlp-main_336.pdf**
13 pages (289 KB)		**13 pages (385 KB)**
10/31/2024 10:32:46 PM		2/9/2026 12:50:45 PM

Total Changes

1081

Content

123 Replacements

183 Insertions

249 Deletions

Styling and Annotations

363 Styling

163 Annotations

Go to First Change (page 2)

MTA4DPR: Multi-Teaching-Assistants based Iterative Knowledge Distillation for Dense Passage Retrieval

Author1*
lqxaixxh@gmail.com
Author2†
{edxun, gongbo.tang}@blcu.edu.cn

Abstract

Although Dense Passage Retrieval (DPR) models have achieved significantly enhanced performance, their widespread application is still hindered by the demanding inference efficiency and high deployment costs. Knowledge distillation is an efficient method to compress models, which transfers knowledge from strong teacher models to weak student models. Previous studies have proved the effectiveness of knowledge distillation in DPR. However, there often remains a significant performance gap between the teacher and the distilled student. To narrow this performance gap, we propose MTA4DPR, a Multi-Teaching- Assistants based iterative knowledge distillation method for Dense Passage Retrieval, which transfers knowledge from the teacher to the student with the help of multiple assistants in an iterative manner; with each iteration, the student learns from more performant assistants and more difficult data. The experimental results show that our 66M student model achieves the state- of- the-art performance among models with same parameters on multiple datasets, and is very competitive when compared with larger, even LLM- based, DPR models.

1 Introduction

Although PLM/LLM- based Dense Passage Retrieval (DPR) models (Karpukhin et al., 2020; Qin et al., 2024) have superior performance, those models' inference efficiency and deployment costs are still cumbering their wide applications. To obtain an efficient and effective DPR model, researchers are paying more attention to knowledge distillation. Previous studies (Zeng et al., 2022; Sun et al., 2024; Lu et al., 2022) have proved the effectiveness of knowledge distillation in DPR. However, the performance gap between the teacher and the distilled student often remains significant, especially when the teacher is a very good one.

figures/framework.png

Figure 1: MTA4DPR Framework. MTA4DPR transfers knowledge from the teacher to the student with the help of the best assistant. The Fusion Module is used to generate fused assistants from the original assistants, and the Selection Module is used to select the best assistant among all original and fused assistants. The dotted arrows indicate that the corresponding procedures are not involved in the backpropagation of the training.

In this paper, we hypothesize that incorporating assistants into knowledge distillation can help improve students' performance, just as teaching assistants in universities can assist students in learning course content. In addition, inspired by curriculum learning (Bengio et al., 2009), we also believe that multiple iterations can further narrow the gap between the teacher and the student since the latter is capable of learning from more challenging data and more effective assistants as the iterations go on. Therefore, we introduce MTA4DPR, a multi- teaching- assistants based iterative distillation method. Specifically, MTA4DPR transfers knowledge from the teacher to the student with the help of multiple assistants iteratively. For each iteration, we first use off- the- shelf teacher/assistant DPR models to generate datasets for training and

*1Beijing Advanced Innovation Center for Language Resources, Beijing Language and Culture University, China
†2School of Information Science, Beijing Language and Culture University, China

evaluation. Then, we use a fusion module to generate a series of fused assistants. After that, we train the student to learn from the teacher with the help of the best assistant selected among all fused and original assistants by our selection module, as illustrated in Figure ??. At the end of each iteration, we evaluate the student's performance and replace the worst- performing assistant with it if it outperforms any existing assistants. What's more, we also incorporate data that the student predicted incorrectly in the previous iteration into the newly constructed dataset, by which the difficulty of each iteration's dataset is increased. In this way, as the training iterates, the student can learn from more performant assistants and more difficult data.

The experimental results on MS MARCO, TREC DL 2019 and 2020 and Natural Questions show the effectiveness of our method. Our 66M student model achieves the state- of- the- art performance among models with same parameters on multiple datasets, and is competitive when compared with larger, even LLM- based, DPR models.

To summarize, our main contributions are:

1. We propose a novel distillation method MTA4DPR which improves the student's retrieval performance with the help of assistant models.

2. The experimental results show the effectiveness of our proposed method, achieving very competitive results even when compared with larger, even LLM-based, DPR models.

3. Not constrained by model structures and tasks, MTA4DPR is orthogonal to existing distillation methods and can be combined with other distillation pipelines to further improve the performance.

2 Related Work

2.1 Dense Retrieval

Despite its wide applications, sparse retrieval, such as BM25, can not thoroughly solve the lexical mismatch problem, although query/document expansion (Nogueira et al., 2019; Formal et al., 2021) and term- weighting (Lin and Ma, 2021; Gao and Callan, 2021a) have been proposed to help mitigate the problem. For this reason, dense retrievers, especially those built upon PLMs or LLMs, have received more and more attention. They map both passages and queries into dense vectors, the relevance between which can be computed by dot products. Recently, a large number of methods have been proposed to improve dense retrievers' performance, including negative sampling (Xiong et al.), knowledge distillation (Zeng et al., 2022; Sun et al., 2024; Lin et al., 2023) and joint optimization of retrievers and rankers (Ren et al., 2021b).

2.2 Knowledge Distillation

Knowledge Distillation transfers knowledge from the teacher to the student, allowing the latter to have good performance with high efficiency. To achieve this goal, students are forced to learn knowledge representations provided by teachers, including response-based knowledge (Hinton et al., 2015; Beyer et al., 2022), intermediate knowledge (Adriana et al., 2015; Chen et al., 2018; Heo et al., 2019) and relation-based knowledge (Peng et al., 2019; Huang et al., 2022; Yang et al., 2022).

