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

Qixi Lu[1,2], Endong Xun[1], Gongbo Tang[2]
[1]Beijing Advanced Innovation Center for Language Resources,
Beijing Language and Culture University, China
[2]School of Information Science, Beijing Language and Culture University, China
{qixi.lu, 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 [**?**, **?**] 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 [**?**, **?**, **?**] 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.

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 [**?**], 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 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 DL2019 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:

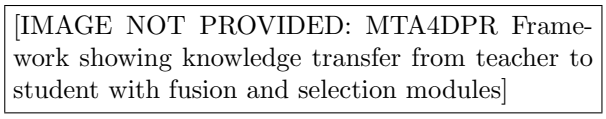

Figure 1: MTA4DPR Framework. MTA4DPR transfers knowledge from the teacher to the student with the help of the best assistant. The Fusion Module is used to generate fused assistants from the original assistants, and the Selection Module is used to select the best assistant among all original and fused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

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, cannot thoroughly solve the lexical mismatch problem, although query/document expansion [**?**, **?**] and term-weighting [**?**, **?**] 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 [**?**], knowledge distillation [**?**, **?**, **?**] and joint optimization of retrievers and rankers [**?**].

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 [**?**, **?**], intermediate knowledge [**?**, **?**] and relation-based knowledge [**?**, **?**, **?**].

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 [**?**, **?**, **?**]. Mirzadeh et al. [**?**] 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. [**?**] 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 specialty. Furthermore, we experiment with iteratively replacing underperforming assistants with better-performing student models, which further improves the performance of the final student model.

3 Methodology

3.1 Preliminary

3.1.1 Task Description

Assume we have a training set $\mathcal{D} = \{(q_i, \mathcal{P}_i, \mathcal{S}_i)\}_{i=1}^{L}$ where q_i is the query, $\mathcal{P}_i$ consists of a positive passage p_i^+ and k hard negatives $\mathcal{P}_i^- = \{p_{i,j}^-\}_{j=1}^{k}$ (passages that are difficult to distinguish from the positive passage) and $\mathcal{S}_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,d}, \ldots\}$ consists of relevance scores computed by the teacher/assistants and $\mathcal{S}_i^d = \{s_{i,d}\}_{d=1}^{D}$ 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 [**?**] 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:

$$s_{de}(q,p) = E_{de}(q)^\top \cdot E_{de}(p) \tag{1}$$

where $E_{de}(\cdot)$ is the dense vector, and $s_{de}(q_i,p_j)$ represents the relevance score of q_i and p_j.

Cross-encoders [**?**] 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 W:

$$s_{ce}(q,p) = W^\top \cdot E_{sp}([CLS;q_i;SEP;p_j]) \tag{2}$$

where $[;]$ is the concatenation operation, and $s_{ce}(q_i,p_j)$ is the similarity of q_i and p_j.

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

$$\mathcal{L}_{cl} = -\log \frac{\exp(s(q_i,p_i^+))}{\exp(s(q_i,p_i^+)) + \sum_{p\in\mathcal{P}_i^-}\exp(s(q_i,p))} \tag{3}$$

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 S for (q,p) pairs, which are then used as the training data for knowledge distillation. To distill the soft labels (scores) from teachers to students, KL divergence $\mathcal{L}_{KL}(tea,stu)$ is used as the loss function:

$$S_{tea,i} = \text{softmax}(s_{tea}(q_i,p_j)) \tag{4}$$

$$S_{stu,i} = \text{softmax}(s_{stu}(q_i,p_j)) \tag{5}$$

$$\mathcal{L}_{KL}(tea,stu) = -KL(S_{tea,i}\|S_{stu,i}) \tag{6}$$

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

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 **??**) which become increasingly difficult as the iterations go on; then, we select the best assistant for each training batch (Section **??**) and train the student model using the teacher together with the selected assistant (Section **??**). 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_i. Then, we merge all retrieved passages together and collect scores from each assistant model for each (q,p) pair. In this way, query q_i has one or more positives 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 $\mathcal{R}$, each ranking r being a permutation of $\{p_1,\ldots,p_d\}$. Then, we use RRF [**?**], Reciprocal Rank Fusion, to re-rank these d passages, taking the top-k passages with the highest scores as the final hard negatives $\mathcal{P}_i^-$ for every q_i:

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

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

Finally, we use the teacher to calculate the relevance score for each (q_i,p_j) pair where $p_j \in \mathcal{P}_i$. By performing the above operations on all training queries, we obtain the base dataset for the current iteration, from which we extract 10% as the evaluation dataset $\mathcal{D}_{eval}$, leaving the rest as the training dataset $\mathcal{D}_{train}$. In addition, inspired by Lin et al. [**?**], we collect the queries for which the teacher can predict the positive as top-1 while the student from the previous iteration cannot 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 [**?**] 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:

$$S_{fus} = \frac{1}{K}\sum_{k=1}^{K} S_{a,k} \tag{8}$$

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

Specifically, say we have $S_{i,A}$, $S_{i,B}$ and $S_{i,C} \in \mathbb{R}^{|\mathcal{P}_i|}$ respectively computed by assistants A, B and C; by just taking the average of $S_{i,A}$ and $S_{i,B}$, $S_{i,A}$ and $S_{i,C}$, $S_{i,B}$ and $S_{i,C}$, and all three assistants, we can obtain four different new score distributions, i.e., $\frac{1}{2}(S_{i,A} + S_{i,B})$, $\frac{1}{2}(S_{i,A} + S_{i,C})$, $\frac{1}{2}(S_{i,B} + S_{i,C})$, $\frac{1}{3}(S_{i,A} + S_{i,B} + S_{i,C})$. 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 Section **??** to select the best assistant model. Then, we use $\mathcal{L}_{cl}$, $\mathcal{L}_{KL}$ to optimize the student model which is also a dual-encoder:

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

where α, β, γ are hyper-parameters, $\mathcal{L}_{cl}$ is the contrastive loss of the student model (see more in Eq. **??**). We also calculate the KL divergence $\mathcal{L}_{KL}(ta, stu)$, $\mathcal{L}_{KL}(tea, stu)$ 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 Appendix A.

4 Experiments and Analysis

4.1 Experimental Settings

We conduct experiments on four retrieval datasets: MS MARCO passage, TREC DL2019, TREC DL 2020 [**?**, **?**] and Natural Questions (NQ) [**?**] datasets. We use the averaged [CLS] representations of the student model's last three layers to represent each query/passage, 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 DL2019 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 [**?**], DeepCT [**?**], GAR [**?**], docT5query [**?**], COIL-full [**?**], UniCOIL [**?**] and SPLADE-max [**?**]; dense retrieval models without knowledge distillation include DPR [**?**], ANCE [**?**], Condenser [**?**], XTR-base [**?**], CotMAE [**?**], GTR-XXL [**?**] and RepLLaMA-7B [**?**]; dense retrieval models with knowledge distillation include RocketQA-v1 [**?**], PAIR [**?**], RocketQA-v2 [**?**], ERNIE-Search [**?**], SimLM [**?**], RetroMAE [**?**], LEAD [**?**], CL-DRD [**?**] and PROD [**?**].

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 [**?**] 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 RocketQA-v1 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 75.3 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 semantics.

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 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

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-reranker or SimLM-base.