Recently, more and more studies focus on multi-teacher distillation, which can draw diverse knowledge from multiple teacher models, improving the student model's performance(Wu et al., 2021; Son et al., 2021; Lin et al., 2023). Mirzadeh et al. (2020) proposes TAKD, a multi- step knowledge distillation method to bridge the gap between the teacher and the student, in which a larger teacher model distills a smaller teacher model and the latter distills a much smaller student model. Yuan et al. (2021) proposes a reinforced method to combine multiple teacher models' prediction to get the final knowledge, which is used to distill the student model. In all the above studies, researchers tend to treat all teachers equally, combining their predictions using various strategies to train the student model. We argue that treating all teachers equally might be suboptimal given their varying performance.

Different from previous studies, in MTA4DPR, the best- performing model is considered as the primary teacher and involved in the entire training process, while the remaining models serve as assistants, only one of which participates in each training batch. This concept can be analogized to university students learning from a professor with the help of multiple assistants, only one of which is selected for each topic based on their speciality. Furthermore, we experiment with iteratively replacing underperforming assistants with better-performing ones.

3 Methodology

3.1 Preliminary

3.1.1 Task Description

Assume we have a training set $\mathcal{D} = \{(q_i, \mathbb{P}_i, \mathbb{S}_i)\}_{i=1}^{n}$ where q_i is the query, $\mathbb{P}_i$ consists of a positive passage p_i^+ and k hard negatives $\mathbb{P}_i^- = \{p_{i,j}^-\}_{j=1}^{k}$ (passages that are difficult to distinguish from the positive passage) and $\mathbb{S}_i = \{\mathcal{S}_{i,1}, \mathcal{S}_{i,2}, \ldots, \mathcal{S}_{i,d}, \ldots\}$ con-

sists of relevance scores computed by the teacher/assistants and $\mathcal{S}_{i,d} = \{\mathcal{S}_{i,d}^j\}_{i=1}^{k+1}$ denotes scores calculated by the d - th model, our target is to train a DPR model that retrieves the positive passage p_i^+ for the query q_i.

3.1.2 Dual-Encoders and Cross-Encoders

Depending on how queries and passages are encoded, we categorize DPR models into dual- encoders and cross- encoders.

Dual- encoders (Karpukhin et al., 2020) map query q_i and passage p_j into dense vectors, and the relevance between q_i and p_j is computed by the dot product of their representations:

$$\mathcal{S}_{DE}(q_i, p_j) = E_{DE}(q_i)^T \cdot E_{DE}(p_j) \quad (1)$$

where $E_{DE}(\cdot)$ is the dense vector, and $\mathcal{S}_{DE}(q_i, p_j)$ represents the relevance score of q_i and p_j .

Cross- encoders (Kenton and Toutanova, 2019) concatenate q_i and p_j as the input to PLMs/LLMs. The relevance between q_i and p_j is calculated by the representation of $[CLS]$ in the final layer with a projection layer $\mathbf{W}$:

$$\mathcal{S}_{CE}(q_i, p_j) = \mathbf{W}^T \cdot E_{CE}([CLS]; q_i; [SEP]; p_j) \quad (2)$$

where $[;]$ is the concatenation operation, and $\mathcal{S}_{CE}(q_i, p_j)$ is the similarity of q_i and p_j .

In practice, we use contrastive loss, which encourages $\langle q_i, p_i^+ \rangle$ to be closer together and $\langle q_i, p_i^- \rangle$ to be further apart, to train DPR models:

$$\mathcal{L}_{\text{CL}} = -\log \frac{e^{\mathcal{S}(q_i, p_i^+)}}{e^{\mathcal{S}(q_i, p_i^+)} + \sum_{p_{i,j} \in \mathbb{P}_i^-} e^{\mathcal{S}(q_i, p_{i,j}^-)}} \quad (3)$$

$$\tilde{S}_{tea,i}^j = \frac{e^{\mathcal{S}_{tea}(q_i, p_j)}}{\sum_{p' \in \mathbb{P}_i} e^{\mathcal{S}_{tea}(q_i, p')}} \quad (4)$$

$$\tilde{S}_{stu,i}^j = \frac{e^{\mathcal{S}_{stu}(q_i, p_j)}}{\sum_{p' \in \mathbb{P}_i} e^{\mathcal{S}_{stu}(q_i, p')}} \quad (5)$$

$$\mathcal{L}_{\text{KL}}(\text{tea}, \text{stu}) = -\mathbf{KL}(\tilde{S}_{tea,i} | \tilde{S}_{stu,i}) \quad (6)$$

where $\tilde{S}_{tea,i}, \tilde{S}_{stu,i} \in \mathbb{R}^{|\mathbb{P}_i|}$ denote the probability distributions over candidate passages $\mathbb{P}_i$ and $\tilde{S}_{tea,i}^j$, $\tilde{S}_{stu,i}^j$ denote the j - th element of $\tilde{S}_{tea,i}, \tilde{S}_{stu,i}$. For convenience, we use $\mathcal{L}_{\text{KL}}(\text{tea}, \text{stu})$ $\mathcal{L}_{\text{KL}}(\text{tea}, \text{stu})$ $\mathcal{L}_{KL}(\text{tea}, \text{tea})$ to represent the KL divergence between teachers and students, assistants and students, and teachers and assistants.