Model	#Params	MRR@10	R@50	R@1k	nDCG@10 (DL19)	nDCG@10 (DL20)
Sparse						
BM25	-	18.7	59.2	85.7	49.7	48.7
DeepCT	-	24.3	69.0	91.0	55.0	55.6
docT5query	-	27.2	75.6	94.7	64.2	61.9
COIL-full	-	35.5	80.7	96.3	70.4	72.1
UniCOIL	-	35.2	86.2	95.8	68.4	69.7
SPLADE-max	-	34.0	87.0	96.5	68.9	68.7
Dense (no KD)						
XTR-base	110M	37.4	87.0	98.0	71.4	73.1
CotMAE	110M	39.4	86.2	98.7	70.4	71.9
GTR-XXL	110M	38.8	87.8	98.8	72.5	74.3
RepLLaMA-7B	7B	41.2	98.4	99.4	73.3	75.3
Dense (with KD)						
RocketQA-v1	110M	38.8	87.0	98.4	70.4	71.1
SimLM	110M	41.1	88.6	98.8	72.5	73.3
RetroMAE	110M	41.6	87.4	98.7	73.3	74.5
LEAD	110M	37.8	86.8	97.4	71.4	72.1
CL-DRD	110M	38.2	87.1	98.1	70.4	71.0
PROD	110M	39.3	88.4	98.7	72.5	73.3
MTA4DPR (Ours)	66M	**41.1**	**88.4**	**98.7**	**71.2**	**71.1**

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

Model	#Params	R@5	R@20	R@100
BM25	-	60.9	73.7	82.7
GAR	-	59.1	74.0	83.5
DPR	110M	74.4	81.9	85.3
ANCE	110M	78.4	83.2	85.4
Condenser	110M	81.9	85.3	87.5
PAIR	110M	83.2	85.4	88.4
ERNIE-Search	110M	85.3	87.5	89.4
MTA4DPR (Ours)	66M	74.5	**83.6**	88.3

Table 3: Ablation results on MS MARCO and NQ.

	MS MARCO MRR@10	MS MARCO R@1k	NQ R@20
Full model	41.1	98.7	83.6
w/o assistants	39.9	98.5	82.2
w/o fusion	40.8	98.7	83.4
w/o iterations	40.1	98.6	82.7

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.

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

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

Iteration	MS MARCO MRR@10	MS MARCO R@1k	NQ R@20
1st iteration	40.1	98.6	82.7
2nd iteration	40.8	98.7	83.4
3rd iteration	41.1	98.7	83.6

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

Method	MS MARCO MRR@10	MS MARCO R@1k	NQ R@20
Random	40.5	98.6	87.6
SF	40.8	98.7	87.9
RBO	40.9	98.8	88.2
KL	**41.1**	**98.7**	**88.4**

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.

4.4.2 The impact of selection methods

We compare multiple methods to select the best assistant, as described in Section **??**. 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.

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.

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. "↑" denotes the improvement compared with traditional knowledge distillation methods.

#Layers	#Emb	#Params	MRR@10	R@50	R@1k
3	384	17M	40.1	86.5	98.4
6	384	33M	39.4↑0.8	88.4	98.4
6	768	45M	41.1	88.6	98.7
12	384	66M	41.8	88.6	98.8
12	768	110M	42.0	89.0	99.0

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.

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

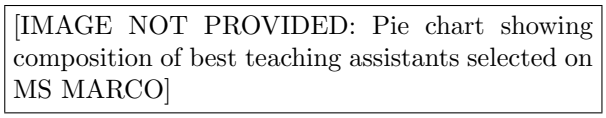

Figure 2: The composition of the best teaching assistants selected on MS MARCO. "R" denotes RetroMAE, "S" denotes SimLM, "M" denotes M2DPR and "R&M" denotes the fusion result of RetroMAE and M2DPR.

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.

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.

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).

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 DL2019 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, 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:

1. 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.

2. Second, at the first training iteration, our method requires multiple off-the-shelf DPR mod-

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 CotMAE, SimLM, RetroMAE and M2DPR, respectively. "C&S" denotes the fusion result of CotMAE and SimLM.