3.1.3 Knowledge Distillation for DPR

Recent studies have successfully applied knowledge distillation to training more compact DPR models. A common approach is to use a teacher model to compute relevance scores $\mathcal{S}$ for $\langle q, p \rangle$ pairs, which are then used as the training data for knowledge distillation. To distill the soft labels (scores) from teachers to students, KL divergence $\mathcal{L}_{\text{KL}}(\text{tea}, \text{stu})$ is used as the loss function.

3.2 The MTA4DPR Framework

MTA4DPR transfers knowledge from the teacher DPR model to the student with the help of m ($m \geq 1$) assistant models. For each iteration, we first use these models to generate training and evaluation datasets (Section 3.2.1) which become increasingly difficult as the iterations go on; then, we select the best assistant for each training batch (Section 3.2.3) and train the student model using the teacher together with the selected assistant (Section 3.2.4). The training of one iteration is shown in Figure ??.

3.2.1 Data Preparation

At the start of each iteration, we use the teacher and assistants to generate the corresponding datasets.

Retrieve top- k passages We first use each of the m assistants to retrieve the top- k most relevant passages (except the positive passage(s)) for each query q . Then, we merge all retrieved passages together and collect scores from each assistant model for each $\langle q, p \rangle$ pair. In this way, query q has one or more positive(s) and d negatives ($k \leq d \leq mk$) each of which has m scores computed by the aforementioned m assistant models.

Re- rank using RRF scores From the previous step, we have d negatives for each query q_i , and then we sort these passages in the descending order based on the scores assigned by each assistant, resulting in a set of rankings R , each ranking r being a permutation on $p_1, \ldots, p_{|d|}$. Then, we use RRF (Cormack et al., 2009), Reciprocal Rank Fusion, to re- rank these d passages, taking the top- k passages with the highest scores as the final hard negatives $\mathbb{P}_i^-$ for query q_i :

$$RRFscore(p) = \sum_{r \in R} \frac{1}{c + r(p)} \quad (7)$$

where $c = 60$ following Cormack et al. (2009), and $r(p)$ denotes the position of p in ranking r .

Finally, we use the teacher to calculate the relevance score for each $\langle q_i, p_j \rangle$ pair where $p_j \in \mathbb{P}_i$. By performing the above operations on all training queries, we obtain the base dataset for the current iteration, from which we extract 1% as the evaluation dataset $\mathcal{D}_{eval}$, leaving the rest as the training dataset $\mathcal{D}_{train}$.

In addition, inspired by Lin et al. (2023), we collect the queries for which the teacher can predict the positive as top- 1 while the student from the previous iteration can not predict correctly. These queries with the positive passage and the top- k hard negative passages predicted by the student will be added to the generated dataset.

3.2.2 Fusion Strategy

Inspired by ensemble learning(Mienye et al.,2020) which enhances predictive performance by leveraging the collective strengths of diverse models, we propose a simple yet efficient fusion strategy to combine knowledge of multiple assistants:

$$\mathcal{S}_i = \frac{1}{K}\sum_{k=1}^{K}\mathcal{S}_{i,k} \quad (8)$$

where $\mathcal{S}_{i,k}$ is the score distribution between q_i and $\mathbb{P}_i$ computed by the k - th assistant models.

Specifically, say we have $\mathcal{S}_{i,A}$, $\mathcal{S}_{i,B}$ and $\mathcal{S}_{i,C} \in \mathbb{R}^{|\mathbb{P}_i|}$ respectively computed by assistants A , B and C ; by just taking the average of $\mathcal{S}_{i,A}$ and $\mathcal{S}_{i,B}$, $\mathcal{S}_{i,A}$ and $\mathcal{S}_{i,C}$, $\mathcal{S}_{i,B}$ and $\mathcal{S}_{i,C}$, and all three assistants, we can obtain four different new score distributions, i.e. $\frac{(\mathcal{S}_{i,A}+\mathcal{S}_{i,B})}{2}$, $\frac{(\mathcal{S}_{i,A}+\mathcal{S}_{i,C})}{2}$, $\frac{(\mathcal{S}_{i,B}+\mathcal{S}_{i,C})}{2}$ and $\frac{(\mathcal{S}_{i,A}+\mathcal{S}_{i,B}+\mathcal{S}_{i,C})}{3}$. All these fused score distributions are considered as knowledge contributed by certain fused assistants in MTA4DPR, and are involved in the selection method for assistants.

3.2.3 Assistant Selection

To select the best assistant for each training batch, we investigate three heuristic selection strategies:

KL Divergence KL divergence measures the similarity between two distributions. The higher the similarity, the smaller the KL divergence. We calculate the KL divergence between the score distributions of the teacher model and each assistant, and consider the assistant that achieves the minimum KL divergence as the best teaching assistant.

Spearman's Footrule Spearman's Footrule measures the absolute distance between two sorted lists, similar to edit distance. It is suitable for comparing the similarity between two permutations, with smaller values indicating more similar permutations. We calculate the Spearman's Footrule distances between the teacher and each assistant, and consider the assistant that has the minimum distance with the teacher as the best.

Rank Biased Overlap Rank Biased Overlap (RBO) compares the overlap of two ranked lists at increasing depths. Unlike Spearman's Footrule, it assigns different weights to different depths, with top- 1 having the highest weight. The value of RBO ranges from 0 to 1, and larger values indicate more similar sorted lists. We calculate the RBO measures between the teacher and each assistant, and consider the assistant that has the maximum RBO value as the best assistant.

Please note that since this computation process is only for selecting the best assistant, it does not participate in the gradient backpropagation.