Assistant Models	MRR@10	R@50	R@1k	nDCG@10 (DL19)
No assistant	39.9	86.8	98.5	69.2
C	40.2	87.3	98.5	69.8
S	40.4	87.3	98.5	70.0
R	40.6	87.7	98.7	70.0
M	40.6	87.6	98.8	70.2
C&S	40.4	87.3	98.7	69.6
C&R	40.6	87.3	98.7	69.6
C&M	40.5	87.6	98.7	70.0
S&R	40.7	87.3	98.7	69.2
S&M	40.6	87.7	98.7	70.1
R&M	40.8	87.8	98.8	70.3
C&S&R	40.7	87.6	98.7	70.8
C&S&M	40.8	87.7	98.7	70.1
C&R&M	40.9	88.0	98.8	70.3
S&R&M	41.0	88.0	98.8	70.6
C&S&R&M	**41.1**	**88.4**	**98.7**	**71.2**

Table 8: The complexity of the training process.

Method	Training Time
Traditional KD	7.53 hours
MTA4DPR	7.12 hours

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	#Params	Index Size	Encoding Time (bs=512)
12	768	110M	25.2G	304.30s
12	384	66M	25.2G	197.06s
6	768	33M	12.8G	163.23s

els, but when there are not enough available models, we need to train teacher/assistant DPR models from scratch, which may increase the training costs.

3. 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.

4. 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 DL2019 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. CotMAE 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_s: 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_s

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

References

[Bengio et al.2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

[Beyer et al.2022] 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.

[Chen et al.2018] 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.

[Cormack et al.2009] Gordon V Cormack, Charles LA Clarke, and Stefan Büttcher. 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.

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

[Craswell et al.2020b] 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*.

[Dai and Callan2019] Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 985–988.

[Devlin et al.2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 417–426.

[Formal et al.2021] Thibault Formal, Benjamin Piwowarski, and Stéphane 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.

[Gao and Callan2021a] Luyu Gao and Jamie Callan. 2021a. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 981–993.

[Gao and Callan2021b] Luyu Gao and Jamie Callan. 2021b. Is your language model ready for dense representation fine-tuning? *arXiv preprint arXiv:2104.08253*.

[Gao et al.2021] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3030–3042.

[Heo et al.2019] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. 2019. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3779–3787.

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

[Huang et al.2022] Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. 2022. Knowledge distillation from a stronger teacher. *Advances in Neural Information Processing Systems*, 35:33716–33727.

[Karpukhin et al.2020] 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.

[Kwiatkowski et al.2019] 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.

[Lee et al.2024] Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, Tao Lei, Iftekhar Naim, Ming-Wei Chang, and Vincent Zhao. 2024. Rethinking the role of token retrieval in multi-vector retrieval. *Advances in Neural Information Processing Systems*, 36.

[Lin and Ma2021] 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*.

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

[Lu2024] Qixi Lu. 2024. M2DPR: A multi-task multi-view representation learning framework for dense passage retrieval. In *NAACL Student Research Workshop 2024*.

[Lu et al.2022] Yuxiang Lu, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhenglie Huang, Shikun Feng, Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, et al. 2022. Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval. *arXiv preprint arXiv:2205.09153*.

[Ma et al.2024] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425.

[Mao et al.2021] 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.

[Mienye et al.2020] Ibomoiye Domor Mienye, Yanxia Sun, and Zhenghui Wang. 2020. Improved predictive sparse decomposition method with densenet for prediction of lung cancer. *Int. J. Comput.*, 1:533–541.

[Mirzadeh et al.2020] 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.

[Ni et al.2022] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855.

[Nogueira et al.2019] Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to doc2query. *Online preprint*, 6:2.

[Peng et al.2019] 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.

[Qin et al.2024] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, et al. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518.

[Qu et al.2021] 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.

[Ren et al.2021a] 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.

[Ren et al.2021b] 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.

[Robertson and Zaragoza2009] Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends@ in Information Retrieval*, 3(4):333–389.

[Son et al.2021] 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.

[Sun et al.2024] 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.

[Wang et al.2023] 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.

[Wu et al.2023] Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. 2023. Contextual masked auto-encoder for dense passage retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4738–4746.

[Xiao et al.2022] 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.

[Xiong et al.2021] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

[Yang et al.2022] 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.

[Yuan et al.2021] 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.

[Zeng et al.2022] 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.