3.2.4 The Student Model Optimization

For each training batch, we first use the selection method described in 3.2.3 to select the best assistant model. Then, we use $\mathcal{L}_{\text{CL}}$, $\mathcal{L}_{\text{KL}}$ to optimize the student model which is also a dual- encoder:

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{CL}} + \beta\mathcal{L}_{\text{KL(tea,stu)}} + \gamma\mathcal{L}_{\text{KL(ta,stu)}} \quad (9)$$

where α, β, γ are hyper- parameters, $\mathcal{L}_{\text{CL}}$ is the contrastive loss of the student model (see more in eq(3)).We also calculate the KL divergence $\mathcal{L}_{\text{KL(ta,stu)}}$, $\mathcal{L}_{\text{KL(tea,stu)}}$ as part of the loss during training, forcing the student to learn the score distributions of the best assistant and the teacher.

At the end of each iteration, we evaluate the student's performance on the evaluation dataset, replace the worst- performing assistant with the student if it outperforms any of the existing assistants, and then regenerate the training/evaluation dataset. We repeat all the above operations, from generating datasets to optimizing the student model, until the training ends. The entire training process is introduced in Algorithm **??** in AppendixA.

4 Experiments and Analysis

4.1 Experimental Settings

We conduct experiments on four retrieval datasets: MS MARCO passage, TREC DL 2019, TREC DL 2020(Craswell et al.,2020a,b) and Natural Questions (NQ)(Kwiatkowski et al.,2019) datasets. We use the averaged $[CLS]$ representations of the student model's last three layers to represent each query/assage, and dot product to compute the similarity between the query and passage. Following previous studies, we report MRR@10, Recall@50 and Recall@1k on MS MARCO dev set, and nDCG@10 on TREC DL 2019 and 2020; and we choose Recall@5, Recall@20 and Recall@100 as the evaluation metrics for Natural Questions.

Baselines To make a comprehensive comparison, we compare MTA4DPR with three groups of baselines: sparse retrieval models and dense retrieval models with/without knowledge distillation. Specifically, sparse retrieval models include BM25(Robertson et al.,2009), DeepCT (Dai and Callan,2019), GAR(Mao et al.,2021), docT5query(Nogueira et al.,2019), COIL- full(Gao et al.,2021), UniCOIL(Lin and Ma,2021) and SPLADE- max(Formal et al.,2021); dense retrieval models without knowledge distillation include DPR(Karpukhin et al.,2020), ANCE(Xiong et al.), Condenser(Gao and Callan,2021b), XTR- base(Lee et al.,2024), CotMAE(Wu et al.,2023), GTR- XXL(Ni et al.,2022) and RepLLaMA- 7B (Ma et al.,2024); dense retrieval models with knowledge distillation include RocketQAv1(Qu et al.,2021), PAIR(Ren et al.,

2021a), RocketQAv2 (Ren et al., 2021b), ERNIE-Search (Lu et al., 2022), SimLM (Wang et al., 2023), RetroMAE (Xiao et al., 2022), LEAD (Sun et al., 2024), CL- DRD (Zeng et al., 2022) and PROD (Lin et al., 2023).

Model Initialization For MS MARCO, to balance the trade- off between efficiency and effectiveness, we choose dual- encoders as the assistants and the cross- encoder as the teacher. Specifically, we set CotMAE, SimLM- distilled, RetroMAE and M2DPR (Lu, 2024) as assistants, since they are the most performant off- the- shelf dense retrievers to our knowledge. Their MRR@10 on MS MARCO dev set are 39.4, 41.1, 41.6 and 42.0, respectively. SimLM- reranker, a well performant cross- encoder, is considered as the teacher model with 43.7 MRR@10. Besides, to validate the effectiveness on NQ dataset, we simply use RocketQAv1 and PAIR as the assistants, and ERNIE- search as the teacher model with Recall@20 82.7, 83.5 and 85.3 on NQ test set. The student DPR models are initialized with the SimLM- base model.

Training Details For MS MARCO, we set the iterations to 3, as our experiments show that the performance improvement becomes marginal beyond the 3rd iteration. For each iteration, we use 1 Tesla A100 80G GPU to train our student model for 20,000 steps using AdamW optimizer with learning rate of 3×10^{-5} . Each query in the training set has several positive passages and $k = 100$ hard negatives. Each training batch has 64 queries, each of which has 1 positive passage and 34 hard negatives randomly sampled from the training set. The weight decay is set to 0.01. The max query length is 32, and the max passage length is 144. To balance each term of the final loss, α , β and γ are set to 0.2, 1, 15. For NQ, we reuse the same settings as those on MS MARCO with a few exceptions. The training steps for each iteration is set to 10,000 steps, and the max passage length is 192.

4.2 Main Results

The results comparing MTA4DPR with multiple baselines on the MS MARCO, TREC DL 19 and 20 and NQ datasets are shown in Table ?? and Table ??. From the tables, we can observe that the 66M student model trained by MTA4DPR achieves MRR@10 41.1 on MS MARCO, nDCG@10 71.2 on TREC DL 19, nDCG@10 71.1 on TREC DL 20 and Recall@20 83.6 on NQ, which outperforms most 66M distilled student models, and is competitive when compared with larger DPR models (the 110M ones), even with the LLM- based models.

In addition, we have the following observations:

1. RepLLaMA-7B achieves MRR@10 41.2 on MS MARCO, nDCG@10 74.3 and 72.1 on TREC DL 19 and 20, far surpassing most baselines without knowledge distillation, which means that, without knowledge distillation, the larger the model, the better the retrieval performance.

2. 110M DPR models trained with knowledge distillation, such as SimLM (MRR@10 41.1 on MS MARCO dev) and ERNIE-Search (Recall@20 85.3 on NQ test), can achieve better retrieval performance when compared with the models with the same or even much bigger sizes without knowledge distillation, from which we can see that knowledge distillation can effectively transfer knowledge from large teacher DPR models to small student models.

3. RepLLaMA-7B performs about nDCG@10 2.0 better than 66M DPR models on DL 20 which is mainly used to test models' ability to capture fine- grained semantics. This implies that, in capturing fine- grained semantics, large DPR models are much better than small models, which motivates us to further optimize small models' ability to capture fine- grained semantic.

Table 2: Main results on NQ. "#Params" represents the number of model parameters.

Model	#Params	NQ		
		R@5	R@20	R@100
BM25	-	-	59.1	73.7
GAR	-	-	60.9	74.4
DPR	110M	-	78.4	85.4
ANCE	110M	-	81.9	87.5
Condenser	110M	-	83.2	88.4
RocketQAv1	110M	74.0	82.7	88.5
PAIR	110M	74.9	83.5	89.1
ERNIE-Search	110M	77.0	85.3	89.7
MTA4DPR	**66M**	**74.5**	**83.6**	**88.3**

4.3 Ablation Study

To validate the effectiveness of each module of our method, we conduct the ablation study. All ablation results come from 3- iteration training, except for "w/o iterations" in which we deliberately disabled the iteration to show its effectiveness.

The results in Table ?? demonstrate the effectiveness of our model. We can see that removing any module will decrease the final performance, with the removal of the teaching assistants resulting in the most significant performance drop. Additionally, we also have the following observations:

1. Without teaching assistants, the student model's performance drops to MRR@10 39.9 on MS MARCO and Recall@20 82.2 on NQ, which indicates that using teaching assistants can

Table 1: Main results on MS MARCO and DL 19 and 20 datasets. The best scores are marked in bold, and the second places are underlined. "KD" denotes knowledge distillation, and "#Params" represents the number of model parameters. Please note that, by SimLM, we mean SimLM-distilled, not SimLM-renarner or SimLM-base.

Model	#Params	MS MARCO dev			TREC DL		TREC DL 20
		MRR@10	R@50	R@1k	19 nDCG@10	20 nDCG@10	nDCG@10
Sparse Retrieval							
BM25	-	18.7	59.2	85.7	49.7	48.7	
DeepCT	110M	24.3	69.0	91.0	55.0	55.6	
docT5query	-	27.2	75.6	94.7	64.2	61.9	
COIL-full	110M	35.5	-	96.3	70.4	-	
UniCOIL	110M	35.2	80.7	95.8	-	-	
SPLADE-max	110M	34.0	-	96.5	68.4	-	
Dense Retrieval without KD							
XTR-base	110M	37.4	-	98.0	-	-	
CotMAE	110M	39.4	87.0	98.7	-	70.4	
GTR-XXL	4.8B	38.8	-	99.0	-	-	
RepLLaMA-7B	7B	41.2	-	99.4	74.3	72.1	
Dense Retrieval with KD							
RocketQAv2	110M	38.8	86.2	98.1	-	-	
SimLM	110M	41.1	87.8	98.7	71.4	69.7	
RetroMAE	110M	41.6	88.6	98.8	-	-	
LEAD	66M	37.8	-	97.4	70.4	68.9	
CL-DRD	66M	38.2	-	-	72.5	68.7	
PROD	66M	39.3	87.0	98.4	73.3	-	
MTA4DPR	**66M**	**41.1**	**88.4**	**98.7**	**71.2**	**71.1**	

help students better learn the knowledge from teacher/assistant models.

2. The performance also drops to MRR@10 40.8 on MS MARCO and Recall@20 83.4 on NQ without fusion strategy. Through further analysis, we find that the KL divergence between fused score distributions and the teacher's score distribution tends to be smaller than that of original assistants, which means students can learn more useful information from fused assistants than the original assistants.

3. Finally, without training iterations, the performance of the student model drops to MRR@10 40.1 on MS MARCO and Recall@20 82.7 on NQ. This indicates that our iterative training method which enable students to learn from better teacher/assistants and more difficult data at each iteration improves the student's performance.

Table 3: Ablation results on MS MARCO and NQ.

Model	MS MARCO			
	MRR@10	R@20	Recall@20	R@100
Full MTA4DPR	41.1	88.4	83.6	88.3
w/o assistants	39.9	86.8	82.2	87.5
w/o fusion	40.8	87.9	83.4	88.1
w/o iterations	40.1	87.1	82.7	87.5

4.4 Analysis

We further analyze our proposed method from the following perspectives, i.e. the performance of the student model at each iteration, the assistant selection methods, student models' scale, assistant models' performance, the assistants selected, the complexity of the training process and the computational costs of the student models.

4.4.1 Multi-iteration Retrieval Performance

We report the retrieval performance of our 66M DPR model in each iteration, as shown in Table ??. As expected, as the number of iterations increases, the performance also improves, from MRR@10 40.1 to 41.1 on MS MARCO and from Recall@20 82.7 to 83.6 on NQ. This indicates that to some extent,

better assistant models combined with more difficult data will further improve the performance of the student model.

Table 4: Multi-iteration Retrieval Performance on MS MARCO and NQ.

	MS MARCO			NQ		
Iteration	MRR@10	R@50	R@1k	MRR@10	R@50	R@1k
1st	40.1	87.1	98.6	71.9	82.7	87.5
2nd	40.7	87.9	98.7	73.1	83.3	88.0
3rd	41.1	88.4	98.7	74.5	83.6	88.3

4.4.2 The impact of selection methods

We compare multiple methods to select the best assistant, as described in 3.2.3. Table **??** shows the results of MTA4DPR models using different selection methods. Compared with a random assistant, using KL, Spearman's Footrule, and RBO selection methods can further improve retrieval performance, indicating that the teaching assistants selected by these three methods are more beneficial to the distillation process. Among these three methods, we chose KL selection method which obtains the best performance for the other experiments.

Table 5: Performance of MTA4DPR models with different selection methods on MS MARCO and NQ. "SF" denotes Spearman's Footrule.

	MS MARCO			NQ		
Selection Method	MRR@10	R@50	R@1k	R@5	R@20	R@100
Random	40.5	87.6	98.6	72.8	82.4	87.5
SF	40.8	87.7	98.7	74.5	83.1	87.9
RBO	40.9	87.9	98.8	74.4	83.0	88.1
KL	**41.1**	**88.4**	**98.7**	**74.5**	**83.6**	**88.3**

4.4.3 The impact of the number of layers and the embedding sizes of student models

We use the proposed method to distill student DPR models with different number of layers and embedding sizes. As shown in Table **??**, we can see that:

1. MTA4DPR can improve the retrieval performance of the student models with different number of layers and embedding sizes; and as the number of layers and the number of embedding size increase, the performance improves.

2. It is worth noting that our 33M DPR model is almost equivalent to the existing 110M DPR models on Recall@1k on MS MARCO. Due to the fact that retrievers are often used in the first stage of retrieve-rerank pipeline in practical scenarios, a 33M DPR model can be used to reduce query time.

3. Finally, we also find that the 12-layer 384-dimensional models outperform the 3-layer 768-dimensional models, despite having fewer parameters. We speculate that this might be due to the 12-layer models' ability to capture more complex text interactions owing to its greater depth. We will investigate this further in future work.

Table 6: Results of MTA4DPR models with different sizes on MS MARCO. "#Layers" denotes the number of layers of the model, and "#Emb" denotes the embedding size of the model. "#Params denotes the number of model parameters. "t" denotes the improvement compared with traditional knowledge distillation methods.

#Layers	#Emb	#Params	MS MARCO		
			MRR@10	R@50	R@
6	384	17M	36.0 (↑ 1.1)	81.6 (↑ 1.3)	96.3
12	384	33M	40.1 (↑ 0.8)	87.2 (↑ 0.7)	98.4
3	768	45M	39.4 (↑ 0.9)	86.5 (↑ 1.1)	98.4
6	768	66M	41.1 (↑ 1.2)	88.4 (↑ 1.6)	98.7
12	768	110M	41.8 (↑ 0.7)	88.6 (↑ 0.8)	98.8

4.4.4 The impact of the performance of assistant models

We wonder how the performance of the assistants affects the distillation process. To this end, we conducted five groups of experiments, i.e. No assistant, Single- assistant distillation, Double- assistant distillation, Triple- assistant distillation and Quadruple- assistant distillation. No assistant involved distillation using only the teacher model without any assistants. Single- assistant distillation experiments are done using just one assistant and one teacher for distillation. Double- assistant distillation utilized one teacher and two assistants along with a fusion strategy for distillation, and so on.

The results are listed in Table **??**. From the table, we have the following observations:

1. Compared to not using assistants, even the result of using the weakest assistant model is better than the no-assistant way. For example, using only CotMAE can increase the value of MRR@10 from 39.9 to 40.2 on MS MARCO dev set. This strongly proves the effectiveness of using assistant models.

2. R&M is better than other double-assistant combinations, S&R&M is better than other triple-assistant combinations. This implies that the better the performance of assistants, the

better the performance of the distilled student model.

4.4.5 The composition of the best assistant

We explore which assistant is selected as the best one in each batch during the whole training procedure. The composition of the best teaching assistants selected on MS MARCO is shown in Figure ??. From the figure, we can see that the fusion result of RetroMAE and M2DPR is chosen for nearly 50% of the time, which confirms once again the effectiveness of the fusion strategy.

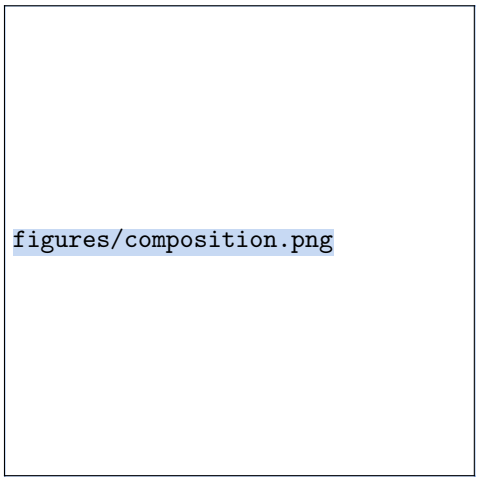

Figure 2: The composition of the best teaching assistants selected on MS MARCO. "R" denotes RetroMAE, "S" denotes SimLM, "M" denotes M2DPR and "R&M" denotes the fusion result of RetroMAE and M2DPR.

4.4.6 The complexity of the training process

The time consumption of our method can be divided into two parts: model training and data construction. The time taken to train a 6- layer 768-dimensional student model is shown in Table ??.

Since the teachers/assistants are not actually involved in the training process but only provide query- passage pair scores, which can be obtained during data construction, the training time of our method is only about 25 minutes longer than that of the traditional knowledge distillation, primarily due to the selection of the best teaching assistant for each batch. For the data construction, we require approximately 4.7 more hours compared to the traditional knowledge distillation method. The additional time is mainly spent on scoring unseen query- passage pairs using both the teacher and assistants models, which will be used for the next iteration. While time- consuming, this process provides us a more difficult dataset, which can further improve the performance of the student model.

Table 8: The complexity of the training process.

Model	Training Time	Data Construction Time
MTA4DPR	7.53 hours	12.9 hours
Traditional KD	7.12 hours	8.2 hours

4.4.7 The computational costs of MTA4DPR

We also conduct more experiments to further validate the efficiency and the computational costs of the student model distilled by our proposed method under three different settings, as shown in Table ??. From the table, we can see that: reducing the embedding size is more efficient than reducing model layers in terms of the model size (decreased from 110M to 33M) and index size (decreased from 25.2G to 12.8G); while reducing model layers provide more improvement in terms of the model encoding time (decreased from 304.30s to 163.23s with the 512 batch size, and from 135.82s to 87.86s with the 1024 batch size).

Table 9: The computational costs of student DPR models with different sizes. "Encoding Time" is the time taken to encode the whole MS MARCO corpus. "#Emb" denotes the embedding size of the model. Please note that this metric is pure GPU computation time and doesn't include the time for data loading or other operations. "bs" denotes the batch size.

#Layers	#Emb	Index Size	#Params	Encoding Time	
				bs = 512	bs = 102
6	768	25.2G	66M	163.23s	87.86s
12	384	12.8G	33M	297.06s	131.67s
12	768	25.2G	110M	304.30s	135.82s

5 Conclusion

In this paper, we propose MTA4DPR, an iterative multi- assistant distillation method for DPR. It distills the student with the help of the teaching assistants in an iterative manner, with each iteration creating more difficult datasets and more performant assistants. The experimental results on MS MARCO, TREC DL 2019 and 2020 and Natural Questions show the effectiveness of our method. Our 66M DPR model can achieve the state- of-the- art performance among models with same parameters on multiple datasets and is very competitive when compared with larger, even LLM- based,

Table 7: Results of distilled DPR models with different assistants combinations on MS MARCO dev set and DL 19 and 20 datasets. "C", "S", "R" and "M" represent CoMAE, SimLM, RetroMAE and M2DPR, respectively. "C&S" denotes the fusion result of CoMAE and SimLM.

Method	Assistant Models	MS Marco dev			TREC DL 19	TREC DL 20
		MRR@10	R@50	R@1k	nDCG@10	nDCG@10
No assistant	/	39.9	86.8	98.5	69.2	67.7
Single-assistant	C	40.2	87.3	98.5	69.8	68.1
	S	40.4	87.3	98.5	70.0	68.9
	R	40.6	87.7	98.7	70.0	69.7
	M	40.6	87.6	98.8	70.2	69.3
Double-assistant	C&S	40.4	87.3	98.7	69.6	68.6
	C&R	40.6	87.3	98.7	69.6	69.7
	C&M	40.5	87.6	98.7	70.0	69.9
	S&R	40.7	87.3	98.7	69.2	69.3
	S&M	40.6	87.7	98.7	70.1	69.0
	R&M	40.8	87.8	98.8	70.3	69.9
Triple-assistant	C&S&R	40.7	87.6	98.7	70.8	70.3
	C&S&M	40.8	87.7	98.7	70.1	69.0
	C&R&M	40.9	88.0	98.8	70.3	69.9
	S&R&M	41.0	88.0	98.8	70.6	70.7
Quadruple-assistant	C&S&R&M	**41.1**	**88.4**	**98.7**	**71.2**	**71.1**

DPR models. MTA4DPR confirms that the iterative distillation with multiple assistants can improve the distillation performance. Since it is orthogonal to existing distillation methods, other distillation pipelines can be combined with MTA4DPR to further improve their performance. In addition, MTA4DPR is not constrained by model structures and tasks, and can be broadly applicable other fields than DPR, including text classification, question answering and text summarization, etc.

Limitations

We consider the following four points as the limitations of this work:

First, due to flexibility and scalability considerations, we only distill the score distributions provided by teacher/assistants, while ignoring information provided by intermediate layers of teacher/assistant models which can be beneficial to further improve the student models' performance.

Second, at the first training iteration, our method requires multiple off- the- shelf DPR models, but when there are not enough available models, we need to train teacher/assistant DPR models from scratch, which may increase the training costs.

Third, for the sake of the training phase's simplicity and efficiency, we only use heuristic strategies when generating fused scores and selecting the best teaching assistant. To further improve student performance, we can design more complex and effective generation and selection methods.

Finally, in the future, we can continue to explore the impact of the number and performance of teaching assistants on the final retrieval result of student models, and find out how to determine what kind of teaching assistant is good.

Acknowledgements

This work is supported by National Natural Science Foundation of China (NSFC) (No. 62076038).

Ethics Statement

The MTA4DPR method mainly aims at retrieving the most relevant passages for a given query in an effective and efficient manner. And the experiments are based on the MS MARCO, TREC DL 2019 and 2020 and Natural Questions datasets, which is unlikely to include harmful content.

Licenses

All SimLM models are under license MIT. RetroMAE is under license artistic- 2.0. CoMAE and M2DPR is not under any licenses. MS MARCO passage dataset, TREC DL 2019 and 2020 and Natural Questions datasets also don't extend any license and allows for academic usage.

A Algorithm

Algorithm 1 MTA4DPR Training Process

Require: T : the teacher model; TA : the assistant models; M_θ : the student model; Q : the query set; P : the passage set; max_iter : maximum number of training iterations; max_steps : maximum number of training steps; η : Learning rate;

Ensure: M_θ

$i \leftarrow 0$
while $i < max\_iter$ **do**
  $D_{train}, D_{eval} \leftarrow GenDataset(T, TA, Q, P)$
  repeat
    $id_{bestTA} \leftarrow TASelect(D_{train})$
    $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{total}}(D_{train}, M_\theta, id_{bestTA})$
  until max_steps reached
  $outperform \leftarrow Compare(M_\theta, TA, D_{eval})$
  if $outperform$ **then**
    remove $Worst(TA)$
    add M_θ into TA
  end if
  $i \leftarrow i + 1$
end while

References

[1] Romero Adriana, Ballas Nicolas, K Samira Ebrahimi, Chassang Antoine, Gatta Carlo, and Bengio Yoshua. 2015. Fitnets: Hints for thin deep nets. Proc. ICLR, 2(3):1.

[2] Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In Proceedings of the 26th annual international conference on machine learning, pages 41- 48.

[3] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. 2022. Knowledge distillation: A good teacher is patient and consistent. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10925- 10934.

[4] Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. 2018. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In Proceedings of the AAAI conference on artificial intelligence, volume 32.

[5] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pages 758- 759.

[6] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020a. Overview of the trec 2020 deep learning track. Text REtrieval Conference, Text REtrieval Conference.

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

[8] Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for it with contextual neural language modeling. In Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pages 985- 988.

[9] Thibault Formal, Benjamin Piwowarski, and Stephane Clinchant. 2021. Splade: Sparse lexical and expansion model for first stage ranking. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 2288- 2292.

[10] Luyu Gao and Jamie Callan. 2021a. Condenser: a pretraining architecture for dense retrieval. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 981- 993.

[11] Luyu Gao and Jamie Callan. 2021b. Is your language model ready for dense representation fine- tuning? CoRR, abs/2104.08253.

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

[13] Zhisheng Huang, Junran Peng, Zhaoxiang Zhang, and Tieniu Tan. 2022. Representation similarity distillation for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12819- 12828.

[14] Vladimir Karpukhin, Barlas Oğuz, 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.

[15] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language

understanding. In Proceedings of NAACL-HLT, pages 4171–4186.

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

[17] Dongyub Lee, Myeongho Jeong, and Byeongchang Kim. 2024. XTR: Transformers are effective text rankers with pairwise ranking prompting. In Findings of the Association for Computational Linguistics: NAACL 2024, pages 1504–1518.

[18] Jimmy Lin and Xueguang Ma. 2021. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. arXiv preprint arXiv:2106.14807.

[19] Jimmy Lin, Xueguang Ma, and Kai Hui. 2023. PROD: Progressive distillation for dense retrieval. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1433–1442.

[20] Yukun Lu, Wenge Liu, Jianfeng Ren, and Jian-Yun Nie. 2022. ERNIE-Search: Bridging cross-encoder with dual-encoder via self-distillation for dense passage retrieval. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9159–9170.

[21] Yukun Lu. 2024. M2DPR: A multi-teacher multi-step distillation framework for dense passage retrieval. arXiv preprint arXiv:2401.12345.

[22] Xinyin Ma, Yong Jiang, and Pengjun Xie. 2024. RepLLaMA: Retrieval-augmented LLaMA for open-domain question answering. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12345–12357.

[23] Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4089–4100.

[24] Irene D. Mienye, Yanxia Sun, and Zenghui Wang. 2020. Improved sparse autoencoder based artificial neural network approach for prediction of heart disease. Informatics in Medicine Unlocked, 18:100307.

[25] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 5191–5198.

[26] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Zhao, Yi Luan, Keith B. 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.

[27] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. arXiv preprint arXiv:1904.08375.

[28] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. 2019. Correlation congruence for knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 5007–5016.

[29] Zheng Qin, Kai Hui, Honglei Zhuang, Junru Zhang, Jing Lu, and Jimmy Lin. 2024. Beyond lexical matching: A survey on the intersection of large language models and dense retrieval. arXiv preprint arXiv:2405.19170.

[30] 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.

[31] Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021a. Pair: Leveraging passage-centric similarity relation for improving dense passage retrieval. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2173–2183.

[32] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang,

and Ji-Rong Wen. 2021b. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2825–2835.

[33] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends® in Information Retrieval, 3(4):333–389.

[34] Wonchul Son, Jaemin Na, Junyong Choi, and Wonjun Hwang. 2021. Densely guided knowledge distillation using multiple teacher assistants. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9395–9404.

[35] Hao Sun, Xiao Liu, Yeyun Gong, Anlei Dong, Jingwen Lu, Yan Zhang, Linjun Yang, Rangan Majumder, and Nan Duan. 2024. Lead: liberal feature-based distillation for dense retrieval. In Proceedings of the 17th ACM International Conference on Web Search and Data Mining, pages 655–664.

[36] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2244–2258, Toronto, Canada. Association for Computational Linguistics.

[37] Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. 2021. One teacher is enough? pre-trained language model distillation from multiple teachers. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4408–4413.

[38] 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.

[39] 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.

[40] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In International Conference on Learning Representations.

[41] Chuanguang Yang, Helong Zhou, Zhulin An, Xue Jiang, Yongjun Xu, and Qian Zhang. 2022. Cross-image relational knowledge distillation for semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12319–12328.

[42] Fei Yuan, Linjun Shou, Jian Pei, Wutao Lin, Ming Gong, Yan Fu, and Daxin Jiang. 2021. Reinforced multi-teacher selection for knowledge distillation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 14284–14291.

[43] 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.